Evolutionary Algorithms for Scheduling Operations

Elena Khmeleva

A thesis submitted to
Sheffield Hallam University
for the degree of
DOCTOR OF PHILOSOPHY

Sheffield Business School
Sheffield Hallam University
September 2016

Abstract

While business process automation is proliferating through industries and processes, operations such as job and crew scheduling are still performed manually in the majority of workplaces. The linear programming techniques are not capable of automated production of a job or crew schedule within a reasonable computation time due to the massive sizes of real-life scheduling problems. For this reason, AI solutions are becoming increasingly popular, specifically Evolutionary Algorithms (EAs).

However, there are three key limitations of previous studies researching application of EAs for the solution of the scheduling problems. First of all, there is no justification for the selection of a particular genetic operator and conclusion about their effectiveness. Secondly, the practical efficiency of such algorithms is unknown due to the lack of comparison with manually produced schedules. Finally, the implications of real-life implementation of the algorithm are rarely considered.

This research aims at addressing all three limitations. Collaborations with DB-Schenker, the rail freight carrier, and Garnett-Dickinson, the printing company, have been established. Multi-disciplinary research methods including document analysis, focus group evaluations, and interviews with managers from different levels have been carried out. A standard EA has been enhanced with developed within research intelligent operators to efficiently solve the problems.

Assessment of the developed algorithm in the context of real life crew scheduling problem showed that the automated schedule outperformed the manual one by 3.7% in terms of its operating efficiency. In addition, the automatically produced schedule required less staff to complete all the jobs and might provide an additional revenue opportunity of £500 000.

The research has also revealed a positive attitude expressed by the operational and IT managers towards the developed system. Investment analysis demonstrated a 41% return rate on investment in the automated scheduling system, while the strategic analysis suggests that this system can enable attainment of strategic priorities. The end users of the system, on the other hand, expressed some degree of scepticism and would prefer manual methods.

Acknowledgements

I would like to sincerely thank my supervisors Professor Adrian Hopgood, Dr Malihe Shahidan, Mr Lucian Tipi and Dr Jonathan Gorst for their guidance and support. Owning to their extremely helpful advice, necessary resources and encouragement, I was able to conduct a very challenging and interesting research project, achieve good results, participate in various competitions and win various awards.

I would also like to express my gratitude to DB-Schenker and Garnett-Dickinson for their participation in my research. My research would not be possible without the data provided by the aforementioned companies. I also appreciate the time all the participants from both companies spent explaining to me their extremely complex processes, sharing their professional knowledge and providing evaluations throughout the development of my research algorithm.

I would like to say a special thanks to Adrian Hopgood for an opportunity to attend external conferences and events, which enabled me to present my research and expand my knowledge outside my immediate research areas.

I am grateful to Mr Lucian Tipi, Malihe Shahidan, Jonathan Gorst and Jamie Rundle for introducing me to teaching, which allowed me to share my knowledge with students.

Finally, I would also like to thank Sheffield Business School for partly sponsoring my research project.

Table of contents

List of figures

List of tables

Chapter 1. Introduction

In recent years technology has been developing at an exponential pace revolutionising business operations and re-shaping customer experience (Kurzweil 2006). Such products as Google Home and Amazon Echo are taking control of homes by being able to understand commands in natural language and remotely operate various appliances and devices (Brandon, 2016). Chatbots like Amelia can answer standard questions and handle a natural conversation with customers, while sophisticated algorithms are capable of predicting future purchases and making personalised product recommendations (Business Wire 2014, Fang, Zhang and Chen 2016, Zhang and Song 2015, Da-Cheng Nie, et al. 2014).

These rapid advancements in AI and their proliferation in day-to-day life are beginning a new technological era. Researchers and practitioners mark this time as the fourth industrial revolution or industry 4.0 (Baur and Wee 2015).

According to the Forrester research agency, $2.06 trillion have been invested globally in software, hardware, and IT services by enterprises and governments in 2013 (Lunden 2013). The number of supplied industrial robots is increasing every year. According to the International Federation of Robotics (2015), it reached 229,000 units in 2014 with eighty percent of executives believing that AI improves workers' performance and creates jobs (Narrative Science 2015).

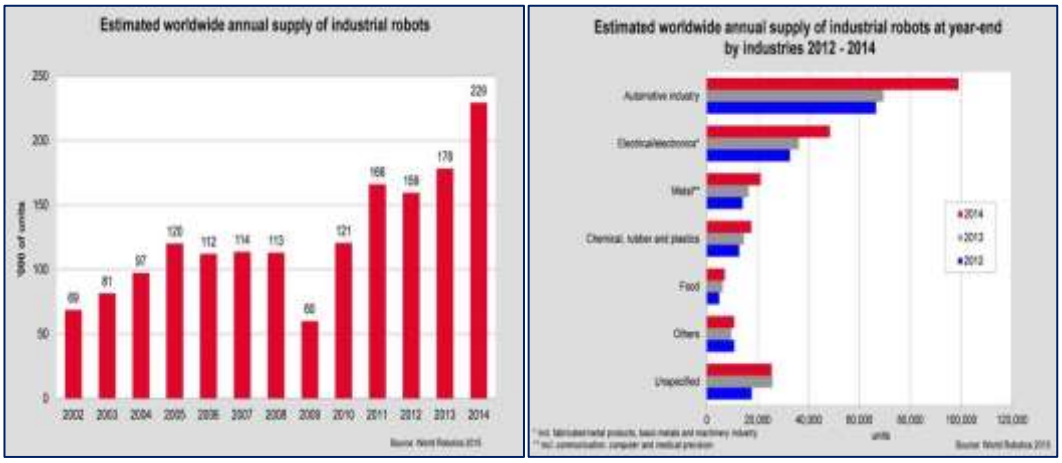


Figure 1 Worldwide annual supply of industrial robots

Source: International Federation of Robotics World Robotics (2015)

Von Rosing and Polovina (2015, p.196) state that "*Business processes are the heart of an organization and the support of the business processes by application systems is central to each organization*". Moreover, in today's rapidly changing business environment and increasing volumes of available information, it is impossible to make an adequate analysis and to take a rational decision without the aid of an information system. Baltzan (2009, p.60) describes this challenge as "*Highly complex decisions involving far more information than the human brain can comprehend must be made in increasingly shorter time frames*".

However, to date, not all companies have a powerful information system, which can assist them with data analysis and decision making. From eighty to eighty-five percent of information remain uncaptured by some of enterprise applications (Polovina 2013). Mckinsey's global survey of 807 executives shows that a large number of respondents expressed dissatisfaction with their current IT solutions. Moreover, respondents claim that their IT is becoming less effective in helping them achieve strategic objectives (Khan and Sikes 2014).

1.1 Planning and scheduling technologies

An example of such analytical decisions are planning and scheduling operations. Planning and scheduling processes are the backbone operations in many organisations. Effective planning and scheduling enable successful assignment of limited resources to required jobs and often determine the overall cost for the project. As the number of regulations and tasks in the value chains of medium and large businesses increases, building an optimal manual schedule becomes an extremely hard, if not impossible task.

While the automotive and electronics industries are relatively automated (Figure 1), the railway and printing industries are lagging behind. For example, Withall, et al. (2011) and Clarke, et al. (2010) observe that such decisions as platform allocation and rolling stock scheduling are made by humans with very limited assistance from information systems. This research will also show that the complex train driver scheduling decisions in the rail-freight industry and job-scheduling decisions in the printing industry are performed manually as well. Various existing commercial scheduling software such as ShiftPlanning, Appointy and WhenToWork provide only a visual representation for very complex

scheduling problems, so the assignment decision itself is still made by humans (WhenToWork 2016, Appointy 2016, ShiftPlanning 2016) .

There are two key questions which one needs to answer when designing a scheduling system. First of all, what algorithm can be incorporated to produce and optimise the schedule? And, secondly, would it be possible to apply the same algorithm for different problems in order to achieve economies of scale and to reduce the development cost?

Traditionally scheduling problems have been solved with linear programming techniques. Linear programmes were formulated during the Second World War by Kantorovich and in 1946-1947 Dantzing proposed a Simplex method for its solution (Wood 1965, Dantzing and Wolfe 1974). Although these methods provide an optimal solution, they are becoming less and less practical as their computation time grows exponentially with the increase in the size of the data. Onwubolu and Babu (200 p.1) state that "*The days when researcher emphasised using deterministic search techniques to find optimal solutions are gone*".

A new generation of algorithms, metaheuristic algorithms, emerged in the 1960s-1970s, which are now becoming increasingly popular (Gogna and Tayal 2013, Kincaid 2008, Gendreau and Potvin 2005, Blum and Roli 2003). By mimicking various natural processes, they are able to tackle large volumes of data and arrive at a sub-optimal solution within an acceptable time frame. For example, Evolutionary Algorithms (EAs) are a class of metaheuristic algorithms, which replicate biological evolution and are guided by the Darwinian principle of "the fittest survives".

One of the algorithms belonging to this class is the Genetic Algorithm (GA) developed by John Henry Holland in 1975 (Holland 1975). Unlike other EAs, a GA uses binary vectors in order to encode the solution. However, similar to many EAs, a GA is "*an artificial intelligence system that mimics the evolutionary, survival-of-the-fittest process to generate increasingly better solutions to a problem. A GA essentially is an optimizing system: it finds a combination of inputs that gives the best outputs*" (Baltzan 2015 p.62). He further states that a GA is best suited to decision making environments in which, thousands, or perhaps millions, of solutions are possible. This is because it can "*find and evaluate possibilities faster and more thoughtfully than a human*" (Baltzan 2015 p.62).

Examples of such environments and problems can include the driver scheduling problem in the rail-freight industry or job-shop scheduling problems in the printing industry.

Several studies such as Azadeh et al. (2013), Shen et al. (2013), Zeren and Ozkol (2012), Ozdemir and Mohan (2001) and Levine (1996) have proposed EAs for the solution of crew scheduling problems. Spanos et al. (2014), Zhang, Gao and Li (2013), Dong Hui (2012), Meeran and Morshed (2012), Qing-dao-er-ji and Wang (2012, Yang et al. (2012) and Jia et al. (2011) have developed the algorithms for the solution of job scheduling problems.

However, despite the popularity of EAs in operations research communities, there are four areas which are under-researched. These gaps are presented below.

1. Most of the EAs have been devised to tackle a specific problem. Minimal research has been conducted regarding applicability of those algorithms across different domains. Moreover, none of the studies investigated the benefits and risks of application of the generalizable algorithm in real life.
2. There is no conclusion amongst researchers regarding the efficiency of genetic operators across different domains.
3. The majority of the developed algorithms have not been tested on real life sets of data and their practicability is not fully known.
4. There is a lack of knowledge of what impact these algorithms would have on a broader business performance if they were built in to real information systems.

This research approaches the problem in a different way. First of all, it works with real business problems rather than their simplified models. Secondly, it tests the effectiveness of standard genetic operations across two conceptually different problems. Finally, it examines the implications of potential utilisation of the devised algorithm in a real business environment.

1.2 Research question and objectives

The key objective of this research is getting an insight into EA capabilities for the solution of real life scheduling problems. An EA will be developed and tested on

two significantly different formulations of scheduling problems in different industries. Its potential impact on immediate and long term business performance will be studied. Given that, the research question and objectives are proposed below.

RESEARCH QUESTION: Can an EA improve scheduling processes in different real-life situations?

RESEARCH OBJECTIVES**:**

1. To understand the complexity of scheduling operations in real life.
2. To investigate the performance of standard EA operators against each other.
3. To develop and evaluate problem-specific operators in the context of the Crew Scheduling Problem (CSP) and the Job Shop Scheduling Problem (JSSP).
4. To understand the limitations and challenges in the design of a generalizable algorithm for different domains.
5. To assess the impact of the automatic crew scheduling system on the operational and strategic performance in a real distribution organisation.

In order to achieve the above-mentioned objectives, collaborations with two companies where scheduling operations play a crucial role have been established. The companies are briefly introduced below, but their operations are considered in greater detail in Chapter 4 and Chapter 6.

1.3 DB-Schenker

DB-Schenker is the largest freight rail operator in the UK (DB-Schenker 2014). Currently DB-Schenker accounts for 39 depots, 1240 drivers and operates 550 trains across the country on a single day. The company transports a wide range of commodities: from coal and chemicals to consumer goods. One of the main challenges the company experiences is the efficient utilisation of train drivers (DB-Schenker Head of HR interviewed on 07/11/2012, DB-Schenker Head of Finance interviewed on 07/11/2012). The problem becomes very complicated in the light of agreements with trade unions, strict restrictions on working and rest

hours, a fixed-hour annual contract with drivers, railway regulations and demand uncertainty.

Preliminary research has found that the current information system (IS) is only able to provide summative historical reports for the team of decision makers. All processes, in particular complex scheduling and rostering operations, are executed manually. IS has only a controlling function on these operations and gives a warning message if the created operation cannot be performed. There is a need of an optimisation engine for IS, which would automatically analyse a large amount of data and suggest an effective way of assignment of the train drivers to the train trips.

1.4 Garnett-Dickinson

Garnett-Dickinson is the second collaborator. This company offers various types of publications ranging from newspapers and catalogues to high quality magazines. The volume reaches hundreds of thousands of magazines, millions of catalogues and brochures a year (Garnett-Dickinson 2012).

Each publication (job) consists of several operations (i.e. printing, folding, stitching). The task is to assign these operations to the relevant machines in the sequence which reduces the completion time and does not violate operational constraints (deadlines, relevance of machines). The effective scheduling processes can reduce the lead time and fully utilise the capacity (Garnett-Dickinson Chief Executive interviewed on 16/10/2012, Garnett-Dickinson Managing Director interviewed on 16/10/2012).

To date, the computer software enables only graphical representation of the job-shop using "drag & drop" technology; however, all the assigning decisions sit on the scheduler (Garnett Dickinson Operations Manager interviewed on 20/10/2012) . That makes the company highly dependent on the planners as well as on their expertise and experience.

1.5 Justification of research collaborators

These companies were chosen for the following reasons:

1. Scheduling operations take a central place in their operational strategies.

2. They do not have an automated scheduling system, so it would be possible to carry out an evaluation of how the companies can leverage the IS solutions based on EAs.
3. The large size of trips, depots, crew, and printing jobs is sufficient to build a credible model and to make generalisations.
4. The companies are interested in the research and agree to cooperate
5. They have granted access to documentation, data and permission to interview members of staff to obtain the required information.
6. The scale of operations and operational conditions are extremely different. Testing the same algorithm on such diverse problems would provide reasonably accurate information with regard to its universal applicability and generalizability.
7. The head offices of both companies are located in South Yorkshire. This offers an opportunity to make regular visits in order to collect a sufficient level of information.

Elaborating on the sixth point, it is necessary to highlight the considerable difference between the problems. The first problem deals with the allocation of the workforce whereas the second deals with the allocation of jobs, thus both have very different scheduling rules and constraints (Pinedo 2009). Moreover, the amount of jobs which need to be allocated to machines in the second problem are significantly lower than the number of train trips in the first problem. As the first problem presents a greater optimisation challenge, this will be the focal example in the thesis.

1.6 Research contributions

This study has made six major academic and practical contributions.

1. The performance of standard genetic operators for two significantly different scheduling problems has been investigated and the most efficient operators have been identified.
2. Novel intelligent genetic operators which allow a practical solution to be found more quickly than conventional operators, while still satisfying all the problem constraints, have been developed.

3. A comprehensive evaluation framework encompassing operational and strategic factors has been developed and applied.
4. The possibility and associated risk of using the same algorithm to reduce the software development costs have been examined.
5. The suitability of an EA for real life scheduling operations has been examined and the operational and economic effect on overall organisation performance has been assessed.
6. A complex research methodology has been designed for combining document analysis, interviews, focus groups, and computational experiments, in order to achieve the above contributions.

1.7 Publications resulting from this thesis

In addition to the aforementioned contributions, the author has produced a conference paper, which has won the Best Refereed Paper Written by a Student award in the application stream at 34th Annual International Conference of BCS Charted Institute for IT in Specialist Group on Artificial Intelligence:

> Khmeleva, E., Hopgood, A.A., Tipi, L. and Shahidan, M. "Rail-Freight Crew Scheduling with a Genetic Algorithm" Proc. AI-2014, Research and Development in Intelligent Systems XXXI, M.Bramer and M.Petridis (eds.), Springer, December 2014.

The full text of the publication and the awarded certificate are presented in Appendix 12.

1.8 Organisation of the thesis

Broadly the structure of the thesis can be divided into three parts. The first part concerns the design of an appropriate methodology for the research. The second part provides an insight into real life scheduling operations and devises a model describing the processes. Finally, the third part deals with the design of the effective EA and assesses its applicability in a real business environment. The summary of each chapter is presented below.

Chapter 1 Introduction	This chapter introduces the research question and research objectives. It also provides an overview of the organisations participating in this research and their scheduling operations.
Chapter 2 Overview of optimisation methods	The key objective of the chapter is to select and justify the algorithm which will be incorporated into the automatic scheduler. In order to accomplish this, several algorithms will be critically reviewed and discussed.
Chapter 3 Multi-disciplinary methods for real life problems	The aim of this chapter is to design an appropriate methodology which would enable the capture of complex real life processes as well as supporting the development of the EA and its evaluation in real life settings.
Chapter 4 Job-shop scheduling in the printing industry	This chapter presents the context of job-shop scheduling in the printing industry and identifies the challenges that schedulers face when assigning jobs to printing presses. The output of this chapter is the model describing the job shop scheduling process.
Chapter 5 Approaches to job-shop scheduling problem	This chapter discusses the approaches developed in the literature for the solution of job-shop scheduling problems and analyses their advantages and disadvantages.
Chapter 6 Crew scheduling problems in the rail-freight industry	This chapter outlines scheduling and planning operations in the rail-freight transportation industry. It focuses on the driver scheduling problems and related health and safety regulations and contractual conditions, which

	must be taken into account when constructing a driver working plan.
Chapter 7 Approaches to crew scheduling problems	This chapter reviews existing approaches to the solution of crew scheduling problems and analyses their effectiveness in relation to real-life scheduling operations.
Chapter 8 EA design	This chapter sets the main principles for effective EA design and proposes two novel operators specifically developed in this research to solve crew scheduling and job-shop scheduling problems
Chapter 9 Comparison of evolutionary operators and strategies: experimental results	This chapter experimentally compares traditional genetic operators and intelligent operators devised within this research. It also empirically validates the joint application of multiple genetic operators within the same algorithm and compares the results obtained for CSP and JSSP.
Chapter 10 Adaptation of the EA to the CSP problem	This chapter establishes the limitations of the application of the same algorithm for conceptually different problems. It enhances the complexity of the chromosome representation for CSP to drive the efficiency of the algorithm. After examination and tests of various decoding procedures, it selects the configuration of the algorithm which provides the most cost-efficient schedule to be applied on the real data set.

Chapter 11 Implication of the research for an organisation	Discusses the adaptation of real life data and EA objective function to real life CSP. It then reports the results of the expert evaluation of the EA produced schedule and analyses the results. The analysis considers the quality and practicability of the schedule, the impact on operational performance and the alignment with organisation strategy.
Chapter 12 Conclusions and future research directions	Provides an overall summary of the conducted research and obtained results. It also outlines some of the research limitations and suggests future research directions.

1.9 Chapter summary

This chapter has demonstrated that despite significant technological progression and increased supply of industrial robots in recent years, complex planning and scheduling operations are still performed manually. The traditional linear programming algorithms struggle to handle a large of amount of data and are becoming replaced by various AI optimisation algorithms. However, the way the companies in the rail-freight industry can leverage AI based on automated planning technologies and their impact on their performance is a significantly under-researched area. The main research question of this study is to devise a proof of concept of such software and conduct a detailed assessment of its effectiveness in a real organisation.

The next chapter introduces and closely examines metaheuristic algorithms, which can be applied to assist companies in solving their crew and job-shop scheduling problems.

Chapter 2. Overview of optimisation methods

2.1 Introduction

The purpose of this chapter is to identify the technique, which can be incorporated into CSP and JSSP automatic schedule builders. In order to accomplish this, several optimisation methods for the solution of combinatorial problems will be reviewed. Their general logic as well as strengths and weaknesses in relation to the solution of the scheduling problems will be discussed. More importantly, because this thesis considers their real-life application and business impact, the development efforts for each algorithm will also be taken into account when selecting an algorithm. Development efforts will be measured on the basis of two dimensions: the algorithm complexity (i.e. how long and sophisticated the code should be) and generalizability (i.e. the possibility to transfer the code to other problems).

This chapter is focused only on the AI algorithms and does not present exact techniques. This is because the majority of real-life optimisation problems have an immense number of constraints and variables and belong to the class of NP-hard combinatorial problems (Hart, Ross and Nelson (1998), Gogna and Tayal (2013)). The exact methods usually struggle with such a large number of variables because they are based on techniques which require generation of all possible combinations. Since it is almost impossible to produce and evaluate all possible solutions for NP-hard problems, they have been rejected due to their impracticability for this research.

2.2 Heuristic methods

Reeves (1993,p.6) provides a definition of heuristic methods.

Definition 1

Heuristic is "*a technique which seeks good (i.e. near optimal) solution at a reasonable computation cost without being able to guarantee either feasibility or*

optimality, or even in many cases to state how close to optimality a particular feasible solution is".

Unlike exact integer programming (IP) techniques, heuristic methods exploit the nature of combinatorial problems rather than their IP formulation (Gendreau and Potvin 2005). Since they do not rely on objective function derivatives they have more chances to escape local optimum as well as handle non-continuous functions and discrete parameters (Haupt 1998). The downside of this is heuristic methods are usually tailored to a particular problem and have a limited applicability to other problems. Design of an effective heuristic method requires substantial knowledge regarding the domain. One of the examples of heuristic methods is greedy randomised adaptive search procedure (GRASP), which is described below.

2.2.1 GRASP (greedy randomized adaptive search procedure)

GRASP is a simple heuristic which consists of two operators: constructive heuristic and local search (Blum et al. 2011). Construction operator assembles a solution element by element with a certain degree of randomization. After that the solution is enhanced with a tailored improvement technique. This is a multi-start method, which means that the described process repeats a certain number of times. The memory preserves all final solutions and at the final iteration the best solution is regarded as a final result (Gendreau and Potvin 2005).

GRASP can provide a solution for a relatively short period of time and can be incorporated into various algorithmic frameworks (Blum et al. 2011). One of the drawbacks of GRASP is that it does not rely on the history of the previously obtained solutions in order to direct the search (Gendreau and Potvin 2005, Blum et al. 2011). Furthermore, the heuristic technique must be specifically designed for a particular domain, which reduces the level of generalizability of the algorithm and requires ample knowledge about the problem.

2.3 Metaheuristic methods

Unlike simple heuristic methods, meta-heuristics are less problem-dependent and are based on a specific algorithm that manages low-level heuristics.

Definition 2

A ***metaheuristic*** is "*an iterative generation process which guides subordinate heuristics by combining intelligently different concepts for exploring and exploiting the search space, and learning strategies are used to structure information in order to find efficiently near-optimal solutions*" (Osman and Laporte 1996, p.1).

In general, heuristic techniques are more flexible and adaptable to real-world situations than exact methods (Gogna and Tayal 2013). Due to having a variety of tools for productive exploration and exploitation of search space, they are more likely to reach the global optima of the function than heuristics methods. Moreover, these techniques, in many cases, allow the algorithm to escape local optimuma. As to their disadvantages, the tuning and design of a low level heuristic might be time-consuming.

Despite the fact that meta-heuristics do not guarantee finding the mathematical optimum, they are highly capable of finding a reasonable solution for a much shorter period of time than exact methods. This fact makes meta-heuristics highly attractive for real life applications (Reeves 1993, Gendreau and Potvin 2005, Gogna and Tayal 2013).

The summary of the benefits and limitations of metaheuristic methods are displayed in Table 1.

Table 1 Advantages and disadvantages of metaheuristic methods

Advantages	Limitations
• More general applicability than heuristic alone • Find global optima • Deal with local optimums • Reasonable computation time • Find global optima • Handle complicating constraints	• Do not guarantee finding the optimum solutions • Not easy to prove efficiency of the algorithm • Requires a significant amount of time to be developed • Analysis and selection of the algorithm for a particular problem is a very challenging task • Requires extensive knowledge about the problem

Adapted from Gogna and Tayal (2013), Gendreau and Potvin (2005)

The majority of metaheuristic methods were inspired by real life processes. For instance, Simulated Annealing mimics thermodynamic processes; Tabu-Search

resembles the brain memory; GA is based on biological evolution and Ant Colony Optimisation imitates ants' behaviour. These methods as well as their advantages and limitations will be discussed below.

2.3.1 Simulated Annealing

The logic of Simulated Annealing (SA) algorithm is derived from the Metropolis's algorithm defining annealing process. Annealing is the chemical transformation occurring in metals when they undergo temperature changes (Reeves 1993). Usually the metal is first heated at a very high temperature till the point when it starts to melt. After that, the temperature starts to drop according to a certain schedule altering the internal energy and making the structure of metal rigid and fixed. The process is often applied in metallurgy in order to produce materials of a high quality with a minimum number of defects (Elhaddad 2012).

Application of annealing principles as an optimisation technique was proposed by Kirkpatrick et al. (1983). In the optimisation context, the energy of the system is equivalent to objective function, state of the physical system is represented by solution and temperature is a control parameter regulating exploitation and exploration phases (Reeves 1993, Gendreau and Potvin 2005, Gogna and Tayal 2013).

The SA algorithm starts from initialisation of the first candidate and selecting a temperature. At the beginning, the temperature should be set reasonably high (Gogna and Tayal 2013). Too low temperature might force the algorithm to converge around a local optimum. However, if the temperature is set too high the convergence of the algorithm can be relatively slow. At the next step, the candidate solution, lying in the neighbourhood of the existing one, is produced. If the formed solution is more cost effective, then it immediately substitutes the existing one. Otherwise it can only be accepted with a certain probability determined by temperature and degree of worsening the solution. In the classic SA algorithm, the probability is computed according to Formula 1. According to this formula the worse solution is more likely to be accepted when the temperature is high and cost increase is low (Gendreau and Potvin 2005).

Formula 1

$$P = \frac{1}{1 + e^{-\frac{\Delta E}{t}}}$$

The temperature reduces at a certain rate a corresponding to the speed of desirable convergence of the algorithm and required completion time. There are several variations of the cooling schedules, which are presented in Table 2.

Table 2 Cooling Schedule

Cooling Schedule	Formula	Comments
Exponential	$T_{t+1} = \alpha^t T_t$	Keeps the system close to equilibrium
Linear	$T_{t+1} = T_0 - \eta t$	One of the most popular strategies to use (Nourani and Andresen 1998)
Logarithmic	$T_{t+1} = aT_0/\ln(d + t)$	Proven to be able to converge to the global minimum, but very slow and is rarely used in practice
Geometric	$T_{t+1} = aT_t$	One of the most popular strategies to use (Nourani and Andresen 1998)
T_0-initial temperature T_t-temperature at the iteration *t* a-cooling speed, (0 <α< 1) d-is usually set to one (Nourani and Andresen 1998)		

Adapted from: Nourani and Andresen (1998), Gogna and Tayal (2013)

SA has a number of benefits which make it a useful optimisation tool. Firstly, acceptance of the worse solution allows the algorithm to avoid being trapped into a local optimum (Blum and Roli 2003). Secondly, owning to a neighbourhood construction strategy, SA thoroughly investigates the search region and can arrive at precise global optimum given that it is located in that region.

However, it would be impossible if the step is too wide as it would be bouncing between different regions rather than between the solutions in the promising region (Nolle, et al. 2001). On the other hand, with too small steps SA might fail in reaching other regions within the allocated timeframe. This challenge is partially caused by the availability of only a single operator which is responsible for the construction of the new solution. Another shortcoming of the algorithm is

that a good solution can be irreversibly overridden by a worse solution as there is no mechanism which stores search history.

2.3.2 Tabu search

Tabu search (TS) was proposed by Glover (1986). The distinctive feature of TS is the memory, which keeps a record of the history of the search (Hopgood 2012).

Like in previously discussed methods, TS starts from building the first candidate. However, unlike SA and GRASP, at the next step it constructs not only one, but several solutions from the neighbourhood of the first candidate. Furthermore, the new solution is only accepted if it has not appeared before (Reeves 1993). This is controlled by *tabu lists* which stores the information about the previously generated candidates. The size of tabu list is called *tenure*. When the tenure exceeds the limit, some of the candidates, usually using a first-in-first out rule are released from the tabu list.

Too small tenure would restrict the algorithm to the exploitation of smaller regions, whereas too large would encourage the search to explore other regions without appropriate examination of each region (Blum and Roli 2003). Since storage of the entire set of solutions requires substantial memory resources, usually only solution parts or movement attributes are saved. However, this might prohibit acceptance of good solution which has some of the forbidden properties unless it possesses some of the *aspiration criteria*. One such criterion is the candidate exibiting the lowest value of the objective function recorded in the process (Blum et al. 2011, Hopgood 2012). Once a group of allowed candidates has been generated, the strongest solution with the minimum cost is selected to become a new current solution and the process repeats.

By preserving the information regarding previous trials, it becomes possible to avoid cycling between the same solutions and thus to save computation time (Gogna and Tayal 2013, Hopgood 2012). Moreover, existence of the long term memory provides the opportunity to recover the best discovered solution even if it was replaced by the worse one (Hopgood 2012).

2.3.3 Ant colony optimisation

Ant colony optimisation (ACO) algorithm imitates ants' behaviour (Dorigo 1992). Searching for the food source and the best direction to reach it, ants leave a chemical compound called ***pheromone***. Once they have reached the foraging area of food, they return back to the nest. As pheromone tends to evaporate over time from the shortest paths, the concentration of the pheromone on the shortest paths becomes higher. Thus other ants, deciding which way to go, are more likely to follow the path with more intense pheromone concentration. Because they also leave pheromone as they walk, the shortest paths get reinforced and the longest are forgotten. It continues until eventually all the ants follow the same route to the food source (Jargen and Guntsch 2005).

Inspired by the ants' skill to find the shortest path, Dorigo (1992) used this mechanism as the logic of optimisation algorithm for the travelling salesman problem. The algorithm begins with initialisation of the set of ants. Then the artificial ants start to construct a partial solution (in terms of what city they will visit next). The decision relies on the previous experience of other ants (pheromone concentration) and the immediate benefits of arriving at the next city. In the classic ACO, the probability is calculated based on the formula below:

Formula 2

$$p(c_r \mid s_a[c_l]) = \begin{cases} \dfrac{[n_r]^a[\tau_r]^b}{\sum_{c_{u \in J(s_a[c_l])}}[n_u]^a[\tau_u]^b} & if\ c_{r \in J(s_a[c_l])} \\ 0\ otherwise \end{cases}$$

where:

a, b signify the relative importance of heuristic information and pheromone value;

$J(s_a[c_l])$-the set of a solution components that are allowed to be added to the partial solution $s_a[c_l]$, where c_l is the last component which was added;

Once all ants have selected their next moves, the pheromone trait increases on the low cost paths while a small fraction is removed from all the paths to mimic the effect of evaporation.

The pheromone mechanism regulates the direction of the search. Concentration of the pheromone increases on certain paths reinforcing the good solutions, whereas pheromone evaporation prevents the algorithm from being stuck in the local optimum. ACO can be very effective for the problems which could be presented in the form of a graph, however, it might be quite hard to fit this algorithm into other models (Gogna and Tayal 2013).

2.3.4 Genetic Algorithm

Genetic Algorithm (GA) mimics natural evolution processes, which is based on the “fittest survives” principle formulated by Darwin (Hopgood 2012). The strongest candidates have a higher probability to reproduce and pass on their good traits to the offspring than weak members of the population. Occasionally some individuals undergo mutation, which alters some of the properties in the organism. Nevertheless, the overall population continues to evolve becoming fitter and stronger.

Despite the idea of application of evolutionary processes in the field of optimisation is being attributed to Holland (1975), the early works on adaptation of genetics concepts for solution of optimisation problems can be found in the studies of Rechenberg (evolutional strategies), Schwefel, Fogel, Owens and Walsh (evolution programming) (Mitchell 1996). The defining merit of Holland's works was the determination of the EA's working principles and formation of schemata and building blocks concept, which are described in greater detail in section 2.7.6.

From the optimisation perspective, the analogy of the individual is the solution itself, the reproduction process is performed by a crossover operator, which recombines the elements of parent solutions according to a certain rule. Mutation operator randomly inverses one or more elements in the solution.

GA is a population-based method that deals with various solutions within the same iteration. The distinctive feature of GA is a crossover operator, which provides an explicit exchange of the parts of good solutions to form a more superior one. In optimisation terms, it allows exploration of the search space and fast reach to previously unvisited regions. At the same time the algorithm is equipped with a local search operator (mutation), which contributes to exploration

of a current region. Thus a well-tuned EA with effective operators is able to return a good solution at a reasonable computation cost (Gogna and Tayal 2013).

2.4 Hyper-heuristic and multi-purpose algorithms

The limitation of the above methods is that being effective for one problem, they might demonstrate opposite results on another one. This is due to their dependency on guiding parameters and domain specific operators. Their adjustment and development can consume a significant portion of time, and if they were to be use in commercial software, this might result in a higher development cost. This section provides an overview of the approaches which were created to be more flexible and easily transferable to new domains. These methods include reconfigurable schedulers and hyper-heuristics.

2.4.1 Reconfigurable Schedulers

Montana,Talib and Gordon (2007) devised a reconfigurable scheduler named ***Vishnu***. Vishnu is able to produce and optimise various schedules with different rules and objectives. It consists of an optimizer and problem representation framework. In the problem representation framework a user must define the optimization criteria and constraints using a language similar to that used in spreadsheets (Montana,Talib and Gordon 2007). Optimiser has embedded GA in its core. GA is based on order-based (permutation of integers) chromosome representation and utilised position-based crossover, which is discussed in great detail in section 2.7.7. Mutation employs the same principle as crossover but without second parent: randomly identified genes are copied from the first parent, and then the rest of the genes are shuffled and placed into unoccupied positions.

The resulting offspring is compared against other individuals. If the same offspring already exists in the population, then it is simply deleted. Otherwise, its feasibility is verified and restored if necessary. The feasibility repair operator is applied after crossover and mutation processes.

The advantage of reconfigurable schedulers is the significant reduction of time and cost required to develop the software (Montana, Talib and Gordon 2007). The software is capable of providing a feasible solution to such problems as Travelling Salesman, Job Shop Scheduling Problem, Vehicle Routing Problem with Time Window, but it does not guarantee the optimality of the produced

schedule (Montana,Talib and Gordon 2007). Furthermore, the optimisation method uses the same parameters for all problems and it is unlikely that they will be effective for various scheduling problems.

2.4.2 Hyper-heuristics

According to Burke et al. (2013, p.64), ***hyper-heuristics*** is "*a high-level approach that, given a particular problem instance and a number of low-level heuristics, can select and apply an appropriate low-level heuristic at each decision point*". Since each heuristic can have its advantages and disadvantages as well as exhibit different performance on various data instances, the main objective of the selection heuristic operator is to automatically identify and apply the most appropriate low level heuristics (Misir et al. 2013). In comparison with meta-heuristics, hyper-heuristics does not have any domain specific knowledge and works with information regarding the operators' performance rather than the solution of the actual problem (Burke et al. 2013).

It is evident that the larger the heuristic set, the more chances that the good solution will be discovered as it is more likely that the appropriate operator will be applied. At the same time testing and managing a large set of heuristics can be extremely time consuming and memory-intensive. Kiraz, Etaner-Uyar and Ozkan (2013) list several operator selection mechanisms:

- Simple Random (all operators have the same probability to be selected)
- Random Descent (applies the same heuristic until the solution is not improving further, then another one is chosen randomly)
- Random Permutation (selects different operators in series following the order they appear in the program)
- Random Permutation Descent (combination of the Permutation and Descent)
- Greedy (use the one, which produces the largest improvement)

The previously considered techniques relate to the selection type of hyper-heuristics, however there is another type of hyper-heuristics called construction hyper-heuristics. Construction heuristics normally starts from an empty template and then fills it in with parts of solutions.

For instance, Hart, Ross and Nelson (1998) designed a chromosome with a complex structure for catching and planning chicken transportation. This problem is characterised as a highly constrained real-life problem. Unlike the traditional GA, where the chromosome only represents a solution, in the developed algorithm chromosome denotes both the elements of solution and the method of its decoding (Figure 2). The first two rows are associated with the orders and their quantities whereas the third and fourth rows dictate which rule must be applied to decode a corresponding order. While all the rows participate in evolutionary processes, different genetic operators are applied to each part of the chromosome.

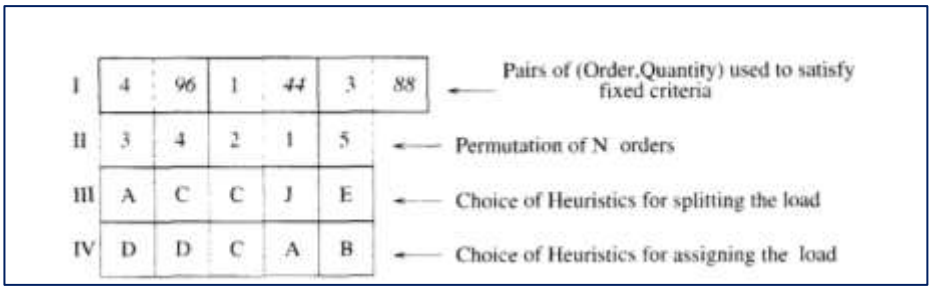


Figure 2 Hyper-heuristics chromosome representation

Source: Hart, Ross and Nelson (1998)

Despite the fact that the algorithm might be perceived as highly problem-specific, Hart, Ross and Nelson (1998) claim that the principles of evolution of both solution and rules can be spread to other problems as well. Moreover, the result comparison with the SA algorithm revealed that EA was able to obtain more practical solutions.

Han and Kendall (2003) developed a hyper-EA for a design of a timetable for events and trainers. Rather than evolving the timetables, hyper-EA evolved fourteen rules (variations of add and remove rules) of assembling timetables. The distinctive characteristic of this algorithm was that the initial timetable remained constant during the algorithm, but various construction rules were evolving resulting in improvement in their abilities to modify the initial schedule. Changes which each operator made were recorded and used to select crossover and mutation operators.

2.5 Algorithm comparison and analysis

The review of the algorithms is conducted from two perspectives: algorithmic configuration and flexibility to accommodate new problems. Table 3 illustrates the level of generalizability in relation to the anticipated development time. Development time refers to the amount of time spent on the coding of the initial algorithm. Level of generalizability denotes the degree of manual adaptation required to fine tune the algorithm for the solution of different problems. The more time required the smaller potential for generalizability the algorithm has.

Table 3 Development time and Generalizability of the algorithms

	Low generalizability	High generalizability
Low development time	Heuristics	Re-configurable schedulers
High development time	Meta-heuristics	Hyper-heuristics

Heuristic methods, mainly consisting of one or two operators, rely on only a small piece of code (hence requiring a relatively small amount of time) which is produced specifically for a certain problem. Application of the algorithm to any other problems would need a considerable redesign of the operators. The same is applied to meta-heuristics with the exception that they are also guided by algorithmic parameters which have to be altered to each problem, hence more time is required to adjust the algorithm to other domains.

Re-configurable schedulers require a small amount of development time as they are based on a single GA and the same set of operators is applied across different problems. In contrast, hyper-heuristics requires development of a collection of low-level operators and the operator selection procedure, which makes the construction of hyper-heuristic more time-consuming than re-configurable schedulers.

Metaheuristic and hyper-heuristic are placed on the same level in terms of the development time. This is because the manual tuning of metaheuristic algorithm compensates for the time required to design and test an operator selection mechanism.

Figure 3 illustrates how the programming efforts are translated into the effectiveness of the algorithm. The reconfigurable methods are the least time consuming in terms of the development. This is because once the general optimisation technique has been developed, the user only needs to formulate constraints and objectives. The algorithm is able to solve scheduling problems as long as the problem can be formulated with available tools. The performance of such an algorithm is unlikely to be high since very general operators are used for all the problems.

For large scale problems, metaheuristic and hyper-heuristic appears to be more effective than heuristics since they have a capability to perform and automatically regulate the exploitation and exploration of the search space. Metaheuristic were placed higher with respect to its performance because it thought to be more tailored to a particular problem and thus is expected to show better results although at the cost of more narrow applicability. Another reason for the high performance is that metaheuristic has faster execution speed since they do not employ automatic adjustments.

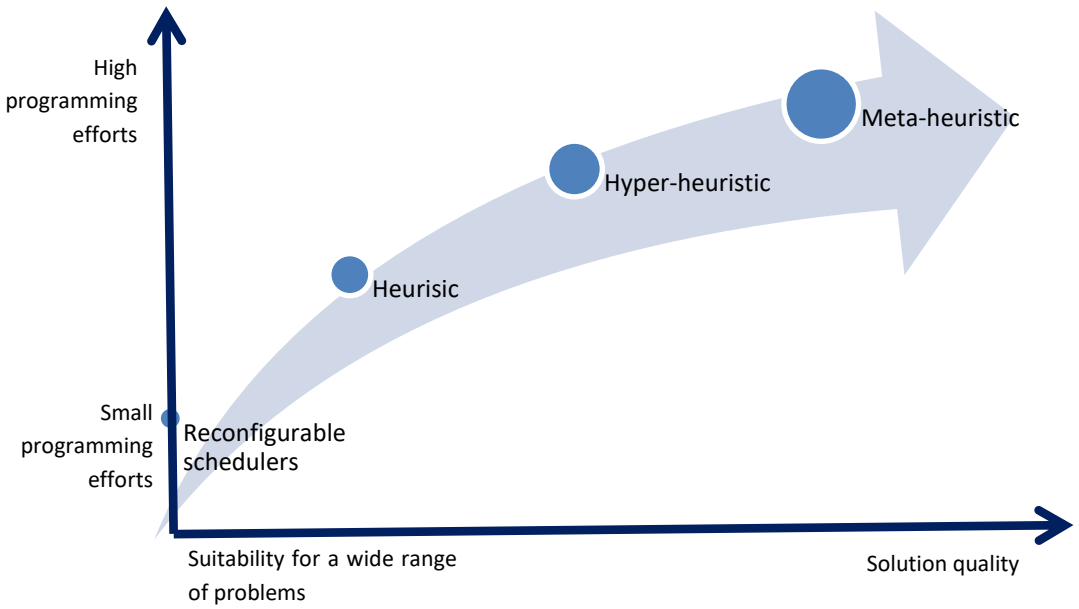


Figure 3 Generalizability of the algorithms level of programming efforts

Since it was concluded that metaheuristic techniques provide a trade-off between the quality of the results and required computation time, the methods belonging to this class will be compared in greater depth. The comparison is based on key

search functions suggested by Gendreau and Potvin (2005) and the details are presented the Table 4.

Table 4 Comparative analysis of the heuristic and meta-heuristic methods

	GRASP	SA	TS	ACO	EA
Construction	Yes	Yes	Yes	Yes	Yes
Recombination					Yes
Random Modification			Yes	Yes	Yes
Improvement	Yes				
Memory Update			Yes	Yes	Yes
Parameter Update		Yes			

Adapted from Gendreau and Potvin (2005)

All metaheuristic algorithms start with the initialisation of a solution and use iterative steps to transfer from one solution to another moving through the search space. The first three algorithms (GRASP, SA and TS) are single candidate methods meaning they improve only one solution at a time. In contrast ACO and EA consider multiple solutions within the same iterations. This allows them to have a more global view of the problem by investigating several regions of the search space.

While each algorithm has its unique benefits and limitations as discussed above, the comparison demonstrated that EA contains more means for the effective search than other algorithms. It comprises of two specialised operators, mutation and crossover, which are responsible for exploration and exploitation of the search space. Moreover, it is a population based method, but unlike ACO, it has an ability to explicitly exchange elements between good solutions (Gendreau and Potvin 2005). At the same time, by the means of mutation, it is capable of performing a neighbourhood search similar to other techniques.

Table 4 also indicates that traditional EA is lacking parameter update stage. However, the more recent EAs have incorporated operators for the adjustments of crossover and mutation probabilities depending on how EA is progressing, for examples Xu and Vucovich (1993), McClintock, Lunney and Hashim (1997) and Chiou and Lan (2002).

2.6 Justification of the selected technique

Having reviewed all the methods, it was revealed that EAs have a number of advantages, which make them suitable for real-life scheduling problems. To summarise, the main benefits and capabilities of EA are:

1. Population-based method. It iterates a set of solutions that leads to better coverage and exploration of the search space (Gogna and Tayal 2013).
2. Contains a mutation operator supporting the search space exploration process and allowing the search to escape a local optimum.
3. Unlike other algorithms, it enables explicit recombination of the attributes between parent solutions via a crossover operator.
4. EA can preserve several solutions for the next iterations to avoid loss of the best solution during the evolutionary process
5. Can perform robustly on the problems with a noise (as it is population base method and one solution is unlikely to change the direction of the search) (Mitchell 1996)
6. Suitable for complex real-life problems (Mitchell 1996)
7. Flexible framework allows hybridization with other methods
8. Can be executed on several processors.

Since EA has been selected as the leading algorithm for the given study, the core operators and underlying principles of its work will be considered in a greater depth in the next section.

2.7 Detailed analysis of EAs

The general framework of one EA's iteration is displayed in Figure 4. The process begins with the initialisation of the chromosomes, and then the algorithm selects parent chromosomes from the population to produce two new solutions. The iteration ends with the replacement of the current population with a new one.

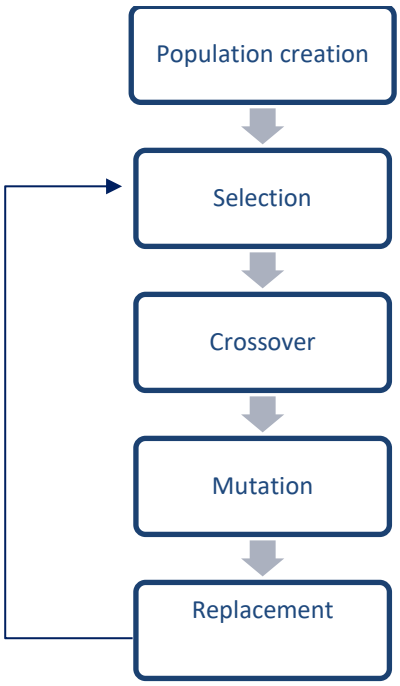


Figure 4 Flow chart of GA

The algorithm repeats until the termination criteria are met. The examples of termination criteria include (Sivanandam and Deepa 2008):

1. Achieved specified number of generations
2. Achieved specified elapsed time
3. The fitness has not changed for a specified number of iterations
4. No improvements have been made within a certain amount of iteration
5. No improvements have been made within a certain amount of time

Each part of the algorithm and the essential principles of its work will be discussed below.

2.7.1 Chromosome representation

Chromosome (or ***individual***) represents a coded solution. The chromosome consists of ***genes***. Position of each gene is denoted by locus and the value as ***allele***.

In traditional GA (a class of EAs) chromosomes are expressed as a binary string of 0s and 1s. However with the further enhancements of EAs and its application to more sophisticated problems, other ways of chromosome representations such as matrix, string of integers, and even computer programmes have emerged (Michalewicz 1996). There are two ways of decoding a solution into

chromosome: ***direct*** and ***indirect***. Direct encoding contains the soltuion itself, whereas indirect consists of the rules for constructing the solution rather than the solution itself (Han and Kendall 2003).

The chromosome can be generated at random as well as based on the already existing feasible, but not the optimal, solutions. Although random generation might increase the population's diversity, it also can construct infeasible solutions. Usually infeasible solutions are tackled in one of the following ways (Michalewicz 1996):

- Automatically deleted from the population (without any further actions).
- Left in the population with the intention that they might contribute to the production of a good solution later.
- The infeasibility is penalised by a fitness function.
- Repair operator which fixes the genes causing infeasibility.

2.7.2 Population

The population is an array of chromosomes. For a stable performance of the algorithm, it is very important to have an adequate population size. The population of a relatively small size can evolve quite fast, but might not capture all the regions of the search space. On the other hand, a too large population can lead to slow convergence (Elmihoub et al. 2006).

2.7.3 Fitness function

Fitness function (maximisation problems) or cost function (minimisation problems) indicates the "goodness of solution". It enables the algorithm to make judgements about particular individuals in the population. The intuitive equivalent of the fitness function is the objective function. However, computation of the value of objective functions of real-life problems can be extremely time-consuming (Elmihoub et al. 2006). To alleviate this, Cooper and Hinde (2003) propose intelligent fitness function, which ensures that the fitness for the same chromosomes will not be tested multiple times. Another possible approach is to use approximation of the fitness function (Elmihoub et al. 2006).

Each cost function can be characterised by the level of epistasis (biological term for genes interaction). According to Haupt (1998, p.168), epistasis is "*the interaction of the coupling between different parameters of the cost function. The extent to which the contribution to the fitness of one gene depends on the value of other genes*". EA works best on medium to high epistatic problems (Haupt 1998).

2.7.4 Selection

Selection is the process of choosing chromosomes for reproduction. Several ways have been developed in order to accomplish this process (Hopgood 2012).

Fitness-proportionate selection. The chromosomes will have a higher probability of being selected if they have higher fitness functions than other chromosomes in the population.

The example is a ***roulette-wheel selection***, in which all the individuals have the section proportionate to their fitness on the abstract roulette-wheel. During the selection, the wheel is rotated and the individual occupying the section where the wheel stopped gets selected. Despite of the fact that the selection probability is being aligned with the fitness, there is still a chance that the individual with low fitness will reproduce more frequently (Hopgood 2012).

In order to mitigate unfair selection, Stochastic Universal Selection, a modification of roulette-wheel selection, can be applied (Hopgood 2012). Stochastic Universal Selection is augmented with a second roulette wheel on which the sections are equally distributed across the wheel. Unlike the roulette-wheel, Stochastic Universal Selection requires only one spin to choose the necessary number of individuals.

Fitness-Scaling. The chromosomes are sorted based on a certain rule, and then topmost individuals are chosen for selection. Methods for ranking calculation include linear fitness scale, sigma scaling, Boltzmann fitness scaling, linear rank scaling, non-linear rank scaling, probabilistic nonlinear rank scaling, truncation selection, transform ranking (Hopgood 2012). The advantage of this method is it allows exploration of the whole search space preventing premature convergence. However, this approach requires more computation time because the scale value of each individual should be estimated every iteration.

***Tournament selection*.** Unlike the previously discussed methods, the probability of the individual to reproduce depends much less on its fitness. It randomly selects two individuals from the population, measures their fitness, and the fittest individual is passed on to the next stage. Only fitness of the selected individuals is calculated meaning that Tournament selection is more time-efficient than the above described techniques. However, as opposed to other methods, the strong individual has smaller chances of being selected.

2.7.5 Elitism

Unlike the standard EA, where a new population completely replaces the current population, elitism prevents strong individuals from being deleted from the subsequent generations (Hopgood 2012). Under elitism strategy, replacement is usually carried out after each individual has been formed (rather than the entire population) by replacing the weakest individual in the population with the new one provided that the new individual is fitter.

2.7.6 Schemata theorem

Before the operators dealing with creation and modification of the chromosomes will be introduced, it is important to understand what happens "inside the algorithm" and why EA works. The schemata theorem proposed by Holland in 1960 is the fundamental principle explaining how EA functions.

Definition 3

Schemata is "*a set of genes values that can be represented by a template*" (Hopgood 2012, p.177) (Figure 5). Asterisk means that the gene can have any value.

0	1	*	1	1	*	*

Figure 5 Schemata instance

Definition 4

Building blocks is a set of schemata. The chromosome matching the schemata are said to be the ***instances of schema*** H (Sivanandam and Deepa 2008).

Definition 5

Order of schemata refers to the number of values on the fixed positions (Sivanandam and Deepa 2008).

Definition 6

Defining length is a distance between the outermost defined bits (Sivanandam and Deepa 2008).

During the algorithm, EA manipulates schemata based on its implicit association with the fitness value. The task of the evolution is to identify those schemata and propagate them further. Building blocks are put together on the same string in the hope of creating a superior string by means of crossover and mutation operators. Good schemata, having a short definition length and a low order, tend to growth very rapidly in the population (Sivanandam and Deepa 2008).

Mitchell (1996) and Sivanandam and Deepa (2008) state that the principles of GA's work are comparable with a sequential, ***two-armed bandit problem***. They explain it as given a set of slot machines ("one-armed bandit machines"), where each machine has a mean value of the award and its variance, but the gambler does not have any information about them. Having N coins the task of the gambler is to maximise the payoff by pulling one of the arms. The common strategy is to conduct some trials by equally pulling each arm and recoding the payoff (exploration). After a certain amount of trials, the gambler selects the arm which gave a maximum pay off and plays on that (exploitation). Clearly the more trials the gambler carried out, the more accurate the decision of which arm to choose would be. On the other hand, he would lose more coins playing on the unprofitable arm.

This problem illustrates the dilemma between exploration (gaining new knowledge and exploitation (obtaining current in-depth and reliable knowledge)

which is a common phenomenon in GA. However, it can be viewed as a multi-armed bandit problem due to the presence of several schemata in the algorithm EA (Mitchell 1996). Formula 3 describes the behaviour of GA according to schema theorem (Hopgood 2012, Mitchell 1996).

Formula 3

$$n(H,t+1) \geq n(H,t)\frac{\bar{f}(H,t)}{\overline{f(t)}}(1-\frac{P_c d(H)}{l-1})(1-P_m)^{o(H)}$$

where H-hyperplane (schema with more than one instance at the previous iteration) ;

$d(H)$- defining length;

l is the total number of genes in the chromosome;

$n(H,t)$ is the number of instances of H at the time t;

$\bar{f}(H,t)$ is a average fitness of H;

$\overline{f(t)}$ is the average fitness in the population;

P_c and P_m are crossover and mutation probabilities respectively;

$o(H)$ is the order of H;

According to this formula the schemata with the higher than average fitness will be occurring more frequently in future generations, whereas the number of schemata with a fitness lower than average will be decreasing (Hopgood 2012, Spears and De Jong 1991). However, this is only applicable when the effect of crossover and mutation is not too disruptive and there is a sufficient sample for reliable estimation of average fitness of H (Spears and De Jong 1991).

Schemata analysis allows not only understanding whether the representation is suitable for EA, but also the overall efficiency of EA as well as prediction of the presence of a certain schema in the next generation (Sivanandam and Deepa 2008, Negnevitsky 2011).

2.7.7 Crossover

Crossover is responsible for production of new chromosomes in the hope that they would be better than existing ones. From the optimisation perspective, the role of the crossover is to facilitate the exploitation of the search by the recombination of building blocks (Mitchell 1996). The schema can survive during

the crossover process if at least one of its offspring is also in its instance and if crossover does not occur within the defining length of the schema (Negnevitsky 2011). Crossovers differ in the way they traverse the searching space.

Single point crossover (Figure 6) recombines parts of the parent chromosomes. The cutting point is usually selected randomly. While it is relatively simple to implement, the limitation of this method is its inability to produce all possible schemata, i.e. it would be impossible to obtain a child with 11*11*1 from the parents 11*****1 and ****11* (Mitchell 1996). Moreover, schemata with long defining length are likely to be destroyed under single point crossover (Mitchell 1996).

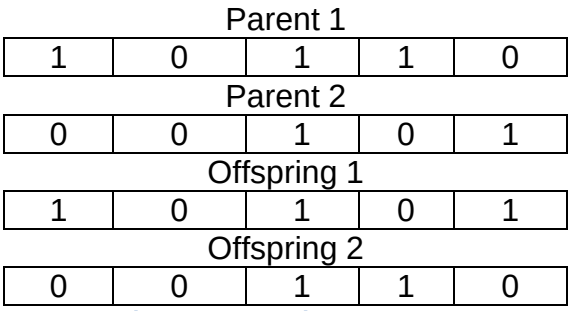


Figure 6 One-point crossover

To tackle this, two-point crossover, with two cut points (Figure 7), has been proposed. However, there can also be schemata that two-point crossover cannot combine.

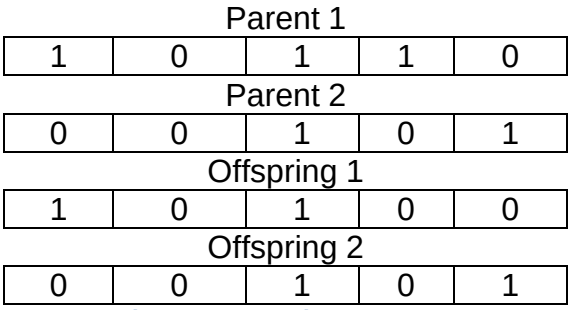


Figure 7 Two-point crossover

Uniform crossover has no "position biases", but can be highly disruptive (Mitchell 1996). Uniform crossover takes a gene from each parent with an equal probability and places it into the same position in s child's chromosome. Figure 8 illustrates the creation of one offspring by using uniform crossover. The second chromosome is formed in a similar way.

Parent 1				
1	0	1	1	0
Parent 2				
0	0	1	0	1
Probabilities				
0.2	*0.7*	*0.3*	*0.9*	*0.2*
Offspring 1				
0	0	1	1	1

Figure 8 Uniform crossover

The above mentioned crossovers are suitable for the binary chromosome representation. However, they might not be appropriate for permutation representation (which is used in many combinatorial optimisation problems) as they can produce duplicate genes. PMX, OX and CX crossovers are very popular and often used for a travelling salesman problem and job shop scheduling problem where permutation representation is used (Sivanandam and Deepa 2008). These crossovers will be presented and explained below.

PMX (partial mapping crossover) works as follows (Haupt 1998, Sivanandam and Deepa 2008). Two cutting points are selected in the Parent 1 and Parent 2 and the alleles between them are copied into Child 1 and Child 2 respectively. Then the rest of the genes are passed according to the exchange map. The following example provides detailed information of how to construct and apply the exchange map.

Parent1	3	8	4	9	2	1	5	6	7
Parent2	8	6	7	2	3	4	9	1	5
Child1	8	6	4	9	2	7	3	1	5
Child2	9	8	7	2	3	1	5	6	4

Figure 9 Partially mapped crossover (PMX)

Assuming that the cutting points are three and five, the segment between them is copied into the same positions in the offspring (Figure 9). The exchange map displays the allelles standing on the same locuses belonging to the selected parts. For the given example the exchange map would be 4->7, 9->2 and 2->3. Then, those genes from Parent 2 that are not in the map are copied into Child1 in the same order as they appear in Parent 2. If they are included in the map then they should be replaced for a corresponding alternative value. For example, 4 will be substituted by 7. In case where if it has to been exchanged for the number which is already in the map (i.e 9 is replaced with 2 and 2 should be replaced with 3), the last value, which is 3, should be inserted.

Kramer and Koch (2007) improved PMX by intelligent selection of the cutting points. Ting, Su and Lee (2010) propose a modification of PMX that can be applied to more than two parents. Both strategies had a positive impact on EA performace, but have been tested only on the traveling salesman problem, thus their effect should still be verified with regard to other problems.

PBX (Position-based crossover) is the equivalent of uniform crossover adapted to the permutation chromosome representation (Cheng, Gen and Tsujimura 1999). It starts with generation of a binary mask, which indicates which parent will provide a gene to the child (0-first parent and 1-second parent). After copying genes from the first parent, the rest of them are passed from another parent to the unfilled positions in a sequence of their appearance in the second parent (Figure 10).

Parent1	3	4	1	6	2	5	9	8	7
Parent2	1	2	3	4	5	6	7	8	9
Binary mask	0	1	0	0	1	0	0	0	1
Child1	3	2	1	6	4	5	7	8	9

Figure 10 Position-based crossover

OX (Order crossover) is similar to the position-based crossover exdef4cept that a part which will be copied is a set of consecutive genes in one parent (Cheng, Gen and Tsujimura 1999). The example of order crossover is shown in Figure 11.

Parent1	1	**5**	**4**	**2**	3
Parent2	4	1	3	5	2
Child1	1	**5**	**4**	**2**	3

Figure 11 Order crossover

CX (Cycle Crossover) forms a child by constantly alternating parents (Haupt 1998). Moreover, unlike the other crossovers the genes are chosen depending on the previously selected genes rather than at random. The following example illustrates the procedure of the CX (Figure 12).

The first round begins with copying the first genes of the first and second parents into the first child and the second child into the first locus (4 and 3). The gene from the first locus of the second parent (4) is found in the first parent, and both genes occupying these positions are copied again (3 and 6). This should be repeated until the cycle is closed up with the same value of the gene as when it has started (6 and 4). Then it starts from the gene which has not been

manipulated yet in the second parent, but now genes from the second parent are going to the first child. Each odd cycle the genes are taken from the first parent to the first child, and each even cycle the genes from the second parent go to the first child. The opposite is applied to the second offspring. If any of the genes are left, but it is impossible to form a cycle, then they are directly dropped down to the relative offspring.

Cycle 1.

Parent1	4	5	1	3	6	8	2	7	9
Parent2	3	2	8	6	4	1	5	9	7
Child 1	4			3	6				
Child 2	3			6	4				

Cycle 2

Parent1	4	5	1	3	6	8	2	7	9
Parent2	3	2	8	6	4	1	5	9	7
Child 1	4	2		3	6		5		
Child 2	3	5		6	4		2		

Cycle 3

Parent1	4	5	1	3	6	8	2	7	9
Parent2	3	2	8	6	4	1	5	9	7
Child 1	4	2	1	3	6	8	5		
Child 2	3	5	8	6	4	1	2		

Cycle 4

Parent1	4	5	1	3	6	8	2	7	9
Parent2	3	2	8	6	4	1	5	9	7
Child 1	4	2	1	3	6	8	5	9	7
Child 2	3	5	8	6	4	1	2	7	9

Figure 12 Cycle Crossover

Cyclic mechanism automatically preserves the legality of the chromosomes. However, there is a probability that the offspring will become an unchanged copy of the parents if all the genes will be inherited in one cycle. In addition, CX was one of the most poorly performed crossovers when tested on TSP problem as it is unable to preserve favourable building blocks (Poon and Carter 1995). Hong

et al. (1995) tested three crossovers: traditional crossover, cycle crossover and two-dimension geographic crossover. Applied with the same probability, Cycle crossover produced an 85% better solution in the beginning, but in the end the performance was equal to all the crossovers.

2.7.8 Mutation

Mutation facilitates exploration of the region. Sivanandam and Deepa (2008) state that by inversion of certain genes it is possible to reduce the defining length of highly fit schema, whereby increase diversity of the population. Because mutation is performed at a very small rate its disruptive power and the extent it affects the solution is not fully understood (Spears and De Jong 1991). There are several ways to perform mutation on permutation chromosome representation.

Insert mutation randomly selects a gene and reinserts it in another locus as illustrated on Figure 13. This type of mutation causes the minimum changes in the chromosome.

Chromosome before mutation	2	5	3	6	4	1
Chromosome after mutation	2	5	3	1	6	4

Figure 13 Insert mutation

Swap mutation exchanges the genes between two randomly identified positions (Figure 14).

Chromosome before mutation	2	5	3	6	4	1
Chromosome after mutation	2	5	1	6	4	3

Figure 14 Swap mutation

Inversion mutation According to Wang and Zheng (2001) inversion mutation arbitrary selects a subsection in the chromosome and reverses the order of all the genes from the selected range (Figure 15).

Chromosome before mutation	2	5	3	6	4	1
Chromosome after mutation	2	6	3	5	4	1

Figure 15 Inversion mutation

Scramble mutation randomly rearranges alleles of the genes on the selected loci (Figure 16). This mutation is more likely to cause a disturbance in the population since it makes the greatest changes in the chromosome. The mutation rate should be relatively low and scramble mutation must be applied to a small

part of the chromosome as it can cause deterioration of the solution quality as well.

Chromosome before mutation	2	5	3	6	4	1
Chromosome after mutation	2	6	5	3	4	1

Figure 16 Scramble mutation

2.7.9 EA parameters

Because crossover and mutation play different roles, it is imperative to find the right balance between them. However, their importance is not static and can change during the course of the algorithm. There are several methods allowing determination of relevant genetic parameters:

- Parameters are identified manually by conducting the experiments for each parameter.
- Carrying out hand optimization. For example, De Jong (1975) calculated optimum parameters for standard problems for the specific population size and EA operators.
- Some researchers evolved encoded parameters together with chromosomes (Hopgood 2012).
- Utilization of fuzzy logic controllers (Varnamkhasti et al. 2012, Neta et al. 2012, Sumer and Turker 2013).

2.7.10 Local search

Local search can have a positive as well as negative impact on the search procedure (El-Mihoub et al. 2006). On the one hand it can direct the algorithm closer to the optimum with less iteration required, whilst on the other hand instigating premature convergence. Furthermore, local searches involve calculation of the fitness function, which is a computationally expensive task for many real life problems. Therefore, it is crucial to determine the correct parameters such as frequency of the local search application, number of the individuals and the method itself to design an effectively functioning search (El-Mihoub et al. 2006).

When used correctly, local search is able to spread the good characteristics of the chromosome to the next generation and is often regarded as learning. There are two ways of incorporation of the learning strategies in the algorithm.

Lamarckian approach integrates the adaptation to the surrounding environment into the genotype properties (Mitchell 1996). Within the algorithm, Lamarckian evolution works as follows. Firstly, the local search mechanism is applied for the selected individual. If the resulting individual is found to be fitter than the original one, then it would replace the existing chromosome. It is said that chromosome representation has a ***Lamarckian*** property if a common genetic operator is able to transfer the good qualities from the parent to a child chromosome. If that is impossible then the chromosome has a ***non-Lamarckian*** property. When only a part of a parent's attributes can be inherited by the future generations, that is called a ***half-Lamarckian*** property (Gen and Cheng 1997).

Baldwinian effect represents learning process during the life time and passing knowledge to the next generations without making any modifications in genetic structures (Mitchell 1996). In the optimisation context, the individual undergoes a problem-dependent local search which usually improves its fitness. However, the change in the original chromosomes occurs only on the phenotype level while the all the genes remain unchanged (El-Mihoub et al. 2006). This procedure signifies the "potential" of the individual that increases the chances of being selected.

Each of these approaches demonstrated its effectiveness when applied to certain domains. Moreover, hybridization of both methods has proven to be efficient for the solution of several real-life problems (El-Mihoub et al. 2006).

2.8 Conclusion

The chapter has presented the optimisation methods commonly used in AI research. The overall logic of each algorithm has been described and their capabilities and limitations have been considered. While each technique has its own advantages and disadvantages, it was shown that EA is more suitable for the solution of the real-life large scale scheduling problems. Therefore, various configurations of this algorithm will be investigated, and then this algorithm will be applied to solve crew scheduling and job shop scheduling problems.

The next chapter will describe the methodology which will be used in order to collect raw data for experimentation and to get an insight into the domains, in which scheduling operations are performed.

Chapter 3. Multi-disciplinary research methods for real life problems

3.1 Introduction

This chapter describes the core methodology for the given research and defines the methods which will be used. Since this research is conducted in two different industries and with the direct engagement of various business stakeholders, it requires utilisation of several data collection and research techniques. These techniques include exploratory semi-structured interviews, focus groups, experiments and investment analysis. All these methods as well as the rationale for their selection are discussed in this chapter.

3.2 Overview of research structure

Research involving real life companies differs from purely academic research in two fundamental ways. First of all, the real life formulation of the problem typically contains a larger number of rules, more possible scenarios and they tend to be more restricted by domain-specific constraints (Hart, Ross and Corne 2005). Secondly, as a sequence of the first one, some of real life systems do not fall into a specific class of standard problems and a tailored model and solution method has to be designed (Jensen 2003). With the intention of studying application and the development of a generalisable scheduling algorithm for real-world scheduling problems, the new research framework, illustrated in Figure 17, has been developed.

The research begins with the collection of pertinent data about the scheduling operations which would enable a good insight into the logic and features of both scheduling problems. At this stage it is crucial to obtain a sufficient amount of information in order to be able to devise a correct conceptual model. In order to achieve this, several visits to the companies will be made as well as a variety of methods of data collection will be employed. At the second stage the collected information will be formally defined and presented in a form suitable for analysis and optimisation.

The third stage deals with examination of a broad range of existing optimisation techniques for the solution of similar models. Analysis of their logic and corresponding strengths and weaknesses will enable the researcher to fulfil the gaps and build on a more powerful technique if necessary.

The fourth stage is the central one as it is responsible for the selection of the technique which will constitute the automatic scheduling system. Several configurations of the algorithm including the standard and problem specific ones will be empirically tested with the aim of identification of the best performing one. Moreover, similar experiments will be repeated for another problem in order to determine the level of robustness and generalizability of the selected configuration.

Once the method has been approved in the "laboratory" environment, a system prototype will be designed before it is presented to a number of industrial experts. In order to conduct a comprehensive evaluation, opinions of various stakeholders will be gathered and discussed. Finally, the essential financial analysis will be carried out in order to assess the economic viability of potential investments in an automated scheduling system.

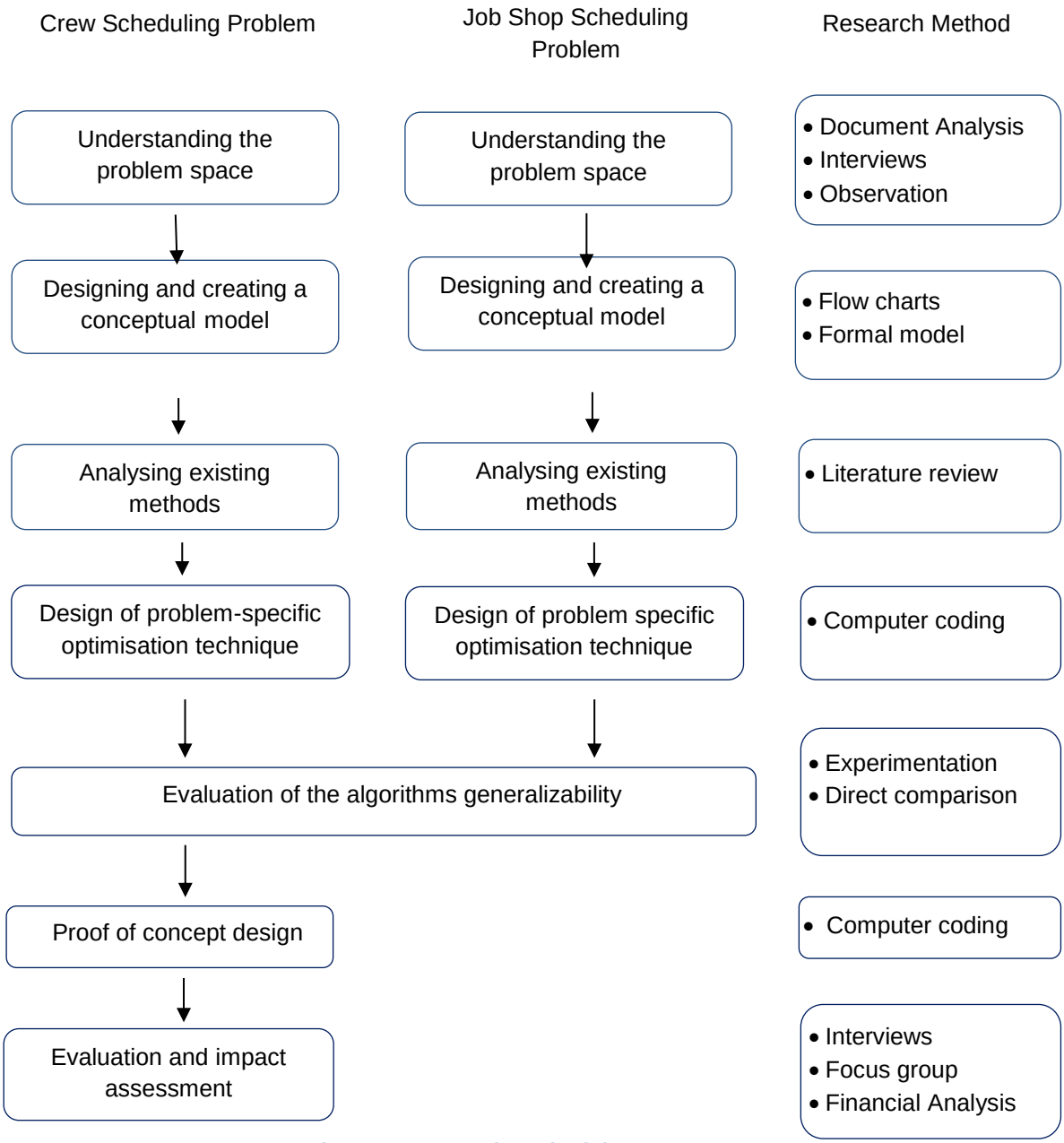


Figure 17 Research Methodology

3.3 Data collection methods

The main objective of data collection is not only an understanding of the scheduling processes, but also acquisition of a sufficient amount of representative historical data for algorithm design and experiments. Wren et al (2003) states that the collection of data in the rail industry presents a very challenging task. The main reason for that is the large number of written and unwritten rules and regulations existing in the industry. Thus, in order to minimise the risk of missing crucial information, the research will rely on two types of data: primary and secondary.

3.3.1 Document analysis

The historical information required for the algorithm testing and evaluation of the results will be collected either during the visits (if available) or sent by email. This process is expected to be convenient for the company managers and the researcher as it does not require a significant amount of time (Martin 1995). However, the main issue is the security of the data. It is also anticipated that the data might be incomplete or not in the appropriate format, hence some data transformation procedures might be required.

The documents explaining procedures and regulations will also be gathered. The advantage of document review is that it allows for receiving complete and structured information with minimal intervention and interruptions (Martin 1995). However, it has some limitations as well. First of all, in order to get an insight into real world operations, some explanation of the industrial terms or operations might be necessary. Secondly, real-world practices, such as those based on schedulers' experience, might be different or not fully explained in the documentation of a company's procedure (Wren et al. 2003). In order to make sure that the information is complete ***triangulation*** will be used. Triangulation is "*the use of different research methods in the same study to collect data so as to check the validity of any finding*" (Gill 2010). Thus interviews and observations will be conducted in order to clarify and expand the information obtained through document review.

The raw secondary data required for this research are presented in Table 5. The methods of collecting the data are discussed below.

Table 5 List of required data

DB-Schenker	Garnett-Dickinsons	Purpose
Historic data set regarding the train trips and drivers	Historic data set regarding printing jobs and machines	• Algorithm development and experiments
Examples of the manual schedules built from the data collected on the previous stage		• Rules extraction and understanding the final schedule. • Assessment of the correctness and quality of the schedule. • Calculation of efficiency/inefficiency of the developed system with the manual practices.
Documents describing the procedures and polices	Not available	• Obtaining knowledge about the background and context in which the process takes place. • Understanding the rules and stages of creating a schedule. • Extraction of the industrial regulations to which the schedule must adhere.

3.3.2 Interview

In addition to the document review, ***semi-structured*** interviews, allowing acquisition of more detailed information, will be employed. Semi-structured interviews will be utilised because they allow modification of the set of pre-defined questions in accordance with previously received responses while still making sure that the discussion covers the key points. Another advantage of semi-structured interviews is a higher response rate, than, for example, in the surveys (Gill 2010).

For this research, respondents will be selected on the basis of the snowball sample: either by recommendation of the previous respondent, or by getting an introduction from the manager of a department. With regard to the sample size, there is no predefined number of participants and it depends on the sufficiency of obtained information, stage of the research and availability of the personnel.

The disadvantage of this method is it is very time consuming for both interviewer and interviewee (Martin 1995). Also interview techniques have always been

associated with subjectivity (Gill 2010). However, in the context of this research the level of subjectivity is predicted to be relatively low since the questions are related to the technical aspects and do not involve personal attitudes and emotions. Another issue with the interview, more common in complex research environments, is that sometimes the managers "*do not want to discourage the researchers by telling all the difficulties at once*" (Wren et al. 2003). Potentially this issue can lead to a researcher designing an inappropriate model and solution. The observation will be carried out in order to mitigate this issue and support already collected data.

3.3.3 Observation

Since during the interviews users might neither be able to explain complex scheduling processes, nor recall all possible circumstances, the observation will complement the information gathered through interviews and validate the researcher's understanding of the process (Roth 2012). The advantage of the observation is that it can reveal the information that the participant omits or takes for granted (Sapsford and Jupp 2006). In terms of information system research, Rees (1992, p.22) states that observation is a very important technique “as watching users working with the system provides more in-depth information than questionnaires or interviews”.

However, observations might affect operations and distract the participants from their work. In addition, observations conducted only a few times might not be a full representation of a typical day (Martin 1995). Moreover, Sapsford and Jupp (2006) state that it is possible to make some inaccuracies in interpretation as all the obtained information still goes through the "prism" of researcher’s understanding and experience.

With regard to this research, during the observation the participant will be asked to build a schedule as they normally do. The schedulers will be encouraged to describe and comment on the processes as they go through the session. In order to avoid any misunderstanding or misinterpretation, the researcher will be asking clarifying questions and summarise the recorded data at the end of the observation.

3.3.4 Problem modelling

Models allow representation of the collected information in a structured and systematic way. A well-devised model can not only reveal clear sequences and the relationship between operations, but also increase the understanding of the complex processes and expose process inefficiencies (Slack 2013, Holt 2009, Becker, Kugeler and Rosemann 2011). However, the limitation of models is they are only an abstract version of real systems and might not address some of real life aspects (Jensen 2003, Becker, Kugeler and Rosemann 2011).

Several process modelling techniques such as flow chart, swim-lane diagram and mathematical models will be employed in this study to describe scheduling processes.

Flow-charts offer a visual illustration of a process and operations within it (Slack 2013). Flow chart enables displaying the process in a form of symbols that is easy to follow and interpret. One of the main challenges is the decision from what level the process should be modelled: a too detailed model can overwhelm with complexity, whereas a simplified model can omit important details (Holt 2009, Becker, Kugeler and Rosemann 2011).

Swim lane diagrams are suitable for the process involving different departments or stakeholders (Slack 2013). The benefit of a swim lane diagram is that it shows responsibilities and functions of each operation owner. Like with a flow-chart, the determination on the adequate level of representation is crucial for accurate process definition.

Mathematical model is "*a tool designed to help solve managerial, planning, and design problems in which the decision maker must allocate scarce (or limited) resources among various activities to optimize a measurable goal*" (Ruhul,Sarker and Newton 2007). There are two types of mathematical models: deterministic and ***stochastic*** (Jensen 2003, Becker, Kugeler and Rosemann 2011). Deterministic models are suitable for representation of the situation where the data and functional relationship between variables are known in advance, whereas stochastic models address the uncertainty of the process in which certain events might occur with some probability.

Ruhul et al (2007) argue that a well-devised mathematical model should meet the following criteria:

- Robust: Should be applicable for all types of potential input variables.
- Adaptive: the changes in the process should be easily incorporated.
- Complete: should contain a sufficient amount of detail.
- User friendly: should be understandable by various users.

3.3.5 Prototyping

The approach "prototype-test-refine" suggested by Hopgood (2012) will be utilised in this research. It starts from the formulation of requirements, followed by designing the general functions, and then obtaining user feedback. The cycle is repeated until the system achieves user expectations and can be implemented. Unlike the traditional "waterfall" model it gives an opportunity for continuous improvement and error corrections at the early stages (Hopgood 2012).

The main purpose of the prototype is demonstration of the algorithm capabilities to the industrial experts. Tangible demonstration allows for avoiding any misunderstanding or misinterpretations. Moreover, Jensen (2003, p.656) states that "v*isualisalisation adds impact to simulation output and often enhances credibility*". It also allows to "*convince sceptic users in its effectiveness and feasibility*" (Montana, Talib and Vidaver 2007) as well as receive immediate feedback from the users (Bocij 2015).

Although the full system implementation could achieve the same objectives, the preferences were given for prototyping rather than the full system implementation for the following reasons (Avison 2006):

- Research deals with core organizational processes, full system implementation will have a high impact on organizational performance, which is risky
- Prototype can allow demonstration of the necessary functions
- Less programming efforts and less time needed to produce a prototype
- Does not require significant financial investments

- Prototypes and experimental simulation provides a means for consideration of different scenarios before their implementation (Jensen 2003)

Prior to expert demonstration, prototype verification and validation will be performed. ***Verification*** is "*a test of design to ensure that the design chosen is the best available and that is error free"* (Bocij 2015, p.396). With the aim of identification of the most effective configuration of the algorithm, several trials will be conducted. Moreover, the correctness of the logic will be examined by performing manual calculations and debugging (Jensen 2003). The test cases will be designed in a way that addresses and evaluates each rule. In addition, a test case that attests all of the rules and trade-off amongst the results will also be executed. According to Montana (2002), it is important to use the real-data set to check the system even though the optimal results might be unknown. Therefore, the algorithm and its behaviour will be studied using both test benchmark data and the real data.

Validation is *"the test of design where we check that the design fulfils the requirements of the business users which are defined in the requirements specification"* (Bocij 2015, p.396). Historical validation where the model output is compared against historical data is deemed to be one of the most commonly used validation methods (Jensen 2003). However, this approach might not be applicable for this research as the automatically produced schedule might differ from the schedule constructed manually, and therefore the expert evaluations, described in the section 3.5, will be conducted instead.

3.4 Algorithm parameter selection

Selected optimisation technique determines the quality of the results and computation time. However, in many cases it is impossible to find the best techniques and its parameters analytically. A series of experiments need to be carried out to discover them empirically.

Experiment is a "*test or a series of runs in which purposeful changes are made to the input variables of a process or a system so that we may observe and identify the reasons for changes that may be observed in the output response*" (Montgomery 2013, p1).

According to Robinson (1994) there are two types of experiments: ***interactive*** and ***batch experiments***. Interactive experiments require the researcher to observe the entire process and make manipulations in the course of the run if required. Batch experiments do not involve the researcher in the process and can perform multiple runs without the need for parameter resetting or process observation. The given research will be based on a series of batch experiments since parameters do not need to be adjusted during algorithm execution and a run can take several hours. All the information regarding algorithm behaviour will be automatically recorded and documented, freeing up the researcher from observation and providing more detailed recording of the algorithm.

One of the key questions related to the batch experiment design is the number of repetitions. Although it is possible to improve accuracy by increasing the number of replications of experiments, there is still no guarantee that the system behaviour will remain similar to what has been demonstrated previously if the experiment will be repeated one more time (Gill 2010,Jensen 2003). In order to achieve a compromise between the amount of replications and time allocated to the experiments, each test case will be executed ten times such as in Liu and Sun (2011), Wanner (2007), Pinto, Ainbinder and Rabinowitz (2009), Wang and Wu (2010), Patel and Padhiyar (2015).

The experimental settings for both problems are defined in section 8.4.

3.5 Algorithm evaluation and business impact assessment

3.5.1 Cost-benefit analysis

Cost-benefit analysis is the "*process of comparing the various costs of acquiring and implementation of IS against the benefits which the organisation derives from the use of the system"* (Remenyi, Money and Twite 1991, p.96). Cost-benefit analysis regards both qualitative and quantitative factors, and that is why this type of analysis is appropriate for a new system evaluation. While it is relatively simple to quantify the cost, the estimation of the IT system benefits is a challenging task. Depending on the role the IS plays in the organisation, Remenyi, Money and Twite (1991) identify the following types of benefits:

Cost displacement approach compares the cost of the investments against the cost the system has saved. It is usually applied in the situations where technology

replaces workers. Although it provides clear financial benefits, this method might not be applicable when the role of the IT system is to add value rather than to reduce the cost.

Impact or time release analysis considers the enhancement of an employee's productivity by measuring the opportunities which might arise from freeing up the employees' time. This can be the involvement of employees in other projects or improved customer relationships which might result in increasing sales.

Unlike cost displacement, ***cost avoidance*** is used when no reduction in the current cost can be achieved, but the introduction of an IS can prevent occurrence of additional cost.

Decision analysis is applied when the responsibility of the IS system is to assist in the decision making process. In theory, this is normally measured as the correlation with organisational performance. However, because the performance is the aggregation of numerous actions and decisions, it might be challenging to establish the pure impact of the IS suggested decisions (Remenyi, Money and Twite 1991).

Nominal breakeven analysis is suitable when the benefits of IS are intangible and are hard to specify. In this case, the employees are asked how much the various services offered by the IS are worth to them. On that basis the decision about the investment into the new IS is taken. It is important to note that it might be quite difficult for a user to put a figure to express the support of an IT system. Furthermore, the opinion might be quite subjective and considerably vary from one user to another.

Decision analysis and cost avoidance approaches will be adopted in this study because the algorithm is designed to improve the scheduling process and will likely reduce the total cost of the schedule. The time release analysis was rejected in this research due to a large number of staff and scarce information regarding the potential tasks they can perform. Although this study considers business process automation, cost displacement analysis is not appropriate as we assume that the number of staff will remain the same. As demonstrated in the following section, some of the benefits of an automated scheduling system are

not possible to quantify, so nominal break even analysis has been rejected as well.

3.5.2 Metrics

Selection of the appropriate metrics is an important decision as the value of the information system, its capabilities and functionality will be judged based on the chosen indicators. There are number of parameters and indicators suggested in the literature which can be utilised for the information system evaluation.

For instance, Hamilton and Chervany (1981) were among the first researchers who provided a collection of indicators for appraisal of information system performance. In general, they can be divided into two broad categories: efficiency-orientated and effectiveness-orientated indicators (Table 6).

Table 6. MIS performance measures

Efficiency-orientated	Effectiveness-orientated
• Requirements definition (compliance to the specification) • Resource consumption (budget, staff) • Production capability (productivity, response time) • Level of investments in the resources (capital expenditure, hardware)	• Information and support provided (response time, quality of information, level of user friendly interface) • User process and user performance (contribution of IS to decision making process) • Organization performance (sales, profit, market share)

Adapted from: Hamilton and Chervany (1981)

While Hamilton and Chervany (1981) state that the first group of indicators is more popular than the second, it might be argued that efficient-orientated indicators on their own do not provide a full picture about the quality of an information system. Likewise, although effectiveness-orientated indicators provide a low level information about IS performance, they do not convey key investment and financial figures which might be of interest to key decision makers.

Several scholars designed indicators specifically for expert and intelligent systems. For example, Sawka (2000) suggests calculating the contribution of IT system in the decision making process as the benefits of those decisions (Luz, Oscar and Claudia 2010). Marin and Poulter (2004) expressed the decision process in financial terms by comparing the cost for consultancy against the cost

of information system (Luz, Oscar and Claudia 2010). The limitation of this approach is it could be very problematic to estimate the actual contribution of an IT system towards the benefits of the decision, since other factors could also contribute to the resulting figures (Remenyi, Money and Twite 1991).

Davison (2001) suggests examining the level of user confidence and the percentage of decisions accepted by users. The disadvantage of this approach is a high level of subjectivity as results might vary from one user to another depending on their personal preferences and experience. Rees (1992) states that users with a lack of experience tend to rely more on the expert system decision, whereas users with a vast experience in the process tend to consult less with the system.

Since each of the analysed methods has its own advantages and disadvantages, a collection of indicators will be used in this research to provide a well-rounded picture about the efficiency and effectiveness of the designed automatic scheduler. These indicators are discussed below.

3.5.3 Operational level

It is important to measure the impact the information system has on the operations, as "*first-order impact of IT investments occur at the process level*" (Tallon, Kraemer and Gurbaxani 2000, p.149). Although there is a vast amount of literature dedicated to improving scheduling mechanisms and measuring organisational performance, much less attention has been paid to measuring the performance of the scheduling operations and defining tailored metrics (De Snoo, Van Wezel and Jorna 2011). For this reason, the standard process performance indicators such as cost, speed, dependability and flexibility, will be defined and applied.

Cost

To date the most common measurement is the total cost of the schedule (Ozdemir and Mohan 2001, Abbink et al. 2011, Azadeh et al. 2013) in the crew scheduling problem and time in the job-shop scheduling problem (Spanos et al. 2014, Chaudhry 2012, Hui 2012, Meeran and Morshed 2012, Qing-dao-er-ji and Wang 2012, Raeesi and Kobti 2012, Thamilselvan and Balasubramanie 2012).

In addition, the opportunity cost defining the revenue from potentially accepting more jobs, which could be taken due to freeing up capacities, will be estimated.

Dependability

Dependability is the ability to fulfil the customer orders on time and according to the negotiated specification (Jones 2012). The dependability of the algorithm will be proven by demonstrating that none of the required jobs are missing in the schedule and there is enough available equipment and staff to perform them.

Quality

High quality operations decrease the cost and increases dependability due to fewer errors and rework (Slack 2013). The quality of the solution will be judged by the industrial experts during the focus group discussion.

This information will be obtained from the focus group solution evaluation and responses.

The evident advantages of this method are that it is not time consuming for both schedulers and the researcher, the data can be quantitatively analysed and modelled. The limitation of this method is the predefined amount of questions and answers, which might restrict the respondents in providing a detailed answer. Therefore, if a user would like to provide additional information with regard to the quality of the diagrams, they will be able to use the space at the end of the questionnaire to do this.

Speed

By performing each operation faster, it is possible to perform more tasks in the same time frame. This would mean that the schedulers would free up some time and assist with other tasks or would have sufficient time to consider different schedule alternatives and chose the best (Kwan 2011). The speed advantage will be measured as the difference between the manual hours to produce a schedule from the certain number of tasks and execution time of the algorithm to produce a schedule from the same tasks.

Flexibility

Flexibility is the ability of the organisation to adjust their processes according to the change in customer orders (Jones 2012). Flexibility increases robustness of operations and raises their dependability (Slack 2013). In the context of scheduling, it is the ability of an organisation to either fit the last-minute schedule or to produce a completely new plan.

A special session devoted to evaluation of the quality of the diagrams with the schedulers will be organised. The primary objective of the session is to determine the extent to which the designed system accommodates business processes and meets industrial regulations. This will be accomplished through ***the focus group method***, in which the schedulers will be asked to discuss a series of questions addressing different aspects of the diagram quality. The plan of the focus group discussion is presented in the Appendix 1.

Unlike surveys, this method is able to provide not only the information about the quality of the diagrams, but also an explanation as to why a particular score was given. In contrast to single participant interviews, focus groups allow multiple experts' opinions to be obtained and then cross validated by the expert themselves rather than the researcher who does the analysis after all the interviews based on his/her understanding.

However, the drawback of this method is that the flow of discussion and the obtained results can be influenced by a number of psychological and social factors such as the domination of one member, unwillingness to express ideas in front of the group and a fear of public speaking (Klein 2003). To minimise the impact of these factors, the researcher will ensure that each member of the group has an opportunity to express his or her opinion and have enough time to do so.

3.5.4 Strategic Level

To enable effective performance of the organisation, IS should support the overall organisation strategy. In order to evaluate the degree of alignment, the key strategic priorities will be studied and the extent to which the automated crew scheduler contributes to them will be identified.

3.5.5 Investment Evaluation

As investment in IS constitutes a large portion of an organisation's capital, the investment decision should be carefully evaluated (Hallikainen, Kivijarvi and Nurmimaki 2002). After the pieces of schedule have been confirmed to be practical, the classic financial indicators such as ROI, NPV, PI and IRR will be computed (Bocij 2015, Remenyi, Money and Twite 1991).

ROI (return on investment) represents the efficiency of investment and is computed based on the formula below.

Formula 4

$$ROI = \frac{Project\ Benefits}{Investment\ Amount}$$

Payback period refers to the amount of time it takes to recoup initial investments. The limitation of both indicators is that they do not consider the time value of the money. In order to overcome this, the NPV value will be determined as well (Atrill 2014).

Net present value (NPV) determines the profitability of the investment project. It compares the cost of investment with the benefits in the form of cost savings over a period of time (Atrill 2014).

Formula 5

$$NPV = \sum_{t=1}^{T} \frac{C_t}{(1+r)^t} - C_0$$

where:

C_0 is the cost of new system.

C_t cost saved in the period *t*.

r is the discount rate, that is the interest rate.

PI (Profitability Index) shows how much money will be earned (saved) per one pound invested.

Formula 6

$$PI = \frac{\sum Present\ Value\ of\ Benefits}{Present\ Value\ of\ Investment}$$

Internal rate of return (IRR) denotes the discounted rate at which NPV is equal to zero. It can be found from the equation displayed below. The higher the IRR, the more desirable the project is.

Formula 7

$$\sum_{t=1}^{T} \frac{C_t}{(1 + IRR)^t} = 0$$

Despite these tools being widely used by investors, they can only give an idea of worthiness of the investment project, rather than a precise number of future financial results. This is because the cash flow is usually based on forecasted values, and in reality can be lower than expected. In addition, the interest rate which depends on inflation might slightly fluctuate as well (Atrill 2014). In this research it will be assumed that the cost savings produced by the IS system will remain the same over its life-cycle period and the interest rate will remain stable.

3.6 Conclusion

The chapter has presented the methodological framework which will be used to guide the research and allow for achieving the objectives. The techniques for data collection, data analysis, model optimisation and algorithm evaluation have been described. Moreover, each technique had been analysed and compared with the similar methods to ensure the most suitable and effective technique is selected and will be applied in this research.

The next chapters will provide an insight into two domains, in which the research will take place.

Chapter 4. Job-shop Scheduling Problem in the Printing Industry

4.1 Introduction

This chapter provides insight into the printing industry and the key processes of handling customer orders. This information derives from the numerous interviews conducted with the managers at Garnett-Dickinson. After detailed consideration of the key operations and relationships between them, the problem is categorised to the relevant class of job shop scheduling problems and the formal mathematical model, which is necessary for the optimisation, is then designed.

4.2 Printing industry overview

The commercial printing industry has changed dramatically over time. Increased competition from digital technology, the internet as a way of spreading the news and advertising, forces printing companies to increase their effectiveness and provide an outstanding service exceeding customers' expectations. Wide availability of printing technology and relatively low entry barriers shift competition in the industry towards price and quality (Datamonitor 2012). Companies usually differentiate themselves and gain competitive advantages by adding other value-added services (Datamonitor 2012).

Customers' expectations of the order fulfilment time fell from 7 to 3-4 days. They also became more demanding in terms of the quality. There is a high variety of products. Each customer requires a unique publication on a specific type of paper and paper size, using different stitching and folded in a certain way. In terms of volume, medium-size companies, such as Garnett Dickinson, do not accept orders of less than 5000 copies, because it is not economically efficient due to fixed set-up cost (Garnett-Dickinson Sales Manager interviewed on 23/04/2013).

Despite the fact that the demand is relatively predictable, there is a considerable surge around the Christmas time period. The demand is based on regular customer orders, which require periodic publications, as well as less expected publications on a single occasion.

While the demand has moderate certainty, the printing process can still be characterised as a high volume and high variations process that makes scheduling operations particularly complicated.

4.3 Job shop scheduling operations in the printing industry

The business process map of handling customer orders is illustrated in Figure 18 and is described below (Garnett-Dickinson Operations Manager interviewed on 23/04/2013).

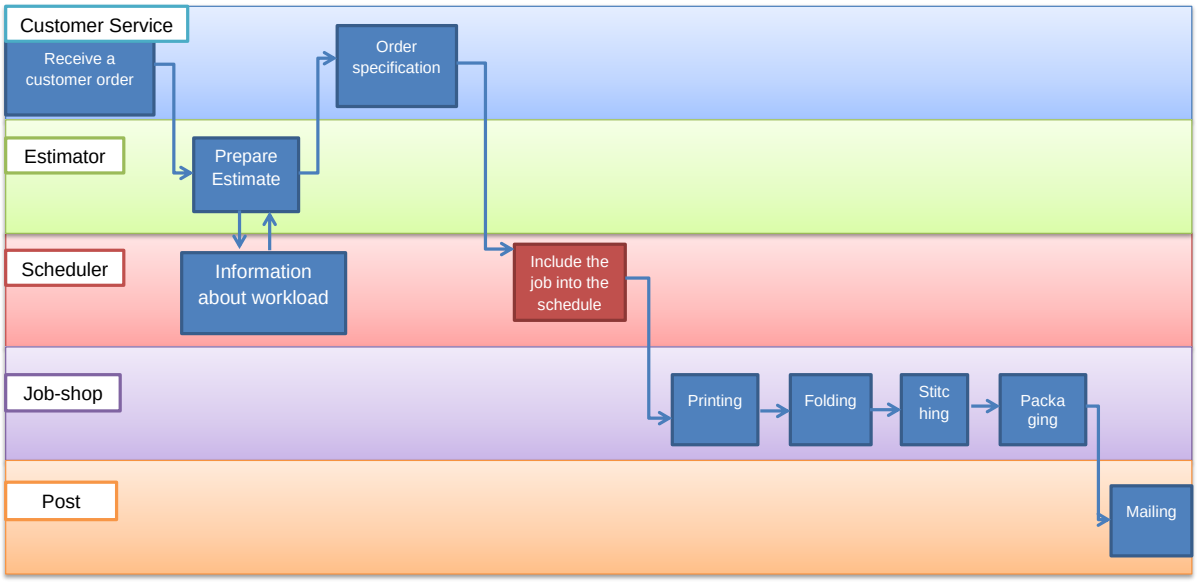


Figure 18 Key business operations in the printing industry

4.4 Estimator

The process starts with taking an order from a customer. At the beginning it is just a general description of what the customer would like to see as a result. Then the task goes to the team of estimators, who suggests different options with regard to paper type and quality, size of the page, finishing style, delivery, destination, packing and the potential cost. Evaluating the cost, the estimator carefully builds a production plan and calculates the amount of required consumables as well as assessing the availability of the printing presses.

The total cost of publication consists of materials, equipment, labour and facilities. The customer usually sees only materials in the estimate (Garnett-Dickinson Estimator interviewed on 23/04/2013). The expected time is calculated on the basis of production speed, which in turn depends on the type of paper (weight

and size), type of fold (single, right angle and etc.). In addition, the lead time depends on other customers' orders, availability of the equipment and processing routes (Garnett-Dickinson Estimator interviewed on 23/04/2013).

Where a customer would like to have publication completed in a shorter timeframe, the estimator contacts the planner to discuss whether there is room to insert the job in the current production plan or identifies if there are other process routes (i.e. by using different machines). Usually it means an increase in the price as well. Once the price has been confirmed, the customer's history is revised and a simple credit check is performed. Upon success the order is put on the plate which divides the colours into magenta, green and cyan. After that the order is forwarded to the planner on the job floor.

4.5 Planner

The planner on the job shop floor deals with low level day to day scheduling problems. He or she breaks the task into operations and assigns them to the corresponding machines. The computer software automatically calculates how much time it will take to perform a certain operation and what resources are needed (Garnett-Dickinson Operations Manager interviewed on 23/04/2013). The planner also, looking at the information provided, collates it with due dates, and then draws operations manually in the Gantt chart (Appendix Appendix 2). The Gantt chart displays the order of operations and the software communicates this to the printing and binding machines. The software checks the sequences of operations and the availability of resources (mostly paper) and displays a warning message if it is impossible to accomplish the task due to the lack of resources or operational precedence constraints violation (i.e. the stitching operation before printing). After determination of the operations, the planner adds a set-up time to the total time of the operation. Set-up time is the time necessary to clean the machine after the previous operation and to load the machine for the succeeding one.

4.6 Job-floor

Appendix 3 shows the pictures taken from the job floor in Garnett-Dickinson. In general, the publication's production route follows five stages: printing, folding, stitching, packing and mailing. Various specialised machines are used to perform

these operations. The machines monitor the work progress, estimate completion time and average speed which are reported back to the scheduler's computer, so he or she would be aware of the availability of the machines and any delays on the job floor (Garnett-Dickinson Operations Manager interviewed on 23/04/2013, Garnett-Dickinson Machine Operator interviewed on 23/04/2013). Machines differ on the basis of type of resources they can handle.

Quality checks are performed during various stages of the process in the form of examination of the random sample (Garnett-Dickinson Operations Manager interviewed on 23/04/2013, Garnett-Dickinson Machine Operator interviewed on 23/04/2013). Although, Garnett-Dickinson sets high quality standards and has ISO accreditation, minor imperfections might occur on the publication. It might be caused by allocation of not enough time for the ink to dry (folding too early) or the glue might leave marks on certain pages. In order to minimise the risk associated with defects and delays, companies produce one percent of extra copies.

4.7 Mailing

Garnett-Dickinson also provides mailing services. Only a sample goes back to the client, while the rest of the publications are inserted into the envelopes and sent directly to clients' clients (Garnett-Dickinson Operations Manager interviewed on 23/04/2013). This allows the company to minimise the finished goods inventory, since the completed jobs leave the facilities without waiting for the client to collect them.

4.8 Importance of the scheduling operations

Clearly the effective scheduling operations can reduce the lead time, increase customer satisfaction and retention. On the other hand, a poorly constructed schedule might increase customer waiting time and cause delays. As a result, businesses might encounter additional expenses such as cost associated with communicating the problem to the customer (paperwork, telephone calls, managers' time); cost for extra set-up times to move the job quicker through the job floor; penalty in a contract or discount; and, the less visible one, lost opportunities (Gere 1966).

This problem relates to the class of job shop scheduling problems and various operation research techniques have been introduced for modelling and finding an effective solution.

4.9 Job-shop Scheduling

Job shop scheduling problem is very common problem in the manufacturing environment (Pinedo 2009). It can be loosely defined as an assignment of jobs to the machines in compliance with all operational constraints.

4.10 Performance indicators

Pinedo (2009) states two dimensions in which the quality of the schedule can be measured. In the first one, the quality of the schedule can be expressed using financial equivalents such as work-in-progress inventory cost, finished-goods inventory cost, cost for the set-up times, utility cost. The second aspect is the time-related parameters such as tardiness, deadline satisfaction, throughput time of a particular job.

The most wide-spread measure of schedule quality is the ***makespan***, which is a total time from the beginning of the first operation to the completion time of the last one. It has been adopted in this study due to availability of benchmark data and results for testing and evaluation.

4.11 Problem modelling and formulation

The given problem relates to the classic JSSP with sequence dependent set-up times. Formalising the description of the Garnett-Dickinson case the main constraints of JSSP can be formulated as:

- The sequence of operations in each job is predefined.
- All operations must be processed without interruptions (once started it should be finished).
- Different operations of the same job cannot be performed simultaneously.
- One machine can process only one operation at a time.
- One job can be processed only at one machine at a time.
- One machine cannot process more than one job at a time.
- One job cannot be processed at more than one machine.

- The set-up times are sequence dependant and are not included in the processing time.
- All the resources (people, consumables) are available.
- All machines are available in the beginning.

4.12 Formal definition of the job shop scheduling problem

Let's assume that a company has a set of machines M={m_1, m_2…m_l} and a set of jobs J={j_1, j_2…j_n}, which need to be assigned to machines in the optimal order and satisfying all production requirements. Each job consists of several operations O_i={O_{i1}, O_{i2}…O_{inm}}. Moreover, each operation has its own processing time, which is T_i={t_{i1}, t_{i2}…t_{imn}}. Since the operations of the same job need to be performed in a specific order (i.e. the folding cannot be done before printing), matrix A defines binary relationship for each operation in O. If (*v*, *w*) €A, then it means that operation *v* should be performed before operation w. S(*v*) denotes a start time for each operation.

Therefore, the objective is to find an optimal schedule with the minimal makespan (Formula 8.1):

Formula 8.1

$$len(S) = \max_{v \in O}(S(v) + \tau(v))$$

Which is a subject to feasibility constraints (Formula 8.2-Formula 8.4):

Formula 8.2

$$\forall v \in O: \qquad S(v) \geq 0$$

This inequality denotes that the start time for all operations should be non-negative.

Formula 8.3

$$\forall v, w \in O, (v, w) \in A: \qquad S(v) + \tau(v) \leq S(w)$$

This constraint requires the previous operation to be completed before the subsequent operation begins.

Formula 8.4

$$\forall v, w \in O, v \neq w, M(v) = M(w): \; S(v) + \tau(v) \leq S(w) \; or \; S(w) + \tau(w) \leq S(v)$$

The inequality three ensures that each machine will not start processing another job until the current job is completed.

4.13 Disjunctive graph representation classic JSSP

Given the graph G= (V, A, E, I). The nodes represent the operations of the jobs (Figure 19). Nodes 0 and 1 are called dummy nodes since they do not consume any time and only indicate the beginning and end of the schedule. The set of arcs A reflects the precedence operations constraints for each job. These arcs are directed and called ***conjunctive arcs*.** The other sets of undirected, disjunctive arcs, shows the operations which should be processed on the same machine. The task is to determine the direction of the disjunctive arcs with the objective of minimisation of the length of the makespan.

Definition 7

Makespan is the ***critical path*** on graph G. Critical path is the longest path starting from the 0 node and ending at node 1. Any operation on the critical path is called *a* ***critical operation***. The critical operation cannot be delayed without increasing the makespan.

Definition 8

The ***critical block*** is a subsequence of operations belonging to the critical path and utilising the same production facilities (i.e. machines).

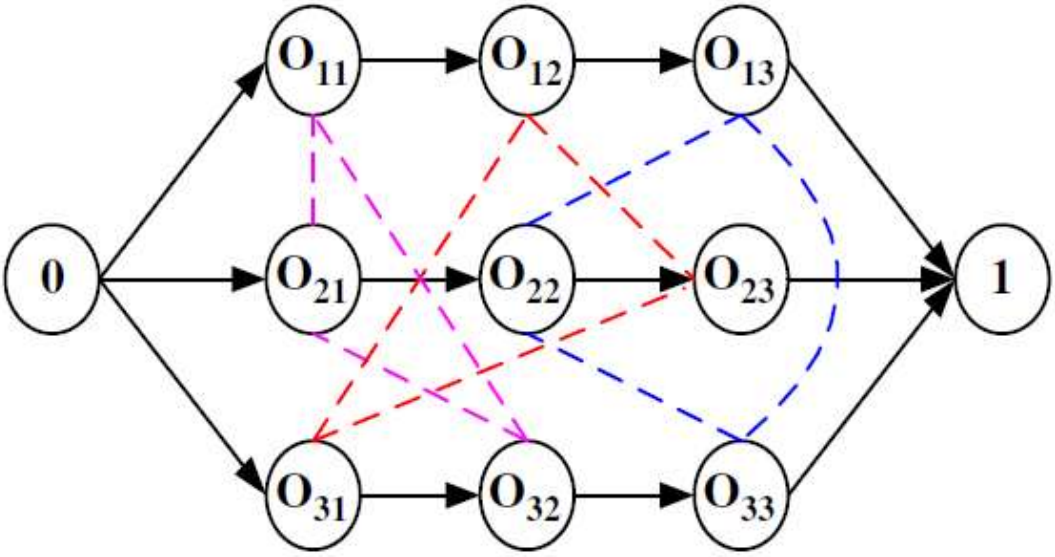


Figure 19 Disjunctive graph representation of JSSP

Definition 9

The schedule is called ***active*** if neither of the operations can be started earlier without delaying other operations.

Definition 10

The ***optimal schedule*** is a feasible schedule in which the maximum interval between any two operations is minimal (Giffler and Thompson 1960, p.489). An optimal schedule is an active schedule. A set of active schedules is much smaller than a set of all feasible schedules (Giffler and Thompson 1960).

4.14 Conclusion

This chapter has described the scheduling processes in the printing industry. These operations have a high degree of variation due to diverse customer preferences, large volumes of publications and very short leading times dictated by customer expectations. Technical constraints and job specifications add more complexity to printing processes. Therefore, it is important to have a software, which is able to build a schedule automatically, or at least to provide a scheduler with an initial solution.

This chapter has established that this problem belongs to the Job Shop Scheduling class and provided mathematical model of this problem. The solution methods for this model, their advantages and limitations are discussed in the next chapter.

Chapter 5. Approaches to Job Shop Scheduling Problem

5.1 Introduction

As established in the previous chapter, the scheduling operations, which Garnett-Dickinson performs, relate to the class of Classical Job Shop Scheduling Problems. The main goal in these operations is to minimise the amount of time required to complete all the jobs, while still satisfying all the production requirements such as the precedent relationship between the operations of the same jobs and assignment of the operations to the correct printing press.

Being one of the most studied problems that has been attracting research attention for over fifty years, JSSP is still not fully resolved (Meeran and Morshed 2012). The following chapter provides the overview of the key algorithms developed to tackle this problem with the main focus on EA, as it has been selected for this research in Chapter 2. After the analysis of various chromosome representations and the corresponding genetic operators developed and utilised in previous research, it reveals gaps in the literature, which will be fullfilled in this research in Chapter 8 and Chapter 9.

5.2 Dispatching Rules

Dispatching rules are one of the most straightforward and first developed methods for attacking JSSP (Gere 1966). The basic dispatching rules are listed below.

- SPT (shortest processing time).
- LPT (longest processing time).
- MWR (Most work remaining - the total processing time of the remaining operations).
- MOR (Most Operations Remain).
- LOR (Least Operations remaining).
- EDD (Earliest due date).
- FCFS (First come, first serve).

The dispatching rules can be applied as follows. All the operations are pre-sorted with respect to the selected prioritization scheme and then they are assigned one-by-one to the machines in the obtained sequence. The advantage of this technique is the ease of implementation and fast execution. However, despite its simplicity, this approach can rarely build an optimal schedule. This is because there is no mechanism which would identify the gaps where some of the succeeding operations can be squeezed in, and thereby would reduce the total completion time.

Priority rules are proven to be effective when they are used as a part of another algorithm. In terms of EA for JSSP, priority rules are often embedded in the chromosome generation process or into the local search operator. Usage of priority rules enables EA to achieve better results compared to the algorithm alone (Mattfeld and Bierwirth 2004, Essafi, Mati and Dauzere-Peres 2008, Zhou, Cheung and Leung 2009, Yang et al. 2012).

5.3 Giffler and Thomson algorithm

The Giffler and Thomson algorithm allows an active schedule to be obtained from the feasible schedule by manipulating the operations lying on the critical path (Giffler and Thompson 1960). It starts with the first operations of each job and assigns them to the relevant machines. Then it takes the second operation and checks if there are any conflicts. According to the definition given by Giffler and Thompson (1960) the ***conflict*** is the situation when the operations assigned to the same machine overlap. In order to resolve the conflict, the successive operation is shifted to the right-hand side on a production plan. This means the completion time of the next operation is the sum of the completion time of the previous operation and duration of the current operation.

This concept can also be extended to the JSSP with precedence constraints where the subsequent operation cannot start until the previous operation of the same job is finished. In this case the starting time is the maximum time between the time when the machine becomes available and the preceding operation of the same job is completed. Likewise, the completion time of the given operation and machine is equal to the starting time plus the duration of the operation.

The Giffler and Thomson algorithm is often incorporated in contemporary algorithms. This is because it enables conversion of any feasible schedule into the active one. This, in turn, greatly reduces the search space and transfers the search to the region where the optimal solution is located. In terms of EA, depending on the chromosome representation, it can be utilised as a decoding procedure, local search operator or feasibility restoring operator (Cheng, Gen and Tsujimura 1999, Gao, Sun and Gen 2008).

5.4 Branch and bound (B&B)

Unlike the aforementioned techniques, in theory B&B is able to solve JSSP to the optimality. However, for real-life problems it requires an enormous computational time that hinders its practical application. Brucker, Jurisch and Sievers (1994) were one of the first researchers who applied this method for the solution of the 10x10 JSSP. Later several enhancements that were able to slightly accelerate the speed of the algorithm were proposed. For instance, Nababan, et al. (2008) combined B&B with a disjunctive programming approach, that enabled B&B to solve 50X20 problems for less than 20 minutes. Liaw (2013) incorporated various heuristics into B&B and obtained a solution of 14 x14 pre-emptive open shop problem for the reasonable amount of time. However, because in reality the size of the JSSP can reach hundreds of jobs and tens of machines, B&B is rarely used to solve JSSP in real life (Lei 2009).

Since in real world applications the optimal solution is not always the main objective, ***beam search strategy*** can be applied to prune unpromising branches earlier in the process (Pinedo 2009). The adaptation of beam search to the JSSP works according to the following principle. Beam search starts from the generation and the evaluation of several schedules. Then, it selects only *w* (beam width) best of them for further branching. To make the restriction of branches even tighter, the additional value *f* (filter width) is defined. Filter width denotes how many branches will be obtained from the current branch. Clearly, the performance will depend on the value on the algorithmic parameters *f* and *w*. Having very low values, it is possible to achieve reduction in the computation time, although the branch which can lead to the optimal solution can be pruned as well. Alternatively, examination of too many branches can be computationally expensive. Thus, like in the majority of the optimisation algorithms, a trade-off

between the solution quality and computing time needs to be experimentally identified.

5.5 Metaheuristic algorithms

Metaheuristic algorithms gain more and more popularity for the solution of JSSP (Abdullah and Abdolrazzagh-Nezhad 2014). Metaheuristic methods can offer a quick solution, which is presummably close to the optimal one, for a reasonable computation time.

Abdullah and Abdolrazzagh-Nezhad (2014) conduct a survey of the popularity of metaheuristic algorithms for the solution of fuzzy JSSP. Their results are presented in Figure 20. Their findings support the selection of the algorithm for this research. The EA appeared as the most popular one followed by Tabu Search, Simulated Annealing and Ant Colony optimisation. These results are similar to the survey results conducted by Lei (2009), Gen and Lin (2014).

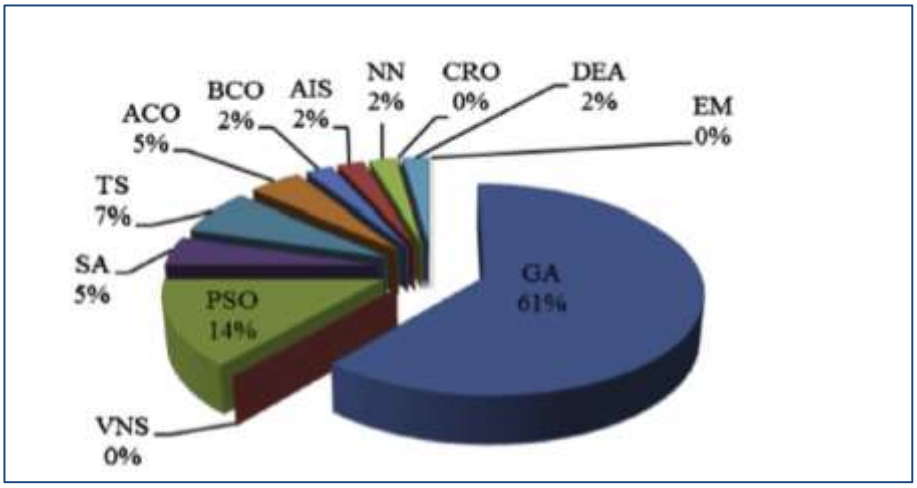


Figure 20 Methods for the solution of JSSP

Source: Abdullah and Abdolrazzagh-Nezhad (2014)

Despite each of these methods having its own advantages and disadvantages and being able to outperform one another in different problems and comparison settings, the majority of researchers agree that the utilisation of a combination of methods can yield much better results than each algorithm on its own (Meeran and Morshed 2012, Spanos et al. 2014, Zhang, Gao and Li 2013, Javadi and Hasanzadeh 2012, Qing-dao-er-ji and Wang 2012, Rakkiannan and Palanisamy

2012). Thus, the application of EA and its hybrids will be considered in great detail in the following section.

5.6 Evolutionary algorithms

5.6.1 Chromosome representations

Various chromosome representations have been designed for the solution of the job-shop scheduling problem. Generally, there are two types of representation: ***direct*** and ***indirect***. Direct chromosome representation encodes the schedule itself, while indirect contains only the rules of schedule deduction. Although the advantage of direct chromosome representation is that the same solution cannot be obtained from different chromosomes, the major disadvantage is that it requires the development of specific genetic operators. The opposite is true for indirect chromosome representation. Although the implementation of the direct encoding is more simple, Corne and Ogden demonstrated that indirect encoding more superior (Hart, Ross and Corne 2005).

In order to implement both representations, an additional procedure, ***schedule builder*****,** should be developed and tailored to both chromosome representations in order to solve JSSP. The role of the schedule builder is to translate the chromosome into a feasible and user-friendly schedule format, which would allow calculation of the makespan and other parameters. The relationship between the schedule builder and chromosome representation is as follows: the simpler the chromosome representation, the higher the burden on the schedule builder and vice versa (Cheng, Gen and Tsujimura 1999).

Hart, Ross and Corne (2005) classify all the existing chromosome representations for the JSSP. The results are presented in Table 7.

Table 7 Chromosome representation of JSSP

Direct representation	Indirect representation
• Operation based • Job based • Job-pair relationship based • Completion-time based • Random-keys	• Preference-list based • Priority-rules based • Disjunctive graph based • Machine based

Source: Hart, Ross and Corne (2005)

Each type and the corresponding schedule builder procedure will be analysed in the next section.

Job-based representation In this type of representation a chromosome consists of a string of integers that represent each job waiting to be scheduled. The example of such representation is presented in Figure 21.

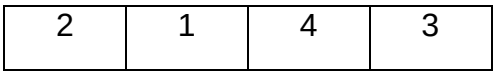


Figure 21 Job-based chromosome representation

There are two major ways of deducing the schedule. The first one is to assign all the operations of the job represented by the first gene, then assign all operation of the job standing at second locus, and repeat this procedure until the end of the chromosome is reached (Jianchao Tang, et al. 2010). Another way to decode this chromosome is to assign first operations of all the jobs in the order they appear in the chromosome, then to assign all the second operations in the same order and so on until all operations are assigned (Amirthagadeswaran and Arunachalam 2006).

Such chromosome representation is able to find a satisfactory solution relatively quickly due to a smaller number of genes, and therefore a small number of possible permutations than in operation-based representation. The experiments carried out by Amirthagadeswaran and Arunachalam (2006) on the 24 standard test instances show that the job-based representation outperforms other representations in 22 cases.

On the other hand, such a robust decoding procedure might prevent formation of the optimal solutions. For instance, it would be impossible to obtain a schedule where the operations of different jobs should be scheduled in different orders. This problem can be tackled with the usage of operations-based representation.

Operation based representation The operation-based representation was proposed by Bierwirth (1995). Given *m* jobs with *n* operations, the chromosome is composed of *mxn* genes. Each operation is represented by its job number. In other words, the job number will appear in the chromosome as many times as the number of operations it contains. The number of a job occurrence in the chromosome denotes the operation's number. The following example with 3 jobs and 3 operations illustrates the chromosome representation and decoding

procedure. The first row in Figure 22 represents the chromosome and the second row explains the meaning of each gene

Chromosome	1	2	1	2	3	3	2	3	1
Job-Operation	1-1	2-1	1-2	2-2	3-1	3-2	2-3	3-3	1-3

Figure 22 Operation-based chromosome representation

Schedule builder is based on similar logic to the Griffler and Thomson algorithm, but uses EA to find an optimal sequence of operations. Based on the given example, the schedule builder firstly allocates the first operation of the first job to the relevant machine, then it placed the first operation of the second job. Decoding gene three, it places the second operation of the first job in the schedule in the best available time, which is the maximum time between the completion of the previous operation of the same job and the time when the corresponding machine becomes available (Amirthagadeswaran and Arunachalam 2006). The process repeats until all the operations appear in the schedule.

Such logic does not violate precedence constraints and produces a feasible schedule. In addition, it can adapt to the changes in schedule rates and uncertain number of operations (Zhu, Chen and Zhang 2009).

It has a half-Lamarckian property, where offspring partially inherit their parent's attributes. However, because each job number appears several times in the chromosome, some standard genetic operators, in particular mutation, might have no impact on the chromosome or produce infeasible solution. Also, with the larger search space compared to the job-shop representation, operation-based representation might attain a better solution, although at a cost of greater computation time.

Job-pair relationship based chromosome representation Unlike aforementioned chromosome representations, the job-pair relationship representation encodes the precedence constraints between operations on different machines (Cheng, Gen and Tsujimura 1996). The chromosome is presented in matrix form, where each row denotes all possible sequences of the operations and each column stands for machines (Figure 23). The binary variable

indicates whether the job *i* is performed before the job *i+1* on a particular machine *j*.

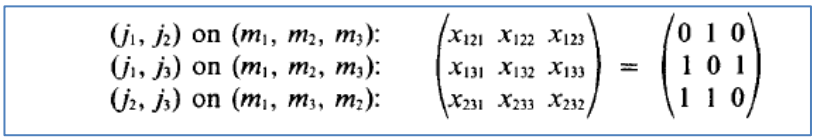


Figure 23 Job-pair relationship based chromosome representation

Source: Cheng, Gen and Tsujimura (1996)

EA designed by Hasan, Sarker and Cornforth (2007) is based on job-pair relationship chromosome representation. It achieves either better or the same as the best known results. However, it is important to notice that the experiments were conducted on small and medium size problems with the data set not exceeding 20 jobs and five machines.

In terms of the development of EA, this representation might be impractical since standard EA operators destroy the feasibility. Moreover, such representation requires significant memory resources. Cheng, Gen and Tsujimura (1996) state that this type of representation causes unnecessary complexity and contains superfluous information.

Completion time-based representation This chromosome representation was proposed by Yamada and Nakano in 1992 (Cheng, Gen and Tsujimura 1996). The length of the chromosome is equal to the total number of operations and each gene *c* denotes the completion time of *i*-th operation of *j*-th job on the *k*-th machine (Figure 24).

c_{111}	c_{123}	c_{132}	c_{211}	c_{223}	c_{232}

Figure 24 Completion time-based chromosome representation

The times are usually obtained through the Giffler and Thomson algorithm and each chromosome is an active schedule (Dahal, Tan and Cowling 2007). The chromosome has no-Lamarckian property (Gen and Cheng 1997). The success of finding the right solution depends on the genetic operators, which are quite complex and have to be devised specifically for this representation. This might be the reason for the little popularity of this representation in the literature.

Random-keys The solution is encoded as a set of random keys (random numbers from 0 to 1). The schedule builder is based on a priority rules mechanism, except that priority rules are determined and managed by EA (Dahal, Tan and Cowling 2007) . The schedule builder assigns operations to the machines in descending order. For instance, for the chromosome shown in Figure 25, sequence of the jobs is the following 2->1->3->4. Clearly, any permutation initiated by genetic operators sustains the feasibility of the schedule. This chromosome has a Lamarckian property, and with the evolutional process the algorithm acquires the knowledge of the relationship between certain chromosomes and the corresponding objective function.

0.35	0.98	0.14	0.03

Figure 25 Random-keys chromosome representation

Vela, Varela and Gonzailez (2010) extends this concept by including the maximum delay times. The length of this chromosome is equal to *2n*, where *n* is the number of the operations. The first *n* genes denote the priorities of the operations, and the second part of the chromosome specifies delay times (Figure 26). The delay times are equal to $gene_g$ x 1.5 x MaxDur (maximum duration of all operations). The principle of the delay times is that if the next operation is not scheduled within the specified time interval (i.e. machine remains idle after the execution of the previous operation), the operation with a lower priority will be placed into the schedule. This enables generation of so-called **parameterized semi-active schedule** (Vela, Varela and Gonzailez 2010).

0.35	0.98	0.14	0.03	650	1030	898	565

Figure 26 Random key chromosome representation with delay times

In the random key chromosome representation, a great burden lies on the complex schedule builder and objective function evaluation. Analogue of the uniform crossover and the mutation, which generates a new member from the same distribution as the original population for this type of chromosome, are introduced by Vela, Varela and Gonzailez (2010).

Such type of chromosome representation is highly reusable and can be applied to other ordering problems (Gen and Cheng 1997). However, the application of random-key representation is more common for problems with fuzzy due dates

and set-up times rather than traditional JSSP (Lei 2010, You-Lian Zheng, et al. 2010).

Preference List-Based Representation In this type of chromosome representation, the chromosome is comprised of m-blocks (*m*-number of machines) and each block stands for a single machine (Figure 27). Every block contains a permutation of the jobs in the order of their priorities (Hasan, Sarker and Cornforth 2007). It needs to be highlighted that this is only a preference list rather than strict scheduling rules. In the example of the chromosome presented on Figure 27, the preference for machine 1 is the first operation of job3, preference for machine 2 is the first operation of job 1, and the second operation of the third job for machine 3. If in some cases precedence-constraints do not allow sequencing the operation in the given order, then the operation is skipped and the schedule builder continues to scan the chromosome from the left to the right and assign only the operations which are permitted to be scheduled. Upon reaching the end of the chromosome, the procedure starts over and assigns unscheduled operations which can now be scheduled. This process repeats until all the operations are allocated. This mechanism provides feasible schedules. However, it is possible to encode the same solution differently, which will result in false competition between chromosomes in the population. However, unlike a situation where the same chromosome can be decoded differently, different encoding of the same solution might benefit the algorithm by contributing to the diversity of the population. It also can resolve itself over a number of iterations.

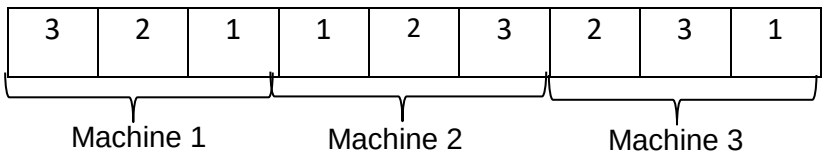


Figure 27 Preference list-based chromosome representation

This representation is commonly used for JSSP with due dates and release times (Cheng, Gen and Tsujimura 1996). Qing-dao-er-ji and Wang (2012), Essafi, Mati and Dauzere-Peres (2008) applied this chromosome representation for the flow shop problem, which is the JSSP without operation-precedence constraints. However, in this case, since the principle of the given chromosome representation is setting the directions of disjunctive arcs, a cycle might occur which in turn leads to an infeasible schedule. A special procedure resolving this

issue is developed by Qing-dao-er-ji and Wang (2012). Although the results reported by Qing-dao-er-ji and Wang (2012) outperform the existing techniques in the literature, especially for large-size problems, it is difficult to certainly state whether the success should be attributed to the chromosome representation scheme or special local search and selection procedures.

Priority rule-based representation In this type of representation a set of priority rules undergoes evolutional process, while the sequence of operations remains static. The length of such chromosomes is equal to the number of operations. Each gene signifies which heuristic rule will be applied to schedule the next operation. With each gene, the schedule builder re-arranges unscheduled operations according to the selected rule and then adds the operation with the highest priority into the schedule (Cheng, Gen and Tsujimura 1999). The examples of these rules are presented in section 5.2.

Operations can also be positioned into the schedule in a probabilistic manner. Firstly, the priorities are assigned to all the operations by using one or more despatching rules. After that one of the operations is probabilistically selected and drawn to the schedule. The operation has a higher chance of being chosen if it has a higher priority than others (Zhang and Wu 2011).

Advantages are ease of the implementation and low time consumption (Abdullah and Abdolrazzagh-Nezhad 2014, Cheng, Gen and Tsujimura 1996). In terms of the disadvantages, the changes indirectly impact genotypes that might cause false competition (Cheng, Gen and Tsujimura 1999).

Machine-based representation. In this type of representation, the chromosome consists of a set of ordered machines (Cheng, Gen and Tsujimura 1996). In order to deduce a schedule a ***shifting bottleneck*** algorithm is used. The main idea of a shifting bottleneck heuristic is to give a priority to bottleneck machines. It works according to the principle as follows. At the first step a schedule builder takes one gene (machine), which has not yet been sequenced and produces a schedule for it. At the next step all the machines that are already in the schedule undergo re-optimization (Adams, Balas and Zawack 1988). These two steps are repeated until the full schedule is obtained.

Disjunctive graph representation. In this representation a chromosome is a list of arcs in the disjunctive graph presented in Figure 19. It shows the processing order between two operations (nodes). The given type of representation can be classified as a variation of job-pair relationships representation. Each gene represents $e_{ij,}$ $e_{ij,}$ is equal to one arc orientated from the node *i* to the node *j*, and 0 denotes the opposite orientation: from the node *j* to the node *i* (Figure 28). This does not guarantee the feasibility of the solution since chromosomes which were randomly generated might contain cycles or violate operation precedence constraints (Gen and Cheng 1997). The critical path algorithm is used in order to conduct a decoding procedure. Cheng, Gen and Tsujimura (1996) state that in the given representation the chromosome is not a solution, which is a schedule itself, but rather a guide for the conflict resolution of the operations competing for the same machine.

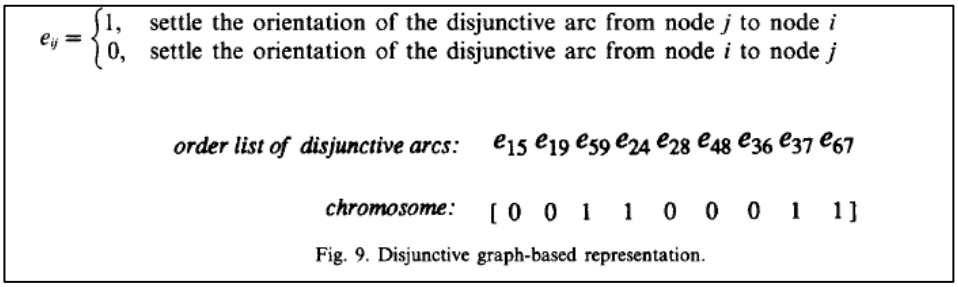


Figure 28 Disjunctive based chromosome representation

Source: Cheng, Gen and Tsujimura (1996)

This chromosome representation presents an extremely large search space as the number of combinations is $(2^{nxm})^m$ (Abdullah and Abdolrazzagh-Nezhad 2014). The possible unfeasibility of the schedule along with the complexity of the searching space might be the reason for the rare utilisation of this representation in the literature.

5.6.2 Population management

The vast majority of EAs for JSSP create the population at random, however various chromosome generation strategies have been introduced as well.

Tang et al. (2010) report that utilisation of Particle Swarm Optimisation in the chromosome generation process can produce better results in terms of quality and stability. Spanos et al. (2014) showed that priority rules can reduce the

number of illegal chromosomes as well as better results in terms of the achievement of augmented goals (i.e. due dates, machine workload).

The advantages of these methods are a high quality of the first population as well as the relatively narrow search space. However, they also might cause premature convergence and poor exploration of the space regions. In order to preserve diversity in the population Javadi and Hasanzadeh (2012) incorporated a neighbourhood check that accepts a new individual only if this chromosome has a predefined distance from already existing solutions. The distance is calculated according to the permutation of operations and machine assignment. However, the main disadvantage of this mechanism is there is no prior knowledge of the number of the regions and thus how many individuals should be in the population. This means that some of the regions might still be skipped.

Defersha and Chen (2010) proposed parallel EA implemented on two processors. They have used the ***island model***, where each processor solves the problem with an EA, but from time to time some individuals can migrate from one island to another. Migration is controlled by a specifically designed operator, which regulates the diversity and migration rates.

5.6.3 Crossover

The fundamental role of the crossover operator is to construct offspring, in the hope that they will be better than precursors as well as to direct the search process to new, as yet unexplored, regions. The type of crossover operator is determined by the chromosome representation. In the EAs designed for the JSSP, traditional crossovers as well as their problem-specific modifications are commonly used. The most popular crossovers for JSSP are outlined below.

Partially mapped crossover (PMX) is one of the widely-used crossovers for permutation encoding. The working principles of PMX were shown in 2.7.7. Jia et al. (2011), Essafi, Mati and Dauzere-Peres (2008) applied this crossover type in their studies of JSSP. Kramer and Koch (2007) improved PMX by intelligent selection of the cutting points.

Position-based crossover (PBX) is the equivalent of uniform crossover adapted to the literal chromosome representation and was utilised in Cheng, Gen and Tsujimura (1999). Figure 10 illustrates the mechanism of PBX crossover.

Order crossover(OX) is similar to the position-based crossover except that a part which will be copied is a set of consecutive genes in one parent (Cheng, Gen and Tsujimura 1999). OX crossover is explained in greater detail in section 2.7.7.

Precedence Operation Crossover was developed specifically for JSSP and can preserve good characteristics from previous generations (Chuanjun Zhu, Yurong Chen and Chaoyong Zhang 2009). This can be seen as an adaptation of PBX for the operation-based chromosome structure. The main principle is to randomly divide all the jobs into two non-empty, non-overlapping sets. Preserving locus, all the operations of the jobs belonging to the first set are transmitted to the first child. Then the operations of the jobs from the second set should be transferred to the first child in the same order as they are in the second parent as shown in Figure 29.

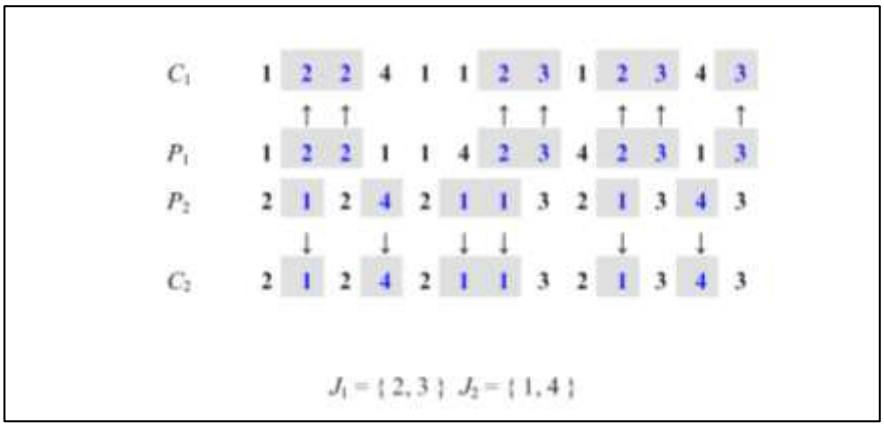


Figure 29 Modified Precedence Operation Crossover

Source: Chen and Zhang (2009)

Subsequence exchange crossover is used with the matrix representation, where a row denotes a series of operations processed by a machine (Figure 30). Usually all rows with odd indexes are passed to the first child from the first parent and even rows are taken from the second parent and copied to the first child. Such a method might produce illegal schedules, which violate the precedence-constraint. The Giffler and Thomson method is commonly used to restore the feasibility (Cheng, Gen and Tsujimura 1999).

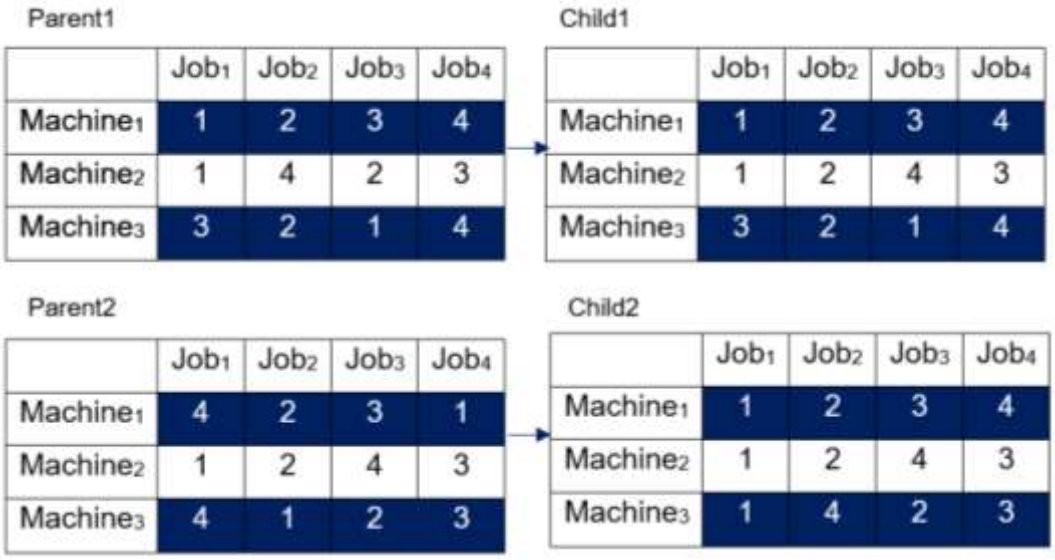


Figure 30 Subsequence exchange crossover

Job-based order crossover is the variation of the subsequence exchange crossover, where the jobs are selected and copied rather than machines (Figure 31). It works as follows. Firstly, a certain subset of jobs is selected in the first parents and copied onto the same positions (preserving machines) to the first child. After that the rest of the jobs are fulfilled in the order they appear in the second parent.

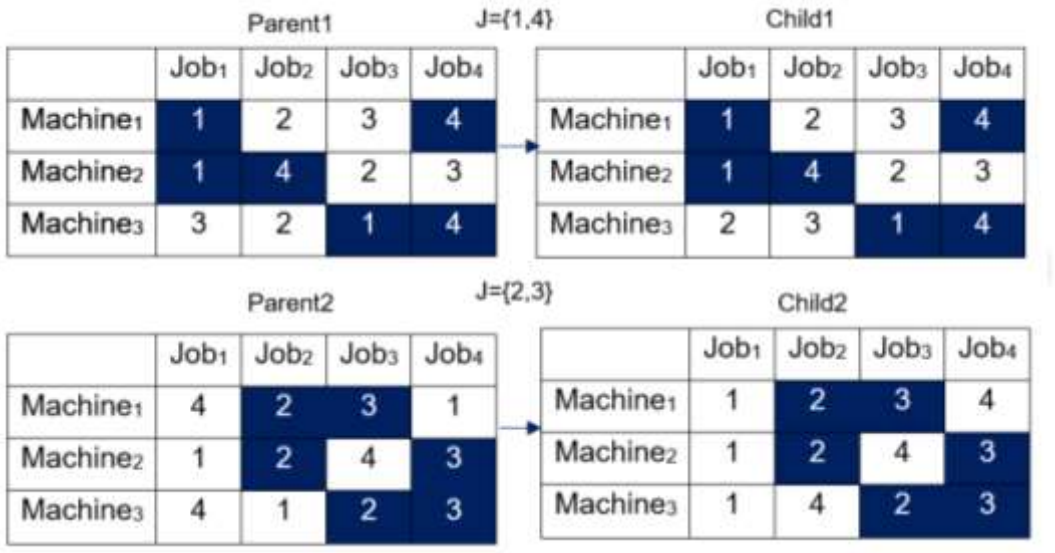


Figure 31 Job-based ordered crossover

5.6.4 Mutation

In order to maintain diversity in the population, a mutation method suitable for the selected chromosome representation should be designed. The most popular

mutation schemes are presented below. However, sometimes mutation is not used at all due to a sufficient diversity in the population caused by effective chromosome representation and crossover procedures (Essafi, Mati and Dauzere-Peres 2008).

Swap mutation exchanges the genes from randomly identified positions as was shown in Figure 14.This type of mutation is the most popular one for the solution of JSSP (Amirthagadeswaran and Arunachalam 2006, Jianchao Tang, et al. 2010, Yang et al. 2012). Swap mutation can be extended to swap the genes responsible for the jobs on the critical path that might lead directly to the attainment of the optimal schedule (Spanos et al. 2014).

Inversion mutation arbitrarily selects a subsection in the chromosome and reverses the order of all the genes from the selected range (Figure 15). Wang and Zheng (2001) employed this mutation type for the solution of JSSP.

Neighbourhood mutation is based on the generation of neighbourhood around a certain chromosome (Figure 32). A few genes usually selected as a basis for the identification of the neighbourhood and several new individuals are produced by permutation of these genes (Chuanjun Zhu, Yurong Chen and Chaoyong Zhang 2009). The best neighbour is accepted.

Parent chromosome						
1	3	2	5	4	6	7
Neigborhood						
1	3	2	4	5	6	7
1	4	2	5	3	6	7
1	4	2	3	5	6	7
1	5	2	4	3	6	7
1	5	2	3	4	6	7

Figure 32 Neighbourhood mutation

Compared to the other aforementioned mutation types, this type allows exploration of a particular region more comprehensively, and therefore the likelihood of solution improvement is higher.

5.6.5 Hybrids and local search strategies

EA has a good capability to explore the entire search space, but it lacks intensive local knowledge (El-Mihoub, Hopgood and Aref 2013). On the contrary, such algorithms as simulated annealing and tabu-search are equipped with the mechanisms of exploration of search space regions, but miss the global perspective. Therefore, the utilisation of EA with the local search technique creates a balance between exploitation and exploration phases. This section considers application of those algorithms in tandem with EA for the solution of JSSP.

5.6.6 Local neighbourhood search

The majority of the local search techniques for JSSP are based on exploiting the neighbourhood structure of the problem derived from the disjunctive graph representation. This is conducted by manipulating the operations lying on the critical path. The most popular method, which can lead to finding an optimum solution, is reversion of disjunctive arcs connecting adjacent operations performed on the same machine (Vela, Varela and Gonzailez 2010, Essafi, Mati and Dauzere-Peres 2008).

Essafi, Mati and Dauzere-Peres (2008) employ additional ILS (iterative local search) procedure. The ILS consists of two stages: the improvement stage and the perturbation stage. The first stage is based on steepest descents and accepts only the moves that add improvements to the solution. When no improvements can be made the perturbation operator repeats the same process with the exception that non-improving moves get accepted as well. This process allows exploration of new regions and avoidance of being trapped in the local minimum.

5.7 Simulated Annealing (SA) and Tabu Search (TS)

SA and TS belong to improvement type of algorithms, that start from an initial coded solution and gradually develops it (Pinedo 2009). The detailed explanation of the principles of their work was given in sections 2.3.1 and 2.3.2.

Various SA algorithms have been devised for the solution of JSSP and its variations (Zhang and Wu 2011, Cruz-Chajvez 2014, Mirsanei et al. 2011, Steinhafel, Albrecht and Wong 1999). The major conceptual differences between

them are temperature cooling schemes and neighbourhood generation mechanisms.

In some cases, SA even outperformed EA. In the experiments of Ponnambalam, Jawahar and Aravindan (1999), SA demonstrated better results in 11 of 20 tests, but at the execution time was significantly longer. It is also important to notice that used in the experiments EA has been designed for the flexible JSSP, whereas SA was tailored to the solution of the classic JSSP. SA also showed superior performance over EA with random key chromosome representation in terms of the quality of the schedule and computation time in the study conducted by Mirsanei et al. (2011). However, based on the observations of the trends in the literature, random key representation is rarely used for the solution of job shop scheduling problem due to the fact that the schedule deduction from the chromosome is a time consuming task, and the majority of researches give their preferences to more problem-specific chromosome representations.

At the same time, several experiments concluded that together EA and SA are able to attain significantly better results (Rakkiannan and Palanisamy 2012, Liu et al. 2011). There are various ways of embedding SA into EA framework. For instance, Wang and Zheng (2001) apply simulated annealing to each new individual in the population until the stopping criteria are met. Then, the algorithm returns the best found solutions in the population to EA. Dong Hui (2012) proposes a crossover operator on the basis of SA algorithm, while Liu et al. (2011) incorporated SA into mutation.

EA can also benefit from the adaptive memory regarding the previous solutions presented in Tabu-search. There are various ways of its integration in the literature. Vilcot and Billaut (2008) utilised TS in order to generate the initial population. The algorithm starts from creating the initial solution and then applies TS in order to produce a neighbourhood and form the rest of the population. Zhang, Gao and Li (2013) apply TS for a certain number of iterations for each individual in the population. In order to reduce the computation time, Javadi and Hasanzadeh (2012) firstly cluster the solutions and then employ a TS as a local search mechanism to a single representative selected from each cluster.

And finally, Thamilselvan and Balasubramanie (2012) implement SA, TS and EA in the same algorithm. EA is a leading algorithm, while TS is performed after the

crossover operator and SA is inserted after mutation. The authors claim that such a combination surpasses EA, parallel SA, and hybrid algorithm of SA and EA.

5.8 Limitations and gaps in the literature

This review of the literature has identified three gaps related to the way the research experiments were conducted and the results were compared. First of all, there is a lack of elucidation and experimental justification of the reasons for the selection of a particular operator in the literature. Secondly, because all the operators were applied in conjunction with one another, it is difficult to determine what operator was responsible for the success or failure of the overall algorithm. Thirdly, there were no direct comparison under the same conditions made, which would allow for establishing efficiency of a particular chromosome representation of a genetic operator.

As modelling the behaviour of the population under certain operators before their implementation is an extremely challenging task (Gendreau and Potvin 2005), the more feasible way to discover the efficiency of each operator is to conduct empirical evaluation. This research provides such evaluation for various genetic operators in the context of JSSP.

5.9 Conclusion

This chapter has reviewed the methods available in the literature for the solution of the Job-Shop Scheduling Problem. They included exact methods such as branch-and-bound as well as heuristic and metaheuristic algorithms which include despatching rules, Giffler and Thompson algorithm, evolutionary algorithm, Simulated Annealing and Tabu-Search.

The main focus was on the configurations of the EAs since analysis of the literature showed that it is one of the popular meta-heuristic algorithms for the solution of JSSP. A wide range of chromosome representations, genetic operators and hybrid algorithms have been discussed.

It was identified that the major limitation of EA research is a lack of empirical evidence of genetic operators' effectiveness. This research will fill this gap by carrying out experimental comparison of the effectives of genetic operators in Chapter 9.

Chapter 6. Crew Scheduling Problem in the rail-freight industry

6.1 Introduction

This chapter concerns the second scheduling problem selected for this research, which is the crew scheduling problem. In order to develop an appropriate and effective solution it is important to have an in-depth understanding of the problem and surrounding business environment. Therefore, the chapter starts by providing an overview of the rail freight industry and its role in the economy.

It then explains the complexity of crew scheduling operations and its importance for the overall business. Close attention is paid to health and safety regulations and contractual terms underpinning the construction of the driver schedule in the real world. Based on this, the formal mathematical model defining driver scheduling processes, which will be used to develop and test the optimisation algorithm, is devised.

6.2 Role of rail freight in the economy

While international trade continues to expand, businesses are striving to increase reliability and reduce their environmental impact (WTO 2014, Eurostat 2015). This has a positive impact on the growth of the demand for transportation (World Energy Council, IBM Corporation and Paul Scherrer Institute 2012, Islam et al. 2015). For example, the number of containers that passed through Felixstowe port, the largest container port in the UK, has increased twice between 2001 and 2011, resulting in a 25% rise in the amount of trains arriving and departing to and from the port (Network Rail 2014).

There are a number of reasons why businesses give preferences to railway freight transportation. These are lower cost with a smaller number of incidents and relatively high quality and reliability (Cacchiani, Caprara and Toth 2010). Rail transport provides higher reliability in terms of the number and length of delays compared with road transport, which is often subject to traffic and congestion. It is estimated that road congestion reduces GDP by £7-8 billion a year (Network Rail 2014). From the fuel cost perspective, rail is almost three times cheaper

than road transportation. For example, using a gallon of fuel it would be possible to move a ton of goods for 246 miles, whereas by road it is only 88 miles (Network Rail 2014). With regard to quality, rail transportation is safer meaning the goods are less likely to be damaged. For instance, this is one of the reasons why the majority of the luxury car brands such as Mini, Jaguar and Land Rover transport 70% of their premium products by rail (Network Rail 2014).

Finally, each train can replace 50 heavy goods vehicles from the road. Effectively this would decrease carbon dioxide emission, number of incidents, congestion and even noise in certain areas making them more pleasant and safer for communities (Network Rail 2010).

6.3 Rail-freight industry overview

Privatisation in 1994 divided the railway industry into two parts: the infrastructure (stations, signalling, tracks) controlled by Railtrack (later National Rail) and private train operating companies (TOC). English, Welsh and Scottish Railways Ltd. (EWS) was one of the biggest freight TOC (Stittle 2004). Freightliner was the second biggest player transporting 17% of the freight traffic. In 2007 EWS was sold to Deutsche Bahn and in 2009 rebranded to DB-Schneker (DB-Schneker 2014b).

Demand for transportation is dictated by overall economic health, international trade and the situation in a particular industry. Figure 33 displays the breakdown by commodity type in the amount of freight moved by railroad based on their weight and distance carried.

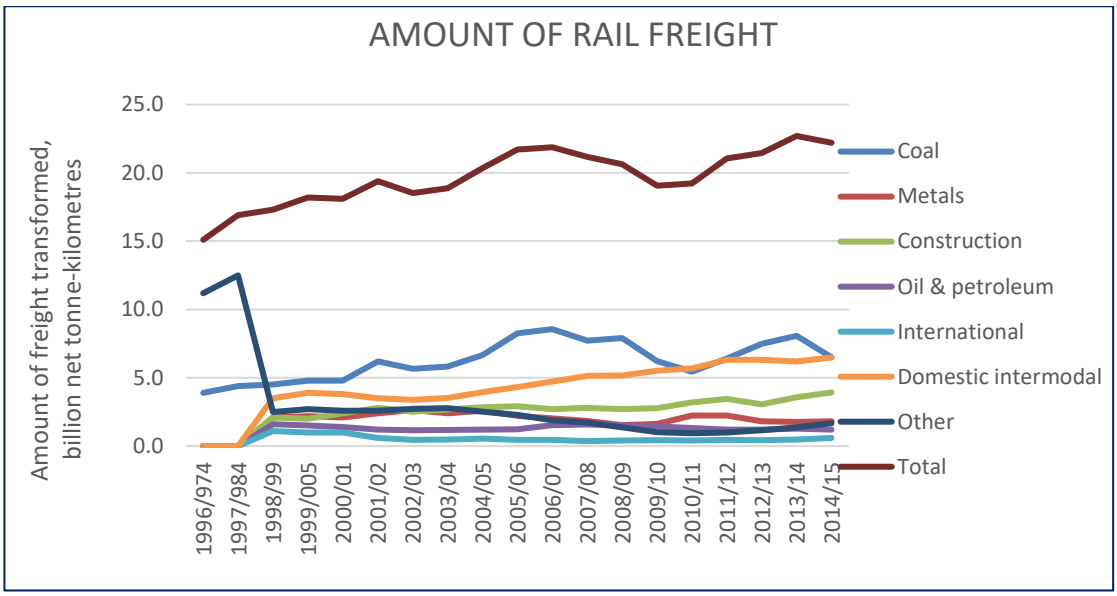


Figure 33 Demand for rail freight transportation by commodity 1996-2014

Adapted from Department for Transport (2015)

Despite the impact of the global recession of 2007-2012, the overall trend in demand for transportation is increasing (Islam et al. 2015). Moreover, the number of trains to serve the industries is predicted to grow even further in the coming years (Marketline 2014). In addition, Network Rail is planning to expand infrastructure which would allow transit companies to have more and longer trains, and operate on a larger number of destinations (Network Rail 2014). Moreover, the HS2 project is expected to "take" the passengers from the standard rail track freeing up capacities for the freight trains.

However, the proportion of commodities in the overall freight dynamically changes. Domestic intermodal category is rapidly growing as the volume of freight passing through the Channel Tunnel is rising (Eurotunnelgroup 2014) owning to the stable relationship with main European trade partners, Germany and France (HM Revenue&Customs 2015). However, the demand for coal transportation has significantly decreased since 2013 due to closure of several power stations and relatively high winter temperatures (Islam et al. 2015). This adversely impacted train operating companies who were forced to reduce the number of their staff particularly in the North of England (BBC 2015, Gazettelive 2015). On the other hand, the demand for biomass, as an alternative to coal, continues to grow compelling the freight TOC to invest in the development of biomass wagons in order to respond to customer needs.

Currently there are seven freight operating companies: Colas Rail, Devon and Cornwall Railways, Direct Rail Services, DB Schenker, Freightliner, GB Railfreight, Mendip Rail (Office of rail and road 2015). The rivalry in the market is assessed as strong (Marketline 2014).

Therefore, in order to effectively adapt to fluctuations in demand and remain competitive, it is paramount for rail freight carriers to have agile business processes. The next section describes the principal planning and scheduling operations (DB- Schneker 2014a).

6.4 Planning operations in the rail scheduling

In order to effectively function and adapt to the changing demand, the rail freight operator needs to solve various problems including the crew scheduling problem, blocking problem, yard location problem, train routing problem, locomotive scheduling problem, train scheduling and dispatching problem (Mu and Dessouky 2011).

As the driver cost is the second largest cost after the fuel cost, the driver scheduling problem has been selected for this research, and the operations influencing crew scheduling will be studied in greater depth (Kwan 2011).

Figure 34 illustrates the operations dealing with customer orders (DB-Schenker Business Manager interviewed on 14/11/12). The process starts with taking an order from a client. Orders vary in terms of the frequency, size and type of commodity. Each order is characterised by the places where the goods need to be collected from and delivered to, volume and tonnage as well as commodity type.

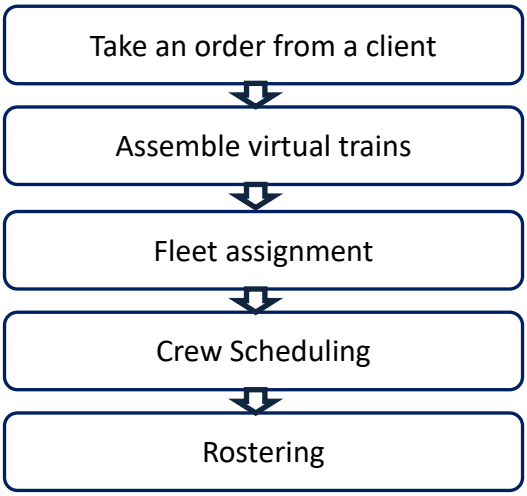


Figure 34. The main operations of the railways freight carrier

Once the orders have been collected, similar commodities which are transported in the same directions and on the same date are grouped together. Then, they are temporarily assigned to the virtual trains (i.e. simulation of the real train). At the next stage, the route of the virtual train is specified and the real fleet is reserved for each virtual train. At this step, a scheduler also adds some ancillary activities such as attachment and detachment of a set of wagons, loading and unloading goods, fuelling a train, freight shunt etc.

The last two stages concern the construction of the crew schedule and assignment of the train drivers to the trips. Crew scheduling operations group a sequence of trips into the shifts. Crew rostering is a process of assigning a driver with the required route and traction knowledge to each shift. Rostering is subject to several industrial regulations. The fundamental constraints are the minimum rest time between shifts and the number of free days. At the rostering stage the planners also make sure that the work is distributed fairly among the drivers.

6.5 Crew Scheduling Problem

At the strategic level, crew management is concerned with depots' capacities and allocation of the depots (Huisman et al. 2005). These decisions are usually based on forecast of demand for the freight transportation and market outlook.

Apart from the demand forecast, another challenge here is maintaining the required level of staff. Hiring new drivers requires a considerable amount of time and financial resources as drivers need to undergo appropriate training before they can start their job. In addition, redundancy is the last option for the company

as it entails difficult negotiations with trade unions and payment of compensation packages (Huisman et al. 2005).

6.5.1 Contractual terms

According to employment contract terms, the drivers are paid the same hourly rate for any time spent on duty regardless of the number of hours they have actually been driving the train. Moreover, in accordance with collectively bargained contracts, each driver has a fixed number of working hours per year, so the company is obliged to pay for all the stated hours in full even if some of the hours are not utilized. Paid additional overtime hours can be worked at the driver's discretion. Thus it is in the best interests of the company to use the agreed driving hours in the most efficient and economical way (DB-Schenker Head of Finance interviewed on 02/09/2013).

From a business perspective, crew management processes are relatively inflexible and any changes in the contractual terms might have serious consequences for the company (i.e. strikes). From the legal point of view, the collectively bargained contract denotes that the company cannot deal individually with each employee (i.e. negotiate amount of working hours). It also heavily restricts the company to freely adjust their workforce in relation to demand.

6.5.2 Crew scheduling processes

CSP in the rail-freight industry deals with the construction of a schedule for a train driver. Each schedule contains instructions for the driver of what he or she should do on a particular day. Within the industry, the driver's schedule is called a ***diagram***. Each diagram should cover all the trains driven by a driver in a given day. It must start and end at the same station and obey all labour laws and trade union agreements. These rules regulate the maximum diagram duration, maximum continuous and aggregate driving time in a diagram, and minimum break time.

All drivers are located in depots where they start and finish their work. Depots are distributed approximately evenly across the UK. Sometimes in order to connect two trips that finish and start at different locations, a driver has to travel on a passenger train, taxi or a freight train driven by another driver. The situation of a driver travelling as a passenger while on duty is called ***deadheading***. The cost

of deadheading varies and depends on the means of transportation and business agreements between operating companies. Despite the potential cost, deadheading is sometimes inevitable and it can benefit the overall schedule (Barnhart, Hatay and Johnson 1995, Jutte et al. 2011).

6.5.3 Operational objectives

The effectiveness of the scheduling operations depends on the degree to which a schedule achieves objectives as follows (DB-Schenker Head of Finance interviewed on 02/09/2013):

1. Minimize the cost of additional transportation, such as a taxi.
2. Minimize the losses associated with unused and excess contract hours at the end of the year.
3. Minimize the spread of durations of the diagrams. All diagrams will therefore be of duration close to the average 8.5 hours, i.e. the annual contract hours divided by the number of working days.
4. Maximize the throttle time, i.e. the proportion of the work shift that is actually spent driving a train. It excludes time for deadheading and waiting between trips.
5. Minimize the deviation of workload distribution across the depots.

6.5.4 Labour rules

In addition, all the diagrams must adhere to various health and safety regulations, such as (DB-Schenker Head of Finance interviewed on 02/09/2013):

1. Maximum diagram duration cannot exceed 11 hours and 30 mins.
2. No driving is allowed after 11 hours of work.
3. For the six to nine hours shift the driver should take either one break of 30 minutes or two breaks of twenty minutes.
4. For the more than 9 hours diagram the break should be one of the following options: one break of 45 mins; 2 breaks of 30 mins each; 3 breaks of 20 minutes each.
5. Maximum aggregate driving should be from 7.30 to 8 hours depending on the class of train. The information about various locomotive types is presented in Appendix 4.

6. Maximum continuous driving time should be from three to five hours depending on the train class and number of stops.
7. All the diagrams with a duration of less than 5 hours are rounded up to the five hours.

6.6 Complexity and size of the problem

Both operational constraints and the size of the problem contribute to the high complexity of the problem (Caprara, et al. 2007). Furthermore, the crew scheduling problem in rail freight is more complex than similar crew scheduling problems in airline and passenger railway transportation. Table 8 compares complexities of CSPs in various industries.

Table 8 Problem Complexity

	Air transportation	Passenger railways	Freight railway
Network structure	Hub-and-Spoke (tree graph)	Acyclic graph	Acyclic graph
Schedule	Cycle. Repeats every week	Cycle. Repeats every week	Based on customer orders
Time	24/7	Day time	24/7
Relief opportunities (Places where drivers can change)	Only at origin/destination	Only at stopping stations	At any passing stations
Deadheads	Planes of their and other companies; Trains between airports only; Taxi connecting the nearly located airports;	Passenger trains Taxi	The same mode of transportations; Passenger trains between all the stations; Taxi connecting nearly located cities;
Geographical coverage	Depends on the scope of the company	Part of the country	The entire country

Source: Adapted from Jutte (2011)

6.7 Importance of effective crew scheduling systems

Given both the intricacy of the problem and its significance in the overall planning processes, it is evident that it would be almost impossible for a human to produce a schedule which would satisfy the objectives stated in 6.5.3. Having an effective

system assisting in the decision making is very important for the following reasons:

1. The crew cost accounts for 20-25% of the total operating cost and is the largest after the fuel cost. Even a 1% improvement can save a company a substantial amount of money (Kwan 2011, Abbink et al. 2005). In the context of DB-Schenker, 1% of crew scheduling savings can be equal to hundreds of thousands of pounds saved a year. This will be discussed in detail in Chapter 11.
2. Unlike the passenger trains, where the route depends on the demand in certain areas, the path of the freight train is also determined by the availability of the train drivers in certain depots.
3. The effectiveness of the subsequent, rostering, stage depends on the quality of the built crew schedule.
4. An effective crew scheduling system might enable a company to be more competitive and support a franchise bid in the UK (Jutte et al. 2011, Kwan 2011).
5. Because crew scheduling is the last operation and is performed in a very short time frame, the work of the schedulers is associated with a great amount of stress. An automatic scheduling system might help to produce an initial schedule and the schedulers would have more time to thoughtfully revise the schedule and possibly conduct a “What-if” analysis (Kwan 2011).
6. Overall, automatic systems would provide more flexibility and agility to the company (Caprara, et al. 2007).

6.8 Mathematical formulation of the CSP

Assuming that the set $T_i = \{t_1, t_2 \dots t_n\}$ represents all the trips to which drivers need to be assigned to, set $K_l = \{l_1, l_2 \dots l_p\}$ contains all possible types of locomotives and set $R_k = \{l_1, l_2 \dots l_q\}$ includes all the routes, each trip t has the following characteristics:

ts_j– start time of the t_i trip.

et_j - end time of the t_i trip.

sl_j- start location (origin) of the t_i trip.

el_j – end location (destination) of the t_i trip.

tl_{il}– is a type of the locomotive that should perform t_i trip; if tl_{il} = 1, then the locomotive which carries out t_i trip belongs to a class l. For each trip only one tl_{il}=1.

rk_{ik}– is a route code; if $rk_{ik} = 1$, then trip belongs to the k-th code of the route. For each trip only one $rk_{ik} = 1$.

There is also a set of drivers $D_j = \{d_1, d_2 \dots d_m\}$ with each driver d having the following properties:

h_j-driver home depot.

tl_{jl}- traction knowledge, if $tl_{jl} = 1$, the d_j driver has knowledge of the l^{th} locomotive type.

rk_{jk} - route knowledge, if rk_{jk} = 1, then the d_j driver has the knowledge of the k^{th} route;

The number of depots is equal to $\mathrm{Ndepots}$. W_{dep} represents all the workload for a depot $\mathrm{dep\ and}\ \overline{W}$ is the average workload of all the depots.

Finally, there is a schedule $S = \{s_1, s_2 \dots s_w\}$ which consists of the w number of diagrams s. In turn, each diagram s consists of a combination of trips and taxi transfers. In order to include the possibility of transporting a driver by a taxi, additional set of taxi trips, Taxi, connecting all locations of origins and destinations as well as depots is created. Each taxi trip has an associated cost taxi proportionate to its duration.

$$\mathrm{Taxi}=\{\mathrm{taxi}_{1,2}, \mathrm{taxi}_{1,3} \dots \mathrm{taxi}_{1,\,nxw}; \mathrm{taxi}_{2,2}, \mathrm{taxi}_{2,3} \dots \mathrm{taxi}_{2,\,nxw};\ \mathrm{taxi}_{n,1}, \mathrm{taxi}_{n,2} \dots \mathrm{tax}_{n,nxw}\}$$

The Formula 9 expresses the main objective, which is minimisation of the schedule cost. The cost of the schedule is composed of four components: labour cost (Formula 9.1), cost for additional transportation (taxi cost) (Formula 9.2), losses from unequal utilisation of drivers' contract hours (Formula 9.3) and loses attributed to unequal distribution of workload amongst depots (Formula 9.4).

Formula 9

$$DriverCost + TaxiCost + DeviationCost + DistributionCost \rightarrow min$$

Formula 9.1

$$DriverCost = \sum_{\forall s \in S} S_{t_{et}-t_{st}} \times HourlyRate$$

Formula 9.2

$$TaxiCost = \sum_{\forall tax \in S} taxi \times HourlyRate$$

Formula 9.3

$$DeviationCost = \sum_{\forall s \in S} \left| S_{t_{et}-t_{st}} - \bar{S} \right| \times HourlyRate$$

Formula 9.4

$$WorkloadCost = \sqrt{\frac{1}{m} \sum_{dep=1}^{Ndepots} \left(W_{dep} - \bar{W} \right)} \times HourlyRate$$

However, this is a subject to the following conditions and constraints (Formula 9.5-Formula 9.9):

Formula 9.5

$$\forall\ t \in T\ t \in S$$

The condition presented on the Formula 9.5 requires all the trips to be included into the schedule.

Formula 9.6

$$tl_{j(tl_{il})} = 1\ \&\&\ rk_{j(rk_{ik})} = 1$$

Formula 9.6 denotes that a driver can be assigned to the trip only if he or she has necessary route and traction knowledge.

Formula 9.7

$$l < t$$

Formula 9.7 ensures that the number of diagrams will not exceed the number of drivers.

Formula 9.8

$$\forall\, s \in S\ 5 < ts_{et} - ts_{es} < 11$$

This constraint requires the duration diagrams to be no more than 11 hours and no less than 5 hours.

Formula 9.9

$$\forall\, s \in S: ts_{sl} = ts_{el} \,||\, ts_{sl} = taxi_{el,thd}\, ts_{el} \,||\, ts_{sl} + taxi_{el,thd} = ts_{el} \,||\, ts_{sl} + taxi_{el,thd} = ts_{el} + taxi_{el,thd}$$

Formula 9.9 denotes that any diagram should start and finish in the same depot. Taxi trips can be used to connect job locations with home depots if necessary.

6.9 Conclusion

The chapter has described the importance of the rail freight for the economy and explained the context surrounding the rail freight driver scheduling problem. The analysis of the health and safety regulations and train driver contract structure has been provided. Given the above information, mathematical model representing the problem has been designed.

The next chapter will consider the approaches developed in the literature for driver' schedules creations and optimisation.

Chapter 7. Approaches to Crew Scheduling Problem

7.1 Introduction

This chapter presents a review of approaches to solving the Crew Scheduling Problem, which was formulated and defined in the previous chapter. Although the research deals with the CSP in the rail freight industry, the crew scheduling algorithms designed for other transit industries will be considered as well because these problems are conceptually similar.

Broadly optimisation techniques for CSP can be divided into exact and heuristics. Exact methods are based on Linear Programming and Column Generation techniques (Lasdon 1970a). The heuristic methods developed for CSP include Simulated Annealing, Ant Colony Optimisation and GA. This chapter presents thoughtful analysis and examination of the effectiveness of each technique. Their gaps and limitations are exposed and discussed in this chapter.

7.2 General approach for the solution of CSP with exact methods

The CSP is usually solved in two stages. At the first stage, all possible diagrams satisfying the industrial constraints and health and safety regulations are enumerated. Typically, the number of generated diagrams reaches 300 000-400 000 for small problems and can be up to 50-75 million for the large ones (Klabjan et al. 2001, Kwan 2004). Diagrams are usually presented in the form of a ***duty matrix*** and modelled as binary vectors where '1' denotes that the trip *i* is included in the diagram *j*, otherwise '0' is inserted. In the rest of the thesis the terms diagram and column will be used interchangeably (Gopalakrishnan and Johnson 2005).

Assuming that the trains in Figure 35 require an assignment of the drivers, some examples of possible diagrams are presented in Figure 36. For instance, diagram one denotes that the driver starts his work at London Kings Cross and drives a train to York, where he changes it and operates another train from York to

Newcastle. In Newcastle he changes the train again and then drives it from Newcastle to the place where he started his work, London Kings Cross.

The second diagram indicates that the work starts in York and a driver should drive a train from there to Newcastle, and then from Newcastle to London King's Cross. As there are no company trains that can deliver him to the home station, he is taking a train to York operated by another company. The deadhead journeys are not explicitly displayed in the duty matrix, but are taken into account when calculating the overall cost. In this example the cost of each diagram is computed on the basis of the drivers' payments and cost of additional transportation only. The cost breakdown and all the deadhead journeys are presented in Table 9 and Table 10.

The generation of the diagrams is performed in a simple and relatively straightforward manner using various graph searching and label-setting techniques, which will be discussed in section 7.2.5.

	Origin	Destination	Departure time	Arrival time
Trip 1	London Kings Cross [KGX]	York [YRK]	13:00	14:51
Trip 2	York [YRK]	Newcastle [NCL]	15:08	16:15
Trip 3	Newcastle [NCL]	London Kings Cross [KGX]	16:59	19:50
Trip 4	London Kings Cross [KGX]	Darlington [DAR]	15:00	17:20
Trip 5	Darlington [DAR]	York [YRK]	17:27	17:54
Trip 6	York [YRK]	London Kings Cross [KGX]	18:30	20:46

Figure 35 Trains in the timetable

Source: National Rail (2015)

	Diagram 1	Diagram 2	Diagram 3	Diagram 4
Trip 1	1	0	0	1
Trip 2	1	1	0	1
Trip 3	1	1	0	0
Trip 4	0	0	1	0
Trip 5	0	0	1	1
Trip 6	0	0	1	1

Figure 36 Diagrams

	Diagram 1	Diagram 2	Diagram 3	Diagram 4
Total Diagram Duration	06.50	07:46	06:42	05:46
Driver Payment, £	273	310	228	230
Transportation cost, £	0	40	97	0
Total, £	273	350	325	230

Table 9 Diagram cost

	Origin	Destination	Departure Time	Arrival Time	Cost
Deadhead diagram 2	London Kings Cross [KGX]	York [YRK]	20:00	21:50	£ 40
Deadhead1 diagram 3	Newcastle [NCL]	Darlington [DAR]	16:25	16:55	£ 97

Table 10 Deadheads

Source:National Rail (2015)

At the second stage, only the set of diagrams that covers the entire schedule in the most cost-effective way is identified (Caprara et al. 1997). Referring to the previous example, the schedule can be covered by diagrams one and three, or two, three and four. It is evident that the first case is more preferable because the schedule has a lower cost and utilises only two drivers. However, because millions of columns might be generated, identification of the set of diagrams which will constitute the schedule is a more complicated task due to a massive number of possible combinations (Gopalakrishnan and Johnson 2005). Kwan (2004) compares this stage with a jigsaw puzzle with an infinite set of pieces, which represent the task to find the diagrams which would not only cover all the trips, but would also effectively fit together.

The problem boils down to the solution of the 0–1 set covering problem (SCP) or set partitioning problem (SPP) (Chu, Gelman and Johnson 1997). Formulas Formula 10.1-Formula 10.3 and Formula 11.1-Formula .3 present mathematical models of SCP and SPP correspondingly. In these formulas, a_{ij} is a decision variable indicating whether a trip i is included in the diagram j; x_j shows if the diagram is included in the schedule; c_j is the cost of the diagram. The only difference between SCP and SPP is that SCP allows to having two drivers on the train, i.e. in the situation when one driver operates the train while another one presents there as a passenger (Formula 10.2 and Formula 11.2) (Caprara et al. 1997).

Set covering formulation	Set partitioning formulation
$Minimize \sum_{j=1}^{m} c_j x_j$ (Formula 10.1)	$Minimize \sum_{j=1}^{m} c_j x_j$ (Formula 11.1)
$Subject\ to: \sum_{i=1}^{n} a_{ij} x_j \geq 1$ (Formula 10.2)	$Subject\ to: \sum_{i=1}^{n} a_{ij} x_j = 1$ (Formula 11.2)
$x_j \in \{0,1\}$ (Formula 10.3)	$x_j \in \{0,1\}$ (Formula 11.3)
$i = 1,2 \ldots n\ trips$	$i = 1,2 \ldots n\ trips$
$j = 1,2 \ldots m\ diagrams$	$j = 1,2 \ldots m\ diagrams$

7.2.1 Column generation

Column generation is one of the most popular algorithms for the solution of CSP. The invention of the column generation is attributed to Dantzig and Wolfe (1960) (Lübbecke and Desrosiers 2005). The first researchers who applied this method for CSP were Lavoie, Minoux and Odier (1988). Column generation remains a very popular method for the solution of CSP nowadays and is used by many authors including Gopalakrishnan and Johnson (2005), Derigs, Malcherek and Schafer (2010), Jütte et al. (2011), Nishi, Muroi and Inuiguchi (2011).

Algorithm 1 Column generation

1: Generate a limited number of columns.
2: Solve Restricted Master Problem.
3: If the solution is feasible then algorithm terminates. If the solution is infeasible go to step 4.
4: Pricing. Using pricing algorithm, find and add new columns. Go to step 2.

Column generation allows obtaining an optimal solution without enumeration of all possible diagrams. The algorithm of the method is illustrated above. In general, column generation consists of two sub-problems: ***master*** and ***pricing***. A master sub-problem is applied to solve a set covering problem from only a limited set of columns (i.e. it is often called a restricted master problem (RMP). The pricing problem is responsible for the production of additional columns. If the new

columns improve the objective function, then the process repeats. If the value of the objective function remains the same, then the process terminates, and it is assumed that the optimal solution is found.

To reduce the number of iterations, Duck, Wesselmann and Suhl (2011) use multiple pricing in order to obtain several columns at the pricing stage and to increase the convergence of the optimisation algorithm. Abbink et al. (2011) introduce the concept of ***fixed columns***. Fixed columns are the columns which demonstrated a small improvement to the previous iteration, and are regarded as a sign that the algorithm is approaching an optimum solution. For this reason, fixed columns remain in the master problem and the pricing problem constructs only the columns which include uncovered by the fixed columns duties.

However, both approaches contain certain limitations. For example, by generating several columns at the same stage, it might not be possible to determine which of them were more beneficial for the solution and should remain. The drawback of the second method is that slight improvement of the objective function can be because the new columns are not significantly better than existing ones.

7.2.2 Master problem

A high volume of constraints presents a significant challenge to the solution of large linear problems (Nemhauser and Wolsey 1988, Hillier 2005, Reeves 1993). For this reason, constraints 10.3 or 11.3 are usually relaxed to non-negativity constraints and imposed later if the obtained solution is not integer. Constraints 10.2 and 11.2 can be relaxed through Lagrangian relaxation. Lagrangian relaxation transforms the constraints into a related penalty function (Formula 12). The penalty coefficients *u* are called ***Lagrangian Simplex Multipliers*** and updated with the sub-gradient optimisation method (Formula 13) (Beasley and Cao 1996). In this formula x_{ij}^{n} is the solution of LP relaxation on the n^{th} iteration, *t* is a positive scalar denoting the step size. The advantage of the sub gradient method is that it is relatively easy to program and it provides good results to practical problems (Fisher 1981).

Formula 12

$$\sum_{i,j} c_{ij}x_{ij} + \sum_{i=1}^{N} u_i(1 - \sum_{j} x_{ij})$$

Formula 13

$$u_i^{n+1} = u_i^n + t_n(1 - \sum_j x_{ij}^n)$$

The relaxed LP solution is solved with various simplex method techniques. The key principle of the simplex method is starting from initial basic solution it iteratively improves it by exchanging basic and non-basic variables. The number of shifts in the initial basic solution usually indicates an upper bound - the maximum possible number of shifts in the schedule (Kwan 2004). Although simplex method is able to identify the optimal solution, it might have a very slow convergence rate due to highly degenerate nature of the problem, according to Jans and Degraeve (2004) (in Duck, Wesselmann and Suhl 2011).

Because CSP is a sheer combinatorial optimisation problem with a large number of variables, the duality attribute (Formulas 14.1 and 15.2) of each linear program is usually exploited (Hillier 2005).

Formula 14

Primal

Minimise cx

Subject to $Ax > b$

$x \geq 0$

Formula 15

Dual

Maximise πx

Subject to $A\pi \leq c$

$x \geq 0$

The ***dual simplex method*** approaches the solution from the infeasible for the primal problem region. It is able to locate the solution for a smaller number of iterations in the problems where the number of columns is significantly larger than a number of rows and is more suitable for the linear programs with integer variables (Klabjan et al. 2001).

Yan and Chang (2002) solved the LP-relaxation with the classic simplex method and used the simplex dual variables from the optimal simplex tableau to modify

the duty arc costs. They also employed sensitivity analysis techniques which determined whether additional columns should be added or not.

With the ***primal dual simplex method***, the optimal solution is approached from both directions: from the primal feasibility and dual infeasibility (Curet 1993). Klabjan, Johnson and Nemhauser (2000) employed this method for CSP and increased the speed of the algorithm by solving linear sub-problems on different processors. After that the dual feasible solutions were considered together in order to identify the direction of a search.

7.2.3 Diagram Generation

The problem of diagram generation is usually modelled as a connection graph (Figure 37). Diagram generation is a very time consuming procedure due to the large number of combinations of trips which can form diagrams to be considered. The breaks and various deadhead opportunities only add the complexity to the existing problem. Therefore, special models and techniques have been proposed which allow for a slight reduction in the intricacy of the problem and handling those activities more effectively.

Unlike straightforward representation, where nodes represent train stations and arcs reflect trains, the majority of the studies utilise nodes to denote activities (trips) and arcs to display possible sequences of activities (Derigs, Malcherek and Schafer 2010, Shebalov and Klabjan 2006, Lu and Gzara 2015). Presenting the problem in this way allows for explicit representation of the constraints (Desaulniers et al. 1997). There can be different types of nodes, depending on the kind of activity they represent (i.e. service node and deadhead node as well as sink and source nodes) (Derigs, Malcherek and Schafer 2010).

Figure 37 shows the graphical illustrations of the diagrams presented in Figure 36. Each node represents a trip or an activity and has such attributes as start location, end location as well as start and finish time. The red nodes indicate the trains to which the drivers need to be assigned and the purple nodes are the passenger trains, which are used as deadheads. The green, source and sink, nodes are the dummy nodes and only symbolise the beginning and the end of the diagrams (depots). In this example the graph illustrates only the nodes participating in the example on the table, but in reality the connection graph

should include all possible transfers including taxi services which can connect all the subsequent work activities.

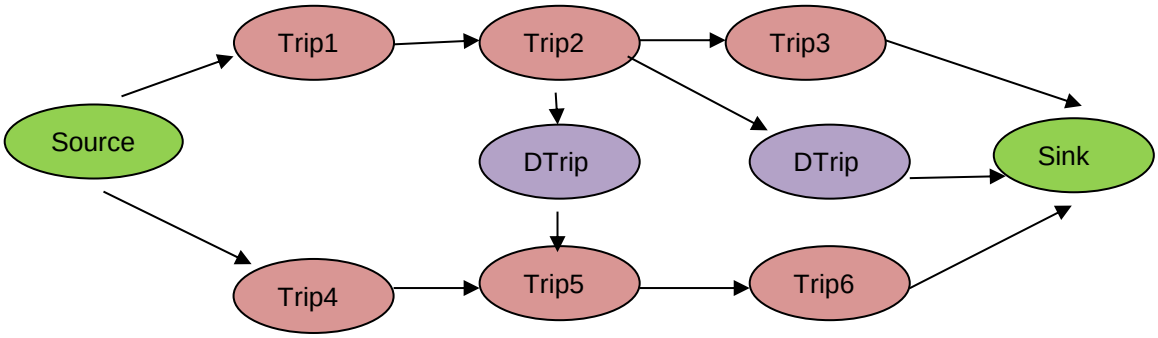


Figure 37 Connection graph

Klabjan et al. (2001), Chu, Gelman and Johnson (1997), Nishi, Muroi and Inuiguchi (2011) employed an alternative graph, **time-space network**, to represent the problem.

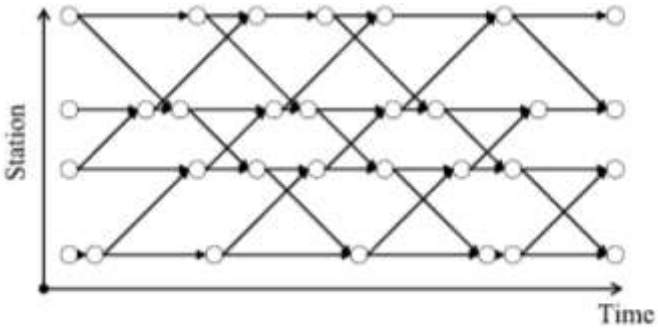


Figure 38 Time space network

Source: Nishi, Muroi and Inuiguchi (2011)

The network is similar to the connection graph except that two nodes instead of one should be allocated for each trip in order to signify the beginning and the end of each trip.

In both graphs, the arc between two nodes exists if it does not violate any of the problem constraints. The typical constraints are: the subsequent trip starts later than the previous trip finishes and there is enough time for a required break or transfer to the next job (Abbink et al. 2011). Since each arc represents the later

in time connection, the graph is always acyclic. Therefore, the task is to partition the graph into non-disjoint parts (Emden-Weinert and Proksch 1999).

The breaks are usually determined once the whole path is built. However, for the typical European rail network this can result in a massive number of break combinations which can complicate the task even further (Drexl and Prescott-Gagnon 2010). Drexl and Prescott-Gagnon (2010) proposed the including of a special node for a break, which would pass this problem to the path generation stage. However, this approach might result in graph cycling and increase the time for path generation (Drexl and Prescott-Gagnon 2010).

Apart from breaks, Jutte et al. (2011) estimate the number of deadhead arcs can reach 20 million for the large crew scheduling problem in Germany, which dramatically increases the number of possible diagrams. In order to reduce their size Jutte et al. (2011) proposed a procedure as follows. Firstly, the benefits of the diagrams containing the arc to the overall schedule are calculated. After that, if the contribution of the column is positive then the arc remains, otherwise the arc is temporally removed from the graph but can be returned at the later iterations.

Abbink et al. (2011) suggest eliminating long deadhead arcs to reduce the complexity. Moreover, in order to speed up the process of column generations, they group the trips performed on the same train into one task. This enabled them to achieve 75% reduction of arcs and nodes on the rail network in the Netherlands.

7.2.4 Diagram generation with pricing problem

The problem of diagram generation for the solution of a linear program formulated in Formula 10 and Formula 11, is called ***pricing problem***. The pricing problem for CSP usually fits into the model of ***finding shortest-path with resource constraints*** (Yan and Chang 2002, Desaulniers et al. 1997, Abbink et al. 2011). The mathematical formulation designed by Desaulniers et al. (1997) is displayed below.

$$\text{Minimize} \sum_{(i,j \in A)} c_{ij}\, x_{ij} \qquad \text{(Formula 16.1)}$$

$$\text{subject to} \sum_{j:(1,j \in A)} x_{1j} = 1 \qquad \text{(Formula 16.2)}$$

$$\sum_{j:(i,j \in A)} x_{ij} - \sum_{j:(j,i \in A)} x_{ji} = 0 \; i = 2,3 \ldots n-1 \qquad \text{(Formula 16.3)}$$

$$\sum_{i:(i,n) \in A} x_{in} = 1 \qquad \text{(Formula 16.4)}$$

$$\sum_{(i,j) \in A} t_{ij}\, x_{ij} \leq T_{shift} \qquad \text{(Formula 16.5)}$$

$$x_{ij}\, \{0,1\} \qquad \text{(Formula 16.6)}$$

The equalities (Formula 16.2) and (Formula 16.4) ensure that the graph starts and ends at the source and sink nodes respectively. The constraint (Formula 16.5) makes it different from the classical shortest path model by imposing the restriction on the maximum path duration to limit the diagram duration (Abbink et al. 2005).

Despite the fact that the graph is acyclic and all the weights are non-negative, it is still extremely challenging NP-hard problem (Pugliese and Guerriero 2013). Various network traversal and pricing techniques were developed to increase the effectiveness of the solution to this problem (Lavoie, Minoux and Odier 1988, Yan and Chang 2002). For instance, Abbink et al. (2011) splits the connection graph, such as the one illustrated in Figure 37, into several parts and solves the shortest path problem with resource constraints on different processors.

7.2.5 Label-setting algorithm for diagram generation

Label-setting algorithm was developed by Desrochers et al. (1988). It is another set of techniques enabling generation of the diagrams, which consist of the sequence of being separated in time jobs.

Label setting algorithm iterates through all nodes in topological order storing the information about the visited paths and resource consumption on each node in a label. ***Resource*** is "an arbitrarily scaled one-dimensional quantity that can be

determined or computed at the vertices of a directed walk in a network" (Drexl and Prescott-Gagnon 2010, p.85). ***Label*** contains information of each path and consumption of each resource, which is updated at each iteration (Pugliese and Guerriero 2013).

With regard to the diagram construction, the labels usually carry such information as driving time, diagram duration, last time of a break etc. (Figure 39). At each iteration a node is selected and all the labels are extended to the successor node. However, the partial path can be extended to the next node if none of the labels violate regulations.

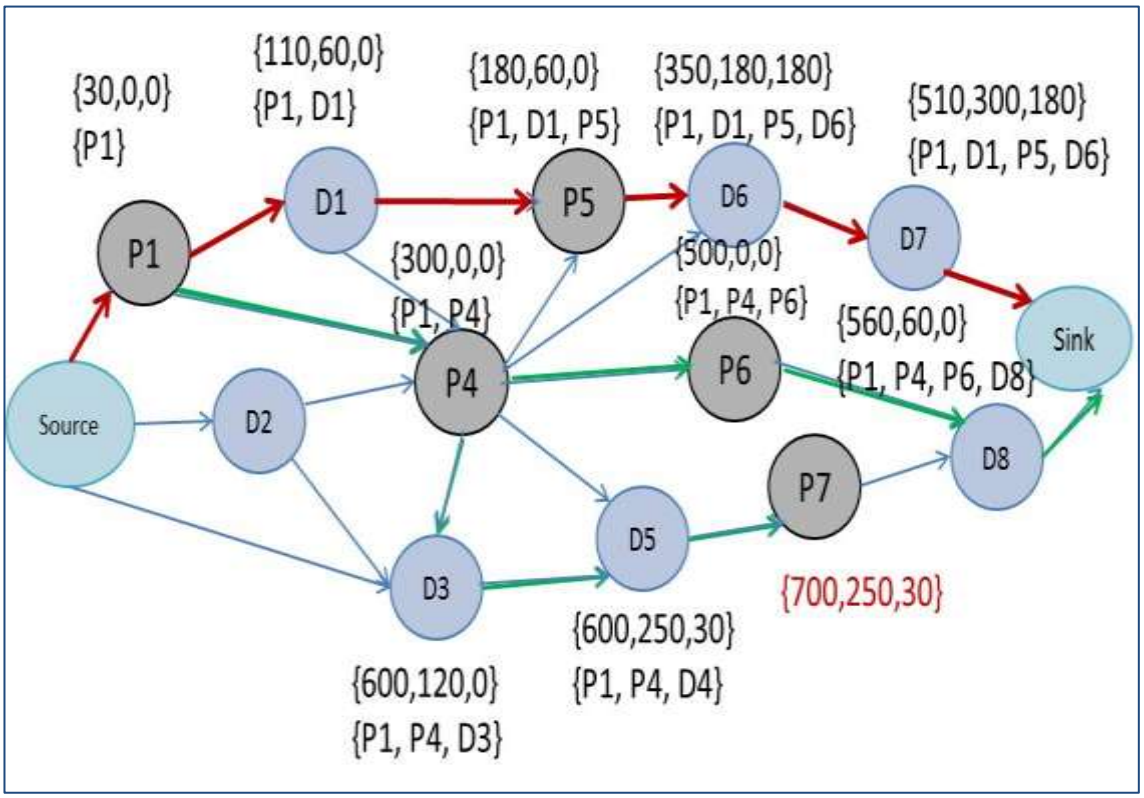


Figure 39 Label-setting algorithm

Figure 39 illustrates the logic of the labelling algorithm. Along with a number of visited nodes each label accumulates the information regarding diagram length, aggregate driving time and time of the last break. The path P1, D1, P5, D6, D7 (red) is legal and satisfies all the regulations. However, the path P1, P4, D3, D5, P7 violates the maximum diagram duration and is backtracked back to P4, where it is extended to P6 and D8.

In order to increase the effectiveness of this technique and reduce CPU resources, Desrochers et al. (1988) included ***dominance rules***. The dominance

rules define the characteristics of the promising (i.e. dominant) paths, which are extended first. Duck, Wesselmann and Suhl (2011) suggest restricting the number of backtracking steps. Klabjan et al. (2001) propose a random diagram generation mechanism and probabilistic node selection. The node to which the path would be extended is selected with a certain probability. The probability is higher if the connection time is less.

Nevertheless, in this case the ideal situation would be if the search could know in advance whether the path would lead to a high quality diagram or not. Having this information, the algorithm would be able to stop propagation of poor quality paths earlier saving a significant portion of computation time. In order to implement this idea, the pruning method has been proposed.

Pruning is "a procedure that fathoms depth-first search of a partial diagram before the diagram is actually obtained" (Makri and Klabjan 2004, p.59). Its key objective is to predict and detect unproductive branches as soon as possible.

To achieve this, Goumopoulos and Housos (2004) introduce an indicator called MAP (minimum available time to complete a partial path). Traversing the graph, MAP is calculated at each node and displays how much time is left in the shift. If the accumulated time is approaching the maximum shift duration and the driver is "too far" from the base depot, then it indicates that it might not be worth extending the path and the algorithm should start building other paths. In order to reduce execution time further, Goumopoulos and Housos (2004) implemented a procedure which places all the rules in an order starting from those which are more often violated in order to check them first and save time on the label calculations.

Although pruning methods reduce the time for generation of new diagrams, they also can possibly reduce the search space and miss a diagram valuable for the schedule. The example of a such situation would be when a path, which could result in a cost-efficient diagram, gets abandoned before reaching to next node. This case precludes finding an optimum solution.

7.3 Diagram set selection

After all the diagrams have been generated using one of the principles defined in sections 7.2.3-7.2.5, the master problem formulated in 7.2.2 is solved through

LP-relaxation. As LP-relaxation removes integrality constraints (Formula 10.3 and Formula 11.3) from formulas 4 and 5 to simplify the computation, in almost all cases of the railway scheduling problem, the solution will contain fractional variables (Figure 40) (Beasley and Cao 1996). While the number of non-zero shifts gives a target number of diagrams and an idea of how many drivers approximately are required to cover the train trips, the non-binary nature of the variables makes the decision whether the diagram would constitute a schedule or not quite ambiguous.

	Diagram 1	Diagram 2	Diagram 3	Diagram 4
Trip 1	1			1
Trip 2	1	1		1
Trip 3	1	1		
Trip 4			1	
Trip 5			1	1
Trip 6			1	1
Solution vector	0.6	0.1	0.25	0.2

Diagram content (1 indicates included trips)

Indicates whether the diagram is included into the schedule

Figure 40 Continuous solution

Identification of the integer solution is usually a computationally intensive task because it requires investigation of a large number of combinations. Moreover, if LP is unable to find a solution within a target number of shifts, the process usually terminates, a shift target increases and the process starts again (Kwan, Kwan and Wren 2001). Below the methods allowing for that and to find the integer solution are presented. They all utilise branch-and-bound methodology devised by Land and Doig (1960).

7.3.1 Branch-and-bound

Branch-and-bound is one of the most popular methods for the solution of integer combinatorial optimisation problems (Hillier 2005). Branch and bound is able to find an exact solution for the small size problems. The main idea of this method is the gradual split of the search space into subsets (branching) and calculation of the (bounds). Bounds indicate how good the solution in the region can be and allows for the elimination of the regions which do not contain the optimal solution. After one of the regions was discarded, the remaining area is split again. The process repeats until the final solution is reached (Algorithm 2). This mechanism

avoids exhaustive search and gradually narrows the search towards the integer solution, while still ensuring that the optimal solution is not discarded.

Algorithm 2 Branch-and-Bound

1: Find $x^{(0)}$ by Solving the initial problem L_0 removing the integrality constraints. It L_0 has no solution then the whole problem does not have the solution.
2: Calculate the lower bound $\xi^0 = f(x^{(0)})$. If $x^{(0)}$ is integer, than $x^* = x^{(0)}$ and $f^* = \xi^0$ and the algorithm terminates. If $x^{(0)}$ is not integer, than $\theta^0=+\infty$, k=1 and go to the step 3.
3: Chose a v-node for the branching (often for which $\xi^v = min\, \xi^i \text{ where } i \subset I$)
4: Select arbitraly one of non- integers $x_r^{(v)}$ and start branching creating L_{2k-1} and L_{2k}
5: Solve L_j, j=2k-1,2k. If L_j does not have a solution then $\xi^j=+\infty$, $\theta^j=\theta^{j-1}$, k=2 go to step7.
6: Find $x^{(j)}$ and calculate $\xi^j = f(x^{(j)})$. If $x^{(j)}$ is integer than algorithm terminates, otherwise $\theta^j=\theta^{j-1}$, k=2 go to step7.
7: Review the nodes and branching stops if $\xi^t=\theta^{2k}$
8: Check the termination condition. if I=Ø then $f^*=\theta^{2k}$, $x^*=x^{(v)}$ where $x^{(v)}$ determined from the $f^*(x^{(v)})= \theta^{2k}$ and the algorithm terminates.
9: Otherwise, k=k+1 and go to step 3.

Christofides, et al. (1979) state that the branch-and-bound concept allows for the addition of various heuristic rules and search strategies, which often are essential elements in search facilitation and acceleration. One of the limitations of this method is that estimation of the bounds can be computationally expensive (Klabjan et al. 2001). Another drawback is that the only way to recognise the optimal solution is to calculate the next solution. If the subsequent solution is not better, then it is concluded that the algorithm attained the optimum. There are no precise instructions of which node should be examined next. Nemhauser and Wolsey (1988) suggest two types of rules that can be applied: ***a priori*** (determine the rule in advance such as for example last in, first out) and ***adaptive*** rules that are based on using information about bounds. Beasley and Cao (1996) observed that the branch with the lower bound value is more likely to lead to the optimal solution.

Nevertheless, due to the large size of real crew scheduling problems, the "pure" branch and bound is still not practical (Gopalakrishnan and Johnson 2005).

Several modifications of this method have been developed in order to improve this method.

7.3.2 Branching strategies

The choice of branching strategy determines the computational cost of the algorithm (Klabjan et al. 2001). The standard approach for branching is to fix one part as one and another as zero (Hillier 2005). However, this is not preferable for CSP because it is a very lengthy approach, which does not exploit the problem structure (Barnhart et al. 1998, Kwan 2004). More effective branching techniques relying on problem specific information developed in the literature are discussed below.

- ***Timeline branching*** has been developed by Klabjan et al. (2001). The set of diagrams containing a certain trip is divided into two sub-sets: the first comprises of all the diagrams where the connection time between the given trip and the preceding trip is lower than a specified value, and the second where the connection time is larger. All the variables from the first set are fixed to zero in the first branch and to one in the second. In a similar manner, all the variables from the second branch are set as one in the first branch and zero in the second branch.
- Rayn and Foster (1981) create ***Follow on branching*** strategies based on the FORCE and FORBID approach. It works as follows. Suppose we have two diagrams with the fractional values in LP relaxation that cover consecutive trips: the first diagram contains the first trip and the second diagram contains the second trip. FORCE branch requires all these trips to belong to one diagram, whereas FORBID branch prohibits it (Gopalakrishnan and Johnson 2005, Derigs, Malcherek and Schafer 2010).
- ***Strong Branching*** is performed as follows. For each variable which has a fractional value two branches are created and a certain number of simplex iterations are performed. The value which demonstrated the better performance on both branches becomes a candidate for branching (Klabjan et al. 2001).
- ***Strong follow on branching*** is a combination of both strong and follow on branching strategies (Klabjan et al. 2001).

- ***Relief opportunity branching*** was developed by Kwan (2004). Relief opportunity is the place between the jobs where a driver changes the train. The relief opportunity branching approach works as follows. If two or more shifts containing the same relief opportunity has a fractional value, then they are all divided into two branches: one having this relief opportunity and the other one not. Because this approach might sometimes fail to provide an exact solution, the FORCE and FORBID approach is then utilised.

7.3.3 Branch-and-price and Branch-and-Cut

Because of a large number of columns and constraints in the CSP, it is almost impossible to process all the diagrams at the same stage. For this reason, branch-and-price and branch and cut methods have been proposed. They enable effective management of constraints and columns and still attain the feasible and mathematically optimum solution for integer programs (Gopalakrishnan and Johnson 2005, Barnhart et al. 1998). Both methods work with a "simplified" version of the problem gradually adding complexity if necessary.

Branch and cut is the combination of branch-and-bound and cutting plane method. Since the more constrained the problem the more difficult it is to solve it (Reeves 1993), branch-and-cut allows omission of the majority of the constraints in the beginning and only adds them if it is necessary (Duck, Wesselmann and Suhl 2011). If the obtained solution with a limited number of constraints is feasible, then it is passed to branch-and-bound to find the best integer solution in that region. Otherwise, additional constraints are added until the feasible solution is identified. The basic concept underlying this approach is that in some linear programs some constraints can be superfluous and would not affect the solution. This situation is illustrated with a trivial example in Figure 41. In that instance the feasible region and thus the area of a solution is bounded by constraint A and B, hence constraint C (blue) would be automatically satisfied and can be removed.

Algorithm 3 Branch-and-Cut

1: Relax the majority of the constraints.
2: Solve LP.
3: If the solution is feasible and integer then the algorithm terminates. If the solution is feasible, but not integer then go to Step 6. If the solution is infeasible go to Step 4.
4: Separation. Using the cutting plane algorithm, find and add some of the violated constraints in order to tackle the infeasibility.
5: Solve new LP. Go to Step 3.
6: Branching. Split the problem into two sub-problems and go to Step 2.

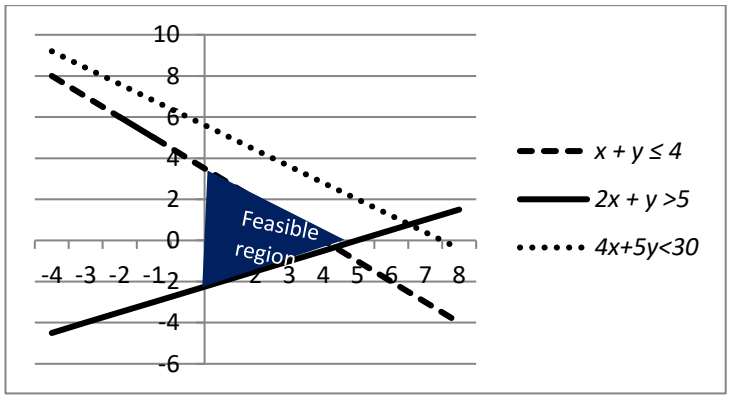


Figure 41 Linear program with constraints

Branch and price is another variation of the branch and bound algorithm (Algorithm 4). Branch and Price is a mixture of the branch-and-bound concept and the column generation approach (Barnhart et al. 1998). Branch-and-Price is applied more often for the solution of Crew Scheduling Problems than Branch-and-Cut. This might be because the number or rows is considerably smaller than the number of columns. Therefore, from the computational perspective it is more efficient to handle hundreds of inequalities than to generate and deal with millions of columns.

Algorithm 4 Branch and Price

1. Generate a limited number of columns.
2. Solve RMP.
3. If the solution is feasible and integer then the algorithm terminates. If the solution is feasible, but not integer then go to Step 6. If the solution is infeasible go to step 4.
4. Pricing. Generate and add new columns.
5. Solve new RMP. Go to Step 3.
6. Branching. Split the problem into two sub-problems and go to Step 2.

In theory, it is possible to combine branch-and-price with branch-and-cut. However, in practice it is quite a challenging task because the generation of new columns can break the row constraints (Barnhart et al. 1998). Duck, Wesselmann and Suhl (2011) designed an algorithm synthesizing branch and price with branch and cut for the airline crew diagram optimization problem. They observed that consideration of rows and column generations enables better decision making in terms of branching and less iterations were required to find a near-optimal solution.

7.4 Metaheuristic methods

Metaheuristic methods allow for finding near optimal solution for large and complex problems considerably faster than exact methods (Alabas-Uslu and Dengiz 2014). This is because they do not require generation of all diagrams and can determine the direction of the search based on the small sample of solutions in that area. This makes them suitable for the solution of real life CSPs (Gogna and Tayal 2013). Such methods as GA, Simulated Annealing and Ant Colony Optimisation were proposed in the academic literature to solve CSP. Their configurations and performance are discussed below.

7.4.1 Simulated Annealing

Emden-Weinert and Proksch (1999) apply the SA algorithm to tackle the diagram generation stage of CSP in the airline industry. A standard SA was augmented with a local search procedure called ***Disturb***. Infeasible solutions were allowed, but significantly penalised. With regard to the SA parameters geometric cooling schedule similar to the one described in Table 2 was used. The initial solution was obtained with the constructive heuristic which orders all consecutive flights.

As the initial solution is usually of a poor quality, the local search operator plays a great role in developing it. Selecting a diagram, Disturb examines all the diagrams with the intention of finding a diagram which can be concatenated with the selected one. Concatenation is only permitted if the succeeding diagram starts at the same location and later than the given diagram. If more than one diagram was detected, then the preferences are given to the diagram with the earliest start. On both tests instances used in the study, SA with Disturb operator showed a better result than SA without it. Such results were attained probably because by combining two diagrams into one, the number of drivers required and the number of deadheads can decrease in the solution. However, the given approach might cause unequal distribution among the depots and drivers, which were not taken into account in the study.

Hanafi and Kozan (2014) combined SA with a constructive heuristic to solve a railway crew scheduling problem. The heuristic procedure firstly analyses the train routes and timetables. If the train starts at one depot, and then returns there within acceptable shift time, then the entire diagram would be based on that train. In the situation, where the train journey is longer than the allowed driver working time, the segment of the trip surpassing maximum diagram time is cut off and placed in the pool for later allocation. At the second stage, all the segments are inspected and grouped into duties of other drivers. Neighbourhood generation in this algorithm is performed by exchanging trips between randomly selected diagrams. The trips can be exchanged only if they start and end in the same locations. Then a random trip from the set of all trips is inserted into another diagram. This only can be a trip arriving and departing from the station with a local connection. Experimental results demonstrate that when this heuristic is incorporated into the SA framework, the algorithm is able to achieve 3%-4% better results than on its own.

This approach significantly reduces the number of trips to be scheduled and thus the size of a problem. The evident limitation of this method is that there might be long breaks in the train schedule (e.g. for maintenance). It is possible that during this time a driver could have been assigned to another train and come back to drive the first train to the home depot. However, such logic would prohibit the assignment of the driver to another train. Another drawback of this idea is that the leftover segments might be geographically separated, which would either

require usage of other means of transportation to keep the number of drivers to a minimum, or to utilise more drivers from various regions.

7.4.2 Ant colony optimisation

Deng and Lin (2011) propose ACO for the solution of the crew scheduling problem in the airline industry. They expressed the problem as finding the shortest path in the graph similar to the Travelling Salesman Problem and applied a standard ACO. The algorithm was tuned by experimentally verified parameters. The algorithm terminates when all the ants use the same path in the solution. The given algorithm showed better results than EA designed in Ozdemir and Mohan (2001), however it is difficult to establish the reason for that as the logic of crossover operator used in EA was not fully presented.

7.4.3 EA algorithm

As shown in the section 7.2, the CSP is usually solved in two stages. The first stage is responsible for the generation of a large number of candidate diagrams. The second stage deals with the selection of the shifts which would constitute a schedule. The majority of the EAs were designed to tackle the second, optimisation, stage of the problem. Only one EA has been found in the literature for the diagram construction step. There were no EAs capable of handling both stages simultaneously. All these types of EAs and their configurations are considered below.

7.4.4 EA for diagram generation

Santos and Mateus (2009) incorporate EA in conjunction with integer programming into column-generation approach for the solution of the pricing problem. The advantage of incorporating EA is its ability to return more than one column at the same iteration as it works with several solutions simultaneously. This accelerates the search and reduces computation time.

7.4.5 EA for optimization

Levine (1996) was one of the first researchers to apply EA for the diagram optimisation stage of a Crew Scheduling Problem. He developed a classic GA with binary chromosome representation and a local search heuristic proposed by Beasley and Chu (1996). The given algorithm was able to find an optimal solution

in 50% of all cases, while branch and cut solved them all. Later numerous EAs were devised for this problem. The main differences between them are the way they represent the solution and the way they perform crossover and mutation operators. These methods are considered in the following sections.

7.4.6 Chromosome representations

In the ***row based representation*** each gene stands for a train trip, thereby the length of the chromosome is equal to the number of rows in the matrix presented in Figure 36. Scanning the chromosome from the left to the right, the decoding procedure selects the column (diagram) which covers the trip and best suits the existing partial schedule. The choice of the column depends on its cost, how many other undercovered rows it covers and how many rows it covers in total.

As Aickelin (2002) noticed, this approach might be biased towards lower cost columns. This might result in search space not being fully explored as some of the columns can never be included in the solution. In order to overcome this issue, Aickelin (2002) added the second part of the chromosome (Figure 42), which contains the weights of column selection criteria. The second part of the chromosome undergoes evolution as well, but the values are set randomly at the first iteration. As the second part of the chromosome has a different structure, the special crossover and mutation mechanisms were employed. The crossover operator copies all the weights from the fittest parent to the child. The mutation replaces an arbitrary selected gene with a random value.

The limitation of the given approach is it might express the same solution in different forms. In addition, Zeren and Ozkol (2012) state that genetic operators often produce offspring with many overcovered rows meaning that too many drivers will be assigned to one trip. This can result in a large number of deadhead trips. Finally, this approach seems to overcomplicate the problem with the evolution of the trips, which in its core evolves the columns.

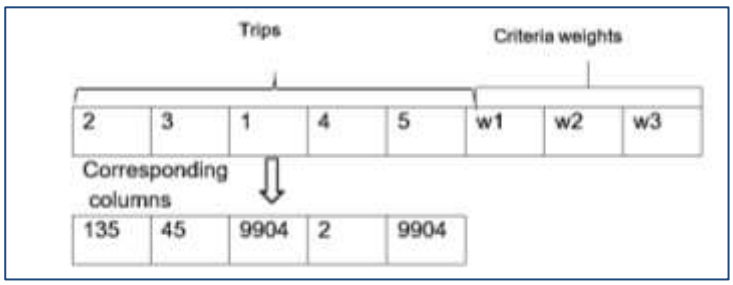


Figure 42 Chromosome representation with weighting criteria

In the ***column-based representation*** (Figure 43) a chromosome is expressed as a binary vector of a length equal to a number of generated columns, which were displayed in Figure 36. The locus of the gene denotes the index of the column (shift). The binary allele indicates whether this shift is included in the schedule or not. Usually the length of the chromosome is very long and consists mainly of zeros. This is because a typical number of generated shifts can be between 30000 and 75000 with only about 100 of them included in the final schedule (Kwan, Kwan and Wren 2001).

1	0	0	1	0	1	0	0

Figure 43 Binary Chromosome representation

The more compact form of column based representation is depicted in Figure 44. The vector of integers contains only the indexes of the shifts comprising the schedule (Kwan, Wren and Kwan 2000, Wren and Wren 1995).

1	4	6

Figure 44 Integer Chromosome representation

The chromosome initialisation stage presents some challenges. An entirely random generation of chromosomes in the population might result in very slow convergence and infeasible schedules. As the number of diagrams is unknown in advance, it is not clear how many genes should be in a chromosome and how many of them should have “1s”.

Levine (1996) used a simple logic to establish the probability of 1s. Observing existing solutions and manually produced diagrams, he determines the average number of trips constituting the diagrams. Thereby, given the total number of trips in the schedule and average amount of trips in each diagram, he obtained an approximate number of diagrams. However, as this approach cannot provide an

exact number of diagrams, he utilised this number as only a probability with every "1" is generated.

Zeren and Ozkol (2012) developed a heuristic aiming at achieving a solution with minimum trip overlapping and number of deadheads when generating a chromosome. For each uncovered job, a randomly chosen diagram containing that job is added to the solution. After each insertion the set of uncovered flights is updated. The process repeats until all the flights are covered in the schedule. The procedure is very similar to the row-based representation developed by (Aickelin 2002).

Kwan, Wren and Kwan (2000) used the continuous relaxed LP-solution as a starting point (Figure 40). Using the information about the target shifts obtained from LP-relaxation, they suggest including 25% of the estimated number of diagrams. However, the diagrams themselves are selected randomly. It is evident, that in many cases randomly selected diagrams might not cover all the trips from the train schedule, so Kwan, Wren and Kwan (2000) apply heuristic FILL and DISCARD procedure to identify the rest of the columns. This heuristic procedure is discussed in more detail in Section 7.4.11.

Shen et al. (2013) apply the same principle of 25%. In addition, because the number of drivers is unknown, they developed an adaptive chromosome representation which allows expansion of the length of the chromosomes in the later generations. The algorithm first starts with the initial length as a lower bound obtained by LP-relaxation. Then if it is unable to return a feasible solution during a pre-defined number of iterations, an extra gene is then added to the chromosome (Figure 45).

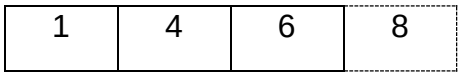


Figure 45 Adaptive chromosome representation

Park and Ryu (2006) introduce the concept of unexpressed genes for the subway crew scheduling problem (Figure 46). Unexpressed genes are the genes which are good on their own, but do not fit well with other diagrams in the schedule. While both expressed and unexpressed parts are involved in crossover and mutation procedures, only the expressed part is used for the calculation of fitness function. The unexpressed part preserves information which might be lost after

the application of genetic operators. Although experimental comparison of a maximum of 634 trips demonstrated that the suggested algorithm outperformed simulated annealing and Tabu search algorithms, EA developed by Zeren and Ozkol (2012) showed better results when tested on a larger data set.

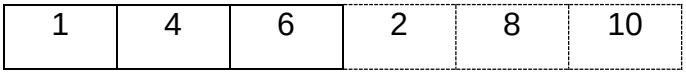


Figure 46 Expressed and Unexpressed genes

To conclude, the column based chromosome representation is fit for purpose as it clearly expresses the problem domain and is capable of reflecting the optimal solution. However, there are two limitations. First of all, a randomly generated solution and chromosomes produced by standard genetic operators usually violate the constraints. Secondly, the exact number of genes is unknown and therefore additional operators are required in order to restore the feasibility.

Another type of chromosome representation, graph-based chromosome representation was devised by Ozdemir and Mohan (2001) and is illustrated in Figure 47. In this case, the genes represent individual trips rather than diagrams. The chromosomes are initialised with graph search procedure which groups the trips into the diagrams. In this EA, mutation employs logic similar to 1PX crossover operator, but performs it on a single individual. The algorithm also includes three crossover operators: set-based, time-based and distance preserving operator. The limitation of this chromosome representation is that not all the trips can be covered after the first round, and thus some restoration procedures are required.

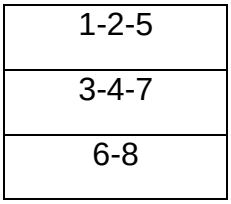


Figure 47 Graph-based chromosome representation

7.4.7 Selection

Binary tournament is one of the most popular selection mechanisms in crew scheduling EAs. It can be found in the works of Ozdemir and Mohan (2001), Levine (1996), Zeren and Ozkol (2012), Chu and Beasley (1998). There are no

empirical results confirming its effectiveness, but the possible explanation of its popularity is its relative simplicity of implementation and speed of execution.

Roulette-wheel selection is a bit less popular, but still has been utilised in crew scheduling EAs in Kornilakis and Stamatopoulos (2002), Kwan, Wren and Kwan (2000), Shen et al. (2013).

In order to increase the efficiency of the selection process, Chu and Beasley (1998) proposed ***matching*** selection for a general set partitioning problem. The first parent is chosen through the tournament selection while the second one is selected on the basis of compatibility with the first one. The ***compatibility score*** is calculated for each individual in the population apart from the one already chosen. The high compatibility score implies that both parents cover the maximum number of trips together, but contain a small amount of the same trips. If more than one chromosome has maximum fitness, than the least fit individual is chosen for reproduction (Chu and Beasley 1998).

7.4.8 Crossover

Due to mostly integer or binary chromosome representations being utilised in the algorithms, standard crossovers operators are often built into the algorithms.

For example, Levine (1996) applied 2-point crossover and Shen et al. (2013)'s EA is based on one-point crossover. Kornilakis and Stamatopoulos (2002) used ***uniform*** crossover. Uniform crossover works as follows. If a certain allele in both parents has the same value, then it is copied to the child. If the values are different, then the only child inherits the gene from one of the parents with the specified probability.

Zeren and Ozkol (2012) used ***fusion crossover*** proposed by Beasley and Chu (1996) for the solution of the general set covering problem. This crossover is very similar to the uniform crossover with the only exception that the probability from which parent the gene will be taken directly depends on the fitness of that parent. The fitter the parent, the higher the probability that it will pass its gene on to a child. The possible limitation of both uniform and fusion crossovers is it will not be able to create different children when the population converges. This might cause premature convergence of the algorithm.

Park and Ryu (2006) designed a crossover which greedily selects genes from both parents and passes them to the child. First this crossover puts all the genes from both parents together. Then it evaluates the value of each gene in relation to the already existing genes in a chromosome. The value is measured by how many uncovered rows can be covered by each gene. The algorithm always gives a preference to genes which have a minimal overlapping with the rest of the genes in a new chromosome.

Cleary, in all of the above-mentioned crossovers there is a high probability that formed offspring will violate some of the problem constraints. Therefore, additional operators and procedures restoring the feasibility are usually applied. They are discussed in section 7.4.11.

7.4.9 Mutation

Park and Ryu (2006) proposed a ***k-exchange*** mutation. It is based on the REMOVE and INSERT principle. It begins by deleting some diagrams with a certain probability. The probability would be lower if the diagram removal will cause a significant increase in the uncovered rows. Then, the insert stage looks for *k* duties to form a candidate pool, which would not only cover the uncovered trips, but would also cause a minimum coverage of already covered trips. While this mutation can contribute to the elimination of redundant diagrams, it requires calculation of the score for each gene in a chromosome every iteration when mutation is performed. In addition, INSERT operation can also consume memory resources as it works with an enormous set of candidate shifts.

As a mutation operator Shen et al. (2013) use a procedure which is very similar to the perturbation operator proposed by Zeren and Ozkol (2012). They remove one of the diagrams from the mutated chromosome and then search for the better replacement though specially designed heuristics.

In order to maintain a number of the diagrams, the selected genes in the chromosome for mutation are inverted to one with the probability equal to the ratio of the diagrams included in the solution to the total number of diagrams in the candidate pool (Kornilakis and Stamatopoulos 2002).

In Kwan, Kwan and Wren (2001) the mutation rate depends on the total number of remaining iterations before the end of the run, the amount of iterations for which

the fittest member survived, and a special control parameter. In general, the mutation frequency increases as the algorithm progresses. This contributes to the reduction of premature convergence possibility. The mutation itself is performed by replacing a random number of genes with an arbitrary selected probability.

Zeren and Ozkol (2012) propose a method for the calculation of the mutation probability (the number of genes in the chromosome to be mutated). This value derives from chromosome length and the number of new chromosomes since the first iteration (Kornilakis and Stamatopoulos 2002)

Because both algorithms have not been compared, it is not possible to conclude analytically which of them would be more effective.

7.4.10 Constraint handling and infeasibility

Since the CSP is the highly constrained problem, developed crossover and mutation operators are unable to always produce legal offspring (Zeren and Ozkol 2012, Kornilakis and Stamatopoulos 2002). Several strategies for dealing with infeasibility have been suggested.

7.4.11 Repair operators

The majority of EAs for CSP employs heuristic operators to restore the feasibility of the chromosomes. They work by the principle ADD and REMOVE first mentioned by Chu and Beasley (1998). The operator starts by scanning the chromosome in order to determine undercovered rows. After that it identifies the diagram from the set of all diagrams, which can cover the trip. Then, this procedure eliminates redundant diagrams from the schedule, where all the trips are already covered by other diagrams.

The enhanced procedure called FILL and DISCARD has been applied by Kwan, Wren and Kwan (2000). Comparing the continuous solution with the final integer solution, Kwan, Wren and Kwan (2000) found that 50%-74% of the shifts included in the optimum solution had non-zeros value in the continuous solution. Furthermore, 75% of non-zero shifts, which were in the integer solution had a value greater than 0.2 in the continuous one. The shifts with this value and above were named ***preferable***. In order to reduce the number of shifts, Kwan, Wren and

Kwan (2000) suggested removing the shifts with ***a relief opportunity*** (a place where drivers can change), which is not included in the preferable shifts.

The benefit of this method is that it not only maintains feasibility, but also performs an operation resembling typical local search. However, the feasibility operator should be applied at each iteration. Given the number of trips and the number of shifts which needed to be scanned and analysed at each iteration, this approach might be computationally expensive for real life problems especially in the rail industry (Kwan 2004).

Instead of treating the constraints on the genotype level as in the case with repair operators, some researchers perform it via penalty function. Allowing infeasible chromosomes in the population, it is possible to maintain diversity, while the penalty function will gradually ensure a non-domination of unfeasible individuals. The main objective is to select an appropriate penalty coefficient and to select the right approach. For example, devising penalty coefficients for CSP formulated as a set partitioning problem, Levine (1996) selected the value proportional to the cost of violation. If the trip occurs more than once in the schedule, then the penalty will be equal to the maximum cost among the diagrams containing the same trip.

However, Chu and Beasley (1998) noticed that while some of the chromosomes can be feasible, they might not be as fit as some of the unfeasible chromosomes in the population. For this reason, Chu and Beasley (1998) suggested separating feasibility from the fitness. They divided all the chromosomes into four groups: low cost and feasible (G4), high cost and feasible (G3), low cost and unfeasible (G2), high cost and unfeasible (G1). In the ranking replacement strategy, the chromosomes from G4 are replaced first, and if the group is empty then the chromosomes from the group G3 are replaced etc.

In order to avoid excessive calculations and verifications, Shen et al. (2013) proposed ignoring the feasibility of the chromosomes until the last iteration. Despite this approach reducing the computation time, there might be the risk that after fixing the feasibility of the chromosome not only the fitness can be lower, but the wrong chromosome could be extracted from the population as a solution.

While both methods can tackle constraint violation, they have several disadvantages. Repair operators are extremely time consuming for the real

problems. In terms of the penalty functions, there is no general formula allowing determination of the right coefficients, and ignorance of infeasibility might not return a practical solution at the end of the algorithm.

7.4.12 EA enhancement

The performance of the algorithm can be improved by design of additional operators.

Zeren and Ozkol (2012) devised a perturbation operator which acts as a local search. It makes a chromosome infeasible by removing one or more genes, and then searches for a replacement from a set of potential shifts with the aim of achieving a lower cost schedule. Despite requiring more CPU time per iteration, it enables EA to converge much faster to reach the solution more quickly.

Kwan, Kwan and Wren (2001) exploit ***combinatorial traits*** in the chromosomes. In general, combinatorial traits are genotype characteristics responsible for the good solution. They state that if the fittest individual survives during a certain number of iterations, then it probably signifies that this chromosome possesses some combinatorial traits. With regards to the CSP combinatorial traits consist of seeding shifts and relief chains.

In order to identify the seeding shifts, for each shift and each trip in it, the algorithm calculates how many shifts contain the same trip. The minimum and average number of trip coverage is calculated for each shift. Based on that, the more unique the shift is, the more chances of it being inserted into the chromosome. The offspring is produced from the candidates of seeding shifts. This concept allows minimisation of the number of overlapping shifts and direct the search to promising areas.

Relief Chain is the sequence of drivers, who operate the same vehicle on different train journey segments. Kwan, Kwan and Wren (2001) observe the correlation between long relief chains and a good solution by comparing the results obtained by ILP and EA. For this reason, relief chains are considered as a combinatorial trait as well. The concept of combinatorial traits allows more effective management of the large number of pre-generated shifts and demonstrated considerably better results compared to the EA without combinatorial traits.

In terms of the fitness function, Li and Kwan (2003) utilised fuzzy logic concepts for its computation. They identified several properties of the effective schedule (i.e. total work time, throttle time, number of diagrams etc.) and designed a membership function for each of them. All the membership functions were then aggregated into the single objective function. Although this approach is able to achieve various objectives, it has two limitations. First of all, accurate design of the membership functions requires conducting a large number of experiments. Secondly, this approach does not explicitly represent the actual cost of the schedule, which might be the ultimate objective for many companies.

7.5 Other metaheuristic approaches

Li (2005) designed a hybrid algorithm named a ***self-adjusting algorithm*** for driver scheduling. This approach is based on population concept coupled with a local search mechanism.

As an equivalent of the fitness function, they use solution ***goodness***. This approach works on a single solution and treats diagrams as individual chromosomes, while the diagrams all together constitute the schedule. The fitness of each diagram depends on other diagrams and it is recalculated at each iteration. Selection mechanism generates a number in the interval from 0 to 1 and removes all the shifts with the lower value. These shifts are returned to the pool of potential shifts. Following this step an analogy for mutation further removes additional genes with a certain probability. This probability does not depend on the number of rows covered or the goodness of the shift. The last step, called Reconstruction, restores the feasibility by returning some of the necessary genes from a pool. Although Li's (2005) approach delivered the solution faster than other EA's and ILP methods, the overall cost of the solution was 0.92% higher and the number of shifts (diagrams) remain almost the same (only 0.01% smaller).

Elizondo et al. (2010) proposed a constructive heuristic for the conductor scheduling problem in underground transport. It starts by generating two diagrams, and then adds the trips from the second diagram to the first in the interval between the trips in the first diagram. If it is impossible to augment the diagram with the trips from the second diagram, then the trips from other diagrams are inserted to fulfil all the gaps. While it does reduce the idle time which can contribute to the reduction of the total number of diagrams, in the context of

the rail industry this approach might be infeasible due to the large geographical distribution. First of all, if diagrams contain the jobs in different parts of the country, the driver might not have the knowledge to operate the trains. Secondly, even if the driver is trained on both regions, he would have to travel for a significant time to arrive for the next job, which would reduce the throttle time and increase the transportation cost.

7.6 Conclusion

The chapter has considered models and approaches for the solution of the CSP. The common approach of tackling this problem is splitting it into two sub-problems: generation of a large set of possible diagrams and then selection of a subset which forms the schedule. While it was shown that the first phase is relatively straightforward and can be searched with various graph traversing techniques, the second phase is more computationally expensive and requires a more sophisticated approach. In the literature. there are two broad categories of methods for attacking this stage: exact and heuristics.

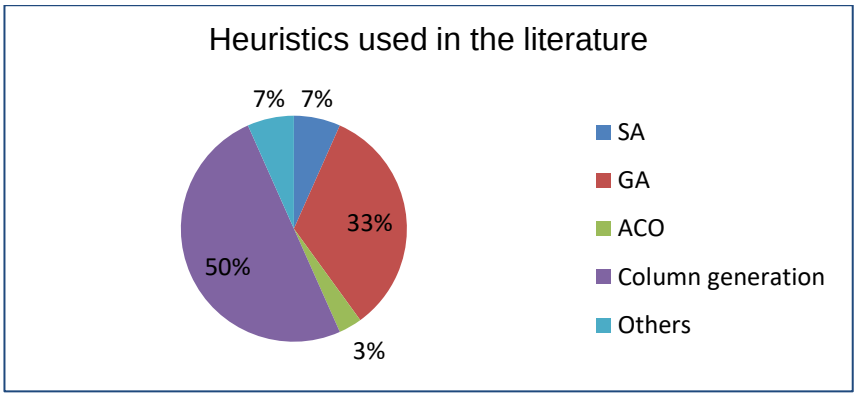


Figure 48 Heuristic used in the literature for CSP

Analysis of the literature has shown that the exact methods based on column the generation are the most popular methods for solution of the optimisation phase of CSP (Figure 48). They offer a number of advantages such as:

- In theory they are able to find a mathematically optimal integer solution for CSP.
- Some of them are able to work with incomplete set of columns and do not require generation of all possible diagrams in the beginning.

However, they also have a number of limitations presented below:

- Even though all the columns do not need to be generated a priori, generation of a subset of the required columns and their re-optimisation is still very time and memory consuming task (Zeren and Ozkol 2012).
- Tailing-off effect of a situation when columns are generated and re-optimised many times without resulting in significant improvements of the solution (Derigs, Malcherek and Schafer 2010).
- Risk of getting into local optima, which can prevent further improvements. The exact algorithms have no mechanisms of detecting and avoiding local optimum (Chu, Gelman and Johnson 1997).
- The general column generation framework works only with the "best" found columns, whereas in some cases a sub-optimal solution can lead to finding the optimal solution more quickly (Deng and Lin 2011, Ozdemir and Mohan 2001).
- Column-generation is only able to develop one solution at a time, compared to the population-based methods which approach the problem from different areas of the search space.
- From the practical perspective, the computer implementation of the branch-and-price algorithm is very complex and requires an expert with a significant prior linear programming knowledge (Barnhart et al. 1998). The person with the right skills might not be in the company.

The analysis showed that while in theory branch-and-price is an effective tool for solving crew scheduling problems, in practice this method might not provide any solution within the required timeframe. In the literature, the algorithms have been tested mostly on the airline data instances with the average size of 500 flights a day. And even there, some heuristic and pruning rules had to be developed to facilitate the algorithm to return at least a near-optimum solution. This is significantly smaller than the data arising in the railway industry.

For this reason, several metaheuristic algorithms were developed. For example, EA is one of the evolutionary computing algorithms, which is rapidly growing in the area of Artificial Intelligence. Kwan, Wren and Kwan (2000) utilised EA to set up parameters for the ILP of TRACS II (crew scheduling system). According to their empirical observations the standard column generation and branch and bound algorithms might fail to find a solution to complex problems with multiple

depots such as in the rail industry. For these problems, EA has been embedded in the system and runs in case standard procedure cannot find a solution. Based on the conducted experiments, the EA alone did not outperform ILP methods, however in some cases if was able to find a solution where other methods failed.

It is difficult to conclude with certainty, which of metaheuristic algorithm performs the best because the reported results were significantly influenced by local search procedures and different data instances.

However, compared to other algorithms, EA has a number of advantages. For instance, unlike SA, EA and ACO are population based methods, which are able to consider more than one solution at the same iteration. In addition, compared to ACO, EA is also able to directly combine the good characteristics of the solution by the means of a crossover operator. Moreover, steady-state EAs ensure that the best solution found in the algorithm will be preserved in the next generations in contrast to SA which can replace a good solution with a worse one with certain probability.

At the same time, despite the aforementioned advantages of EA, the analysis of the literature identified some limitations of EA based on so called Generate and Select approach. They are as follows:

- EA has a lack of control of the shift generation. No matter how powerful EA is, if the best diagrams have not been captured in the generation stage, it would be impossible to obtain a good solution without them.
- In addition, providing too small number of columns, the optimal solution might not be within them. On the other hand, if the number of columns is too large, it would take too much time to perform the search among them.
- They all contain an additional operator or other means to restore the feasibility given the chromosome representation and main genetic operators. These processes are very time consuming due to the size of the problem and the number of combinations which needs to be processed.
- Chromosome length and how many drivers should be in the solution are unknown and not all of the algorithms are able to dynamically change it.
- In addition, these approaches do not consider drivers' traction and route knowledge. This might result in limited practical applicability of the solution.

The research reported in this thesis aimed to exploit the numerous advantages of EAs while overcoming the limitations listed above. The method designed within the given research addresses all the discussed problems by, first of all, tackling the problem with only one stage (instead of generate and select approach). Secondly, the utilised chromosome representation and decoding procedure ensures feasibility during all time of the algorithm. This prevents wastage of the computation resources and time, which otherwise would be spent on the restoration of the chromosome legality.

Chapter 8. Evolutionary algorithm design

8.1 Introduction

This chapter describes the process of the algorithm development for the solution of CSP and JSSP. CSP deals with assigning the train drivers to the train trips in accordance with a large number of health and safety regulations. JSSP is concerned with assigning the jobs given the industrial process routes and technical constraints.

In order to develop an appropriate and efficient algorithm, rules guiding the design of an effective EA are established. After that, both problems are conceptually analysed in order to identify the components which they could share within the algorithm and those which need to be tailored to each particular problem. Based on that, the relevant standard genetic operators are selected and problem-specific ones developed. The framework, which will be utilised for operators and algorithm evaluation, as well as test instances are also presented in this chapter.

8.2 Key principles of EA design

Several researchers proposed the principles which allow design of an effective EA. Aickelin (2002) stated that one of the main objectives should be the minimisation of the number of iterations required to yield a good solution. This can be achieved through the construction of efficient genetic operators as well as ensuring that they are relevant for the chosen chromosome representation. According to Aickelin (2002), the quality of genetic operators can be measured using three characteristics:

- Computation efficiency (fast speed of execution).
- Determinism (the same permutation yields the same solution).
- Ability to discover the optimal solution in the solution space.

Furthermore, Davis (1991) states that incorporation of the domain-based heuristics and problem specific information can also increase the productivity of the search.

8.3 Analysis of the CSP and JSSS

In order to design a successful reconfigurable algorithm, it is important to understand the commonalities which both the problems share as well as any differences that exist between the two problems. Table 11 presents a conceptual comparison of CSP and JSSP.

Table 11 Conceptual comparison of CSP and JSSP

	Garnett-Dickinsons Production scheduling	DB-Schenker Logistics/transportation
What task needs to be assigned	Printing job, which consists of several operations.	Trip or any other train operating activity. This is a single job and cannot be broken down to smaller operations.
Who performs the task	Appropriate printing press.	Driver with relevant route and train type knowledge.
Rules	• Machines are available all the time • The operation of the same job cannot start until the previous operation of the same job has finished.	• Subset of drivers who are available on a particular day (who are not on annual leave, sick or performed a shift in the last 12 hours). • The trips and activities should begin at the specified time.
Variations	• No restriction on machines' maximum work time. • Maintenance is very rare and no break between the jobs required. • Machine can perform only one job at the time. • Jobs changeover is already included in the duration of the jobs and this does not need to be individually scheduled.	• Driver can work only a certain amount of hours. • Driver is required to have a break after a specified amount of time. • Driver can drive only one train, but can travel as a passenger on another train trip. • Different modes of transport can be used in order to get to the next job (travel as a passenger by passenger/freight train, taxi).
Objective function	Minimising makespan time.	Minimise total cost of the schedule.

Initial consideration shows that both problems belong to the class of assignment problems and they can be encoded into permutation chromosome representation. In such representation, each chromosome contains of a vector of integers, where

each integer (gene) stands for the task which needs to be scheduled. There are a number of reasons for selection of this chromosome representation:

- This style of representation is suitable for combinatorial optimisation problems (Sivanandam and Deepa 2008).
- It is one of the most popular representations for the scheduling problems (Hart, Ross and Corne 2005).
- As discussed in section 7.4.6, permutation chromosome representation is more compact than binary chromosome representation.
- Unlike symbol chromosome representation, it is not too problem specific and different problems can fit into it.

However, both problems have domain specific assignment rules which prevent application of the same algorithm to both problems. This requires the design of separate decoding procedures and unique formulas for fitness function calculations. But because the representation itself remains unchanged, similar crossover and mutation operators can be applied.

8.4 Proposed EA test framework

The framework of the algorithm which was designed to conduct the trials is presented in Figure 49. The EA, which is the core optimisation part, is shared by CSP and JSSP, while encoding and decoding procedures as well as fitness function are customised for each problem. Some initial trials showed that the 90% crossover probability and 40% mutation probability produced the best results. Similar values were used by various researchers including Amirthagadeswaran and Arunachalam (2007), Majumdar and Bhunia (2011), Mattfeld and Bierwirth (2004), Pezzella, Morganti and Ciaschetti (2008). These values will be used during all the experiments in this study.

Since there is no clear conclusion in the literature regarding the efficiency of genetic operators, several standard permutation operators will be tested using two procedures. The first set of trials will be conducted for a single crossover and mutation operator embedded into the algorithm. The second series of trials will investigate a performance of the synergies of the mutation and crossover operators, which are described in detail in section 8.4.4.

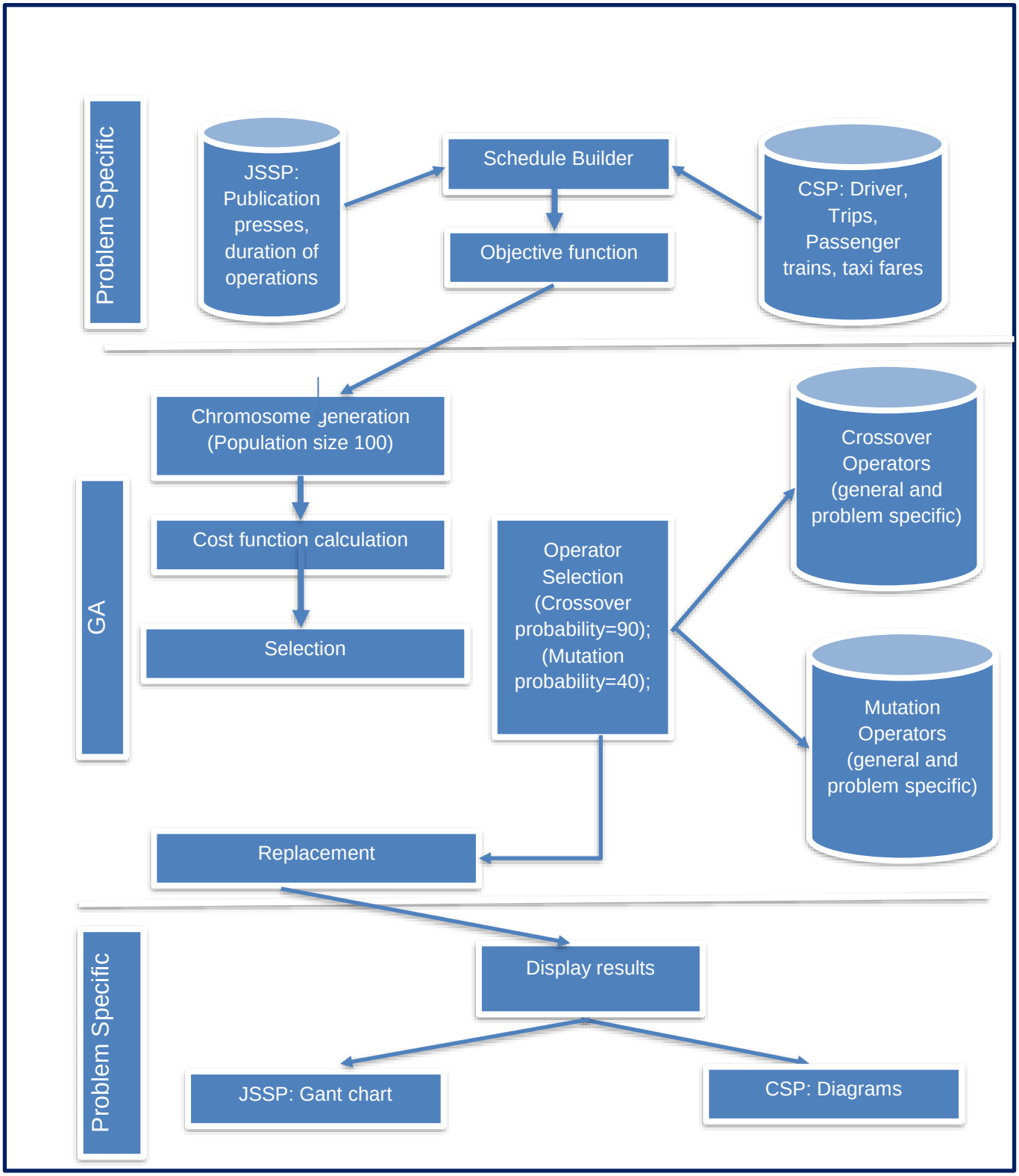


Figure 49 EA for CSP and JSSP test framework

8.4.1 Crossover operators

Since the selected chromosome representation does not contain genes with repetition, the traditional genetic operators such as one or two-point crossover can cause infeasibility to the chromosome. Therefore, only the operators suitable

for the permutation chromosome representation such as PMX, CX, LOX and PBX will be tested in the given research.

PMX, LOX and CX are the most popular crossovers for permutation problems (Xu, Xu and Gu 2011, Wiese and Glen 2003). PBX has also been included because it is a variation of the popular uniform crossover (Gen and Cheng 1997). It is interesting to investigate its behaviour on the non-binary chromosome representation.

Moreover, they all apply the principles of good block preservation and genes exchange between the parents in a different manner. LOX and PMX copy the entire substring from parents to children, whereas CX and PBX swap the information gene by gene. In addition, LOX preserves the relative position of the genes while CX, PMX and PBX absolute position (Croce FD, Tadei R, Volta G 1995, Bo, Hua and Yu 2006).

To the date there is no evidence to support which approach is the most effective for different domains (Elaoud, Teghem and Loukil 2010), therefore several experiments will be conducted in the next chapter in order to identify it.

8.4.2 Mutation

With the similar objective in mind as in the case of crossover operators, the most popular mutation operators will be investigated. Due to specifics of scheduling problems, it is important that the mutation operator does not create duplicate genes, which might violate the problem constraints and cause reduction or repetition of the jobs.

Following this rule, three mutation operators have been selected for testing: Simple, Swap and Scramble. They all cause a distinctive level of disturbance in the chromosome, which will have a different effect on the evolution process. For example, Simple mutation only exchanges two neighbourhood genes and causes very little alteration in the chromosome structure. Swap mutation switches the genes from different parts of the chromosome resulting in moderate changes. Finally scramble mutation rearranges all the genes on the specified interval and therefore the level of disturbance is estimated to be high.

It is anticipated that simple mutation will be more efficient on small data sets, whereas scramble mutation might be more appropriate for the large problem instances. This is because the scramble mutation affects a large number of genes and makes significant changes to the chromosome. This gives the algorithm an opportunity to quickly move from one solution to another inspecting diverse combinations of the genes, which can lead to finding the optimal solution faster. On the contrary, the simple mutation can be beneficial for smaller size problems, where movement from one solution to another can be accomplished by an exchange of the positions of two adjacent genes. The scramble mutation might be disturbing for such problems and prevent the convergence of the algorithm.

Finally, swap mutation might perform the best on the small and medium data. However, these hypotheses need to be empirically validated in order to provide a definitive answer.

8.4.3 Operator selection mechanism

Since the evolutionary search is orchestrated by genetic operators and each of them has their own unique way of schemata manipulation, Ming, Cheung and Wang (2004) and Kumar, Contreras-Bolton and Parada (2015) state that application of different operators can improve the process of optimisation.

Three diverse strategies of application and management of the operators have been devised in order to test the effectiveness of the operators' synergy.

Strategy1. In this strategy each operator has an equal probability of being selected at every iteration. The advantage of this approach is that operator selection time is negligible and implementation of various operators might boost diversity in the population due to their technical differences in the chromosome formation.

Strategy2. Unlike the previous strategy, the second strategy is focused on finding the two best chromosomes which can be constructed from selected parents by available genetic operators. This method applies all the operators to the same pair of chromosomes and keeps a record of temporarily created offspring and their fitness. Once the process has completed the production of new chromosomes, the replacement procedure adds only the two fittest individuals to the population and deletes the others.

The rationale of this method is that it enables the maximum performance from the operators. The down side of this strategy is that the time spent on each iteration might be much longer compared to the previous strategy.

Strategy 3. The third strategy is a trade-off between the two previous strategies. It tests each crossover in the same fashion as strategy two for a predefined amount of trial iterations, and then applies the best-performing operator for a series of subsequent iterations. The amount and ratio of the trials to the number of standard iterations will depend on each problem and data size and will be discussed in the related sections.

The motivation for this strategy relies on the concept that the landscape of the search region changes as the algorithm progresses. This implies that the effectiveness of the operators changes over time as well (Elaoud, Teghem and Loukil 2010). The benefit of this strategy is that it is less computationally expensive than strategy two, but at the same time it takes into account the effectiveness of the operators unlike strategy one. The potential limitation is that the operator that has been identified as superior during the trial iterations might not be the most powerful at successive iterations.

8.4.4 Selection

Preference was given to binary tournament selection as it is a comparatively simple and non-time consuming selection mechanism. It is also a popular selection strategy that is used in numerous EAs for CSP and JSSP (Park and Ryu 2006, Kwan, Kwan and Wren 2001). Binary tournament selection can be described as follows. Two individuals are selected from the population at random and the fittest amongst them is selected as a first parent. The same process repeats in order to obtain the second parent. The tournament selection has been chosen because of its efficiency as it can select two individuals without calculation of the fitness for all the population. It is also relatively unbiased towards the high-fitness individuals unlike other fitness scaling selection techniques. This is important because some individuals with poor fitness might still possess a valuable combination of genes.

8.4.5 Replacement

Elitist replacement strategy was implemented in the algorithm. The motivation of this is that it enables preservation of good chromosomes during the course of the algorithm. Elitist algorithms demonstrated a good performance in various research studies, and was incorporated into the popular NSEA-II (Leno, Sankar and Ponnambalam 2013, Liang and Leung 2011, Kim and Ellis 2008, Deb et al. 2002)

8.4.6 Data Instances

Finally, this section describes the data sets on which the developed algorithm will undergo their assessment. Following the research objectives of testing EAs on two conceptually different problems with significantly different numbers of tasks for assignment, two data sets for experimentation presented in Table 12 and Table 13 will be used.

Table 12 Data sets for CSP experiments

Data size	Data	Number of Trips	Number of Depots	Execution time
Small	CSP_780	780	12	390 minutes
Medium	CSP_1260	1260	21	630 minutes
Large	CSP_1980	1980	33	990 minutes

Since there are no available benchmark data in the literature which match the CSP problem defined in section 6.8, three data sets comparable to the real data in respect of quantity of trips, proportion of the drivers and depots have been arbitrary created. The real data are not used at this stage due to format and quality issues that can affect the experiment result. This issue is discussed in detail in Chapter 10. The computation time depends on the size of that data set and increases by 30 minutes with increase of data size by 60 trips.

Table 13 Data sets for the JSSP experiments

Data size	Data	Number of Jobs	Number of Machines	Number of iterations
Small	JSSP_20	20	12	100
Medium	JSSP_31	31	21	300
Large	JSSP_50	50	33	500

On the contrary, three JSSP benchmark data instances, Lawrence 20x10, Lawrence 30x10, Storer, Wu, and Vaccari hard 50x10, will be used for JSSP experiments. Due to the size of the data, the computation time is anticipated to be very short, therefore the number of iterations will act as a termination criterion.

8.4.7 Adaptation to the CSP

As was established in section 8.3, application of the reconfigurable EA requires development of the schedule builder and decoding procedure individually for each problem. Along with these functions, a specialised crossover and mutation operator will be developed for each problem. This section provides details on these aspects in relation to the CSP problem.

8.4.8 Decoding procedure

In the given algorithm the solution is encoded as a vector of integers, where each integer represents a job which needs to be assigned to the schedule. In the context of CSP a job denotes either the task of driving a train or performing an ancillary activity such as fuelling the train, loading and unloading wagons.

Once the chromosomes have been randomly generated, jobs are allocated in series to the diagrams according to the following logic (Figure 50). Starting from the leftmost gene, the procedure finds the first driver in the database (the position of the driver is displayed in grey on the picture) who has the necessary route and traction knowledge to operate that trip and creates a new diagram for him or her. Then the procedure checks if the same driver is able to drive on the next journey (i.e. the second gene). If it is possible, then that trip is added to his or her diagram. If the station of origin for the current trip differs from the destination station of the previous trip, the algorithm first searches for passenger trains and the freight company's own trains that can deliver a driver within the available time slot to the next job location, e.g. Diagram 1, between trips 3 and 8 (Figure 50). If no such

trains have been found but there is a sufficient interval between the trips, then the algorithm inserts a taxi journey.

The information regarding driving times and the current duration of the diagrams is stored. Before adding a new trip, the algorithm inserts a break if necessary. If the time expires and there are no trains to the home depot that a driver can drive, the deadheading activity completes the diagram, as in Diagram 2 (Figure 50). If a trip cannot be placed in any of the existing diagrams, the procedure takes the next driver from a database and creates a new diagram for him or her.

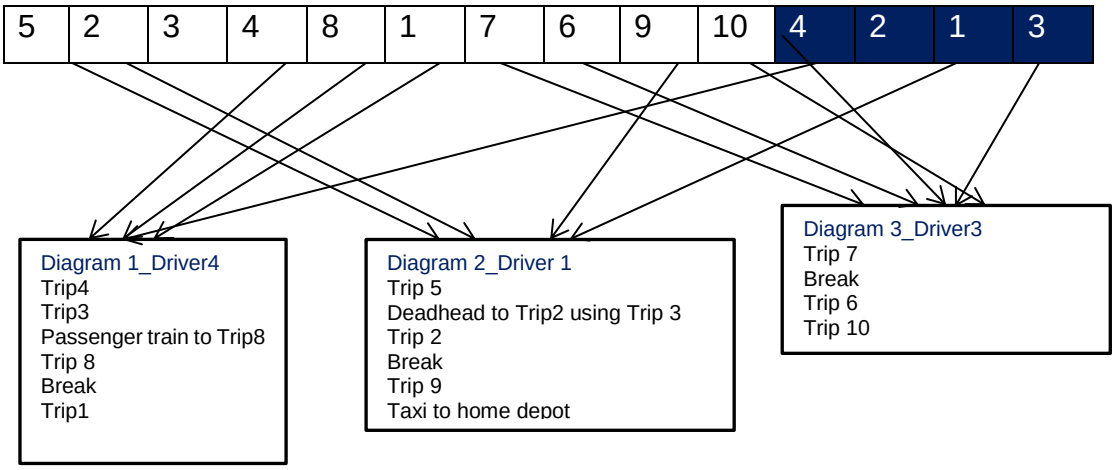


Figure 50 Chromosome representation and decoding logic

On rare occasions, a few diagrams might be left with only a few trips and a duration that is less than the minimum. This is due to the fact that other drivers are either busy at this time or located at different stations. This problem is also seen in some examples of the real diagrams. According to the company's regulation, in this situation the company pays a driver for five-hour work. This practice is also suitable for GA as it will act as a penalising mechanism for the short diagrams.

The given representation has a visual resemblance to the flight-graph representation suggested by Ozdemir and Mohan (2001), but the decoding procedures are different. The flight-graph representation generates trips based on a depth-first graph search, whereas in the proposed EA they are produced at random. Random generation is beneficial since it does not exclude situations where a driver can travel to another part of the country to start working in order to have an even workload distribution across the depots, while depth-first search usually places only geographically adjusted trips together.

The advantage of the proposed chromosome representation is that it creates both the diagrams and schedule within the same algorithm, thereby giving the EA greater control over the solution. It also does not require the generation of a large number of diagrams at the beginning. In addition, this representation does not leave under-covered trips and ensures that no unnecessary over-covering happens. It is possible that at the beginning of the algorithm this chromosome representation might produce schedules with a high number of deadheads. However, due to the specific fitness function and genetic operators, the number of chromosomes containing deadheads decreases rapidly with evolution.

However, until Chapter 10, the position of the drivers in the data base remains fixed. This limitation enables utilisation of the same chromosome representation for JSSP. The configurations, where the position of the drivers is manipulated will be discussed further in sections 10.2 and 10.3.

8.4.9 Fitness Function

An adequate solution of the CSP requires the achievement of several objectives: reduction of driver and additional transportation costs, equal distribution of the workload amongst the drivers, reduction of the losses associated with unbalanced diagram lengths, and increase in driver utilisation. There are also two conflicting objectives: high throttle time and low deviation from average diagram lengths. It is evident that with the increase in throttle time, the deviation from the average diagram length will be increased towards a minimum diagram length. This is due to the algorithm attempting to allocate a diagram for a single trip in order to achieve 100% throttle time.

As it is possible to find a single financial equivalent for all goals, the single-objective EA, whose aim is to reduce the total cost of the schedule, will be applied. The formula below displays the logic for the fitness function calculation, where the first summand represents the driver payment, the second part of the function is the cost of a taxi and the third and the fourth are the potential losses from the deviation from the target shift length and unequal workload distribution amongst depots.

Formula 17

$$ScheduleCost = \sum_{i=1}^{N} T_{diagram_i} \times C_{HourlyRate} + \sum_{i=1}^{N} C_{deadheading_i} + \sum_{i=1}^{N} (T_{diagram_i} - 8{,}5) \times C_{HourlyRate} + \\ + \left| \overline{T}_{depot} - T_{currentdepot} \right| \times C_{HourlyRate}$$

8.4.10 Problem-specific crossover for CSP

This section presents the crossover operator which has been developed specifically for CSP and which preserves and propagates high-quality diagrams accumulated throughout evolution. Figure 51 illustrates the main steps in creating the offspring.

The process starts from calculation of throttle time for each diagram in Parent 1 (the genes constituting the diagrams with high throttle times are colour coded in the darker shade on Figure 51). At the second step, 25% of all the diagrams are copied to the first child. Finally, the missing trips are added in the same order as they appear in the second parent. The same procedure is then used to form the second child.

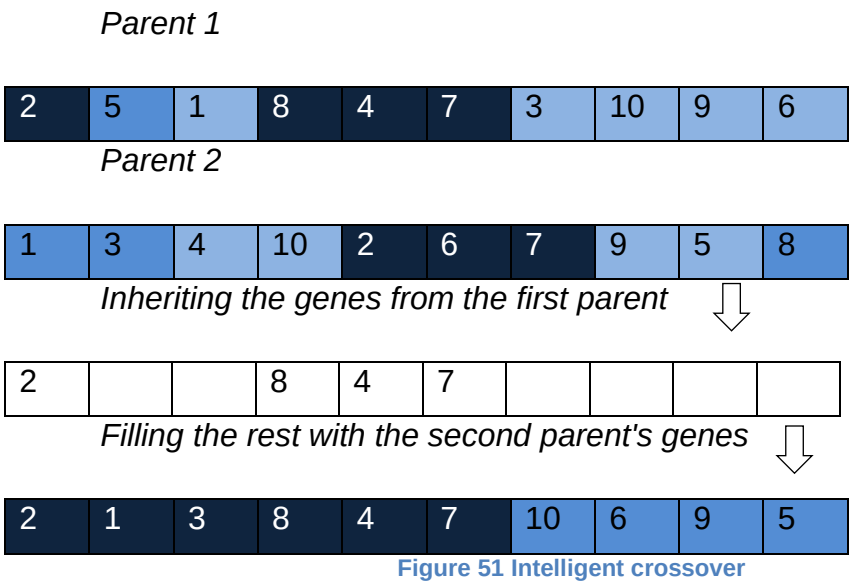


Figure 51 Intelligent crossover

8.4.11 Problem-specific mutation for CSP

Likewise, the devised mutation for CSP operates with the throttle time of the diagrams. The idea behind mutation for CSP is the trips placed in poor quality diagrams would make a cost efficient schedule if they were correctly re-inserted into other diagrams. This process is depicted in Figure 52.

It begins with computation of diagrams' throttle times (the genes constituting the diagrams with a high throttle time has a darker shade in Figure 52). Then, it selects a random gene from the diagram with the least throttle time and identifies the place where this trip can be inserted. The trip can only be placed between other trips, where the time of arrival of the preceding trip is earlier than the departure time of the selected trip and the arrival time of the given trip is no later than the departure of the subsequent trip. The same is applied to the locations. The departure station of the re-inserted trip should be the same as the arrival location of the previous trip and the departure of the next trip should be the same as the arrival place of the new trip. If this condition has not been met by any pair of consecutive trips, then the selected gene is re-inserted randomly.

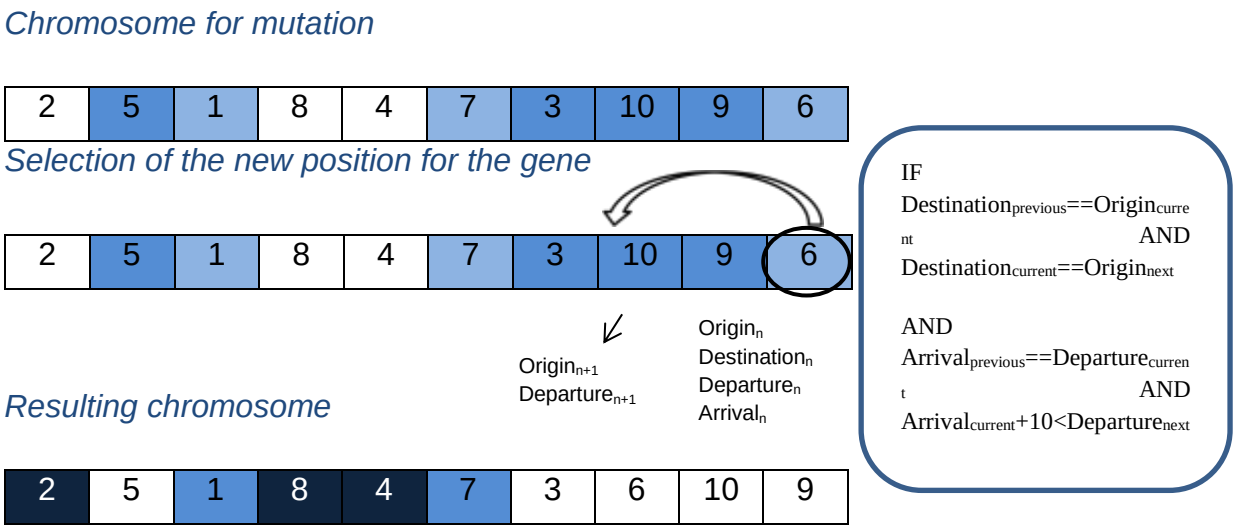


Figure 52 Intelligent Mutation

8.5 Adaptation to the Job-Shop Scheduling Problem

8.5.1 Chromosome representation and decoding procedure

Preference was given to the job-based chromosome representation for the following reasons:

- Unlike operation-based representations, the job-based chromosome representation does not contain repetitions of the genes and hence fits to the CSP as well.
- Attaching a number to each operation of each job in order to avoid repetitions is not an option because the operations should be performed

in a specific order. The selected genetic operators will be unable to preserve this sequence.

- A machine-based representation has been rejected due to the fact that not all machines can perform different operations and there is no information available of which operation can be performed on a different set of machines;

The fundamental difference between decoding procedures for job-based representations is the order in which the operations get assigned to machines. There are two techniques of the schedule deduction. First one assigns all the operations of the job to corresponding machines and only then moves to another job (Algorithm 5). Another way of accomplishing it is to assign only one operation from each job in the order they appear in the chromosome. Once all the first operations were scheduled, the procedure starts to assign second operations. This is repeated until operations of all the jobs have been assigned (Algorithm 6).

Algorithm 5 First Decoding procedure

```
1: FOR i=1:Njobs
2: FOR j=1:NOper[Job[i]];
3: CurrentMachine=Machines[Job][i]
4: JobDuration=Duration[Job][i] ;
5: MachineCompletion[CurrentMachine]=
max(MachineCompletion[CurrentMachine]
6: JobCompletion[Job])+JobDuration ;
7:JobCompletion[Job]=MachineCompletion[CurrentMachine];
8: j++;
9: END
10: i++;
11: END
```

Algorithm 6 Second decoding procedure

```
1: FOR i=1:NOper;
2: FOR j=1:NJobs;
3: CurrentMachine=Machines[Job][i]
4: JobDuration=Duration[Job][i] ;
5: MachineCompletion[CurrentMachine]=
max(MachineCompletion[CurrentMachine]6:JobCompletion[Job])+JobDuratio
n ;
7: JobCompletion[Job]=MachineCompletion[CurrentMachine];
8: j++;
9: END
10: i++;
11: END
```

In order to identify the most effective procedure, a series of ten experiments testing each algorithm has been performed for each data set.

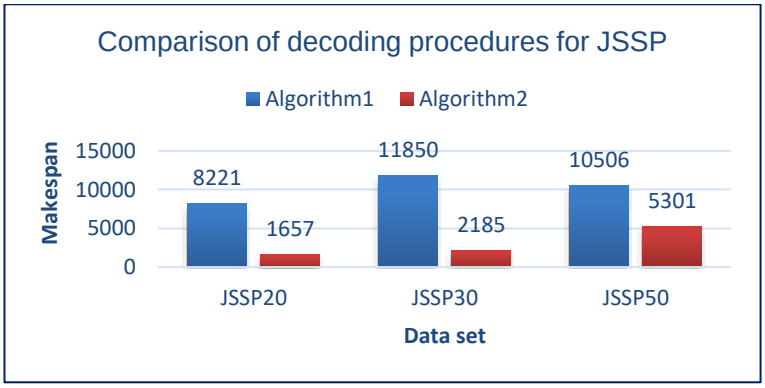


Figure 53 Comparison of decoding procedures for JSSP

Figure 53 illustrates that the second approach provided a better schedule for all test instances, and thus will be implemented in the decoding procedure.

8.5.2 Problem-specific crossover operator

The main objective of JSSP is the minimisation of the total time required for processing all the jobs. Since the job duration is fixed and cannot be minimised, the only way to attain it is to keep the idle time between them to the minimum. With this in mind the problem-specific crossover has been designed. It is based on the principle that if the idle time between two jobs is relatively small than the relative position of these two jobs should be preserved. The example of execution steps of the crossover is outlined below.

Step1**.** Calculate the total idle time between each pair of jobs in the first parent. The idle time for each operation is computed by calculating the absolute difference between the time when the machine became available from the time when the preceding operation of the same job has been completed.

Parent 1

2	1	3	5	4
Operation 1: 5 Operation 2: 7 Operation 3: 1 Total: 13	Operation 1: 1 Operation 2: 1 Operation 3: 2 Total: 4	Operation 1: 1 Operation 2: 5 Operation 3: 2 Total: 8	Operation 1: 1 Operation 2: 3 Operation 3: 3 Total:7	

Parent 2

3	4	1	2	5

Step2. Identify 30% of the jobs with the smallest idle time in between.

2	1	3	5	4

Step 3**.** Copy the given genes to the first child preserving their position.

	1	3		

Step 4. Fill in the empty genes with the genes from the second parent sustaining their order.

4	1	3	2	5

Figure 54 Problem-Specific Crossover for JSSP

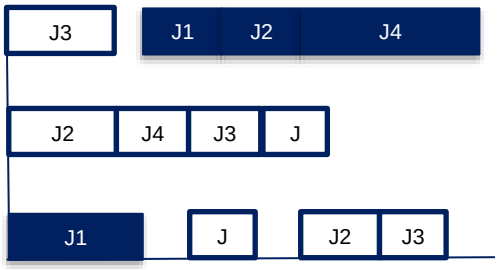
8.5.3 Problem-specific mutation operator

The proposed problem specific mutation is based on the critical path (the longest path in the graph connecting sink and source nodes) properties. Since only permutation of the operations lying on the critical path can improve the schedule, the mutation which deals with permutation of those operators has been embedded in the algorithm. A similar concept was incorporated in the local search mechanism in Zhang, Rao and Li (2008), Raeesi and Kobti (2012) and demonstrated positive results. Figure 55 demonstrates three steps of proposed mutation.

Chromosome for mutation

1	2	3	4

Step1. Identify the critical path and the jobs lying on the critical paths. In the given example these are jobs one, two and four.



Step2. Reinsert randomly selected job from the critical path between other jobs lying on the critical path. Assuming this is job number four, so it will be reinserted between job 1 and job 2.

Mutated chromosome

1	4	2	3

Step3. If there are only two jobs, then only a swap operation should be performed. If there is only one job, then re-insert it into the randomly selected position.

Figure 55 Intelligent Mutation

8.6 Conclusion

Based on the conceptual analysis conducted in this chapter, the framework for algorithm investigation has been proposed. It has been identified that both problems can be encoded into a permutation chromosome representation and identical genetic operators can be utilised. Four popular crossover operators and three mutation operators were chosen for their distinctive features in manipulation of schemata.

Along with traditional genetic operators, specific crossover and mutation operators have been devised for each problem. Unlike common permutation operators, they do not blindly exchange the genes between two parent chromosomes, but rather attempt to identify and propagate useful parts of the chromosome as well as to make local improvements in the schedule.

The effectiveness of these techniques individually and in conjunction with each other is empirically tested and discussed in the next chapter.

Chapter 9. Comparison of evolutionary operators and strategies: experimental results

9.1 Introduction

It is crucial to identify a set of the most effective genetic operators, which should be embedded into the automatic crew scheduling system. The main objective of this chapter is empirical validation of the effectiveness of traditional permutation genetic operators and problem-specific operators devised in Chapter Chapter 8.

The chapter begins with comparison of the standard permutation genetic operators with each other as well as against the operators which have been specifically developed for the CSP and JSSP within this research. The second part of this chapter explores whether the joint use of the operators (i.e. application of several different mutations and crossovers) within the same algorithm can improve the results.

For consistency of the experiments, in the CSP all the operators will be applied to the first part of the chromosome which corresponds to the trips, leaving the second part, which characterises the position of the drivers, fixed. The second part of the chromosome will be directly copied from the first parent to the first child and from the second parent to the second child in the crossover operator and will not be modified during mutation. This assumption enables equal comparison of the operators between both problems.

9.2 Crew Scheduling Problem

9.2.1 Single Crossover and mutation experiments

The purpose of the given series of the experiments is to investigate the performance of crossover and mutation operators. Because it is important to take into account the collective effect of both mutation and crossover, each combination of crossover and mutation has been tested. The experiments were carried out on the three sets of data: small (CSP 780), medium (CSP 1260) and

large (CSP 1980). Each experiment has been repeated 10 times. Graphs in Appendix 6 demonstrate one of the runs of each configuration.

9.2.2 Crossover Performance

All four standard crossovers along with customised crossover for CSP have been tested with each type of mutation ten times, therefore the total number of runs for each crossover reached 50 on each data set. The average results for each crossover are displayed in Figure 56-Figure 58.

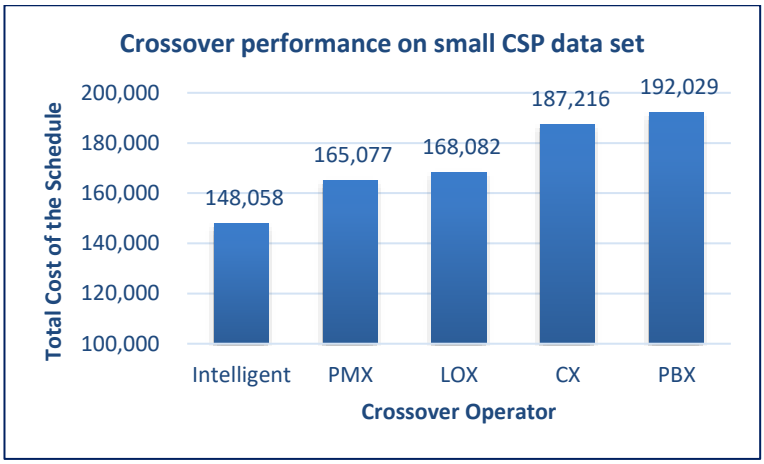


Figure 56 Performance of different crossover operators on a small CSP data set

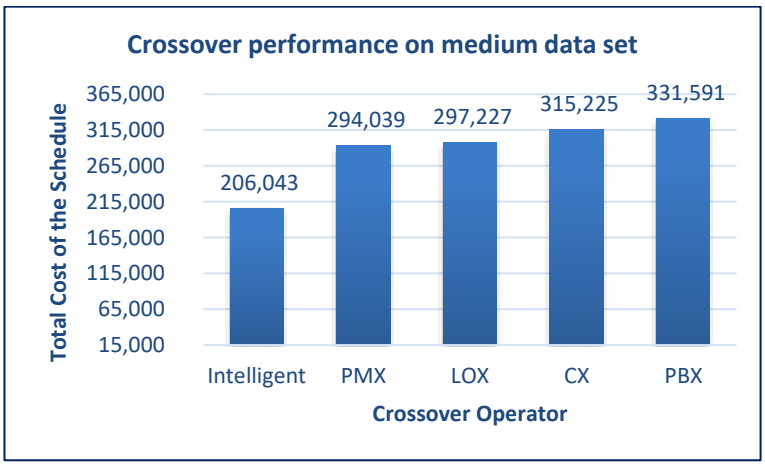


Figure 57 Performance of different crossover operators on a medium CSP data set

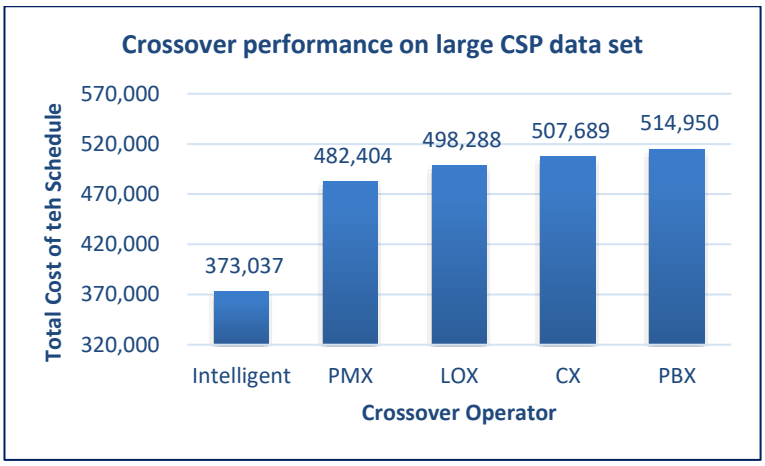


Figure 58 Performance of different crossover operators on a large CSP data set

The intelligent crossover demonstrated 11%, 30% and 23% correspondingly better results on the small, medium and large size problems than the PMX, the next best performing crossover. The graphs in Figure 136-Figure 138 in Appendix 6 display that Intelligent crossover was able to reach a better solution for the smaller amount of iterations. This is especially noticeable on the medium and large data sets, where Intelligent crossover could locate a promising region significantly faster than other crossovers.

The second best performing operators are PMX and LOX. Their performance is relatively similar with a maximum observed difference of 4 % on the large data set whereas the difference on the small data set constituted 2%.

Interestingly, although LOX crossover has some similarities with Intelligent crossover as they both preserve the position of the genes in the same way during crossover stage, Intelligent crossover achieved a12% better result on a small data set, 30% on the medium and 25% on the large data set. This suggests that intelligent selection of cutting points can improve the efficiency of the operator.

CX and PBX demonstrated the poorest performance amongst all crossovers tested on the CSP. They showed relatively similar performance with the maximum deviation in final results reaching 5% with CX crossover delivering a better schedule. On average the solution produced with CX crossover is 6% worse than LOX crossover and 27% worse than Intelligent Crossover.

Such results obtained by CX can be explained by its inability in some cases to create a new solution which can be the reason for premature convergence and

poor exploitation of the search space as demonstrated in section 2.7.7. In its application to CSP, PBX crossover showed a very slow convergence process (Figure 145-Figure 147 in Appendix 6). This may be because of the way the crossover exchanges the genes. Referring back to the principles of its mechanism, it swaps random trips between the parents preserving their absolute position. However, from the decoding perspective this does not protect good diagrams and leads to poor convergence. On the other hand, crossovers which preserve the substring, i.e. PMX, and the relative position of the trips (LOX) have proven to be more effective in the context of CSP.

9.2.3 Mutation Performance

This section compares the performance of standard permutation mutations against each other as well as against the problem specific mutation, which is described in section 8.4.11. Each of the mutation types was applied with every crossover operator ten times, and in total has been tested for fifty times on each data set. Figure 59-Figure 61 present average mutation results on three different data sets.

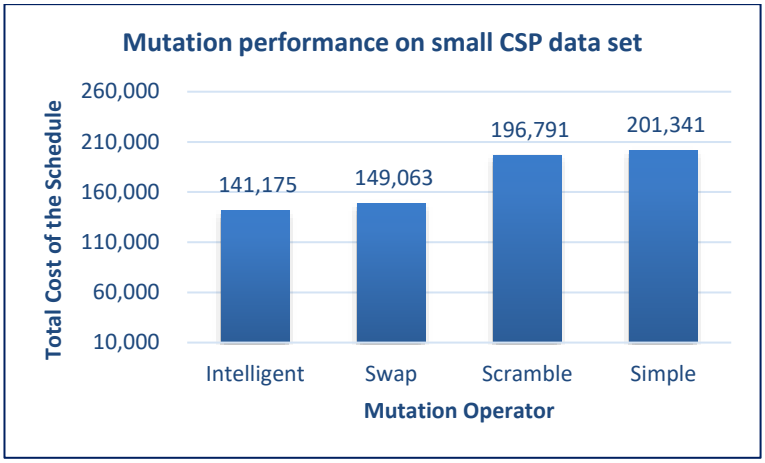


Figure 59 Performance of different mutation operators on the small CSP data set

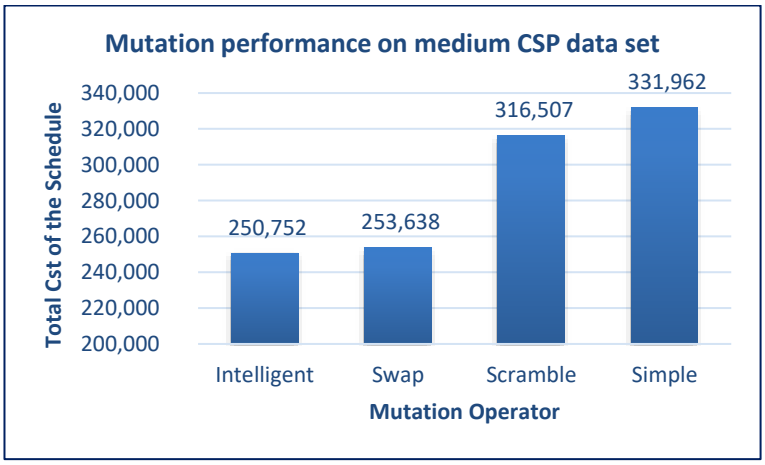


Figure 60 Performance of different mutation operators on the medium CSP data set

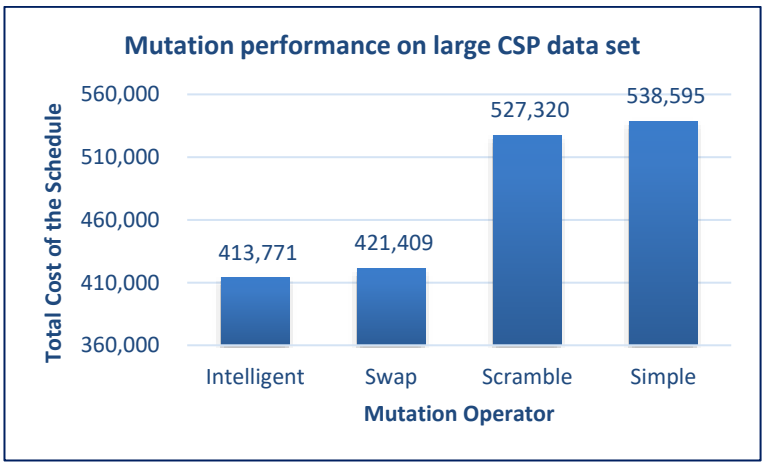


Figure 61 Performance of different mutation operators on the large CSP data set

In general, the behaviour and final results of Intelligent mutation are similar to Swap mutation, whereas Simple mutation resembles Scramble mutation (Appendix 6).

The average difference in the results between Intelligent and Swap mutation constitutes 6 % on CSP 780, 2 % on CSP 1260 and CSP 1980. This may be because they share similar logic: if the intelligent mutation could not find the trips which can be re-inserted, it swaps two randomly selected genes. Unlike crossover, the intelligent selection of the genes and position where they need to be re-inserted provided only marginal improvement in the mutation. This may be due to the impact of putting consecutive trips together, which has increased workload distribution and possibly deviation from the target shift length.

The difference in the produced results between Swap and Scramble mutation is more significant and achieves 24%, 20% and 21% on the small, medium and large data set respectively. The difference in performance of Scramble and Simple mutation is not significant. Scramble mutation produced schedules 3%, 5% and 3% better than Simple mutation on CSP780, CSP1260 and CSP1980 correspondingly.

The overall effectiveness of Swap and Insert mutation operators can be explained by the fact that they provided an opportunity for genes, which represents the trips, to migrate from one diagram to another, which resulted in the trips which start one after another at the same location to be connected without deadheads.

Although the initial hypothesis was that Scramble mutation would work well on large data sets because of its ability to change the position of a larger number of genes than other mutation operators, the results have proven the opposite. This might be because the mutation caused significant disruptions in the schedule and obstructed the convergence of the algorithm. Figure 62 demonstrates how the shuffle of the genes performed by Scramble mutation can increase the number of diagrams. Although the initial schedule in the chromosome consisted of two diagrams (each diagram is displayed in a different colour and pattern), the mutated chromosome has four diagrams as it reversed the consecutive trips on the positions between two and four.

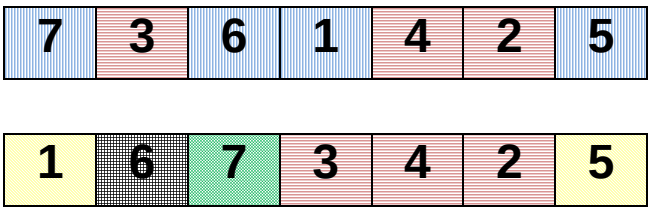


Figure 62 Example when scramble mutation deteriorates the schedule

Mutation Simple which dealt with two adjacent genes did not provide a significant improvement to the solution. This can be due to the fact making the changes on the genotype it failed to make changes in the phenotypes. This could happen when the genes were allocated to different diagrams anyway, so their position in relation to one another did not affect the cost of the schedule. Figure 63 provides a demonstration of such a case.

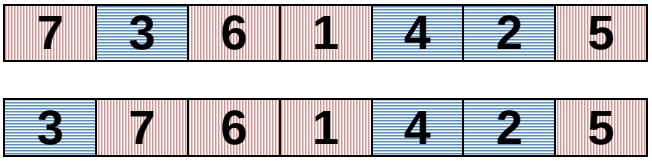


Figure 63 Example when a simple mutation does not change a driver schedule

The schedule on that picture consists of two diagrams, which are presented in different colours. If mutation occurred at position two and swapped the trip number seven and trip number three, it would not change schedule as these trips belonged to different diagrams anyway. The schedule would only be affected if trip three started before trip seven and there would be a sufficient transfer time between the trips, or alternatively the trips which belonged to the same diagram would be permutated.

9.2.4 Crossover and mutation

Because the average performance of crossovers and mutations does not provide information regarding their collective performance, the graph presented in Figure 64 and data in Table 14 display the operators' effectiveness in ascending order starting from the most powerful pair.

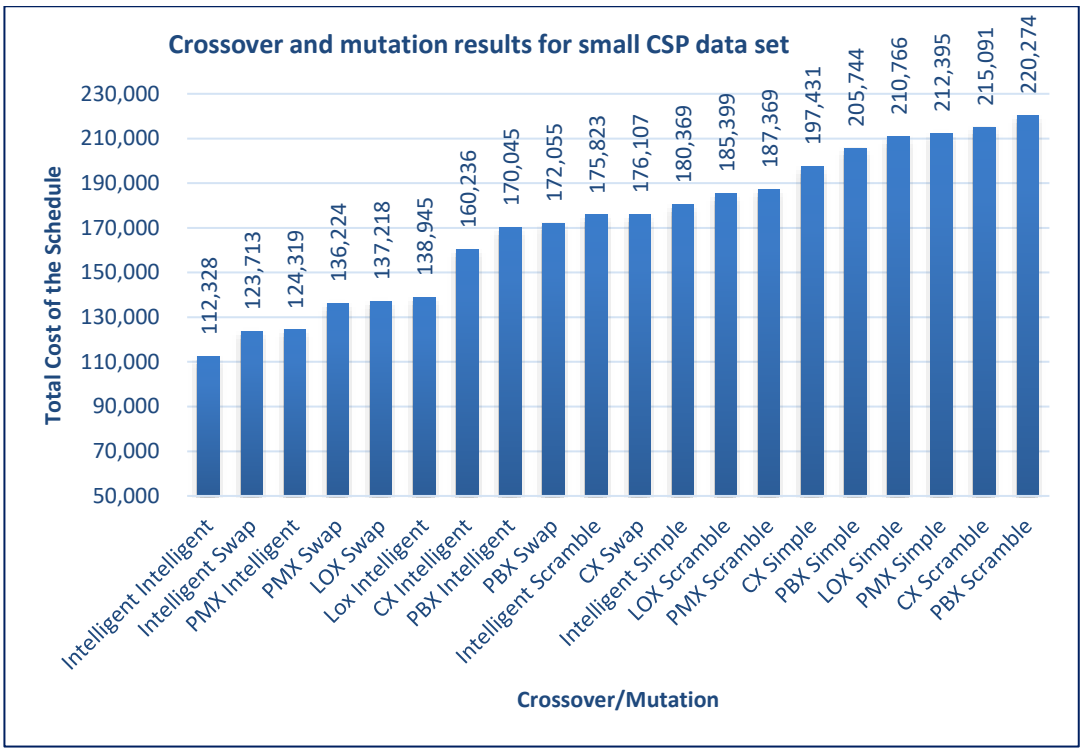


Figure 64 Performance of the pair of crossover and mutation on the small data set

Table 14 Average results of crossover and mutation performance for CSP780

Average of Cost Function	Mutation			
Crossovers	**Intelligent**	**Swap**	**Scramble**	**Simple**
Intelligent	112,328	123,713	175,823	180,369
PMX	124,319	136,224	187,369	212,395
LOX	138,945	137,218	185,399	210,766
CX	160,236	176,107	215,091	197,431
PBX	170,045	172,055	220,274	205,744

With regard to the small data set, the pair of Intelligent crossover and Intelligent mutation outperformed the next best performing pair of traditional genetic operators PMX and Swap by 18 % and the worst performing pair PBX and Scramble by 50%. Despite the general tendency in the effectiveness of each operator presented in the previous sections, crossover CX and PBX outperformed PMX and LOX when applied with Simple mutation. In addition, Swap mutation obtained better results with PBX crossover than with CX. Lastly, CX and LOX crossovers were more effective with Scramble mutation than PMX crossover with the same mutation.

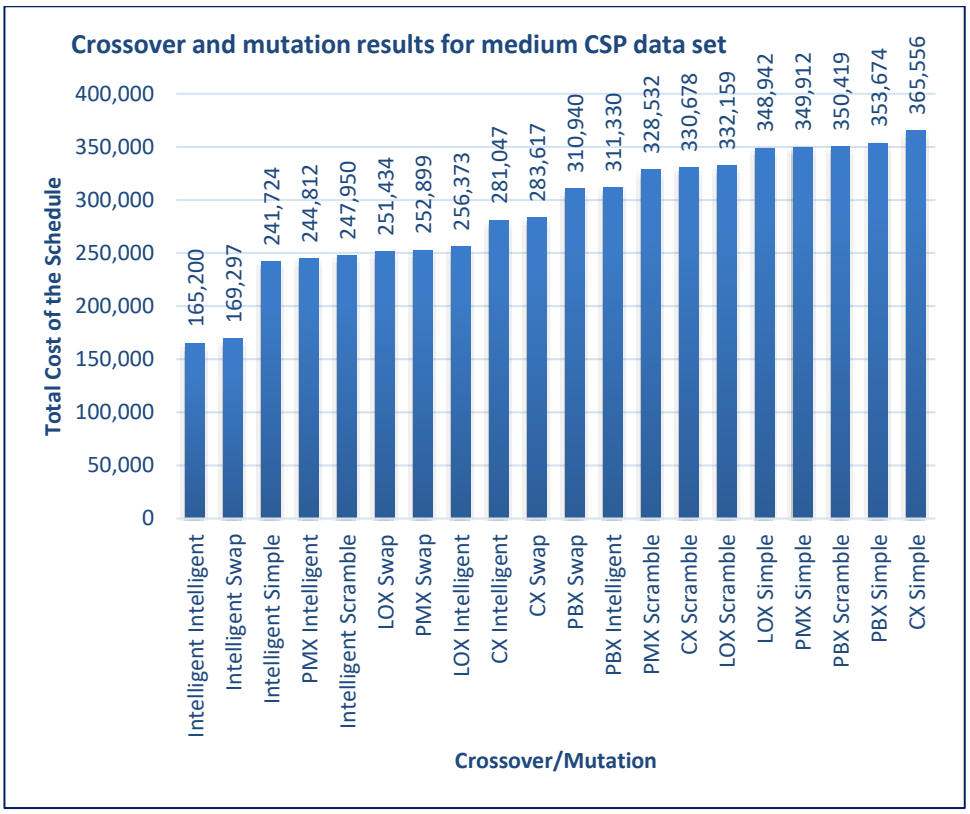


Figure 65 Performance of the pair of crossover and mutation on the medium data set

Table 15 Average results of crossover and mutation performance for CSP1260

Average Cost function	Mutation			
Crossover	**Intelligent**	**Swap**	**Scramble**	**Simple**
Intelligent	165,200	169,297	247,950	241,724
PMX	244,812	252,899	328,532	349,912
LOX	256,373	251,434	332,159	348,942
CX	281,047	283,617	330,678	365,556
PBX	311,330	310,940	350,419	353,674

In terms of the medium data set (Figure 65, Table 15), the combination of the Intelligent Crossover and Intelligent Mutation demonstrated superior results compared to other pairs of operators. It outperformed the next best performing combination of standard operators LOX and Swap by 34% and the worst performing combination of CX crossover and Simple mutation by 54%. With almost all mutation operators, Intelligent crossover outperformed other configurations of EA, and with three out of five crossovers Intelligent mutation produced more cost efficient schedules than other mutations (Figure 65). The

only exceptions are LOX and PBX crossovers, where the best results were achieved with Swap mutation. Also it has been noticed that LOX crossover outperforms PMX crossover with Swap and Simple mutations, while CX is more powerful than LOX when applied with Scramble mutation. In addition, with Simple mutation LOX performs better than PMX, and PBX is more effective than PBX when applied with Simple mutation.

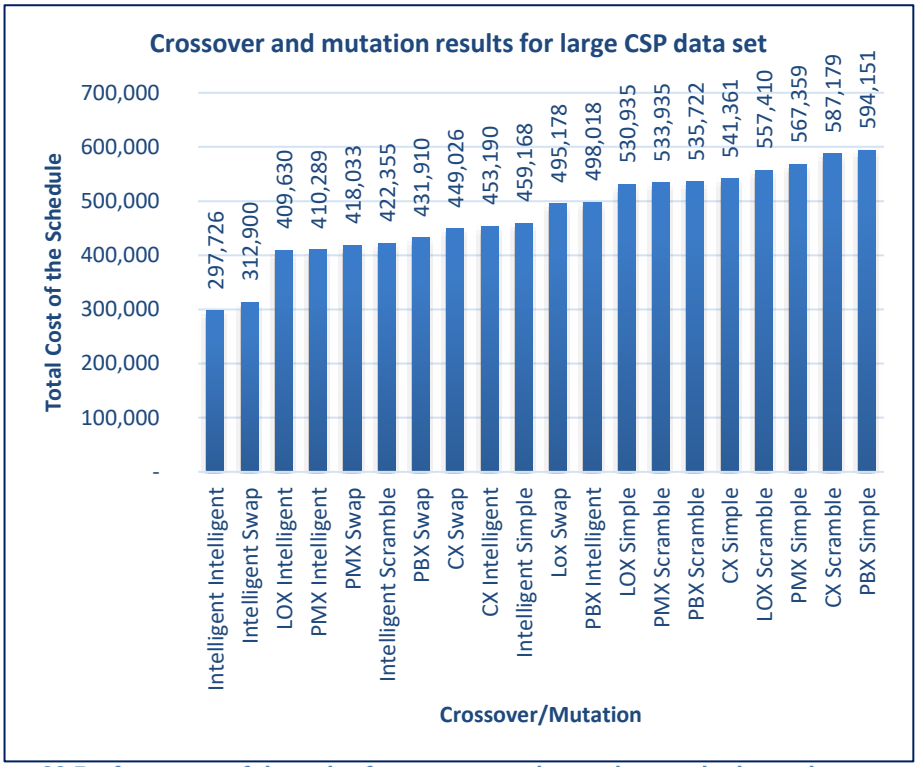


Figure 66 Performance of the pair of crossover and mutation on the large data set

Table 16 Average results of crossover and mutation performance for CSP1980

Average of Cost function	Mutation			
Crossover	**Intelligent**	**Swap**	**Scramble**	**Simple**
Intelligent	297,726	312,900	422,355	459,168
PMX	410,289	418,033	533,935	567,359
LOX	409,630	495,178	557,410	530,935
CX	453,190	449,026	587,179	541,361
PBX	498,018	431,910	535,722	594,151

On the large data set (Figure 66, Table 16), again Intelligent crossover and Intelligent mutation obtained 27% better results than conventional operators, PMX and Swap, and 50% better than the weakest operators PBX and Simple. With Intelligent mutation, LOX crossover performs slightly better than PMX. In

general, the tendency of crossover and mutation performance remains the same. However, crossovers CX and PBX outperformed LOX when used together with Swap mutation. In addition, PBX showed better results with the Scramble mutation than PMX, LOX and CX. Finally, crossovers LOX and CX performed better with Simple mutation than PMX.

9.2.5 Multiple operators

Since Hong, Kahng and Byung Ro Moon (1995) argued that each crossover and mutation operator traverse the search space in a different manner, we wanted to empirically validate how application of several crossovers and mutations together would affect the search process. Three strategies of operator selection and application described in section 8.4.3 were applied to the data instances of various size and this section reports the obtained results.

9.2.6 First Strategy

Strategy one selects genetic operators that will be used at random. This is accomplished by generation of two random numbers at each iteration, which represent which crossover and mutation will be applied to the population at that iteration. The algorithm also measured the level of improvement each crossover makes, which is expressed as the change in the fitness function. The algorithm ran ten times and the average contribution of each crossover and mutation is exhibited in Figure 67.

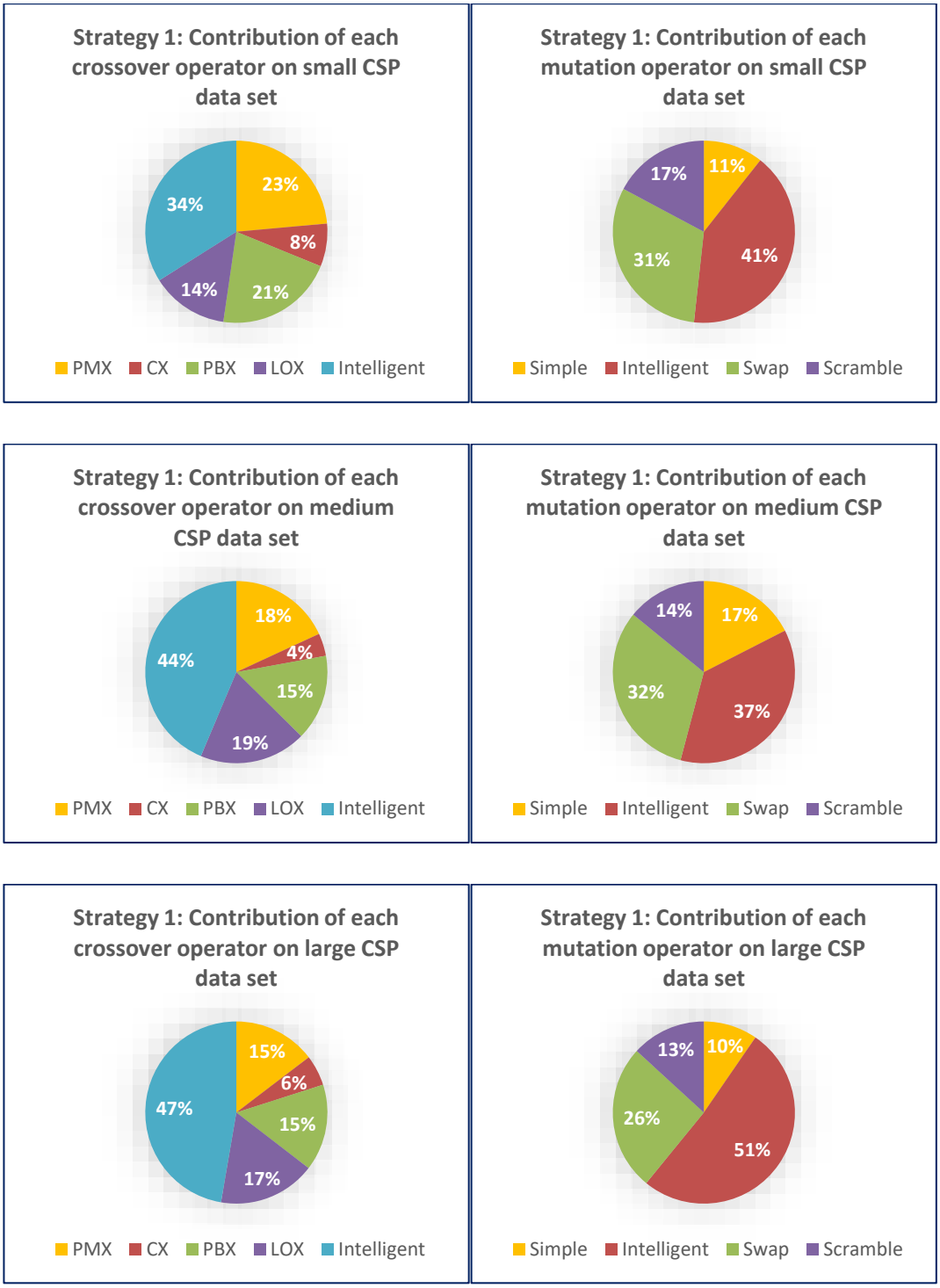


Figure 67 Contribution of each operator in Strategy 1

The Intelligent crossover demonstrated the largest contribution towards improvement of the solution compared to other crossover operators. Its impact varies from 34% to 47% and increases with the data sets. This is in line with the results reported in section 9.2.2, where Intelligent Crossover outperformed other crossovers on all data sets.

PMX crossover made 9% larger improvements than LOX crossover on a small data set. However, LOX outperformed PMX by only 1% and 2% on the medium and large data set correspondingly, despite PMX showing slightly better results than LOX when applied alone.

PBX crossover displayed a comparative performance when applied together with other crossovers. It made a greater reduction in the cost of the schedule than LOX on a small data set and achieved the same average result as PMX crossover on the large data set. Unlike the case with other crossovers, PBX crossover performance is different to its performance on its own on the CSP problem instances. Application with other crossovers possibly enabled it to have a good starting solution at different iterations.

Comparing performance of PBX and CX crossovers it can be noticed that PBX produced better solutions by 7%, 11% and 9% than CX on small, medium and large data sets, however when PBX was applied on its own it created 1%-5% worse solutions than CX.

With regard to the performance of the mutation operators, the level of effectiveness of mutations is relatively consistent with their performance as single operators. The largest improvement of 41%-51% was accomplished by the Intelligent mutation operator, followed by Swap mutation which optimised the cost by 26%-31%. On the small and large data sets, Scramble mutation showed better results than Simple mutation which reflects their individual capabilities, however on an average data set the simple mutation outperformed scramble by 3%.

9.2.7 Third Strategy

Unlike the first strategy, the third strategy deliberately selects operators on the basis of their performance. For this purpose, it conducts a series of the trial tests during which it assesses each operator and records its improvement towards the cost function. Then it applies the most successful one for a specified number of

iterations. Thus the frequency of the operators' utilisation as an additional indicator of effectiveness has also been considered. The reason why two operators, total improvements and frequency, rather than only average improvement are calculated is because in later stages of the algorithm when it converged, the operator can still be selected, but cannot make any improvements. This might deteriorate their average improvement score and will not reveal the actual efficacy. The graphs below (Figure 68) display the average results obtained during the experiments.

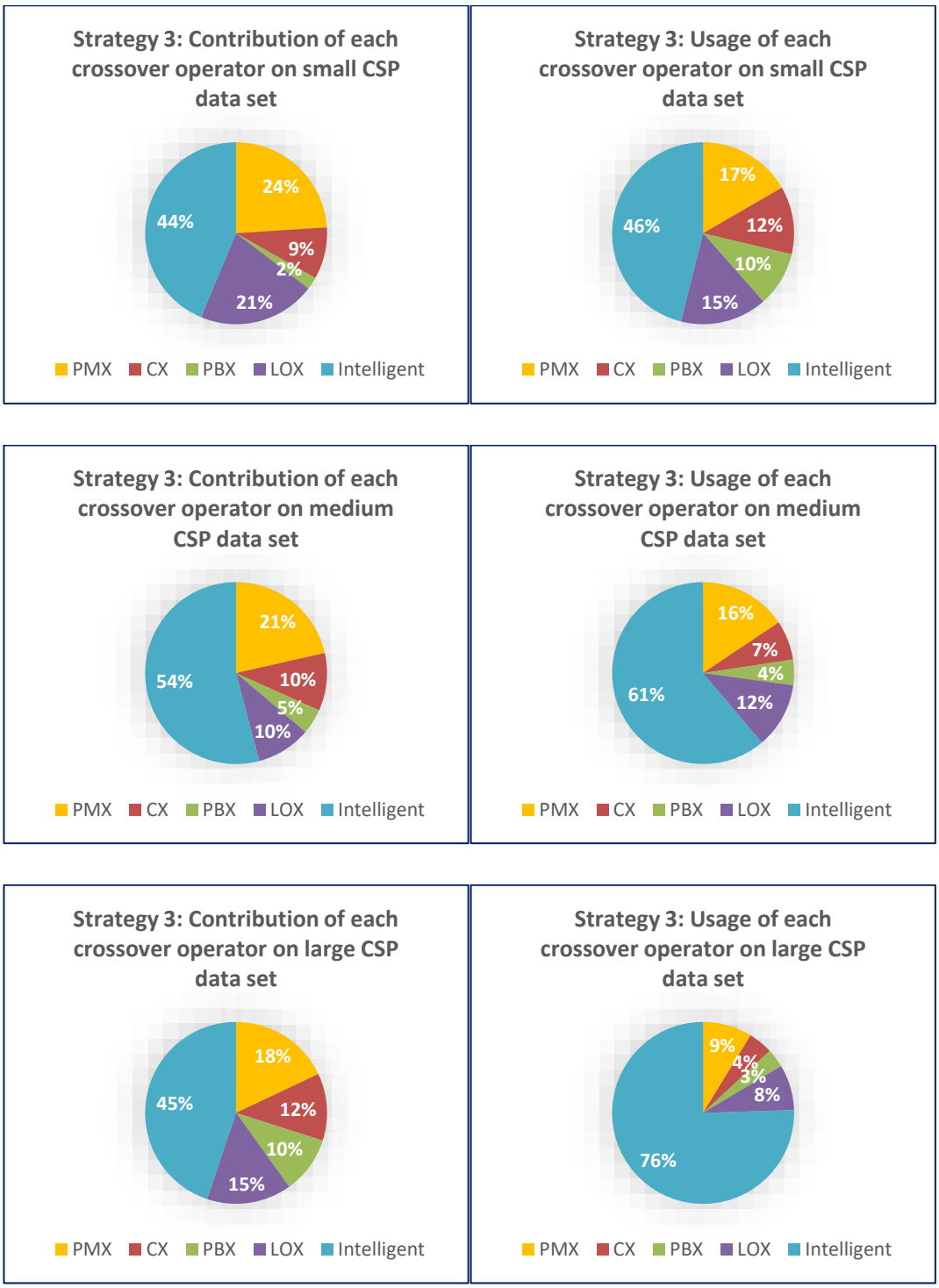


Figure 68 Strategy3: Contribution and utilisation of each crossover operator for CSP

As with the previous results, the Intelligent crossover made greater improvements than other crossover operators. The contribution towards reduction of the cost of the schedule varies from 44% to 59%. This is echoed by the number of times it has been used, which is in the range of 46% to 75%. Noticeably, the utilisation of Intelligent crossover is greater on the large data sets.

The efficiency of the PMX oscillates between 18% and 24%, and its utilisation between 9% and 17%. The portion of the improvements made by LOX is in the range of 15%-21% with the usage ratio of 8%-15%.

The contribution of CX grew from 9% to 12% despite utilisation dropping from 10% to 6%. Similar to that, PBX effectiveness increased from 2% to 10%, however its utilisation reduced from 10% to 3%.

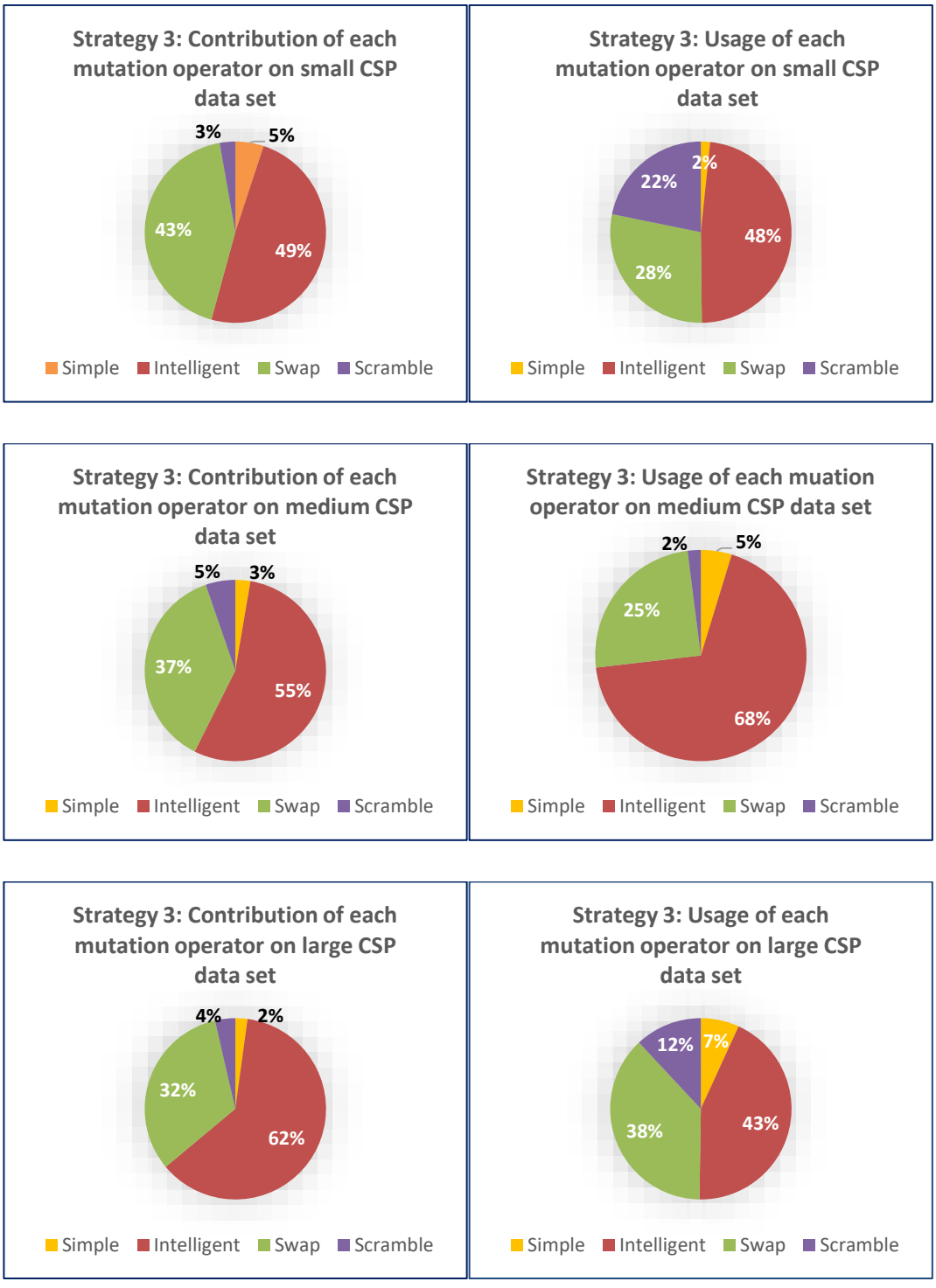


Figure 69 Contribution and utilisation of each mutation operator for CSP

With regard to mutation (Figure 69), the Intelligent mutation was selected on average 53% of the time. The Intelligent mutation is responsible for 49%-52% of the fitness function improvements. Swap mutation demonstrated a strong ability to make a large improvement as well, its total cost reduction ranges from 32% to 43% while it was called in 25%-38% of all occasions. The contribution of Scramble and Simple mutation is relatively small and does not exceed 5%. The utilisation of such mutations varies from 7% to 24%.

9.2.8 Comparison of all strategies

This section compares the average final results delivered by three strategies. The first strategy has selected the operators with equal probability, the second strategy applied all of the operators and then selected the one which produced better offspring, and third one tested each crossover every 50 iterations and applied the most effective one. The graphs on the Figure 70 - Figure 72 demonstrate how the function evolved during the optimisation process.

Despite being the most time consuming and thus evolving a fewer number of times, the second strategy descended much faster than other techniques. The first strategy seems to fail to converge perhaps because application of each operator led to a wider exploration of the search space. The third strategy exhibited an average performance by descending faster than the first strategy, but slower than the second strategy.

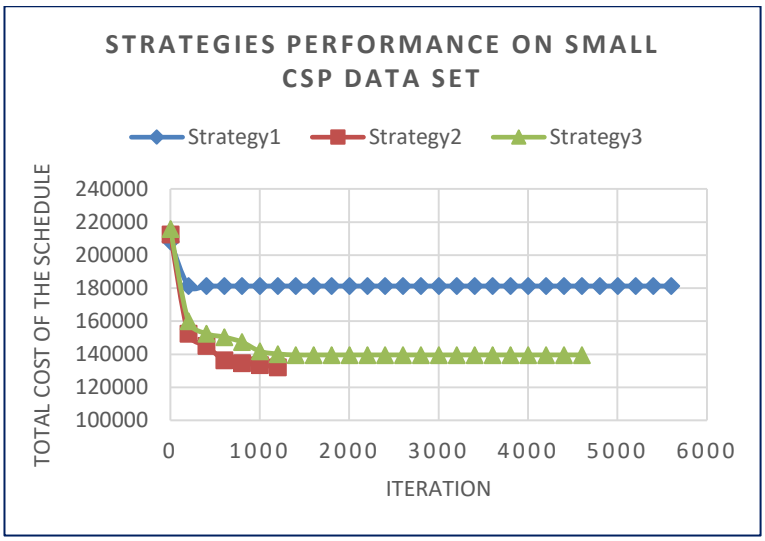


Figure 70 Performance of different strategies on a small data set

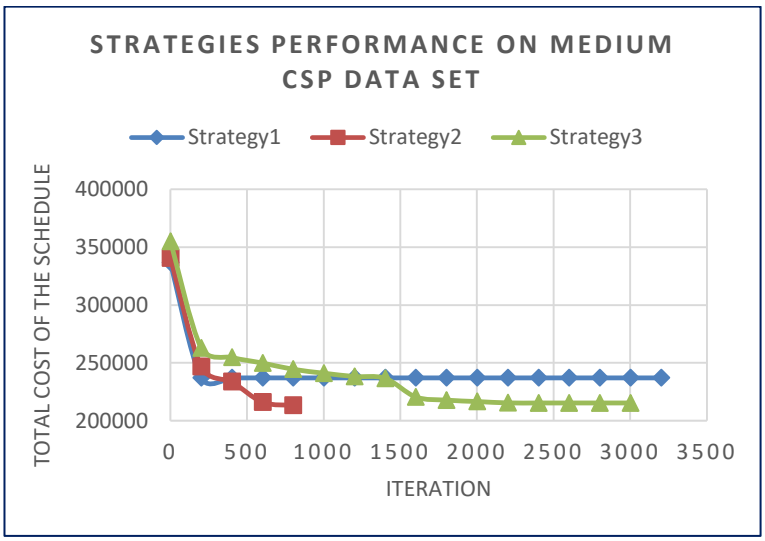


Figure 71 Performance of different strategies on a medium data set

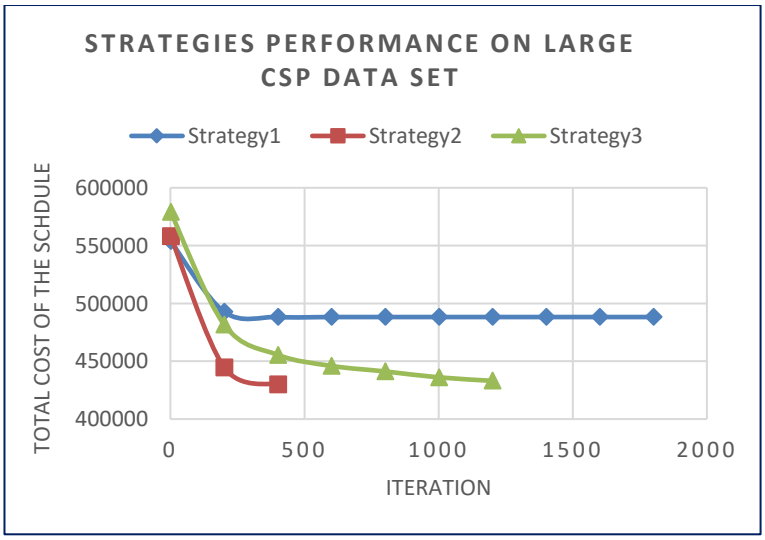


Figure 72 Performance of different strategies on a large data set

The average final results (Figure 73-Figure 75) indicate that application of all crossovers is the most efficient strategy of managing several incorporated genetic operators. The obtained average results for 10 runs are 21%, 13% and 23% better than the method with random selection of operators for CSP780, CSP1260, CSP1980.The difference in the results produced by application of the best crossover at each iteration and every 50 iterations is less significant and is 5% for the small data set, and 9% for the medium and 15% for the large data sets.

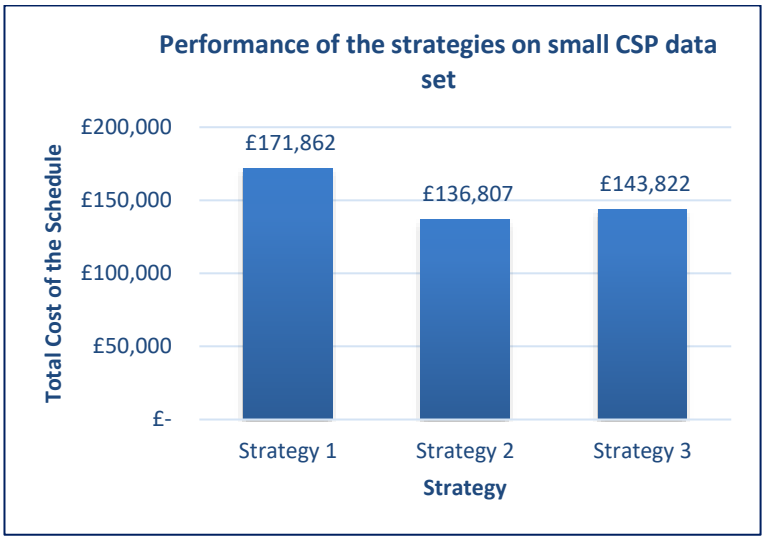


Figure 73 Final results produced by each strategy on a small data set

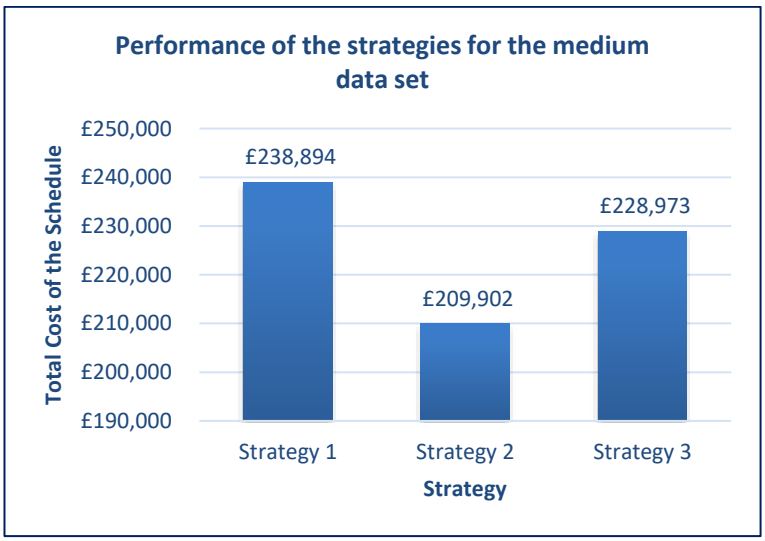


Figure 74 Final results produced by each strategy on a medium data set

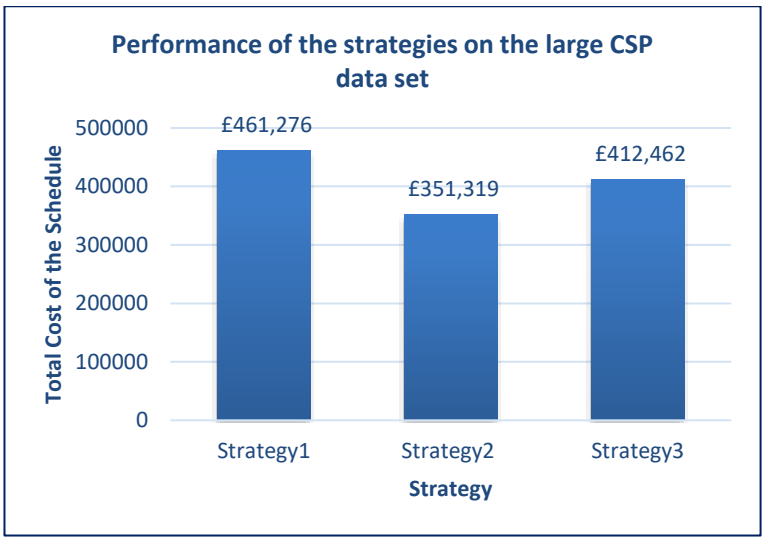


Figure 75 Final results produced by each strategy on a medium data set

However, when compared with the results of the single operator performance declared in section 9.2.4, the application of several crossovers did not surpass some of them. On the small data set, the second strategy performed better than 16 configurations, third strategy was better than 14 and the first strategy better than 12. On the medium data set all the strategies delivered a better solution than 18 out of 20 combinations of crossovers and mutations. And finally, on the large data set, the second strategy was better than 18 pairs of crossover and mutation operators, while the third strategy outperformed 16. The first strategy produced better results than only 9 combinations of crossovers and mutations. This may be due to the following reasons:

1. The time spent per iteration was longer due to application of several operators and their periodical evaluation in Strategy two and three.
2. The algorithm kept exploring the entire search space without landing on a particular region in Strategy one.

9.4 Classic Job-Shop Scheduling Problem

This section analyses the effectiveness of genetic operators when applied to the classic Job Shop Scheduling problem in a similar fashion to as it was carried out for the CSP. First of all, the performance of conventional crossovers and mutations will be presented, and then their role and impact in joint application will be studied.

9.4.1 Crossover Performance

When all the operators had been tested, the results were calculated and aggregated. Figure 76-Figure 78 display the summary of the crossovers performance.

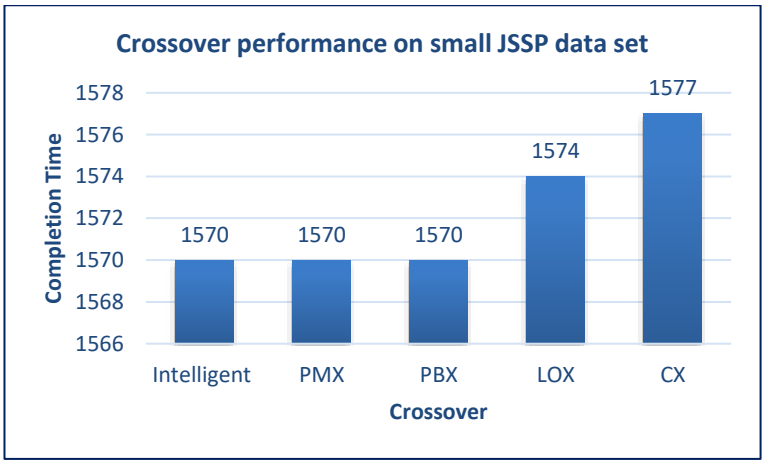


Figure 76 Performance of crossover operator on small JSSP data set

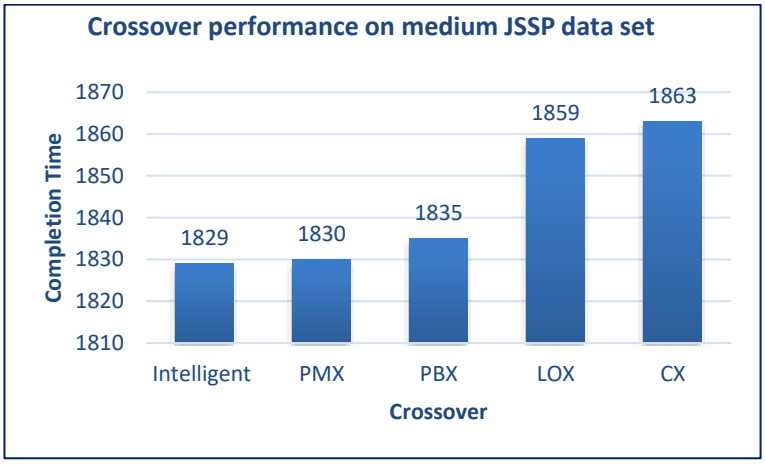


Figure 77 Performance of crossovers on a medium JSSP data set

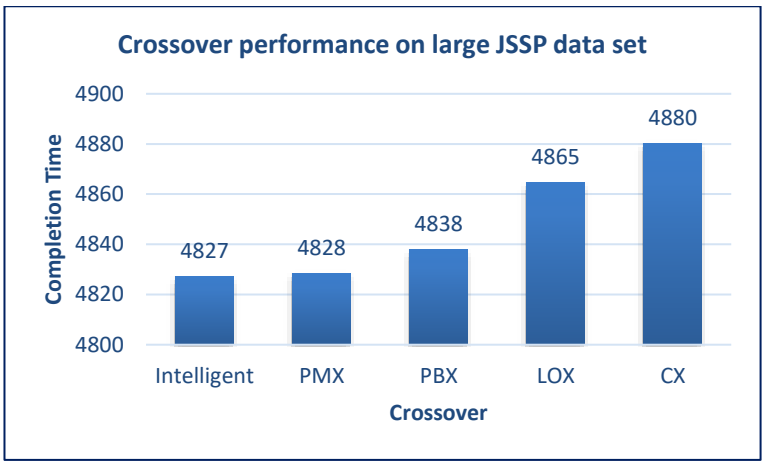


Figure 78 Performance of crossovers on the large JSSP data set

On a small data set, three crossover operators, Intelligent, PMX and PBX achieved the same average results. LOX and CX demonstrated a worse performance with the average result being correspondingly three and seven minutes longer than the best obtained result of 1570 on a small data set.

On the medium and large data instances, Intelligent and PMX crossover produced almost the same schedule on average with only a one-minute difference in favour of Intelligent approach. On the same data sets, performance of PBX crossover was poorer than PMX.

LOX crossover produced a longer schedule than PBX across all data sets, but outperformed CX by 3, 4 and 15 minutes on the JSSP20, JSSP31 and JSSP50.

It can be noticed that while Intelligent and PMX crossovers came first and second as in the CSP, PBX crossover was more powerful on the JSSP and outperformed the LOX operator. Moreover, CX was the least effective crossover when applied to JSSP, although it was second from the end on the CSP.

9.4.2 Mutation Performance

Similar to the previous section, the average results of each mutation have been calculated. The graphs in Figure 79-Figure 81 exhibit the comparative performance of mutation operators on three data sets. Appendix 7 illustrates one run of each mutation with different crossover operators.

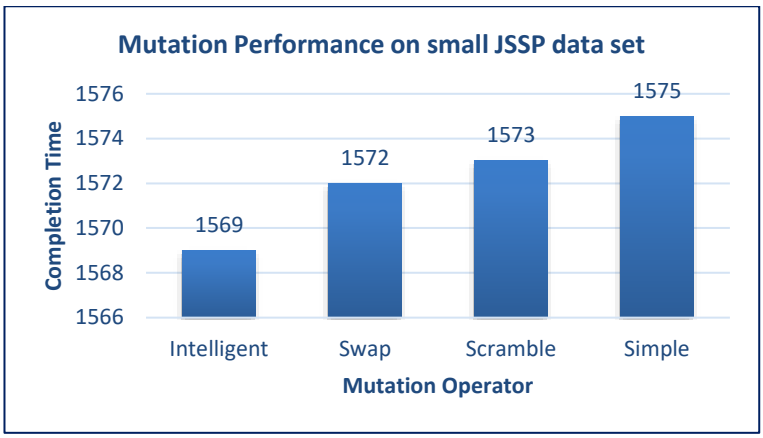


Figure 79 Performance of different mutation operators on small JSSP data set

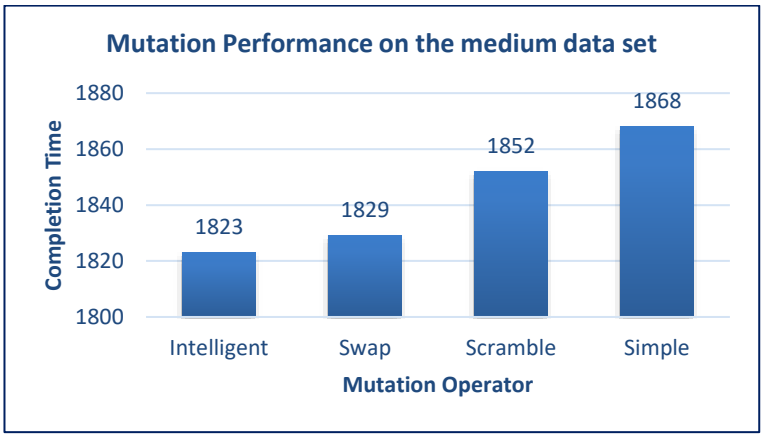


Figure 80 Performance of different mutation operators on medium JSSP data set

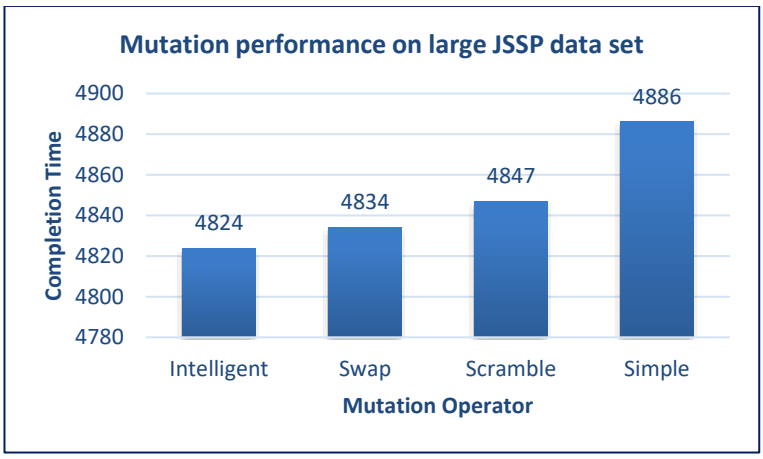


Figure 81 Performance of different mutation operators on large JSSP data set

Based on the experimental results, it can be concluded that the Intelligent mutation was the most effective one amongst tested heuristics for mutation. Swap mutation comes second as its average resulting schedules were 3 (JSSP20), 6 (JSSP30) and 10 (JSSP50) minutes longer than mutation developed specifically for JSSP. Scramble and Simple mutations were the poorest performing operators

on JSSP data instances, with Scramble mutation being able to produce a shortened schedule by 2, 6 and 39 minutes than Simple mutation on the JSSP20, JSSP30 and JSSP50 respectively. The gap between Swap and Scramble mutation varies from one minute on a small data set to 13 minutes on large and medium data sets.

The effectiveness of mutations in relation to each other when applied to JSSP corresponds to the results obtained on the CSP.

9.4.3 Crossover and mutation

This section examines the combined performance of both crossovers and mutations. The bar chart and the table below demonstrates the average results of each combination placed in ascending order starting from the most efficient one.

From Figure 82 and Table 17, it can be seen that on a small data set, more than half of the algorithms reached the schedule with the makespan of 1569. In particular, all the crossovers supported by Intelligent mutation attained the best solution in these experiments.

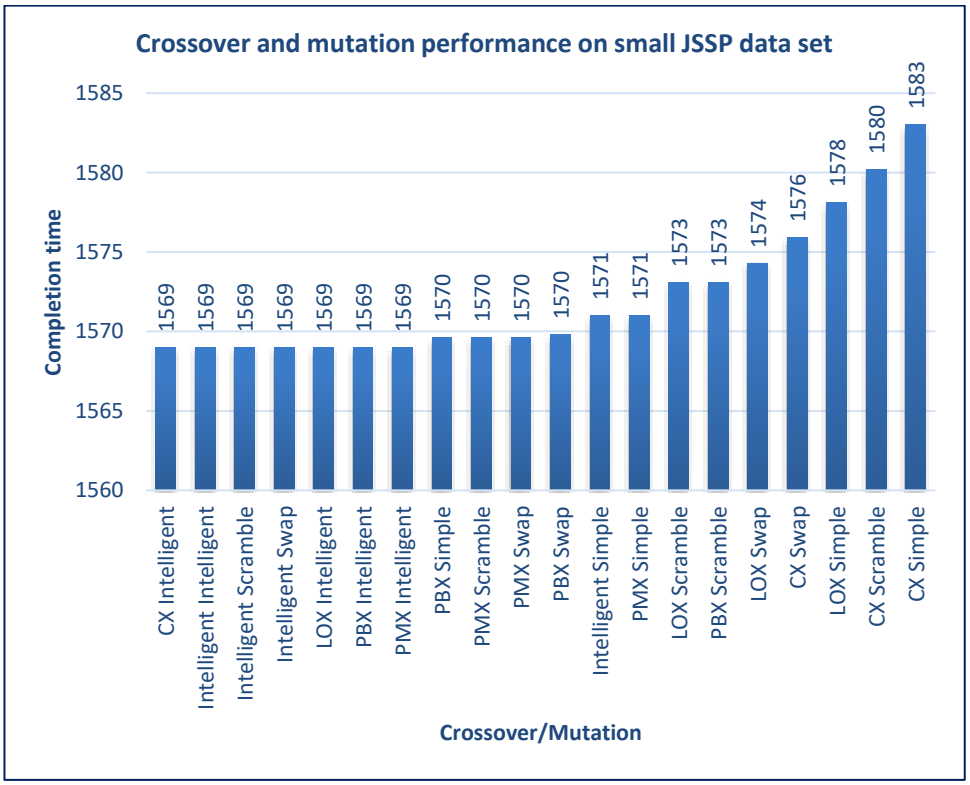


Figure 82 Average results achieved by a combination of crossovers and mutation on small JSSP data

Table 17 Average crossover and mutation result for the small size JSSP

Average of Fitness Function	Mutation			
Crossover	Intelligent	Swap	Scramble	Simple
Intelligent	1569	1569	1569	1571
PMX	1569	1570	1570	1571
PBX	1569	1570	1573	1570
LOX	1569	1574	1573	1578
CX	1569	1576	1580	1583

Figure 83 and Table 18 display the average results obtained for the medium JSSP consisting of 30 jobs. The best average results of 1820 were achieved by Intelligent crossover supported by Intelligent and Swap mutations and PBX crossover with Intelligent mutation. CX and LOX together with Simple mutation produced the poorest results on this data set and the difference with the best results constituted 95 and 101 minutes respectively.

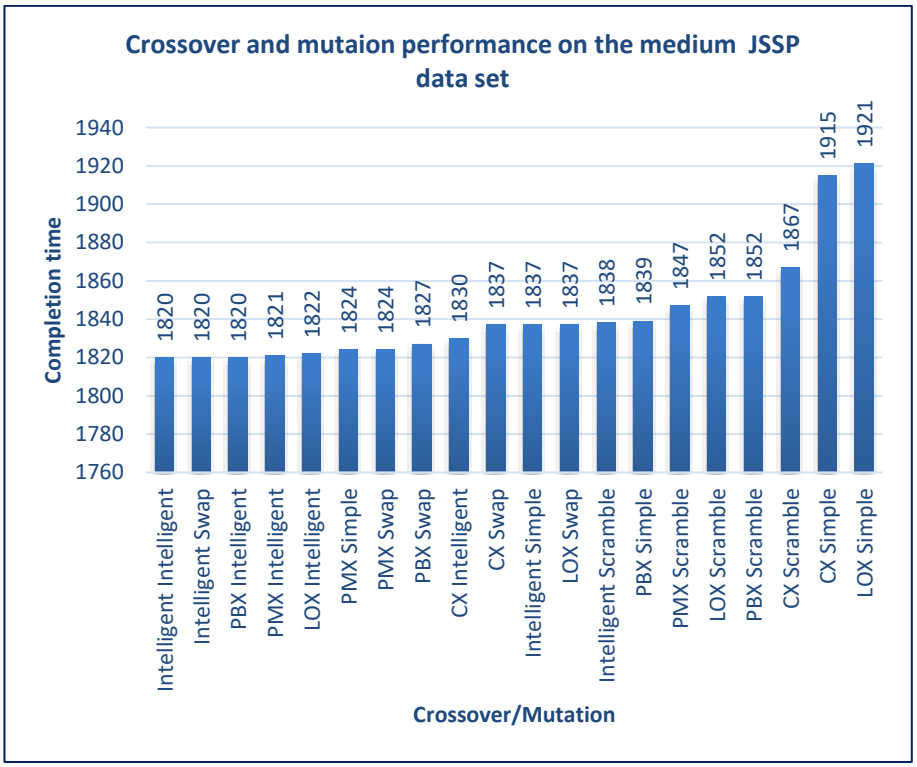


Figure 83 Average results achieved by a combination of crossovers and mutation on medium JSSP data

Table 18 Average results obtained by crossover and mutation operator on medium JSSP

Average of Fitness Function	Mutation			
Crossover	**Intelligent**	**Swap**	**Scramble**	**Simple**
Intelligent	1820	1820	1838	1837
PMX	1821	1825	1848	1824
PBX	1821	1828	1853	1840
LOX	1823	1838	1853	1921
CX	1831	1837	1867	1915

The computation results for the large data set are aggregated in a similar fashion and exhibited in Figure 84 and Table 19. These results are more diverse and the performance of the operators becomes more apparent. Intelligent crossover and mutation produced the best results. Likewise, in the small data set results, all of the combinations containing Intelligent mutation are situated in the first half of the league, while Simple and Scramble are located at the end. The graphs show that the crossovers supported by Swap mutation are spread across the bars. This suggests that they depend more on the effectiveness of the crossover operator they are used with.

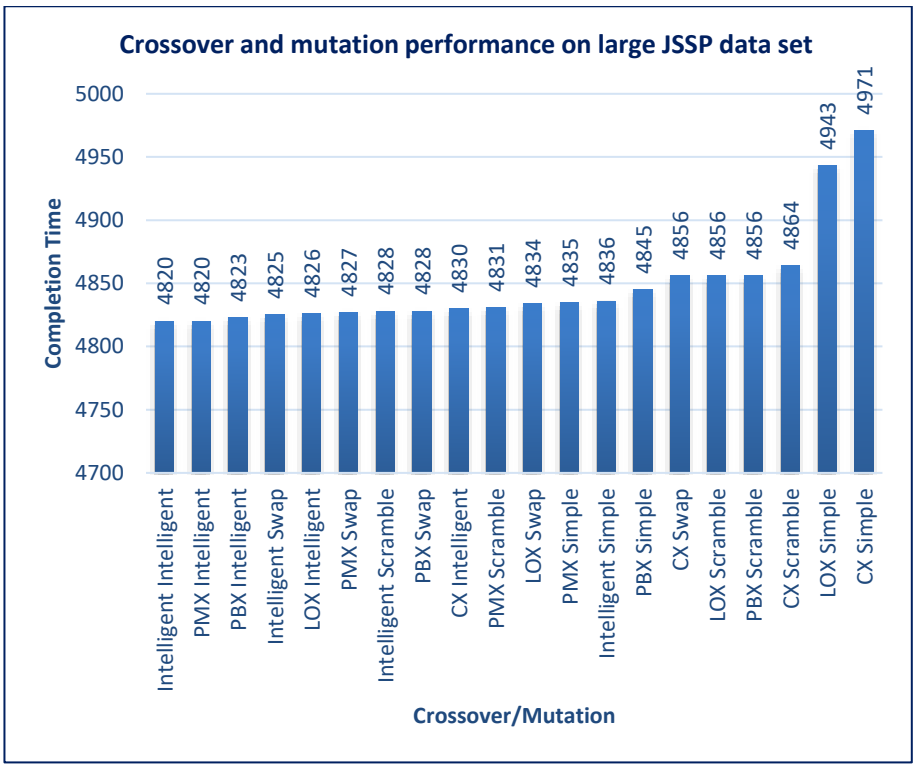


Figure 84 Average results achieved by a combination of crossovers and mutation on large JSSP data

Table 19 Average results obtained by crossover and mutation operator on large JSSP

Average of Fitness Function	Mutation			
Crossover	**Intelligent**	**Swap**	**Scramble**	**Simple**
Intelligent	4820	4825	4828	4836
PMX	4820	4827	4831	4835
PBX	4823	4828	4856	4845
LOX	4826	4834	4856	4943
CX	4830	4856	4864	4971

9.4.4 First Strategy

This section examines the performance of all operators together under the same algorithmic framework. In strategy one, only one randomly selected mutation and crossover operator is applied at each iteration. Figure 85 displays the average contribution of each operator towards the solution. Since a uniformly distributed random number generator has been used, each operator has been called approximately an equal amount of times.

Despite several crossovers and mutations yielding similar results on the JSSP20, their contribution when they were applied together revealed some differences. Intelligent and PMX crossover were the most valuable for the algorithm as they reduced the time of the schedule by 63%. LOX and PBX crossover showed slightly inferior performance by decreasing time by only 29%. The remaining 8% were dropped by CX.

On the medium data set, the contribution of the operators stays relatively the same with the exception of PMX and PBX crossovers, where PMX outperformed PBX by 4%.

On the large, JSSP 50 data instance the influence of the operators on the solution is more balanced than on a smaller data set. Despite Intelligent crossover still playing a leading role, the impact of CX crossover rose to 11%, while PMX and PBX shared the similar result of 21%.

In terms of performance of the mutations, Intelligent and Swap mutation together contributed to approximately 78% of the improvement. The remaining 22%, were made by Simple and Scramble with Simple mutation optimising the solution more effectively. While Intelligent mutation remains the leading operator, the effect of the Intelligent mutation decreases with the increase of the data set.

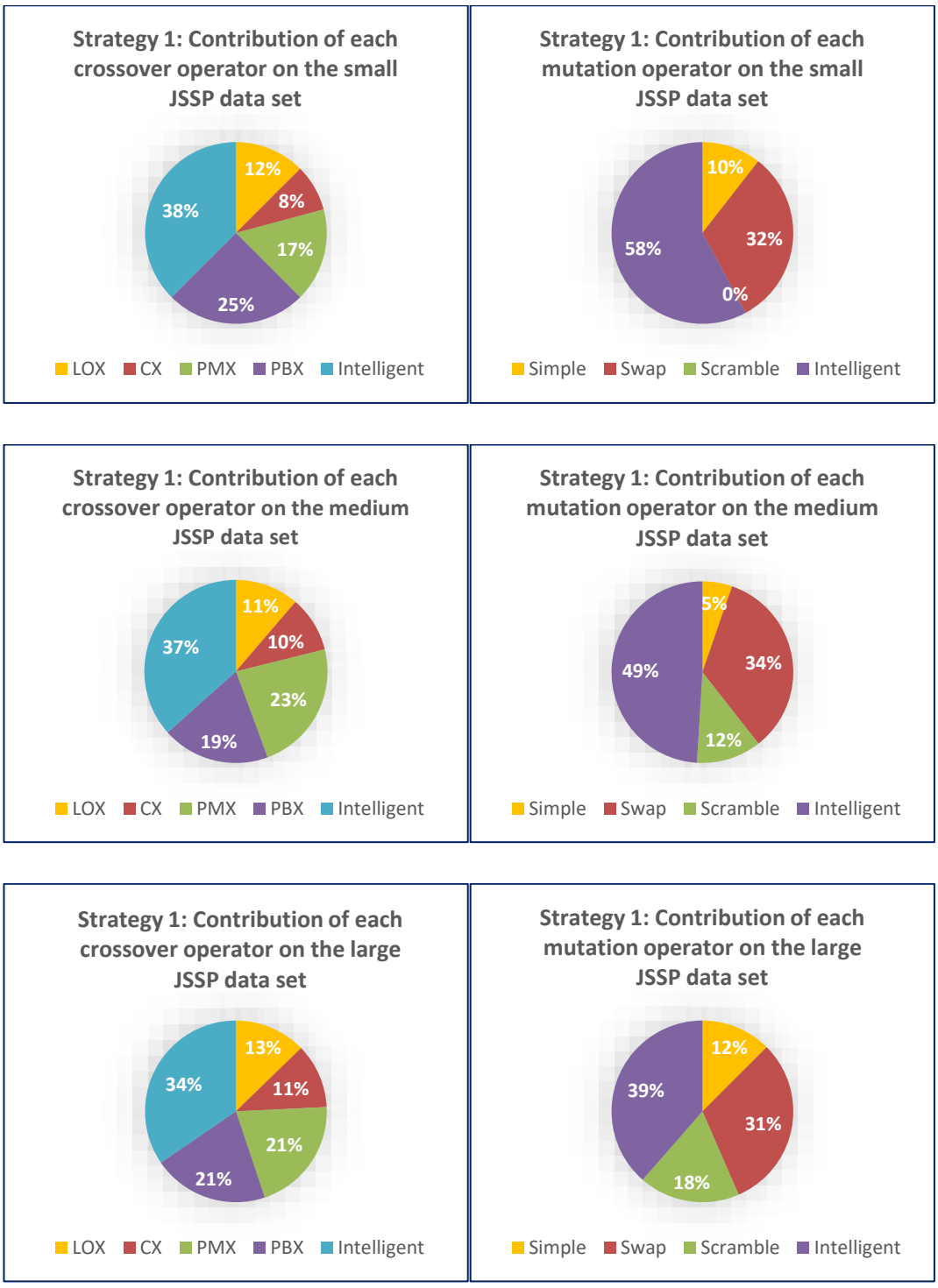


Figure 85 Experimental results of the first strategy applied to JSSP

9.4.5 Third strategy

The third strategy is performed by testing each crossover first on the limited amount of iterations and then applying the operator, which produced the best results during the testing phase for the course of the subsequent iterations. For the operator performance analysis, the algorithm also kept track of how many times each crossover has been utilised as well as by how much time it decreased

the makespan. Aside from total contribution, which can be low if the operator has been applied when the algorithm was converging, the amount of times each operator has been used will also be considered.

As the number of iterations in JSSP is considerably smaller than CSP, the frequency and proportion of trial and non-trial iterations has been adjusted for this problem. The parameters are exhibited in Table 20.

Table 20 Strategy 3 Parameters for JSSP

Data	Number of iterations for trials	Frequency of trials
JSSP20	1	20
JSSP31	1	30
JSSP50	1	50

The aggregated results of the contribution and usage frequency of crossover and mutation operators are presented in Figure 86.

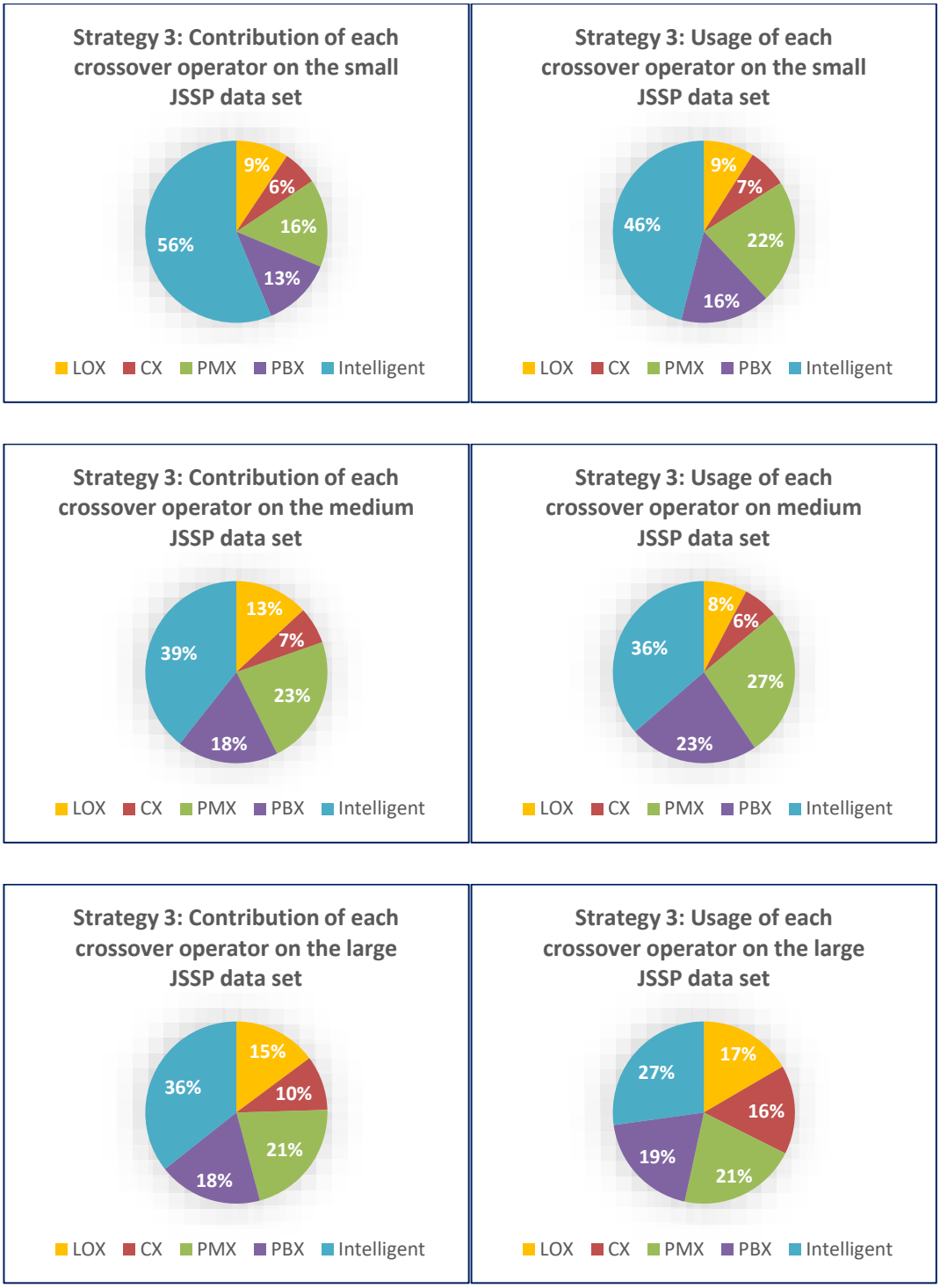


Figure 86 Strategy3: Contribution and utilisation of each crossover operator for JSSP

The contribution and utilisation of the Intelligent Crossover was the largest compared to other operators. This is the most apparent on the smallest data set, where Intelligent Crossover on average reduced the makespan by 56% and has been successfully selected on approximately 4-5 tests. PMX crossover came second as its performance on average constituted 20% while being selected in

23% of cases. Again, PBX outperformed LOX in terms of the contribution by 4% on average and utilisation by 8%. CX was the poorest performing operator with an impact of only 6%-10% and utilisation on average 10%.

In terms of mutation (Figure 87), the greatest reduction (35%-52%) in the time of the schedule was caused by Intelligent mutation, which was used in a third of the iterations. Similar to the previously declared results, it was followed by Swap mutation, whose contribution is estimated at around 30%. The joint impact of the Simple and Scramble mutations varies from 20% to 25%.

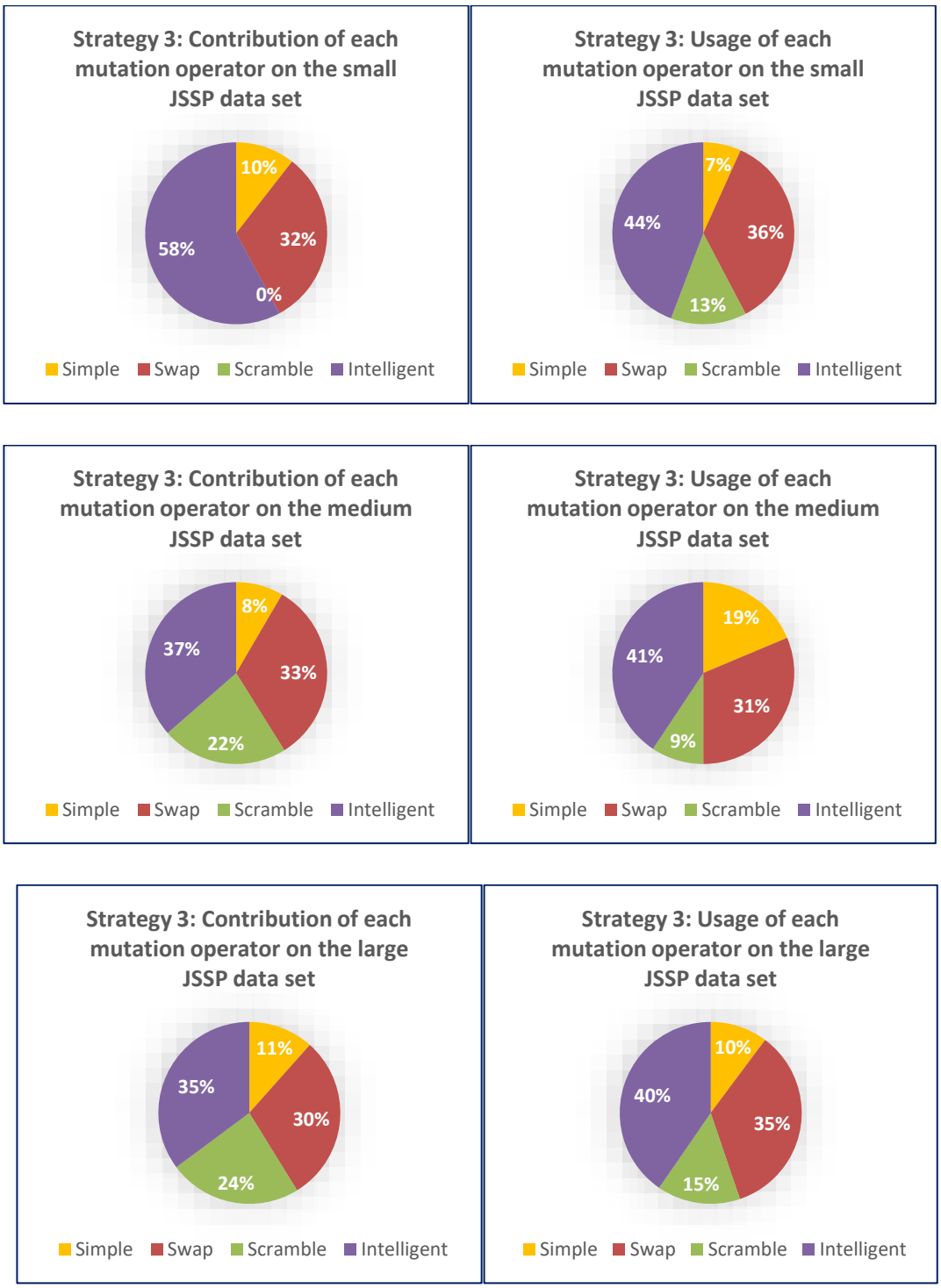


Figure 87 Contribution and utilisation ratio of mutation operators in the Strategy 3 for JSSP

9.4.6 Comparison of all strategies

This section compares the final results obtained by the three strategies on the JSSP. The bar charts in Figure 88-Figure 90 show the average final results after ten runs of each strategy.

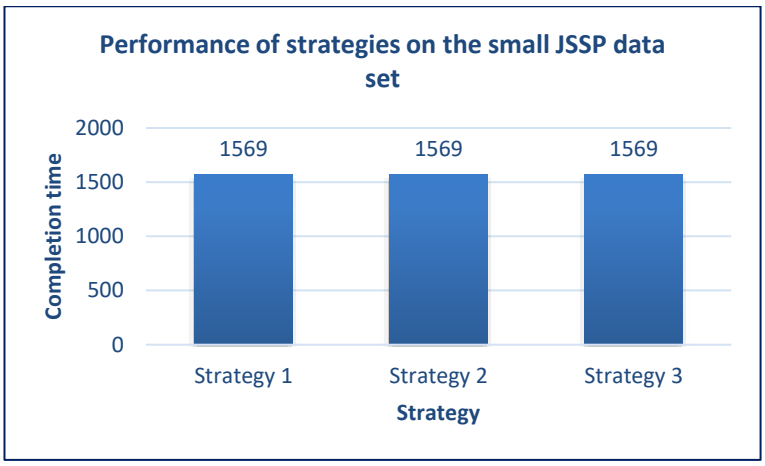


Figure 88 Performance of strategies on the small JSSP data set

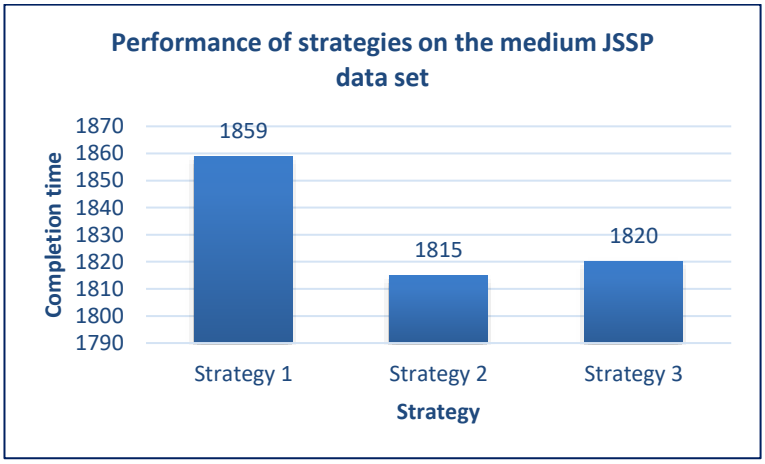


Figure 89 Performance of strategies on the medium JSSP data set

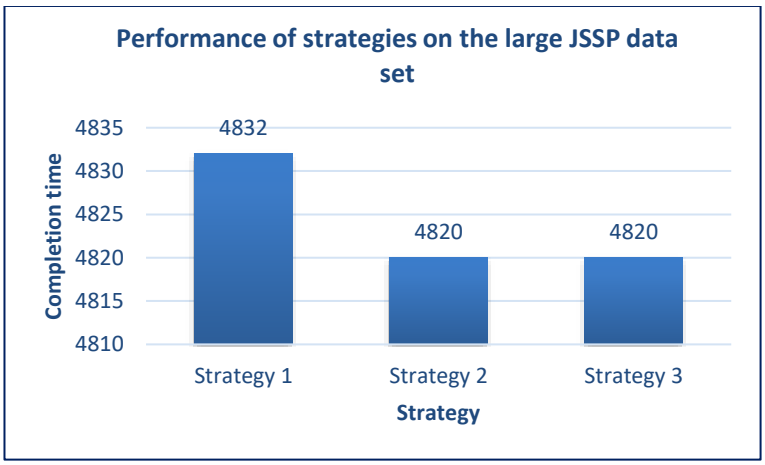


Figure 90 Performance of strategies on the large JSSP data set

With regard to the small data sets, all the strategies achieved the best recorded result in this research. This can be attributed to the strong operators incorporated into the algorithm, which possibly compensated the poor performance of others.

However, with an increase in the size of the data, the difference between strategies became more noticeable. By deliberately selecting the best performing operator at each iteration, Strategy two managed to produce a better solution than Intelligent crossover and mutation on their own. Strategy three reached the same makespan as the best performing heuristics, possibly because it was greatly dominated by them. Finally, the first strategy obtained the worst result which was due to the inclusion of operators that were not well matched to this problem.

With regard to the large data sets, the third and second strategies produced the same results while the first strategy produced a 12 minute longer schedule.

Figure 91-Figure 93 illustrate one of the runs of each strategy with respect to three data sets varying in sizes. It can be seen that the second strategy was more efficient and converged faster than the others, whereas the first strategy was the slowest.

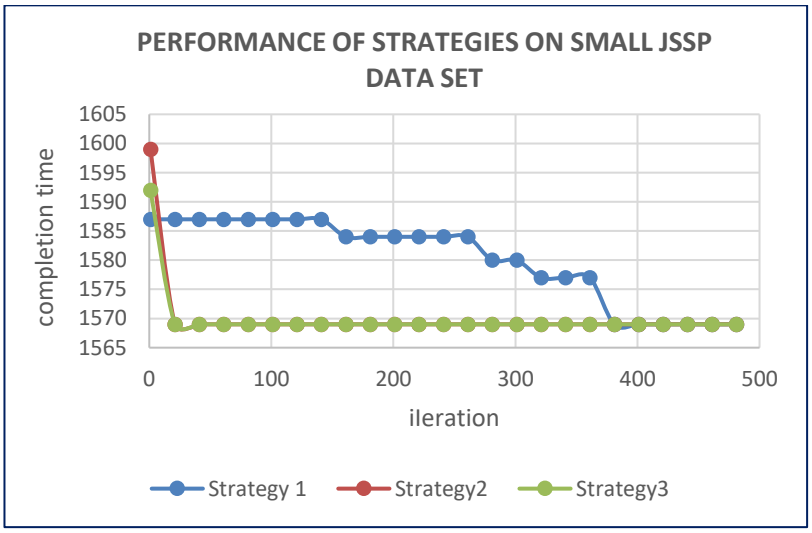


Figure 91 Performance of different strategies on a small JSSP data set

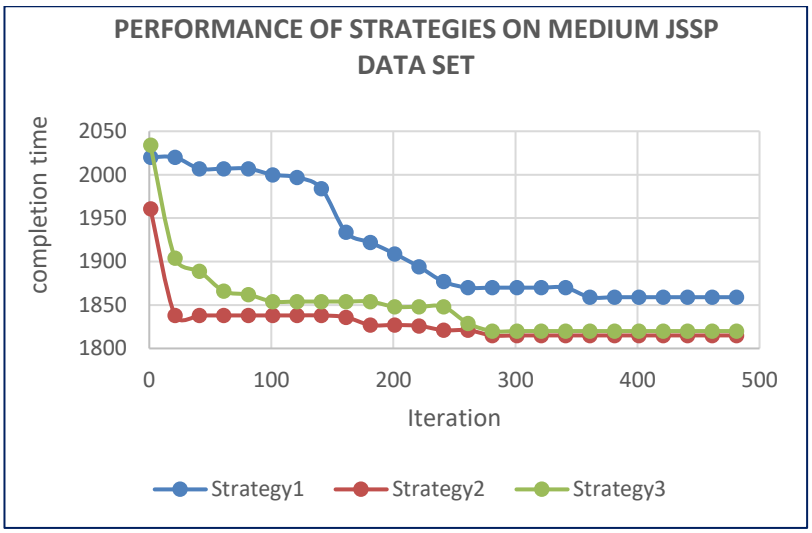


Figure 92 Performance of different strategies on a medium JSSP data set

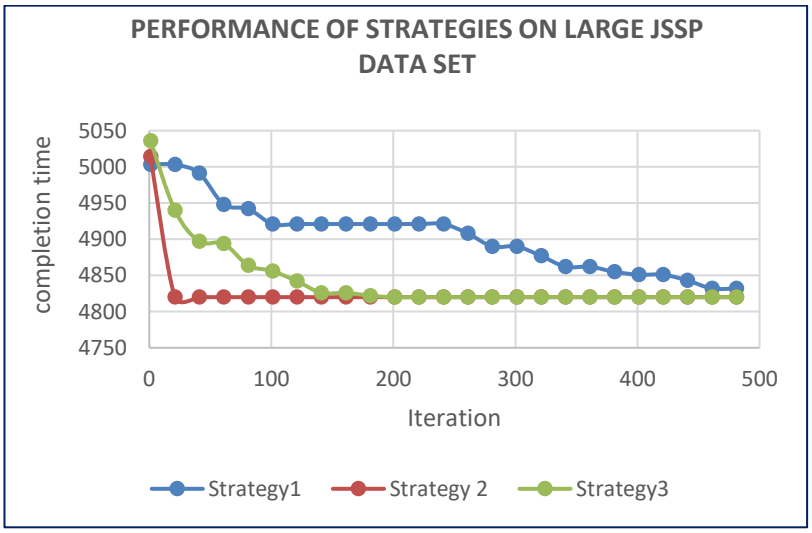


Figure 93 Performance of different strategies on a large JSSP data set

9.5 Comparison and result discussion

9.5.1 Single operator performance

The aim of this chapter was to answer the question whether the same permutation operators can be effective across different domains. Based on the conducted analysis, the summary of the results is presented below.

- The problem specific operators showed a superior performance compared to typical permutation operators across all data sets of both problems.
- Despite a fundamental difference between CSP and JSSP in the chromosome structures, decoding procedures and fitness function

calculations, PMX crossover consistently demonstrated good results after Intelligent heuristics.

- PBX came third on JSSP, whereas LOX on CSP.
- CX performed relatively poorly on both problems: it was the fifth on the JSSP and fourth on CSP.
- The performance of the traditional mutations was consistent across all the experiments. In order of efficiency, they can be sorted as Swap, Scramble and Simple.

Although there is no explicit comparison of the same operators on the JSSP and CSP available in the literature, a comparison of the obtained results will be made with other problems solved by EA with the use of permutation chromosome representation and traditional crossovers.

There are several studies that showed that the modified or problem-specific operators are more beneficial for the algorithm. This has been proven across different domains such as Electric distribution network problem (Carrano et al. 2006), flow shop scheduling problem with multiple factories (Gao, Chen and Liu 2012), capacitated vehicle routing problem (Nazif and Lee 2012), corridor allocation problem (Kalita and Datta 2014) and Cloud Infrastructure Management (Pascual et al. 2015).

PMX is regarded as one of the most popular crossovers for permutation encoded chromosomes (Kumar, Gopal and Kumar 2013). The effectiveness of the PMX crossover can be confirmed by the fact that it was applied to a wide range of domains, for instance in Project Management (Yuan and Zhi-Ping 2006), Assignment problem (Sahu and Tapadar 2007), Packing non-identical circles within a rectangle with open length problem (He and Wu 2013) and many others.

In the comparative studies, PMX outperformed PBX and CX crossovers on the facilities layout design (Chan and Tansri 1994). It also showed better results than PBX in 9 out of 11 instances of the TSP with the population size set to 100, but showed opposite results when the population size was reduced down to 50 (Kumar, Gopal and Kumar 2013). In other studies, PMX came third on the Travelling Salesman Problem (Tagawa, et al. 1998) and movie distribution problems (variation of TSP) (Zhang and Zheng 1995).

In direct comparison with LOX, PMX delivered almost identical results when they were applied to solve a university timetabling problem (Kumar, Gopal and Kumar 2013). The effectiveness of PMX and LOX crossover can be explained by their ability to preserve substrings which stand for diagrams. However, PMX's ability to exchange the genes by mapping them in both parents has proven to be more effective than preservation of the relative position of the trips from the second parent.

Interestingly, LOX crossover was more effective than PBX crossover on the CSP, whereas PBX produced a better schedule than LOX on JSSP. This contradiction can be explained by the problem structures and applied decoding procedures. Figure 94 presents a special case which distinctively demonstrates the different effect of LOX crossover on both problems. The genes belonging to different diagrams are represented in different colours and have different textures.

Following LOX logic, the genes occupying the positions on the intervals from one to two and from six to eight were passed on from Parent 1 to Child 1. The missing genes were copied from the second parent keeping their relative position.

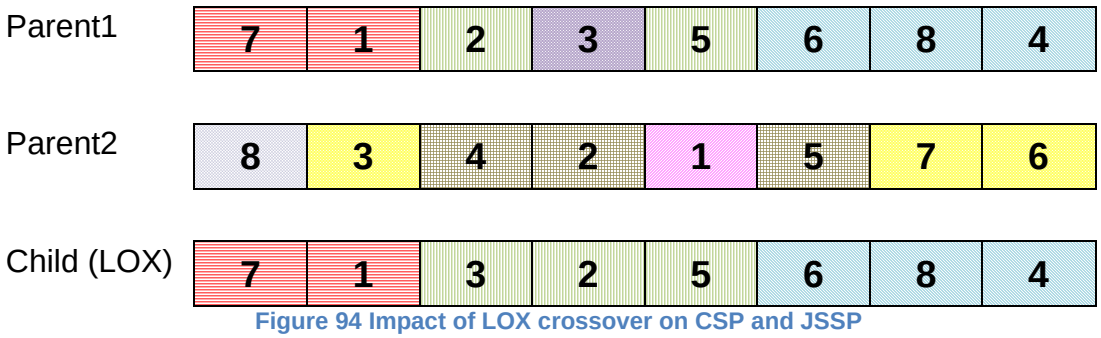


Figure 94 Impact of LOX crossover on CSP and JSSP

It can be seen that two diagrams (7,1) and (6,8,4), were preserved and, by re-arranging the position of the other genes, two diagrams (3) and (2,5) were combined into one (3,2,5). Thus the number of the diagrams reduced from four (in Parent 1) to three (in Child 1). A decrease in the number of diagrams typically causes a drop in the driver and taxi costs, which are usually the largest costs constituting cost function. However, such permutation might not make such a significant impact for the job shop schedule, where it would only swap job 3 and 2.

In contrast, the effect of PBX crossover, illustrated in Figure 95, is directly opposite to LOX. Assuming the trips 7, 2, 3 and 8 were passed from the first

parents and the rest were copied from the second parents, it can be noticed that a large portion of the trips have exchanged their relative positions. In the context of CSP it denotes that consecutive trips were swapped and can no longer be placed into the same diagram resulting in the increase of the overall number of diagrams, and subsequently the total cost of the schedule. However, the given exchange of job sequence will not cause such a disturbance in the schedule of JSSP.

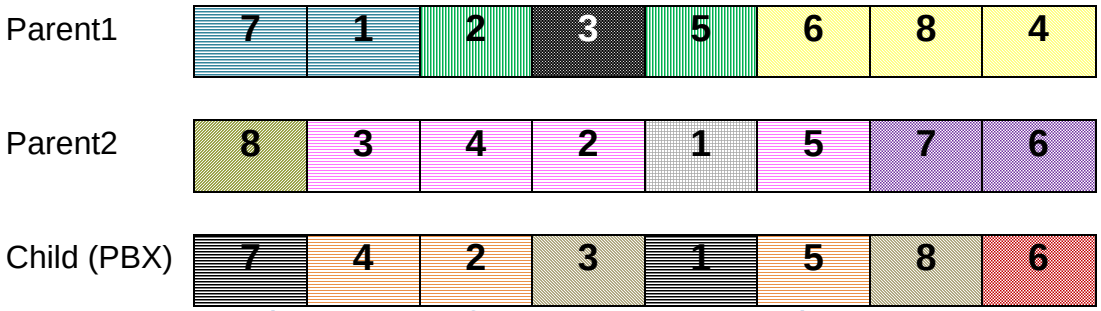


Figure 95 Impact of PBX crossover on JSSP and CSP

In addition to its suitability for JSSP, PBX crossover showed superior results mostly on the travelling salesman problems (Kumar, Gopal and Kumar 2013, Abdoun and Abouchabaka 2012, Sharma and Tapaswi 2013). However, no such comparison was found in other domains.

CX crossover exhibited poorer performance than other operators in a variety of the experiments, for instance in Tagawa, et al. (1998), Zhang and Zheng (1995), Xu, Xu and Gu (2011), Kumar, Gopal and Kumar (2013), Abdoun and Abouchabaka (2012). However, it outperformed PMX and LOX on the University design timetable problem (Chinnasri, Krootjohn and Sureerattanan 2012) and the RNA folding problem (Wiese and Glen 2003).

To conclude, there is strong evidence that the heuristic which is based on domain specific knowledge tends to outperform more general operators. As for the other operators, the evidence is weaker due to comparison with other research not being made in absolutely identical settings. The factors such as utilisation of different genetic parameters such as population, size, crossover and mutation rates, number of iterations the algorithm ran for, use of additional techniques (repair operators, special procedures to population initialisation) could have a significant impact on the final solution.

9.5.2 Strategies

The summary of the key results of strategies is presented below.

- The second strategy delivered better results than the first and third strategies.
- In five out of six cases, the single operator algorithm was more effective than any of the strategies;
- The contribution of the operators in the third strategy corresponds to their effectiveness when they are applied on their own.

However, the fact that single operator EA in general performed better than multiple operator EA contradicts some of the studies reported in the literature.

For example, Zhang, Wang and Zheng (2006) showed the positive impact of the application of synergy of operators on the performance of EA when applied for the Flow Job-Scheduling problem. Along with LOX and PMX crossovers, they employed the less popular C1 and NABEL. They also embedded Swap, Insert and Inverse (special case of Scramble) mutation operators. Their proposed selection strategy was somewhat between Strategy two and Strategy three. They tested each crossover at every iteration similar to Strategy three, however the accumulated operators scores were kept through the entire evolution (Strategy three used only the scores obtained at each iteration). Despite its effectiveness, the possible limitation of this approach is that crossover which has been productive at previous iterations might not be effective at the current iteration.

Similar to this, in the study conducted by Elaoud, Teghem and Loukil (2010) the selection process was similar to the third strategy, but the operators were probabilistically selected according to their scores.

Hong, Wang and Chen (2000) used similar logic in operator selection and reported positive results as well. However, binary chromosome representation and corresponding operators were employed while the algorithm was tested on linear and non-linear functions.

Unlike strategy 2, Kim, Gen and Yamazaki (2003) applied each combination of crossovers and mutation to each individual before they were placed back in the

population. In addition, they have incorporated a fuzzy logic controller for the adjustment of the operator probabilities.

This discrepancy in the results obtained in the discussed studies and given research can be due to the following reasons.

1. The different set of operators was included in the algorithm.
2. The execution time was not taken into account.
3. The operator selection mechanism has been tailored to the selected operators.
4. The algorithms have been evaluated on different problems.

9.6 Conclusions

The last two chapters presented experimental results of the comparative performance of various genetic operators. The main purpose of the conducted experiments was identification of effective crossover and mutation operators which can be incorporated in the EA-based automatic scheduling system.

It was found that domain specific Intelligent heuristics which explicitly preserve good building blocks is more valuable than those which do it implicitly. As for the standard crossover and mutation operators, PMX crossover and Swap mutation are proven to be the most effective conventional genetic operators for both JSSP and CSP problems.

Another observation which has been made is that PBX crossover tends to be more efficient for the problems where the genes are decoded consecutively such as Job Shop Scheduling Problem and TSP. On the other hand, LOX is more suitable where the relative position of the genes plays a greater role and where some genes can be temporarily skipped during the decoding procedure, such as Crew Scheduling, Timetabling and various bin packing problems.

CX crossover showed poor results in the given experiments as well as in other trials conducted in the literature, although it managed to be more effective in two other domains. This means that it needs to be carefully examined before application in new domains.

Although the incorporation of multiple operators was beneficial for a large number of studies in the literature, this was not confirmed in the current research. With regard to the strategies' performance in general, the second strategy performed the best even at the cost of longer time spent per iteration, followed by the third and first. However, Strategy three demonstrated an interesting property: the ratio of the contribution of each crossover corresponded to the results of their single tests. This attribute can be used for selection of operators when designing a new EA as strategy three can replace time consuming trial-and-error methods, but still provide stable results.

Chapter 10. EA Adaptation to the CSP problem

10.1 Introduction

In Chapter 8 and Chapter 9, it was proven that it is not efficient to apply the same chromosome representation for the solution of both CSP and JSSP. It was shown that job-based chromosome representation does not allow for having different orders of operations' assignment in JSSP and to alternate the sequence of drivers in CSP. In order to design an efficient algorithm for evaluation by industrial experts, this section returns to the leading problem in this research, which is CSP, and suggests ways of enhancing the proposed algorithm. The CSP has been selected due to higher complexity expressed in a large number of rules and regulations. Evaluation of the algorithm using this problem enables a more accurate conclusion as to whether EA can be used in the real life settings.

The analysis of the chromosome structure utilised in section 8.4.8 suggested that, in addition to the position of the trip in a chromosome, the location of the driver in the chromosome plays an important role and can affect the formation and cost of schedule. Therefore, an additional series of the experiments is carried out in order to investigate whether manipulation of the position of the drivers in the chromosome is beneficial.

Two types of experiments are conducted in this chapter. The first part examines the effect of additional operators on the evolution of the second chromosome component. The second part of the trials will explore whether some modifications in the decoding procedure related to the assignment of the drivers can improve the construction of the schedule. Since the experiments conducted in the previous chapter proved the high efficiency of the customised operators, they will be adopted in this chapter.

10.2 Chromosome representation supporting evolution of drivers' role

The limitation of the previously used universal chromosome representation is that the order in which drivers were encoded into chromosomes was the same for the entire population (blue part of the chromosomes on Figure 96) and it remained unchanged in the course of the evolution. Given that the number of available drivers usually exceeds the number of trips, drivers at the end might not be reached by the decoding procedure unless the preceding drivers did not have sufficient route and traction knowledge. This can lead to the situation of a trip being assigned to a sub-optimal driver and the production of a more expensive schedule.

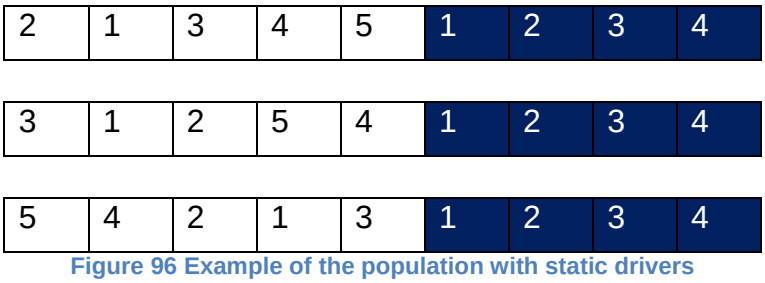


Figure 96 Example of the population with static drivers

Figure 97 displays the new population for the EA with driver and trip evolution for CSP consisting of five trips and three drivers. Like the trips, drivers' genes are generated at random.

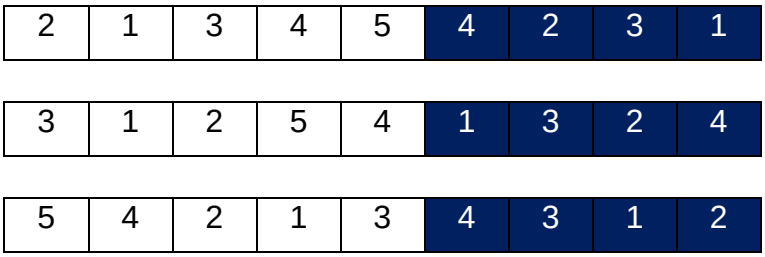


Figure 97 Example of the population with evolving drivers

In addition to the improvement in the decoding procedure, it is expected that the evolution of the second part will also enhance the optimisation process as the solution will be approached from two different directions: drivers and trips. The possible downside of the proposal is it might interfere with the speed of the evolution and thus convergence might be slower due to the expanded search space.

10.2.1 Genetic operators and their effectiveness in the driver evolution

This section presents the operators which will support the drivers' evolution and evaluates their effectiveness. In order to achieve a reasonable convergence of the algorithm, it was decided that only mutation operators would be employed to evolve the second part of the chromosome. The mutations which will be operating on the driver's part are Swap, Insert, Scramble and Simple. The best performing in the previous chapter, Intelligent mutation and Intelligent crossover, are applied to the first part of the chromosome responsible for the trips.

The trials have been carried out ten times for each mutation type. The data used for the experiments are the same and as specified in Table 12. The graphs presented in Appendix 8 illustrate the evolutional process with drivers participating in mutation.

In terms of the behaviour of the algorithm, no explicit differences were identified. The functions across all data sets have converged relatively quickly at the beginning of the algorithm and have not substantially evolved after that point. This might be due to two factors. Firstly, the Intelligent Crossover and Intelligent mutation strongly dominated the process and the effect of the driver change was less significant compared to the evolution process of the trips. Secondly, the expansion of the search space caused by the increased number of possible combinations of drivers and trips prolonged the exploration part of the algorithm. Therefore, the conclusion of the efficiency of tested operators will be drawn from their average results, displayed in Figure 98-Figure 100.

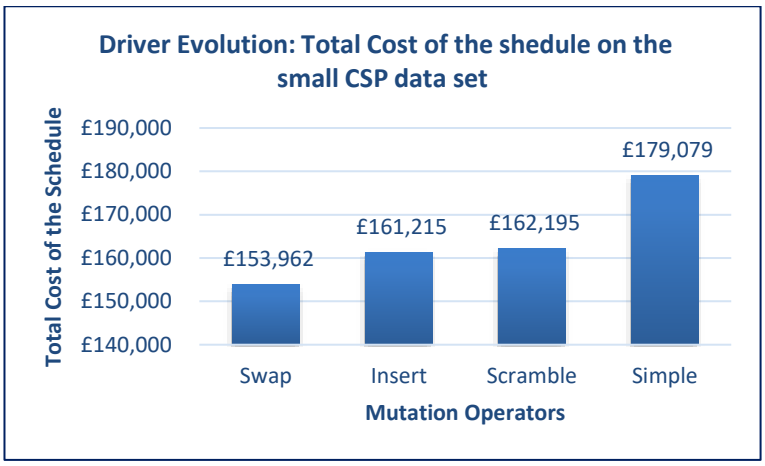


Figure 98 Final results of various driver evolution operators on a small data set

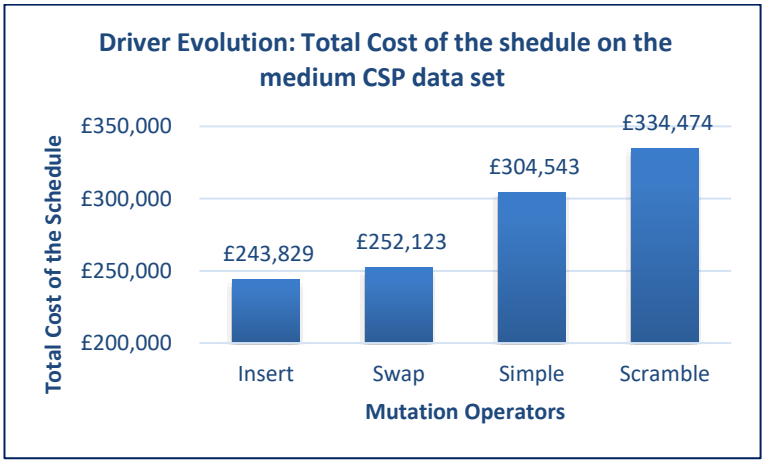


Figure 99 Final results of various driver evolution operators on a medium data set

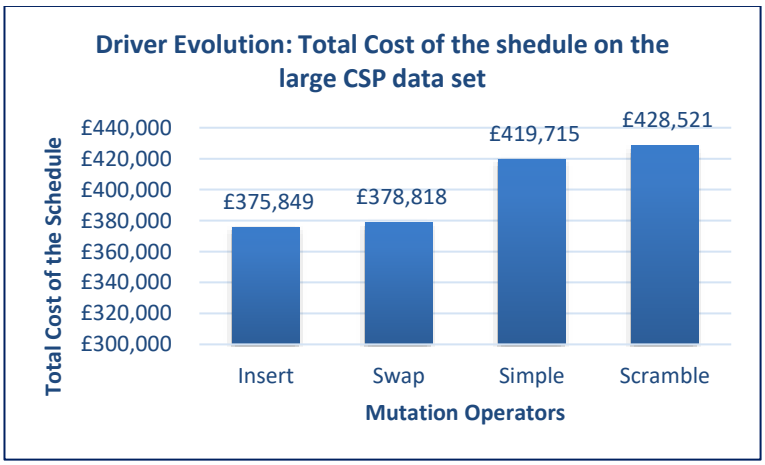


Figure 100 Final results of various driver evolution operators on a large data set

From the obtained results it can be noticed that Insert and Swap operators performed better than Scramble and Simple mutations. Insert mutation outperformed Swap mutation on the average and large size data sets by 4% and

1% respectively, but showed worse performance on the CSP 780 producing on average a 5% inferior schedule.

The gap between the Scramble and Swap mutation is more noticeable on the medium and large data sets, where the difference reaches 18% and 10% correspondingly. The observed difference on the small data set is not significant.

The Simple type of permutation performed the worst on all data sets regardless of their size. The gap between Simple and Scramble mutation is 10%, 9% and 3%. The gap between the best performing mutation and Simple mutation constitutes 15%, 28% and 11% on a CSP780, CSP1260 and CSP1980 respectively.

The initial comparison with the results reported in section 9.2.4 demonstrates that within the same amount of time and using the same data, the additional mutation of drivers showed worse results than the algorithm with the fixed drivers position. This issue will be investigated in greater depth in section 10.4.

10.3 Nearest Driver

Although the Evolving driver strategy discussed in the previous section is able to move the position of the driver in the chromosome, it still has two limitations. First of all, it does not guarantee that the “right” driver will always be on the loci from which it will be reached by the decoding procedure and, secondly, this approach is more computationally expensive as it deploys additional operators.

In order to tackle the inefficiencies of the previous approach, another method has been devised. However, unlike evolution of the drivers, it deals with the decoding procedure rather than genetic operators.

10.3.1 Fitness function adaptation for the Nearest Driver algorithm

The major difference to the existing decoding procedure discussed in section 8.4.8 is that the driver is selected depending on his proximity to the first trip in the diagram rather than his position in the chromosome. This means that looking for the driver to operate a first trip in the new diagram, the algorithm first identifies the driver who is located closely to the trip and then verifies whether the driver has been trained for that route and train type (Algorithm 7). If the driver does not

have the knowledge of that trip or train, another closest driver is checked. This process repeats until the driver has been found.

The advantage of the given logic is that all the drivers can be considered in the assignment process. However, the possible disadvantage is that this procedure can be more time consuming and might involve a driver who would be better suited to other trips.

Algorithm 7 FIND NEAREST DRIVER

```
1: DriverFound=FALSE;
2: FOR i=0; i<NDrivers;
3: DriverDistance[i]=TaxiTimes(Driver[i].Depot, Train[j].DepartureStation)
4: END
5: WHILE DriverFound=!TRUE
6: Position=FIND MIN(DriverDistance)
7: IF Driver[Position].RouteKnowledge==Trip[j].RouteKnowledge &&
8:Driver[Position].Tractionknowledge==Trip[j].TractionKnowledge;
9: DriverFound=TRUE;
10: ELSE
11: DriverDistance[Postion]=max(DriverDistance)+1; //so this driver would
not participate in the consideration again
12: END
13: END
```

Thus in order to empirically evaluate the effectiveness of such a procedure, ten tests have been run using three data sets defined in the Table 12. The parameters of the algorithm remain unchanged and the search will be guided by the Intelligent crossover and mutation operators since their results outperformed other operators.

10.3.2 Nearest Driver results

The graphs in Appendix 9 illustrate one of the runs of the algorithm with the incorporated procedure of finding the Nearest Driver and compare it against the standard decoding procedure. As the logic of genetic operators has not been affected, the behaviour of the functions remains very similar. However, as can be seen from the graphs, the starting solution is on average 20 % smaller than the standard decoding procedure. The Nearest Driver procedure also has converged quicker which is due to the fact that the search space is reduced by

fixing the driver to the first trip in the diagram. The average final results for the three data sets are presented Table 21 and they will be discussed in depth as well as compared with other successful techniques in the following section.

Table 21 The Nearest Driver experimental results

Data Set	Total Cost of the Schedule
Small (CSP_780)	91 425
Medium (CSP_1240)	137 381
Large (CSP_1980)	246 058

10.4 Comparison of all successful techniques

This section provides a detailed comparison among the best performing techniques from each experiment section: single operator, multiple operators, evolving and Nearest Driver.

In terms of single operator, Intelligent crossover and mutation are the problem-specific genetic operators which achieved the best results when compared with the other four standard crossover operators and three standard mutations, and their result will be used for comparison. With regard to the multiple operators, the second strategy where all the crossovers and then mutations operators were applied together outperformed the strategy with the random selection of operators and embodied operator trials.

The algorithm with the Insert mutation delivered better results on two out of three data sets than other driver evolution mechanisms and will be included in the comparison as well. Finally, the Nearest Driver approach with Intelligent crossover and mutation in its core will be included in the evaluation.

The comparison and analysis are based on a wide range of crew scheduling objectives such as the daily cost of the schedule, number of diagrams, workload distribution, deviation from the target shift length and throttle time.

Actual Cost of the Schedule. Actual Cost represents the day cost of the schedule, which is made up of driver payments and taxi costs. Figure 101-Figure 103 illustrate the cost breakdown in the solution obtained by four algorithm configurations on the small, medium and large data sets.

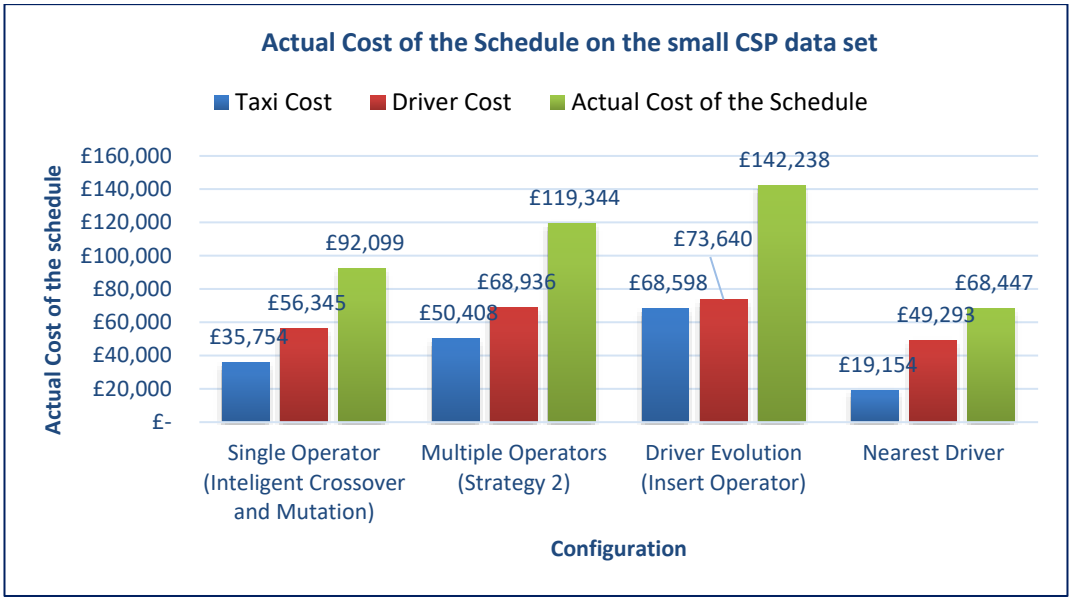


Figure 101 Comparison of EA configurations: Actual Cost of the Schedule of the small CSP data set

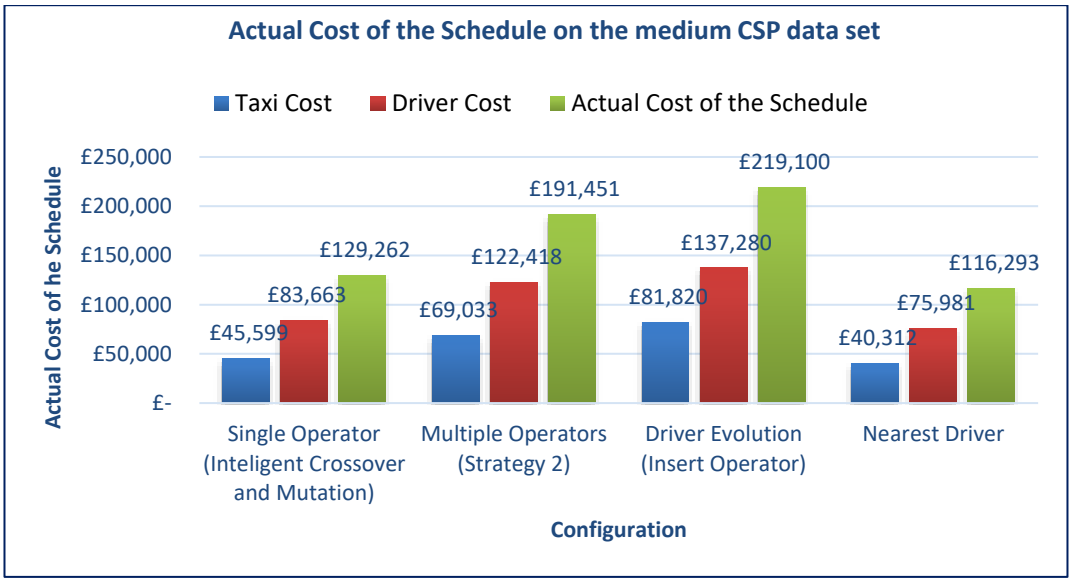


Figure 102 Comparison of EA configurations: Actual Cost of the Schedule of the medium CSP data set

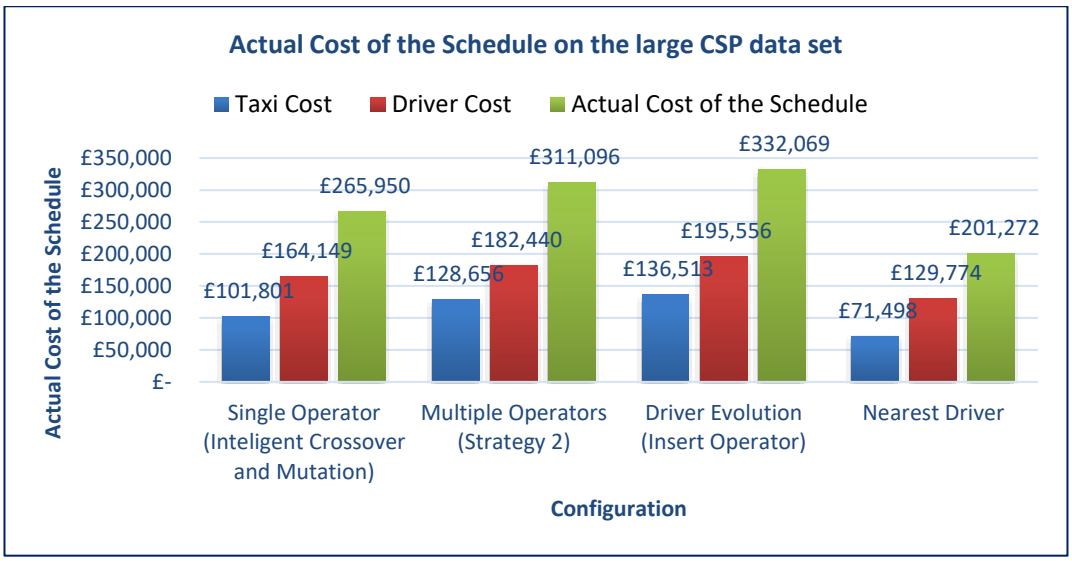


Figure 103 Comparison of EA configurations: Actual Cost of the Schedule of the large CSP data set

The Nearest Driver approach produced better results in terms of both taxi and driver costs, followed by the single operator approach. The multiple operator approach (Strategy 2) showed better results than the Driver Evolution approach on the small and medium data size, but performed worse than Driver Evolution on the large data set.

The taxi cost in the Nearest Driver approach was smaller on average by 29% than the Single Operator approach with the standard decoding procedure. This shows that the decoding procedure which takes into account the proximity of the trip to depot is more effective than that which considers only route and traction knowledge of the driver. Furthermore, it does not only reduce the cost of the taxi by shortening the deadhead trip from depot to the remote station, but also minimises the driver cost by 14% since the driver has to spend less time on the transfer trip. This reduced the day cost of the schedule by 20%.

Although the second strategy dealt with several operators and had the means to select the best offspring which could be generated, the day cost of the schedule is greater by 24% on average than the schedule constructed by the algorithm with only one operator. There are two possible explanations for this phenomenon. Firstly, the observed large taxi cost suggests that the algorithm has not been run for a sufficient number of iterations (as the algorithm progresses, fewer deadhead trips are left in the diagrams). Secondly, the poor performance of the second strategy could be due to embedded poor performing operators (such as CX, PBX

crossovers and Simple and Scramble mutations) as they can lead the search to the wrong region by generating a solution which has slightly better phenotype than the worst individual in the population, but a poor phenotype which other operators will struggle to improve in subsequent iterations.

The driver evolution approach produced a 12% worse day than Strategy 2. The taxi cost exceeded the cost in the schedule produced by the second strategy by 15% and the driver cost was higher by 8%.

***Number of diagrams*.** Despite not being the explicit objective, the reduction in the quantity of diagrams, which is the number of required drivers, implies not only a cost reduction, but also the acceptance of more customer orders which make a positive impact on the revenue. A comparison of the average number of diagrams constituting the schedules across different data sets is displayed in Figure 104-Figure 106.

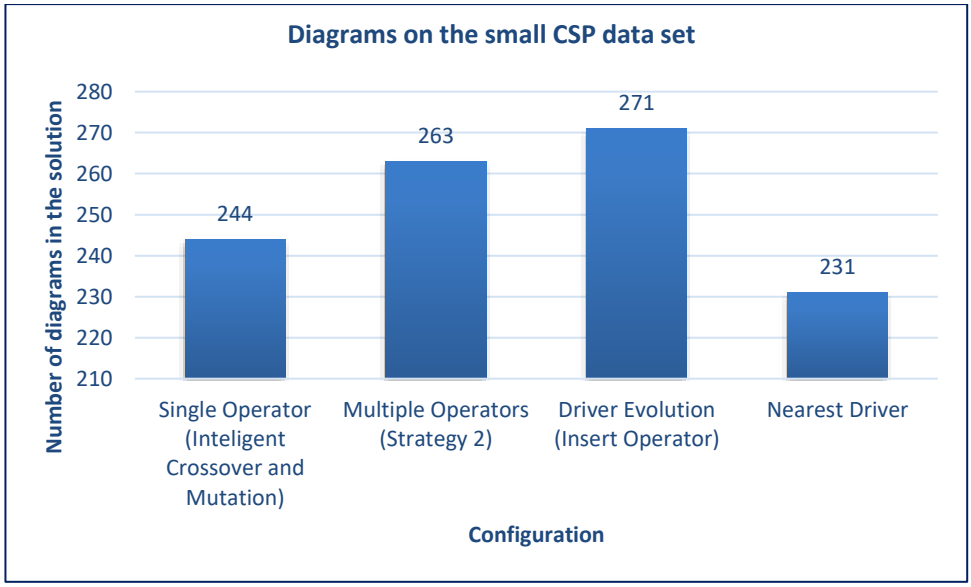


Figure 104 Comparison of EA configurations: Number of diagrams on the small CSP data set

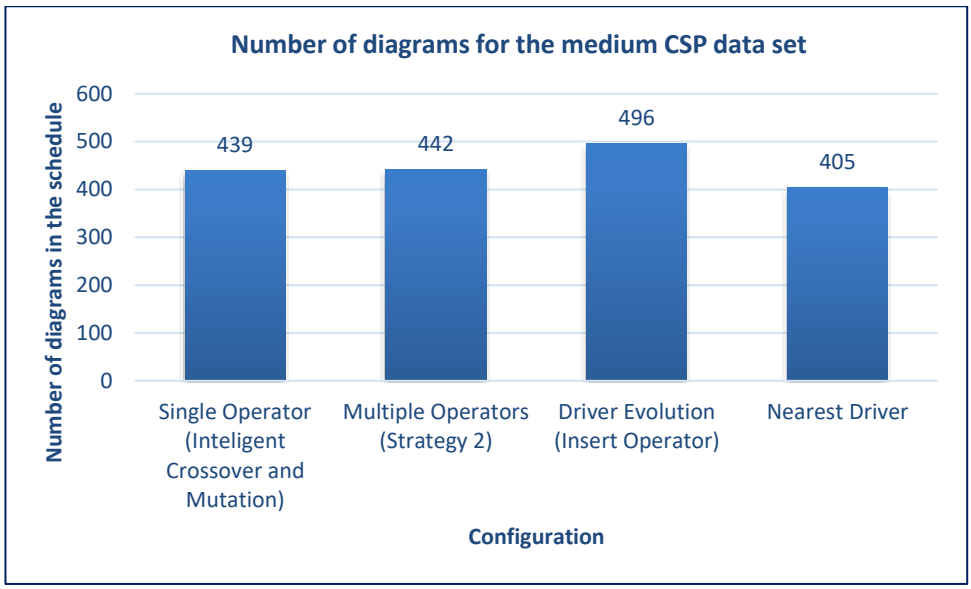


Figure 105 Comparison of EA configurations: Number of diagrams on the medium CSP data set

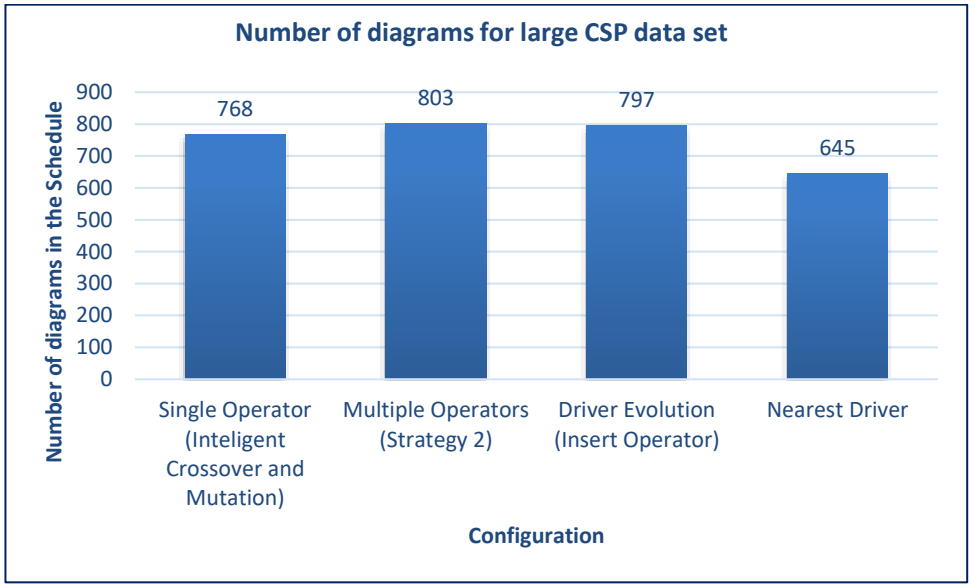


Figure 106 Comparison of EA configurations: Number of diagrams on the large CSP data set

The number of diagrams resulting from the application of the Nearest Driver strategy on average was significantly smaller than other algorithms by 13, 34, 123 than the single operator algorithm; 32, 37 and 158 than strategy 2; and 40, 91, 152 than Driver Evolution. This was caused by the reduction in the transportation time at the beginning of the diagrams, so more trips could fit into the working time, therefore fewer diagrams were needed to cover all the trips.

Moreover, this factor had a positive impact on the throttle time (Figure 107-Figure 109). With the application of the Nearest Driver procedure, the throttle time on

averge rose by 5%, 4% and 9% (for CSP 780, CSP1260 and CSP 1980 respectively) in comparison to the standard procedure.

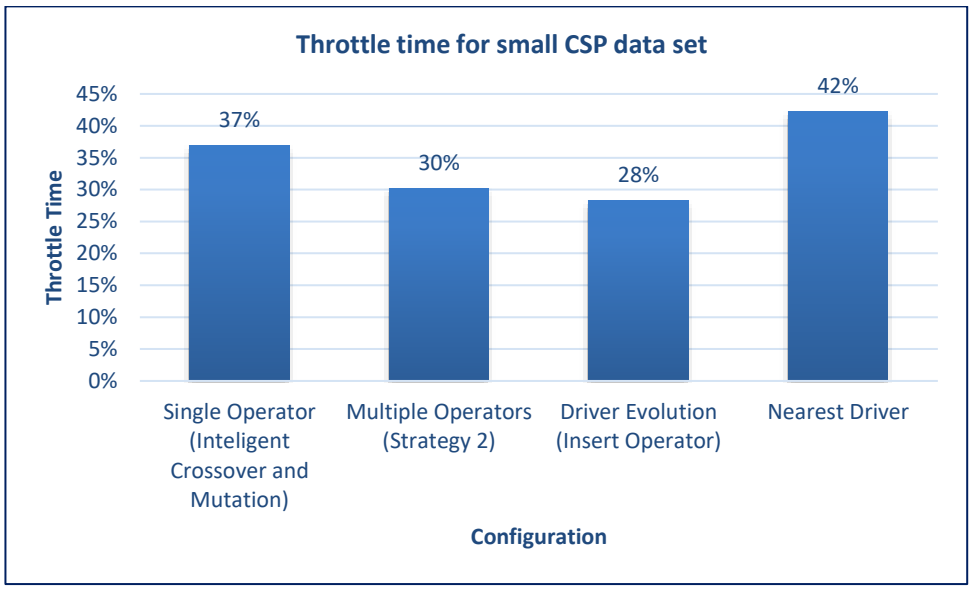


Figure 107 Comparison of EA configurations: Throttle time on the small CSP data set

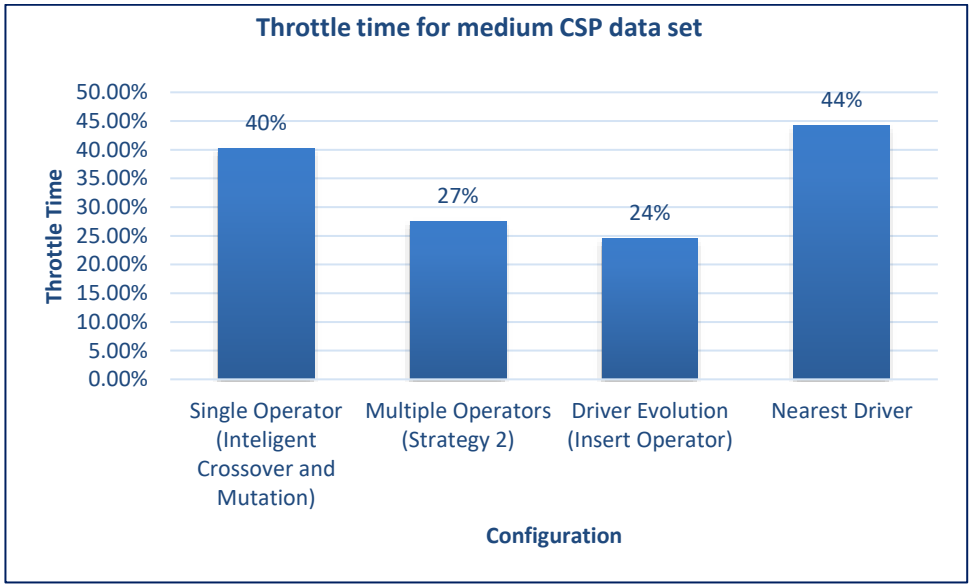


Figure 108 Comparison of EA configurations: Throttle time on the medium data set

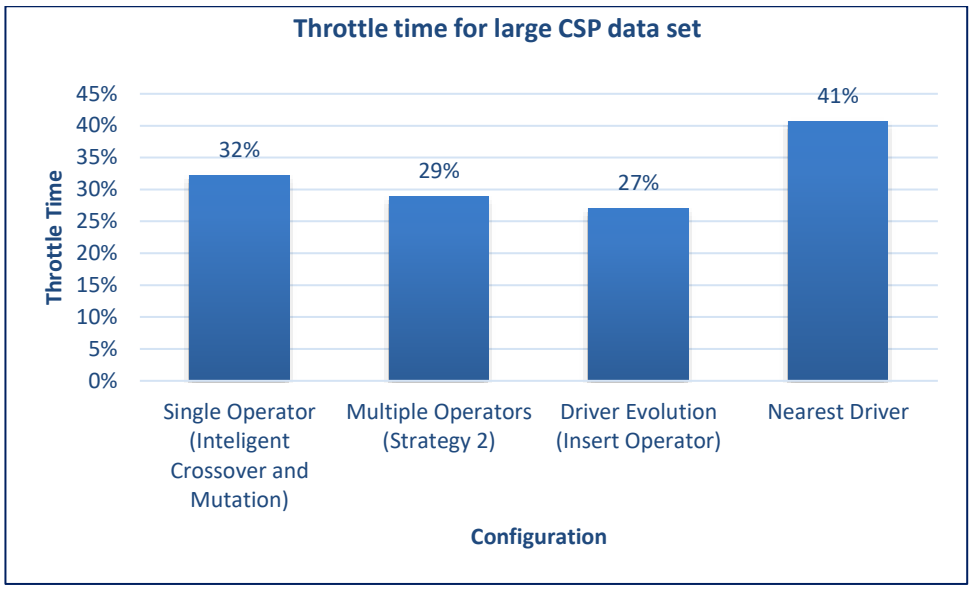


Figure 109 Comparison of EA configurations: Throttle time on the large CSP data set

The next group of indicators are oriented towards the achievement of the long term objectives.

Target shift length. As established earlier, the presence in a schedule of diagrams with a length of 510 minutes (8 hours and 30 minutes) helps to ensure that the number of hours left or in excess of the contract at the end of the year will be minimised. As the cost of unused and excess hours is the same, an absolute deviation from the target shift length has been calculated. The bars on Figure 110-Figure 112 demonstrate the average deviation of diagrams from the 510 mins.

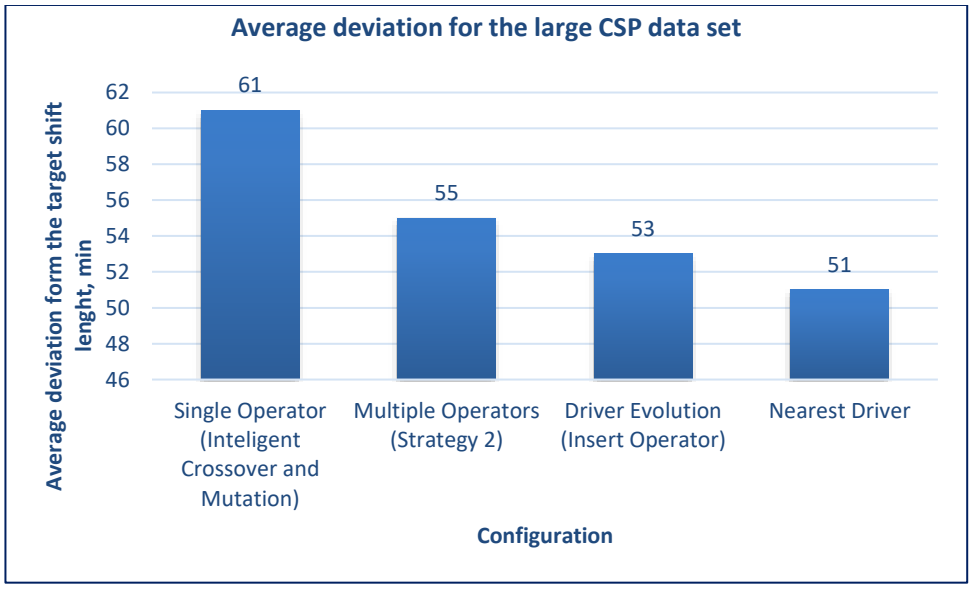


Figure 110 Comparison of EA configurations: Average deviation on the medium CSP data set

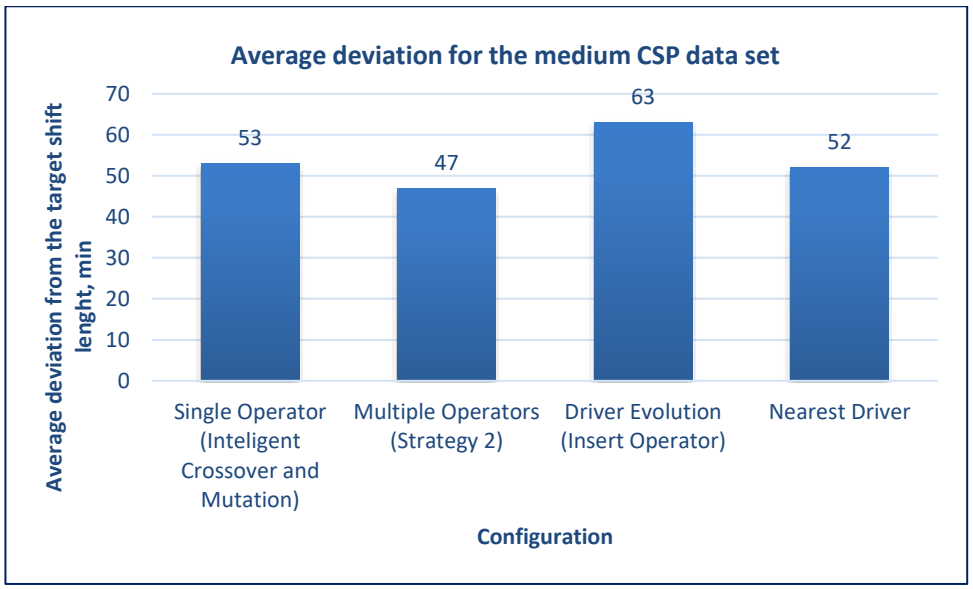


Figure 111 Comparison of EA configurations: Average deviation on the large CSP data set

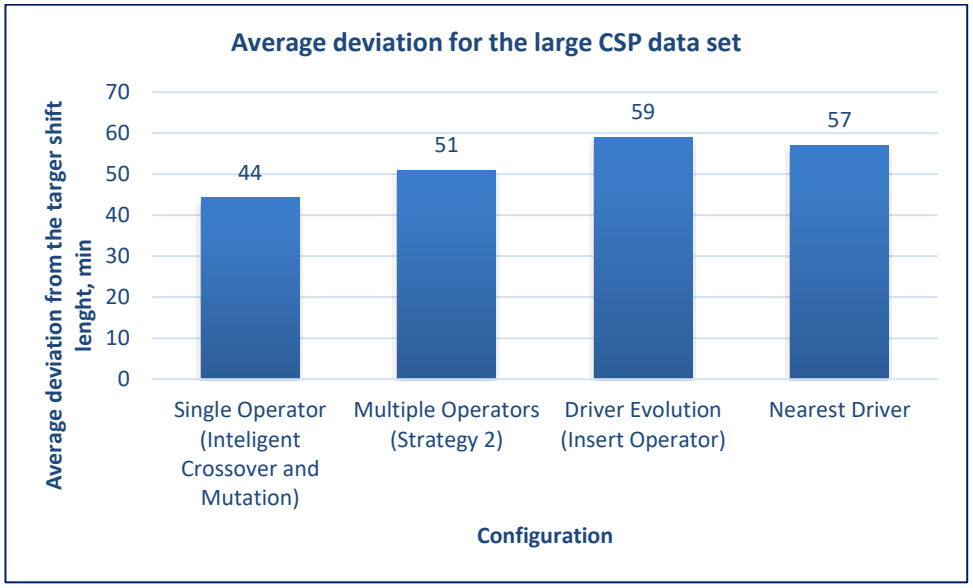


Figure 112 Comparison of EA configurations: Average deviation for the large CSP data set

There is no conclusive evidence with regard to the comparative abilities of the tested algorithms to produce a schedule with a certain deviation from the target shift length. In general, the average deviation in the solutions fluctuates between 44 and 63 minutes, but there is no tendency or correlation with data set or data size. This is because the fitness function plays only a secondary role in regulation of the deviation from the target shift length. The decoding procedure plays a greater role in this process as it decides whether a new trip can be accepted into the diagram or not. This part of the decoding mechanism is the same across all the algorithms which explains the absence of any significant difference in the results.

Workload distribution. Another aspect of the schedule which ensures that the number of the excess and unused contract hours will be minimized is the workload distribution amongst the depot. The graphs in Figure 113-Figure 115 present the average standard deviation in the workload distribution among depots.

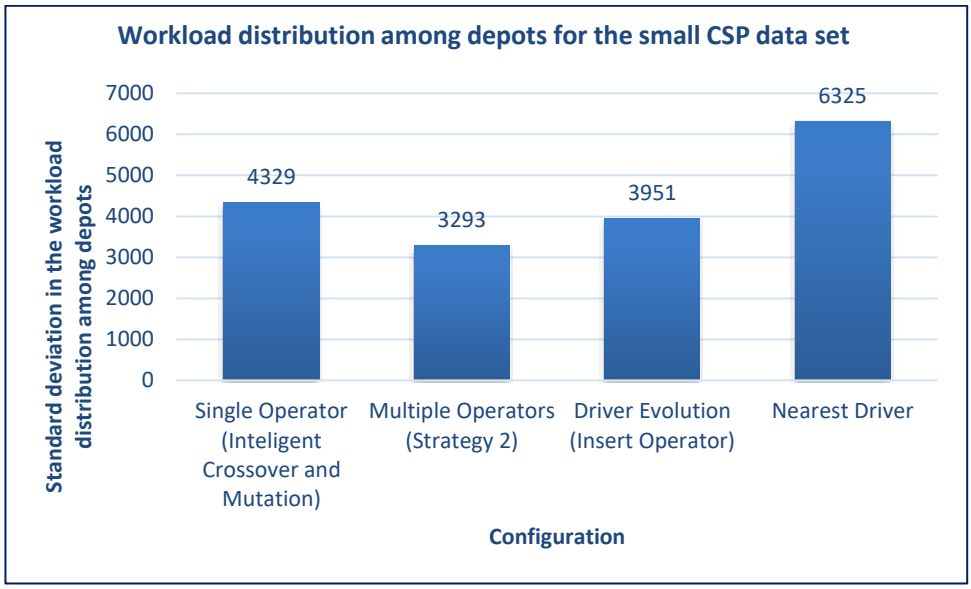


Figure 113 Comparison of EA configurations: Workload distribution among depots for the small CSP data set

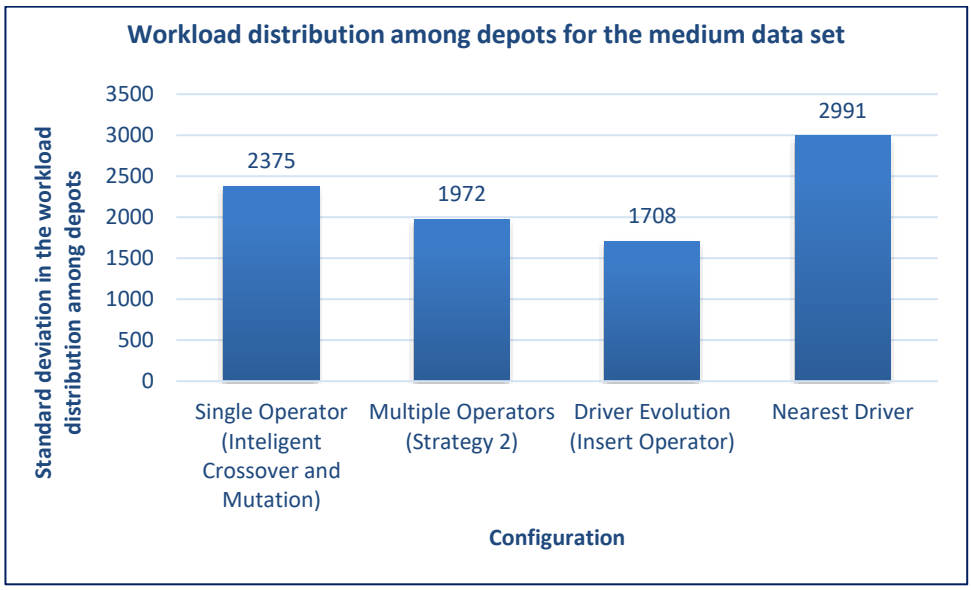


Figure 114 Comparison of EA configurations: Workload distribution among depots for the medium CSP data set

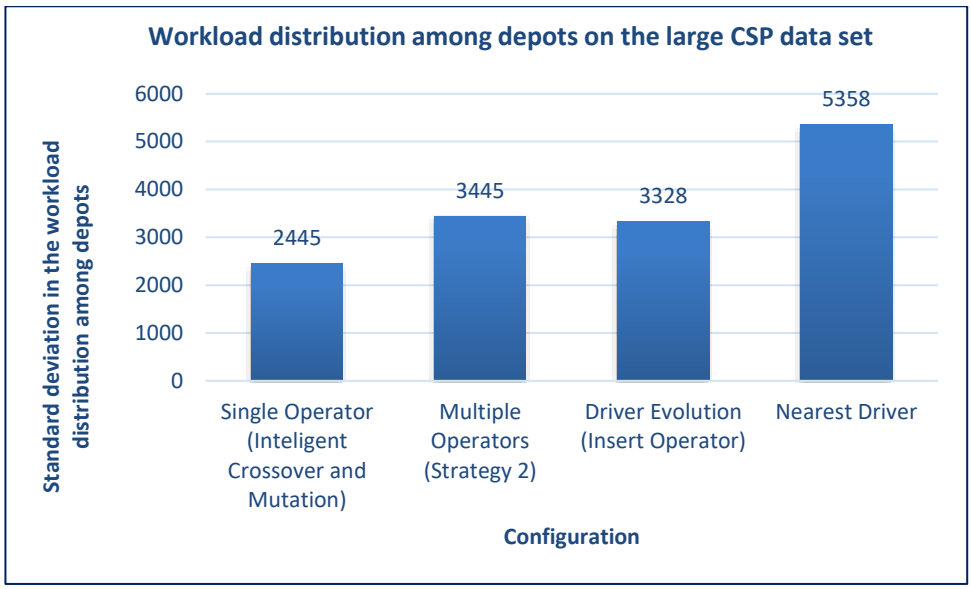


Figure 115 Comparison of EA configurations: Workload distribution among depots for the large CSP data set

Unlike the short-term cost indicators, the Multiple Operator approach and Driver Evolution approach delivered better results on two out of three data sets from the long-term perspective on the schedule. This can be explained in two possible ways. On the one hand, it may show their capabilities for production of a geographically balanced schedule. This might be especially relevant for the Driver Evolutionary strategy because it has the capacity to adjust driver position in response to the fitness function. On the other hand, taking into account the large taxi cost, it might be a sign of a poorly converged algorithm as the taxi cost has a higher weight in the fitness function (£120 per hour for a taxi, against £40 per hour for unequal workload distribution).

The previously successful Nearest Driver strategy achieved the worst balance in the workload allocation. This showed that selecting the driver who is closer to the trips is not always ideal for satisfaction of all objectives, since it takes some work from the remote depots leaving the drivers with different workloads in different locations.

***Fitness function*.** Finally, the algorithm which will be applied to the real data will be selected on the cost function basis. The bar charts in Figure 116-Figure 118 illustrate the total cost of the schedule including the daily and the penalty costs.

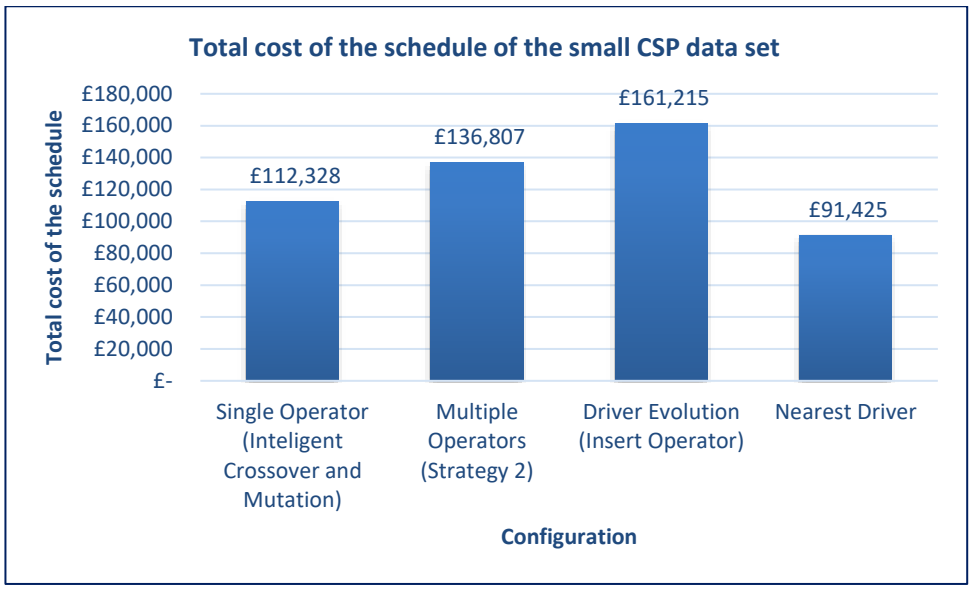


Figure 116 Comparison of EA configurations: Total Cost of the Schedule of the small CSP data set

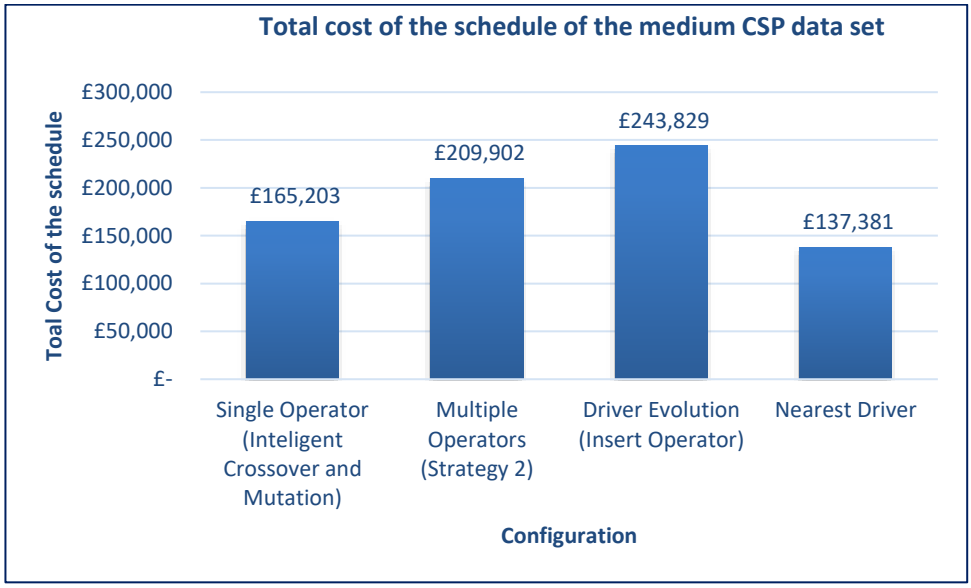


Figure 117 Comparison of EA configurations: Total Cost of the Schedule of the medium CSP data set

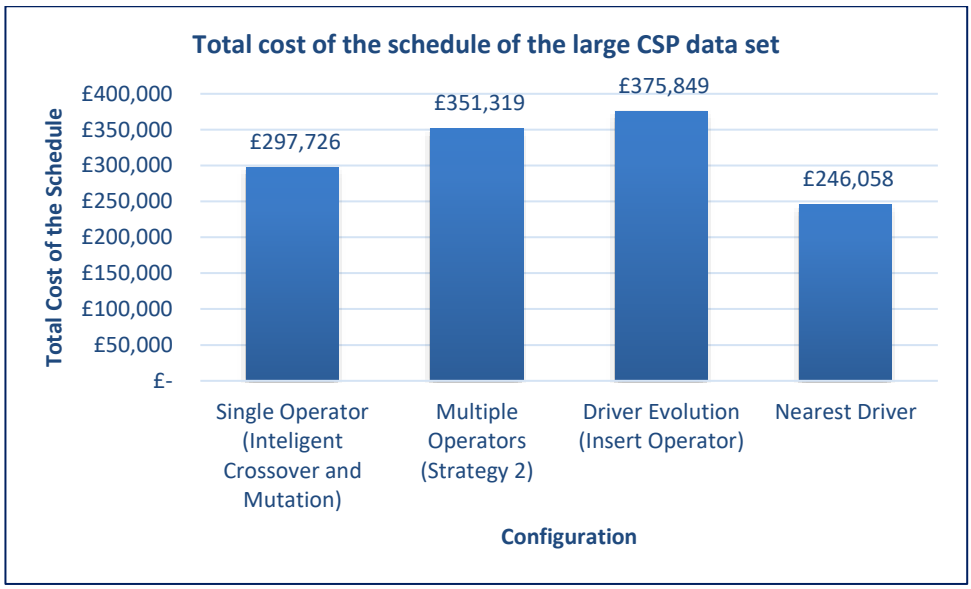


Figure 118 Comparison of EA configurations: Total Cost of the Schedule of the large CSP data set

The Nearest Driver approach constructed the most cost effective schedules, whereas Evolving Drivers produced the poorest ones. In general, the Nearest Driver approach produced better solutions in respect of short-term, daily objectives, while Evolving Operator ensured better work allocation achieving better long-term goals. Figure 119 presents an example explaining these structural differences in the constructed schedules. In this figure, the trips included into the same diagram and corresponding drivers are presented in the same colour and have the same texture.

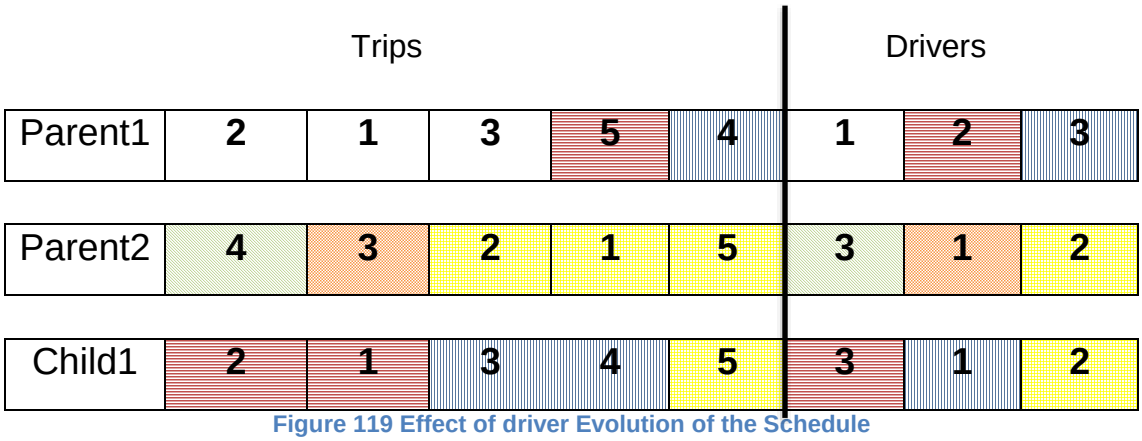


Figure 119 Effect of driver Evolution of the Schedule

Assuming that the first diagram in the first parent (2,1,3) has the highest throttle time, it is copied to the first child. Trips four and five become reversed, and if they were sequential they potentially could be placed into the same diagram and the number of diagrams would be reduced. However, the third driver has also been reinserted, and possibly cannot operate trip three. It might be because he has not been trained for this task or because he is from the remote depot and would run

out of time if he or she performed this trip. This situation results in splitting a good diagram into two, which might have a knock on effect on the rest of the schedule. At the same time, this creates three diagrams and, having three drivers in the data, it can be assumed that the standard deviation in the workload distribution would be less than if two drivers were assigned to the given trips.

The results of multiple and single operator EAs lie between Nearest Driver and Driver evolution strategies. This is because they are able to adjust the trips to the driver position in the chromosome, but might not be able to reach some of the drivers. EA with single domain specific heuristics performed better than the algorithm equipped with the same heuristic as well as other general operators. This is because it required less time for each iteration and embedded a powerful mechanism of preservation of high quality diagrams, which allowed construction of a schedule faster by avoiding a "random walk" search.

As the satisfaction of various objectives was incorporated into the cost function, cost-wise the Nearest Driver approach outperformed Single operator, Multiple Operator and Driver Evolution by approximately 17%, 32% and 40% respectively. For that reason, it will be incorporated into the EA which will be used to produce a schedule using real life data.

10.5 Conclusion

The aim of this section was the design of an effective algorithm for the solution of CSP. It took previously successful algorithms with Intelligent genetic operators and enhanced them with two mechanisms: evolution of the drivers and assignment of the first trip in the diagram to the driver from the previous depot.

The comparison of the above approaches showed the effectiveness of the later one, even though it did not achieve a balanced workload allocation. However, it gained advantage in terms of the low daily cost which compensated for inequality in the workload distribution.

The next chapter will consider its adaptation and deployment on the real data sets.

Chapter 11. Implication of the research for an organisation

11.1 Introduction

This chapter discusses the development of the proof of concept of the automatic scheduling system and its effectiveness and applicability to the real organisation. It explains how the incorporated EA has been adapted in order to accommodate some problem-specific aspects of the real CSP as well as how real life data have been transformed in order to meet the format requirements of the algorithm. Then, it moves on to examination of the benefits and risks of application of the standard and generalisable EA for the optimisation of crew schedules.

Once all the modifications were made and the most-cost efficient algorithm generated a solution from real life data, it has been evaluated by the industrial experts at different levels. The evaluation procedure sought to find out about the implication of such a system for different business aspects ranging from day to day operations to overall strategic performance. In addition, the analysis of the software investment project was carried out to demonstrate the financial value of the system.

11.2 Overview of the adaptation process

The application of the developed algorithm on the real data sets requires execution of three adaptation stages which are shown on Figure 120. The first stage deals with the data preparation and conversion to the format accepted by the algorithm. At the second stage the fitness function of the algorithm is modified in order to accurately reflect the cost structure. The third stage begins after algorithm completion and is responsible for deduction of the diagrams from the chromosome and their presentation in a user friendly format. These steps are explained in detail in the following sections.

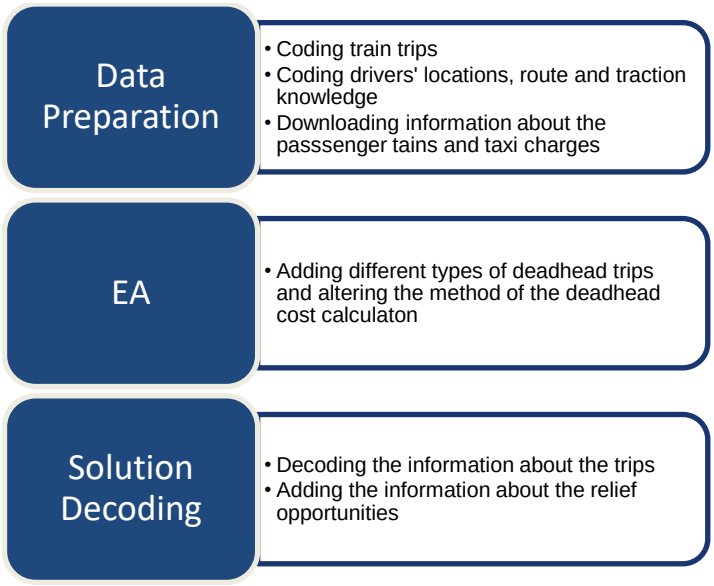


Figure 120 The process of testing EA on the real data set

11.3 Data preparation

11.3.1 Passenger trains and taxis

Interviews with the scheduler indicated that a driver can use various modes of transportation including vans (company cars), taxis and passenger trains. The fitness function usually requires this information when it attempts to connect two spatially separated trips. There are two ways in which it can access the given information: online (when the algorithm requires it) or offline (the information entered before the algorithm starts).

The advantage of the first method is that no additional procedures need to be performed before the beginning of the algorithm and only the trips included in the schedule will be downloaded. However, from a practical perspective this method can be extremely time consuming and its avoidance is usually recommended if possible (Google 2016). Conversely, the offline download of all deadheads will not impact the actual time of running the algorithm, but entails design of a specialised downloading procedure and storage of large sets of data.

Because the fitness function is calculated thousands of times during the course of optimisation, the preference was given to the offline method. Furthermore, in

order to avoid download of a full set of trains across the UK, only the trains which can connect the train trips in the schedule (those that depart and arrive within the time window between the trips and necessary stations) will be obtained. Moreover, in order to limit the number of inefficient deadheads, the maximum time window is restricted to five hours. The sources and methods of gathering this information are considered below.

Manual distance calculation. Given the latitude and longitude of both locations, it is possible to compute the distance between two points on the sphere. Although this method can provide results for a very short period of time, the accuracy of the journey duration between two locations might be quite poor since the roads are not straight and the driving speed varies on the different parts of the path. Imprecise information about deadheads can lead to a situation when a driver misses one of the trains causing disruptions to the entire schedule. In addition, this approach cannot provide data on the passenger train timetable.

National Rail. As a part of the information transparency initiative, Rail Network shares comprehensive data regarding passenger trains (Network Rail n/d). To date, each data feed encompasses each single train, its stops, platforms, and arrival and departure times. Despite plans to release in 2015 the Darwin system, which supports journey planner functions and can find the possible connections between the trains, the deployment has been postponed and the system is still not in full operation.

The currently available system might be used to obtain the timetable only of direct trains between two locations. However, an additional rather complex procedure must be designed to perform a network search to identify potential train connections. Moreover, its execution as a part of the fitness function would considerably increase the time of the algorithm as graph search techniques are usually NP-hard (I-Lin, Johnson and Sokol 2005, Pugliese and Guerriero 2013). On the other hand, the exclusion of journeys with changes would force the system to assign taxi trips instead of passenger trains.

Aside from that, there are two serious drawbacks of this system which reduces its suitability for the given algorithm. First of all, the route is provided only from one train station to another disregarding possible walking and car rides to and from the train station. The second shortcoming is that it does not consider the

possible combinations of transport modes, i.e. when one part of the route is performed by taxi and another by train.

Google maps. Google maps developer's service fulfils the problem requirements more closely: it contains the data about the rail and road transport, it is able to insert walking directions and is based on a fast and powerful mechanism to quickly discover the optimal route between two locations. Its response contains step-by-step journey instructions and realistic durations of each part.

Another benefit of using Google Maps is that it provides up-to-date information about infrastructure objects, traffic, and public transport service availability at a particular date and time. This is crucial as reliance on the incorrect data about deadhead trips might cause disruptions to a driver's diagram, which might, in turn, have a knock-on effect on the rest of schedule.

However, unlike other sources this is a commercial service and has an annual subscription cost (Google 2016). But it can be noticed that Google Maps' capabilities to provide more precise transfer information than other tools, and subsequently reduce the risk of rail freight train delays, justify the cost of the subscription. Therefore the Google Maps service will be used in this research.

11.3.2 Company trains

The import of real-life data into the devised system was one of the greatest challenges. Due to significant differences in data structures used in the company and the one embedded in the prototype, two different sets of data have been obtained from the company. They consist of freight train schedules and the diagrams covering that schedule. Information from the train timetable comprises the start and end locations as well as departure and arrival times. However, it does not specify all the stops of the trains and the required activities.

Another solicited data set provides already scheduled diagrams which display driving and ancillary activities, however it is based on missing, unstructured and inconsistent data (Figure 121).

DIAGRAM	DIAGRAMSTARTDATET	DIAGRAMENDDATETIM	ASACTIVIT	ARRIVALDATETIME	DEPARTUREDATETIM	LOCATION	HEADCOD
AC0154	05/13/2015 11:23:00	05/13/2015 22:13:00	WAR			Work as required in Acton Yard &	
AC0154	05/13/2015 11:23:00	05/13/2015 22:13:00	REL			Relieve / Place all Late Running	
AC0154	05/13/2015 11:23:00	05/13/2015 22:13:00	REL			WY0032 at 1131	
AC0154	05/13/2015 11:23:00	05/13/2015 22:13:00		05/13/2015 11:31:00		Acton Yard	7A09
AC0154	05/13/2015 11:23:00	05/13/2015 22:13:00	FS			Freight Shunt as Req'd	
AC0154	05/13/2015 11:23:00	05/13/2015 22:13:00	IMM			Immobilise Loco	
AC0154	05/13/2015 11:23:00	05/13/2015 22:13:00	MOB			Mobilise Loco	
AC0154	05/13/2015 11:23:00	05/13/2015 22:13:00	FS	05/13/2015 12:05:00	05/13/2015 12:35:00	Acton Yard	7C77
AC0154	05/13/2015 11:23:00	05/13/2015 22:13:00	RELD			by WY2932 at 12.33	
AC0154	05/13/2015 11:23:00	05/13/2015 22:13:00	PL			Prepare Loco	
AC0154	05/13/2015 11:23:00	05/13/2015 22:13:00			05/13/2015 13:40:00	Acton LHS	0L26
AC0154	05/13/2015 11:23:00	05/13/2015 22:13:00	ATT	05/13/2015 13:50:00	05/13/2015 14:10:00	Acton Yard	6L26
AC0154	05/13/2015 11:23:00	05/13/2015 22:13:00	RELD			by AC0019 at 1410	
AC0154	05/13/2015 11:23:00	05/13/2015 22:13:00	FB45			or 2 x FB30 as convenient	
AC0154	05/13/2015 11:23:00	05/13/2015 22:13:00	BOFF	05/13/2015 22:13:00		Book Off	
AC0159	05/13/2015 07:10:00	05/13/2015 17:45:00	WAR			Work as required in Acton Yard &	
AC0159	05/13/2015 07:10:00	05/13/2015 17:45:00	REL			Relieve / Place all Late Running	
AC0159	05/13/2015 07:10:00	05/13/2015 17:45:00	MOB	05/13/2015 08:35:00	05/13/2015 08:45:00	Acton Yard	6A62
AC0159	05/13/2015 07:10:00	05/13/2015 17:45:00	RELD			By AC0048 at 0845	
AC0159	05/13/2015 07:10:00	05/13/2015 17:45:00	FB30			Facility break for 30 minutes	
AC0159	05/13/2015 07:10:00	05/13/2015 17:45:00	MOB			Mobilise Loco	
AC0159	05/13/2015 07:10:00	05/13/2015 17:45:00			05/13/2015 13:00:00	Acton Yard	7A12
AC0159	05/13/2015 07:10:00	05/13/2015 17:45:00	FS	05/13/2015 13:05:00		Acton Foster Yeoman	
AC0159	05/13/2015 07:10:00	05/13/2015 17:45:00	DISC			Discharge To Be Continually Manned	8Q99
AC0159	05/13/2015 07:10:00	05/13/2015 17:45:00	RELD			by AC0157 as convenient	
AC0159	05/13/2015 07:10:00	05/13/2015 17:45:00	FB30			Facility break for 30 minutes	
AC0159	05/13/2015 07:10:00	05/13/2015 17:45:00	REL			AC0157	
AC0159	05/13/2015 07:10:00	05/13/2015 17:45:00	DISC			Discharge	
AC0159	05/13/2015 07:10:00	05/13/2015 17:45:00	FS		05/13/2015 17:25:00	Acton Foster Yeoman	7A12
AC0159	05/13/2015 07:10:00	05/13/2015 17:45:00		05/13/2015 17:30:00		Acton Yard	
AC0159	05/13/2015 07:10:00	05/13/2015 17:45:00	IMM			Immobilise Loco	
AC0159	05/13/2015 07:10:00	05/13/2015 17:45:00	BOFF	05/13/2015 17:45:00		Book Off	

Figure 121 Example of the data format

As can be seen, the time of the activities, for example in row one, is not stated. In addition, in the location column, the activity is placed together with the location. Given the size of data (more than 9000 entities), the manual correction and fulfillment of the data was impractical. To tackle this, the missing information was repaired either based on the subsequent or previous activities (when the finish time of the current activity is unknown) or based on the average data regarding the duration of certain activities. However, for some tasks there were insufficient data to restore the missing attributes. Loshin (2013) argues that filling in missing values can be counterproductive and dangerous as it can lead the analysis in a completely different direction. For this reason, several activities with missing attributes which could not be determined are combined into blocks if they are between the tasks where all the required characteristics are known.

In order to make sure that all the activities were extracted from the schedule, a comparison against the full train data has been made (Table 22).

Table 22 Real life train data set comparison

	Trip data	Blocks data
Number of pieces of work	431	905
Total duration, min	114026	93005
Average duration, min	264 min	102 min

The analysis indicates that 18% of the activities were left unaccounted for in the transformation process and further refinement and manual attempts to fix it were unsuccessful. For this reason, the adjustment of 18% will be made when comparing the results of the EA based solution against the manually produced schedule.

11.3.3 Drivers

The data about the number of available drivers in each depot has been collected as well. In total there are 1024 drivers who are located in 39 depots. At the moment, there are no electronic data linking the drivers' route knowledge with train trips. Together with the Head of Service it was established that the drivers are familiar with the area within a 200 mile radius of their depot location. Although this approach might produce only a sub-optimal solution because it will prohibit the diagrams where the drivers with knowledge of other regions can travel further, the permission to drive on all the routes can lead to an unpractical solution or prevent the algorithm from convergence as it would widely enlarge the search space.

11.3.4 EA modification

In the experimental version, only one type of deadhead has been used. This simplification was made because the type of a deadhead does not influence the configuration of the algorithm and its performance. However, for the real life data there are four possible modes of transportations and they are defined in Table 23.

Table 23 Types and specification of the deadhead transportation

Mode	Duration	Cost	Comments
WALK	Less than 15 minutes	0	When two locations are situated within 15 minutes from each other.
PASSENGER TRAIN	8mins-5 hours	0	The cost for the passenger is fixed and assumed to be zero in the fitness function calculation
TAXI	Up to 5 hours	£ 2 per minute	The taxi trips have been restricted to five hours based on operation and cost considerations
VANS	Up to 5 hours	£0.19 per minute	One van is located at each depot and can be used by drivers to transport themselves between different locations. Each van should be returned to the depot at the end of the working day.

In order to embed the given travelling opportunities into the fitness function and decoding procedure, the following elements have been added.

Deadhead selection. The search for the appropriate deadhead mode is carried out in the following order: walk, passenger train, and taxi. The search stops when the suitable transportation method is found. The vans are not considered at this stage as they depend on the entire diagram which might not be finalised at this stage. This relies on the fact the duration of the taxi trip is the same as the van trip and therefore will not affect the further construction of the diagrams.

***Taxi trips replacemen*t**. Once the entire diagram has been created, the procedure starts identifying the taxi trips, which can be substituted for a van. The principle of this procedure is to scan the diagram from the beginning to the end and notice the places where the taxi trip starts and ends. The trips where the driver travels to one destination and then returns from the same destination to the same place of origin is replaced. The taxi trip can be converted into van driving

if it will not exceed the maximum driving time allowance. The cost difference is then subtracted from the total cost.

Selection process. In addition to the reduction of cost, it has been mentioned that the company is interested in the minimization of the number of drivers as well. Because the revenue associated with the number of drivers is hard to model, it was decided not to include this information in the fitness function. However, in order to reflect this preference, some adjustments were made in the selection process. In the case when two chromosomes have the same cost, the selection gives higher preferences to the chromosome with the minimum number of diagrams.

11.3.5 Solution construction

Once the solution has been identified, the schedule builder assembles the diagrams from the chromosome and enhances them with supplementary details of the deadhead directions and non-time consuming activities (i.e. relieving the drivers, booking on and off). These activities are added at the end because they do not affect the schedule, but would consume a significant portion of computation time if they were a part of the algorithm.

11.3.6 Proof of concept design

In order to demonstrate the automatic crew scheduler logic, a proof of concept has been designed. The screenshots are illustrated on Figure 122-Figure 124.

The window in Figure 122 exhibits the coded data denoting all the trips and geographical locations, as well as the drivers and their knowledge. The panel on Figure 123 allows the selection of EA parameters. Finally, the window and charts in Figure 124 help to explain how the process of the evolution works. The charts present how the cost components develop under the evolution process. They show how the overall cost reduces as well as the changes caused to the actual cost of the schedule and the losses associated with unequal workload distribution and deviation from the target shift length.

The average diagram parameters are presented above the graphs and indicate the average throttle time per diagram, average deviation of the diagram from the

target shift length and the average number of deadhead transportations in one diagram.

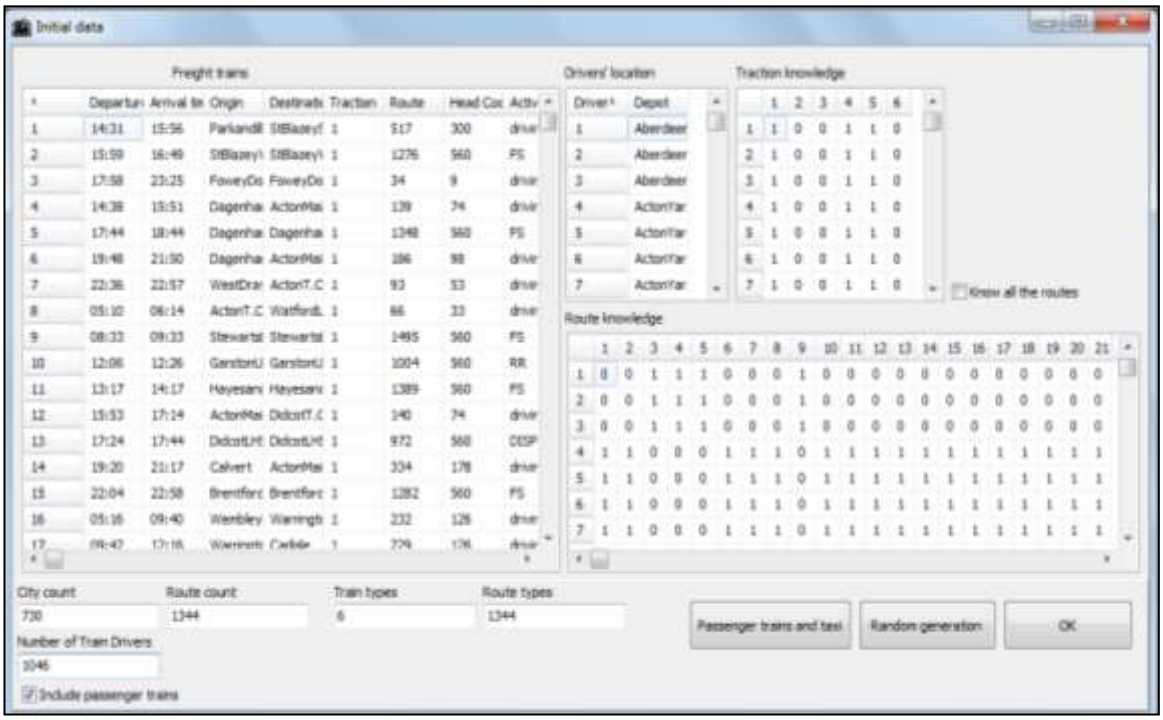


Figure 122 Demonstration of the input data

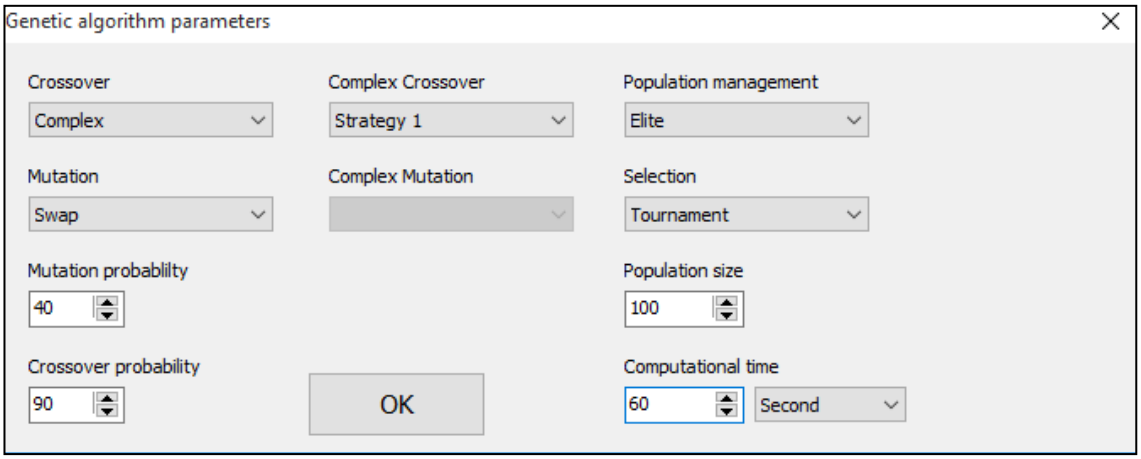


Figure 123 Screen shot of the prototype: Selecting EA parameters

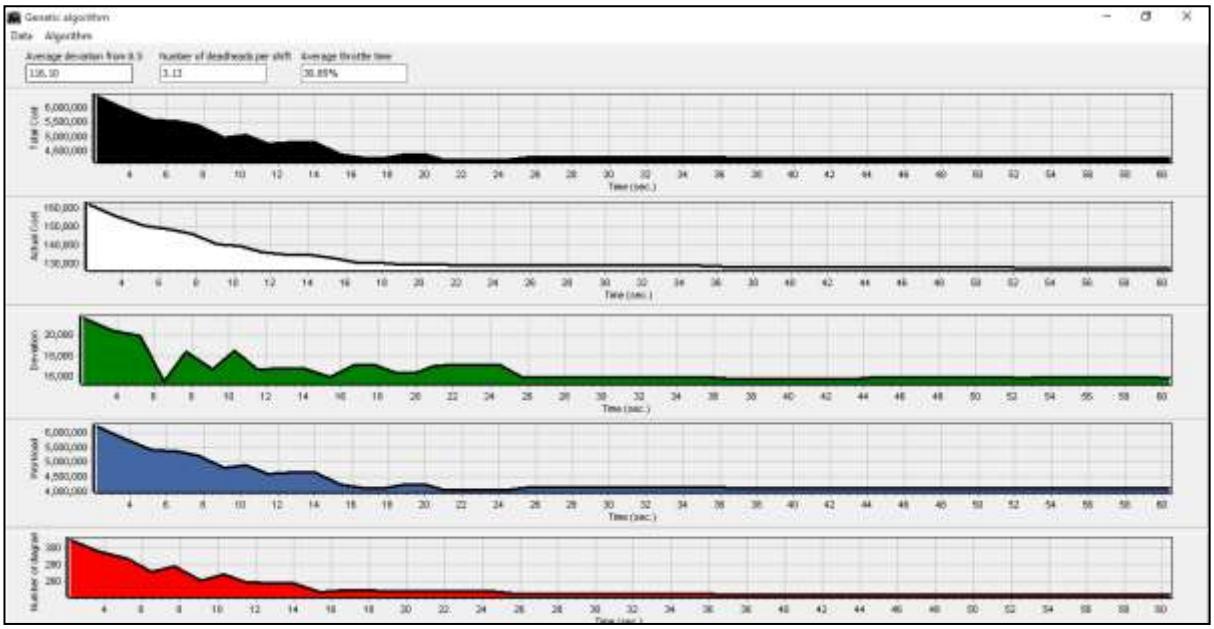


Figure 124 Screen shot of the prototype: evolution process demonstration

11.4 Optimisation of the real-data set with Nearest Driver EA

Once all the data have been transformed into the appropriate format, the algorithm with the parameters and operators described in section 10.3 is executed. The graphs in Appendix 10 exhibit one of the runs of the algorithm and show the improvements of the cost function and corresponding changes of its components.

Each run lasted 24 hours and the algorithm was repeated 10 times. Table 24 presents the average results. The allowance of 18% was made in order to compensate for the missing data. It is important to note that this represents the worst case scenario as if these activities were scheduled separately. It is anticipated that scheduling all activities together would lead to a better solution due to the consolidation of the short diagrams and the opportunity to travel as a passenger on the freight trains.

Table 24 Average cost of EA- produced schedule on the real data

	EA solution	EA solution with 18% adjustment
Driver Cost	£116,769	£137,787
Taxi Cost	£12,601	£14,869
Vans cost	£880	£1,038
Number of diagrams	348	410
Cost of deviation	£20,192	£23,827
Workload cost	£15,022	£17,726
Daily cost	£130,250	£153,695
Long-term cost	£35,214	£41,553
Total Cost	£165,464	£195,248

11.5 Cost-benefit analysis of the generalizable algorithm

Throughout this research several configurations of the algorithm have been considered. These configurations can be presented on the customisation spectrum, where standard EA lies on the one side of the spectrum and EA with specifically developed chromosome representation and tailored genetic operators on the other. Two EAs in the middle of that spectrum are the EA with multiple operators and the EA with intelligent operators, but standard chromosome representation.

11.5.1 Cost

The cost of producing various algorithm configurations is measured based on the development time. The time estimate was derived from the actual research time which was spent on performing each activity. The table below displays the time per each development stage and the total time required to produce each configuration of the algorithm.

While each algorithm required the same efforts for problem understanding and analysis, the solution design, development and testing times vary significantly across the configurations. Interestingly the difference between completing EA with standard and multiple operators is only one week. This is because the code production time for the multiple operators is compensated by the reduction in the test time for multiple operators (as the algorithm tests the operators and selects

the best automatically). Customised algorithms, on the other hand, required creation of the novel operators and more complex chromosome representation which significantly increased the amount of design and build efforts.

Table 25 Algorithm development efforts

	Analyse	Solution Design	Build	Test	Total
Standard EA	16 weeks	2 weeks	2 weeks	8 weeks	28 weeks
Multiple operators EA	16 weeks	4 weeks	6 weeks	3 weeks	29 weeks
EA with intelligent operators	16 weeks	7 weeks	5 weeks	5 weeks	33 weeks
Customised EA	16 weeks	9 weeks	7 weeks	9 weeks	41 weeks

The development cost of each algorithm is calculated based on the duration of each stage presented in Table 25 and the UK average labour cost. The average salary for the specialist with the required expertise was obtained from job search web-site, Glassdoor (2016). The fixed cost of programming software and equipment is omitted at this stage as it is identical for all configurations and will have no impact on the results' comparison.

Table 26 Algorithm development cost

	Analyse (Business Analyst)	Solution Design (Software Engineer)	Build (Software Developer)	Test (Software Tester)	Total
Annual Salary	£40,000	£61,074	£39,155	£25,712	
Standard EA	£12,308	£2,349	£1,506	£3,956	£20,118
Multiple operators EA	£12,308	£4,698	£4,518	£1,483	£23,007
EA with intelligent operators	£12,308	£8,222	£3,765	£2,472	£26,766
Customised EA	£12,308	£10,571	£5,271	£4,450	£32,599

11.5.2 Benefit

The schedule cost estimates have been produced using the results about the relative efficiency of each of the algorithm configuration presented in Chapter 9 and real life schedule cost delivered by the Nearest Driver configuration (section 11.4). The estimated schedule costs for the other configurations are displayed in Figure 125. The bar chart shows two values: the estimated average cost obtained

with the certain technique and the percentage difference with the lowest cost solution produced by the Nearest Driver configuration.

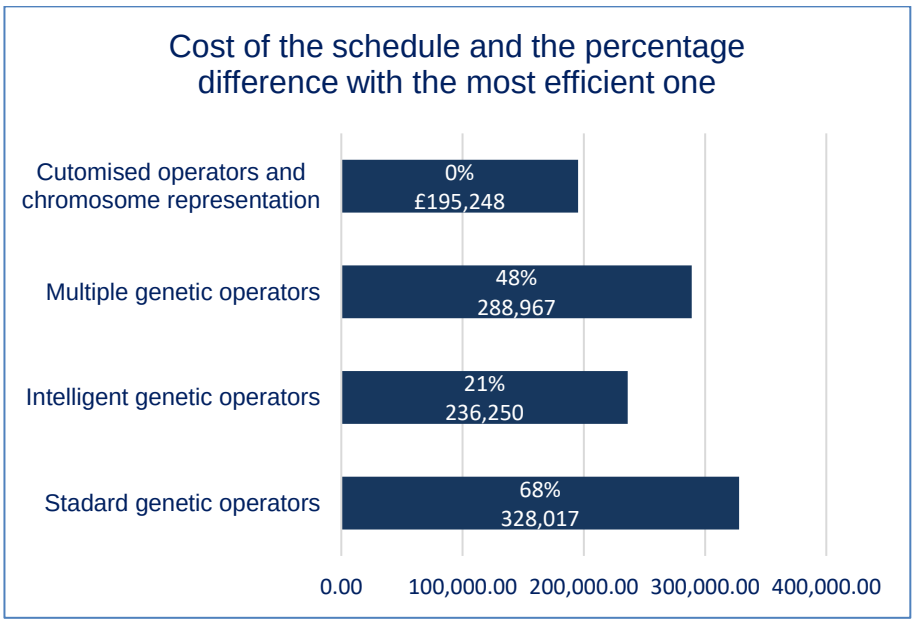


Figure 125 Cost of the schedule and the percentage difference with the most efficient one

The cost comparison of the above results with the actual solution (Figure 126) revealed that non-fully customised algorithms do not outperform the manual schedule. This is because their solutions utilise a larger number of drivers which has an adverse impact on additional revenue streams. This is discussed in detail in section 11.7.2.

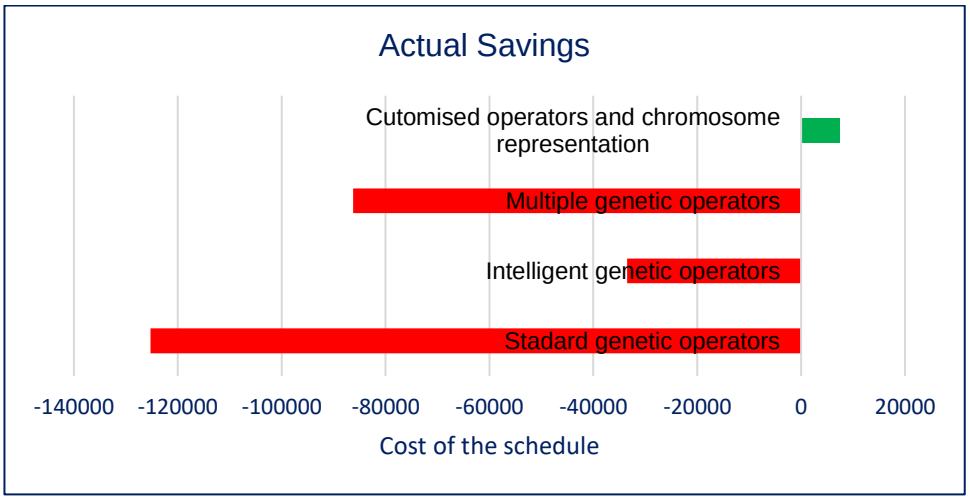


Figure 126 Cost difference between the manual schedules and EAs

In order to continue comparison and obtain a more in-depth understanding of cost-benefit ratios of various EA configurations, the benefits are evaluated based on the comparison with the obtained results rather than comparison with real life

schedule. Assuming that standard EA does not provide any benefits, the cost saving advantages of using other algorithms have been measured against the EA with standard genetic operators. The graph on Figure 127 shows the cost differences between the standard EA and its various configurations.

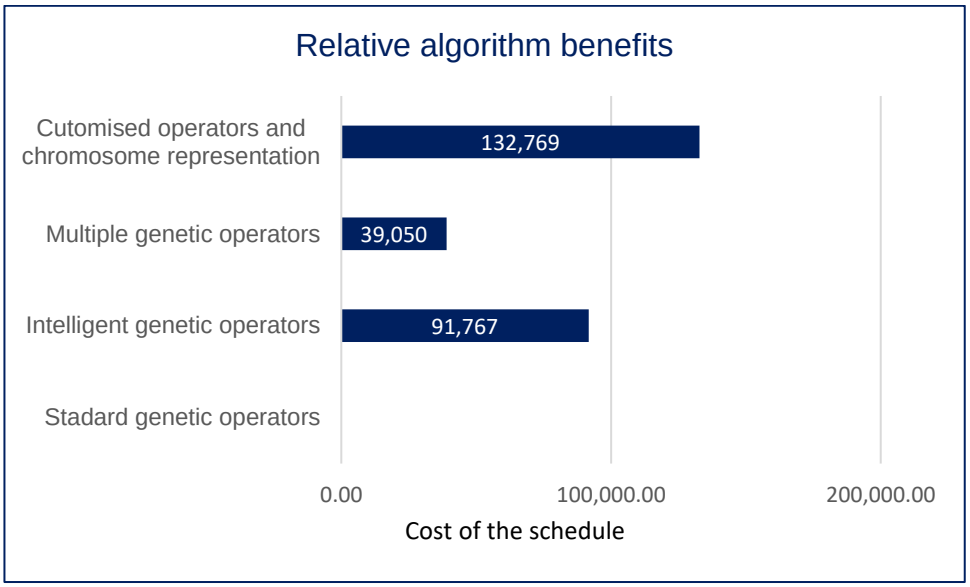


Figure 127 Relative benefits among the algorithm

The results indicate that while application of multiple operators provide certain benefits to the standard algorithm, the most economic schedule was produced by the algorithm with unique chromosome representation and tailored genetic operators. The next section will assess the practicality of the development of such an algorithm in the real life by adding the development cost to analysis.

11.5.3 Cost-benefit analysis

Since both the benefits and development cost depend upon algorithm configurations, the profitability index is calculated using the formula presented in section 3.5.5 in order to compare the cost-efficiency of the algorithms. The profitability index expresses how many pounds will be saved per one pound invested.

The results displayed in Figure 128 demonstrate that despite the development of the customised algorithm requires more time and financial resources, the derived benefits outweigh the benefits for the lower cost investment in the less customised algorithm.

This means that although EA with multiple operators appears to be more transferrable across different problems and the companies can achieve significant economies of scales by re-using the same algorithm for various optimisation problems, the investment in design of fully customised EA for each problem would be the most profitable solution.

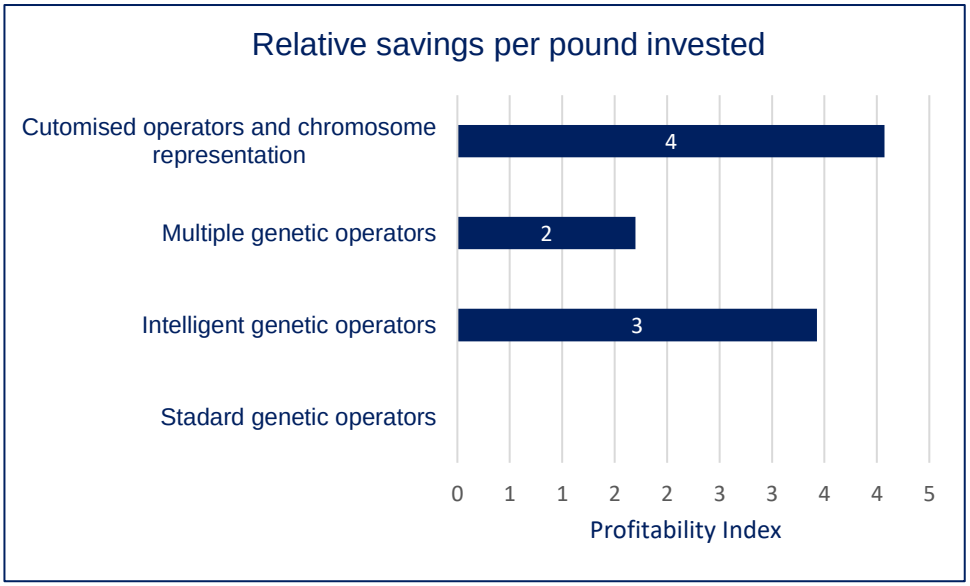


Figure 128 Relative savings per pound invested

11.6 Results Evaluation and Discussion

Since it was established in section 11.5.3, that the fully customised algorithm with Nearest Driver configuration is the most profitable amongst other EAs, further analysis will be solely focused on that technique.

Table 27 illustrates the detailed comparative results of the automatically and manually produced solutions. The manually constructed solution consists of 414 diagrams with the total duration of drivers' work of 3595 hours and 40 minutes which is equivalent to a cost of £143,827. The schedule also contained the chargeable deadhead trips by taxi and vans with the total estimated cost of £13,691. The analysis of the existing diagrams showed that the standard deviation in the workload distribution amongst the depots reached 3245 minutes, which might have a potential cost of £20,265 at the end of the year. Likewise, the average deviation from the shift length of eight hours and thirty minutes constituted 91 minutes, which might cost a company £24,948 if it is not stabilised on other days.

Table 27 Comparison of the cost of manually and automatically produced schedules

	EA solution with 18% adjustment	Manual	Difference
Driver Cost	£137,787	£143,827	£6,040
Taxi Cost	£14,869	£13,038	-£1,831
Vans cost	£1,038	£653	-£385
Number of diagrams	410	414	4
Cost of deviation	£23,827	£24,949	£1,122
Workload cost	£17,726	£20,265	£2,539
Daily cost	£153,695	£157,518	£3,823
Long-term cost	£41,553	£45,214	£3,661
Total Cost	£195,248	£202,732	£7,484

It can be seen from Table 27 that while the cost for the drivers has been significantly reduced, the cost of the taxis has increased substantially. This was caused by the amalgamation of the trips by the means of the deadhead transportation, which was necessary in order to include more trips into the diagrams and to reduce the total number of drivers. In most cases this was achieved by the insertion of a taxi trip because the train trip did not fit into the available interval of time or the connection was required at late or early times when passenger trains do not operate. Nevertheless, the overall crew day cost has been reduced by £3,823 on average.

Moreover, the automatically produced schedule distributed workload more equally amongst the depots as well as producing shifts with reduced deviation from the ideal shift length. Thus the savings from the balanced work allocation constitute £3,661, while the total savings a day reach £7,484.

In the produced schedule we could not find any solutions similar or identical to the manual diagrams. Therefore, the automatically constructed diagrams were presented to the crew scheduling experts in DB-Schenker, who were asked to give their opinion on each diagram and evaluate its overall quality and feasibility. The next section explains the evaluation procedure and reports the results from the assessment session.

11.6.1 Diagram sample evaluation

In order to verify real life feasibility of the diagrams and their conformance to all industrial regulations, a randomly extracted subset of the diagrams was shown to the scheduling experts. These diagrams are included in Appendix 11. The comments regarding each diagram are outlined below.

Diagram 1. The unnecessary mobilisation of train activity (11:47-12:03) was spotted in diagram one. The activity is not required because the driver relieves the previous driver meaning that it is very likely that the engine has already been started. Another comment regarding this diagram is that the assigned Walk activity (19.31-19:34) from Immingham Lindsey Refinery to Immingham Hit Coal Facility is impossible as it is an industrial area and there is no footpath. This implies that either a taxi trip or van should be inserted instead.

Diagram 2. This diagram received three comments. The first comment is that the driver is expected to walk to the Immingham TMD (traction maintenance depot) first to get the necessary equipment before performing the attachment of the wagons. The second comment is that the break (18:55-19:10) was superfluous and is not required by the health and safety regulations. The third observation was that there is no area to take a break at the Immingham SS (19:34-20:50).

Diagram 3. This diagram was approved by all schedulers because it not only satisfied all the regulations, but also effectively combined the trains from two different manual diagrams into one.

Diagram 4. According to the specifics of the operations, a FS (freight shunt) activity should be between PR (propel) and Driving (08:42-08:48). The train 6J94 could not be verified as it does not run any longer, but it was suggested that the driver cannot leave a train in Goole unattended, so the driver should have spent all day at that station. The assigned break in Immingham SS at 09:00-10:09 cannot take place due to the absence of the required facilities.

Diagram 5. This diagram was given a similar comment to Diagram2: the driver should have visited Mossend TMB before performing the attachment of the wagons (09:03-09:23). In addition, due to the large length of train 6G25, attaching the wagons takes more time as those wagons should have been inserted in the middle of the train, so the train should be decoupled first. This means that more

time should have been allocated for the activity. Another remark is that MOB (mobilise train) activity is usually performed after loco preparation (PL at 17:43-17:58) and before driving. It was also noted that the train trip effectively replaced a part of the taxi trips (10:48-11:53).

Additional Comments. During the discussion additional information came up, which highlighted the limitations of the current algorithm, and in particular the schedule builder functionality.

The sequence and duration of the activities depends on a large number of factors: terminal infrastructure, commodity types, types of wagons, and even sometimes client preferences and capacities. For instance, the freight shunt activity can be performed either by the driver or by the freight recipient company. This activity also varies in duration, which is determined by the level of automation at the client's site as well as the amount and type of loads. Likewise, attachment and detachment depends on the number of wagons and train configuration. Moreover, some activities can only be executed at specific stations. For example, reversing the train engine can be performed only at certain terminals with a special rail track infrastructure, and a driver can have a break at the stations or passenger trains with necessary facilities.

These limitations resulted from the built model, which did not incorporate the above-mentioned rules. However, as Rensburg (2011, p.1710) pointed out, the main objective of the model is to solve the problem and in order to do so, sometimes it is necessary to "eliminate those real-world details that do not influence the relevant goals of the problem". Another factor that contributed to these limitations is that the schedulers did not mention the rules at the interview process. According to Freeze and Schmidt (2015, p254), this represents tacit knowledge, which is "unspoken relations and patterns that help individuals store, organise and retrieve relevant information at appropriate times". The extraction and documentation of such knowledge usually present a great challenge, since people might struggle to explain or to recall all the details of the process.

11.6.2 Overall impression

Once all the comments were obtained, the schedulers were asked a number of questions regarding their opinion on the diagrams and the automatic system overall. The questions and the answers are discussed below.

Question: In your opinion, to what extent do the diagrams comply with all the regulations (i.e. maximum diagram duration, maximum working time, minimum breaks etc.)?

Answer: Diagrammers came to the conclusion that the duration of the breaks, driving times and the diagram duration satisfy the health and safety regulations, and in fact the system placed more breaks than required. However, it was pointed out that some of the locations do not have necessary facilities and need to be reconsidered.

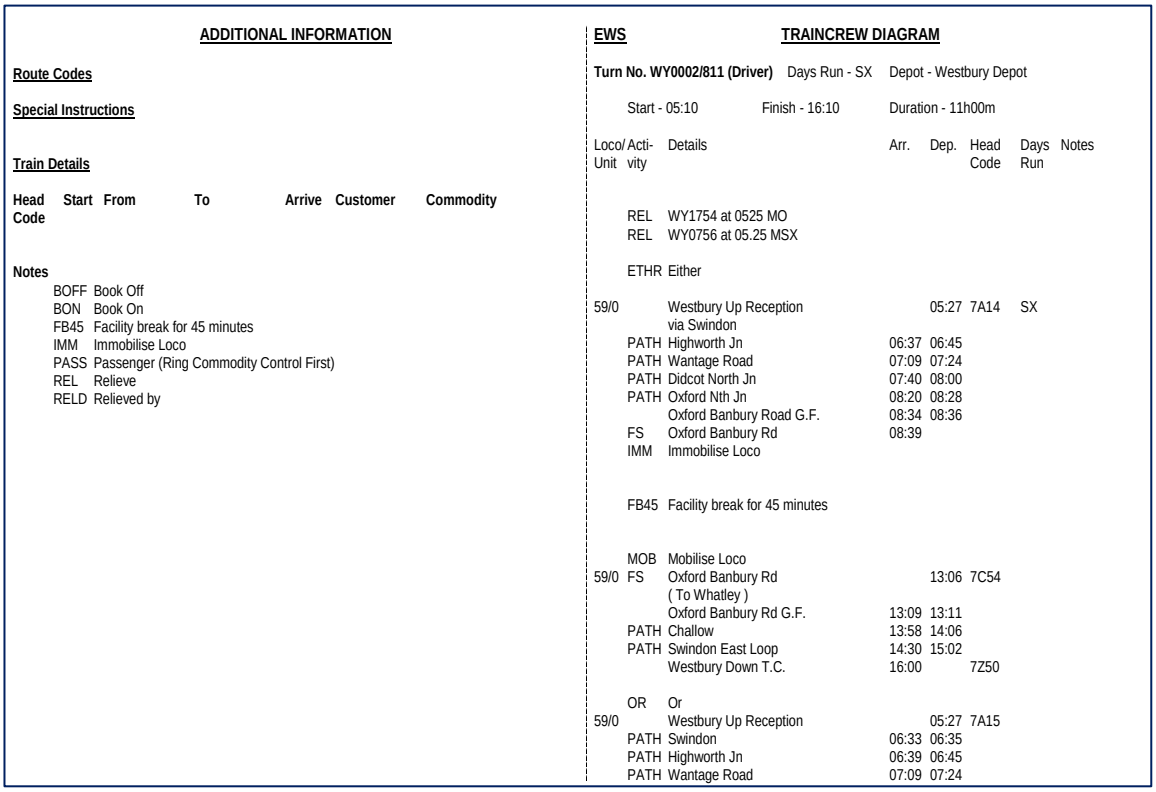
Discussion: Indeed, at the moment the system does not take into account the location when assigning a break. However, it is possible to resolve this in future research by collecting the information about the stations' and trains' facilities and integrating them in the schedule builder. Concerning the larger number of breaks, this happened because the driver could not be assigned to any tasks at the particular time, so the system reserved it for a break in case that there would be no time slot for a break later. This is relatively simple to fix by designing a scanning procedure, which would check the diagrams and remove unnecessary breaks once all the diagrams have been constructed.

Question: Does it contain all the necessary information for the drivers to perform the tasks?

Answer: Verification of all the abbreviation used in the diagrams showed that they are correct and recognisable by drivers. The location and time of the trains were also verified and approved. It was also noted that the different layout of the diagrams (Figure 129, Figure 130) might be a bit unusual for the drivers and would take time to get used to, but is not a major problem.

Head Code	Activity	Start	End	Origin	Destination
	book on	07:04	07:14	NewportDepot	NewportDepot
	walk	07:14	07:17	NewportDepot	NewportSig.NT1273
6V49	driving	07:17	07:41	NewportSig.NT1273	LeckwithLoopNorthJn
6V49	driving	07:41	08:21	LeckwithLoopNorthJn	MargamKnuckleYard
	taxi	08:21	08:48	MargamKnuckleYard	Port Talbot Parkway
Arriva Trains Wales	PASS	08:48	09:01	Port Talbot Parkway	Bridgend
	taxi	09:01	09:20	Bridgend	AberthawReceptionSdgs
	MOB	09:20	09:25	AberthawReceptionSdgs	AberthawReceptionSdgs
4F69	driving	09:25	10:26	AberthawReceptionSdgs	NewportA.D.JnSdgs
	OP	10:26	10:32	NewportA.D.JnSdgs	NewportA.D.JnSdgs
	DORR	10:32	12:35	NewportA.D.JnSdgs	NewportCourtybellaSdg
	OP	12:35	12:43	NewportCourtybellaSdg	NewportA.D.JnSdgs
	Relieved by driver num	39			
	walk	12:43	12:46	NewportA.D.JnSdgs	NewportDepot
	book off	12:46	12:56	NewportDepot	NewportDepot

Figure 129 Example of the diagram produced by the algorithm



ADDITIONAL INFORMATION

Route Codes

Special Instructions

Train Details

Head Code	Start	From	To	Arrive	Customer	Commodity

Notes

BOFF Book Off
BON Book On
FB45 Facility break for 45 minutes
IMM Immobilise Loco
PASS Passenger (Ring Commodity Control First)
REL Relieve
RELD Relieved by

EWS **TRAINCREW DIAGRAM**

Turn No. WY0002/811 (Driver) Days Run - SX Depot - Westbury Depot

Start - 05:10 Finish - 16:10 Duration - 11h00m

Loco/ Unit	Acti-vity	Details	Arr.	Dep.	Head Code	Days Run	Notes
	REL	WY1754 at 0525 MO					
	REL	WY0756 at 05.25 MSX					
	ETHR	Either					
59/0		Westbury Up Reception via Swindon		05:27	7A14	SX	
	PATH	Highworth Jn	06:37	06:45			
	PATH	Wantage Road	07:09	07:24			
	PATH	Didcot North Jn	07:40	08:00			
	PATH	Oxford Nth Jn	08:20	08:28			
		Oxford Banbury Road G.F.	08:34	08:36			
	FS	Oxford Banbury Rd	08:39				
	IMM	Immobilise Loco					
	FB45	Facility break for 45 minutes					
	MOB	Mobilise Loco					
59/0	FS	Oxford Banbury Rd (To Whatley)		13:06	7C54		
		Oxford Banbury Rd G.F.	13:09	13:11			
	PATH	Challow	13:58	14:06			
	PATH	Swindon East Loop	14:30	15:02			
		Westbury Down T.C.	16:00		7Z50		
	OR	Or					
59/0		Westbury Up Reception		05:27	7A15		
	PATH	Swindon	06:33	06:35			
	PATH	Highworth Jn	06:39	06:45			
	PATH	Wantage Road	07:09	07:24			

Figure 130 Example of the diagram produced manually

Some issues were spotted which would restrict the performance of some of the activities included in the diagrams. For instance, in some locations walking is impossible and highly dangerous (e.g. an oil refinery terminal). In addition, despite the short geographical distance, some locations require longer transportation times to the trains due to security checks and gates.

It was also recommended that it would be worth specifying at what time the driver should relieve another driver as it does not always happen on the arrival of the trains as the train can stay at the terminal a long time.

Comment: While the last comment can be tackled with the utilisation of the train timetable information, the procedures at different locations and their characteristics might be hard to obtain and would require a comprehensive analysis of the sites and stations.

Question: To what extent can the automatically generated diagrams help you in your daily work (i.e. as a starting solution)?

Answer: In terms of the practical use, the general reaction was “probably rather no, than yes”. They elaborated by saying that “it would be good to know it in theory how it should be, but in practice…I don’t think I would use it”. They also said that the schedule is currently constructed “bit by bit” meaning that every time a customer order changes, the trains are just added or removed to and from an existing schedule.

Although this approach can be less disruptive, over time, such modifications can result in a highly inefficient schedule. The schedulers agreed on that point and added that it could be possible to re-optimise it from time to time but not daily.

Comment. This response was slightly unexpected since the system was aimed at providing a better solution and it was intended to simplify the daily tasks of the users.

Markus (1983) analysed the factors which can lead to unacceptance of organisational changes, and in particular the resistance to information systems. She categorised them into three groups: people orientated, system orientated and integration orientated.

In terms of the people orientation, psychology research claims that such factors as age, gender, culture, background and technological experience have a significant impact on the perception of an IT system (Morris, Davis and Davis 2003, Freeze and Schmidt 2015, Laumer et al. 2016). However, Davis and Songer (2009) found that in architecture, engineering, and construction industries, where their research has been carried out, no correlation was found between age, education and technology resistance. Having no information about the schedulers’ background and controversy in the research conclusion, this concept cannot be used to explain the rejection of the system.

The second principle is the system oriented approach, which is applicable when a new IT system does not meet the expectations of the users or cannot correctly and timely perform its functions (Davis and Songer 2009, Misir et al. 2013). There are a number of limitations of the current prototype such as a basic user interface, the diagrams that were unfamiliar to the users, and the small location errors. These could significantly undermine the core value of the system and its usefulness. Although these problems can be tackled in future versions, the negative reaction can be explained by the short-term focus concept defined by Oreg et al. (2008). He states that short-term focus "involves the degree to which individuals are preoccupied with the short-term inconveniences versus the potential long-term benefits of the change"(Oreg et al. 2008, p936) .

Furthermore, the proposed system also breaks the standard "bit by bit" diagram construction pattern and replaces it with the verification of the already created diagrams. Oreg (2006) states that in general people tend to have a negative opinion about the novel systems and the things they were previously unaware of. Several researchers showed that users who have been working with the current system for a long time are less likely to recognise the business needs of updating it (Lapointe and Rivard 2005, Haerem and Rau 2007, Bhattacherjee and Hikmet 2007, Klaus et al. (2010), Bhattacherjee and Hikmet 2007).

Another approach which can explain the reaction of the experts is the integration orientation, which suggests that rejections and scepticism of a new IS are caused by potential consequences of its implementation such as alterations in job structures, reduction of autonomy and change of power. Potentially the negative response was caused by a subconscious fear that they might have less involvement in the familiar diagram construction process and change in the responsibilities to the inspection and quality checks of the created diagrams.

Not the least important aspect was the role the introduction of the research played. The title of the research and the utilised algorithm could create an impression that the automatic system is complex and can "replace their work". Oreg (2006) states that such predispositions tend to form negative perceptions of a new system. Furthermore, Baruch and Hind (1999) and Probst (2003) argue that employees tend to react negatively to an organisational change if it threats their job security.

Question: Could you please evaluate the standard of the diagrams in general?
Answer: The diagramers came to the conclusion that at the moment the diagrams are not feasible and cannot be safely operated. In addition to the above mentioned comments, they were concerned that the diagrams do not provide different route options, which are useful at the rostering stage. Since drivers have different sets of route knowledge, the rostering staff would be left with more options for driver assignment.

Comment: The relevance of the comment is twofold. On the one hand, the automatically produced schedule ensures that the driver with the corresponding route and traction knowledge is available. On the other hand, it does not deal with the rostering operations and does not possess the information about the previous working time of a particular driver.

11.6.3 IT perspective

According to the Application and Projects Manager (interviewed on 09/12/2015) the currently existing IT system is not up-to-date and has a limited scalability. However, it is undergoing revision and drastic improvements, after which the algorithm can be integrated as a back office process into the new platform. In such a case, the level of changes and required efforts are assessed as moderate and the duration of the implementation project is estimated at six weeks. Since the algorithm can function together with the existing system, the risk of the system failure is considered as relatively low (Application and Projects Manager interviewed on 09/12/2015).

The major difficulty is the transformation of the data into the new format. The problem of misspelt and missing data discussed in section 11.3.2 was caused by the manual data entry.

11.6.4 HR perspective

The potential impact of the new system on the staff will be considered from two perspectives: from the driver perspective and from the schedulers' perspectives.

Although the automatically generated diagrams group the work activities in a different manner compared to the existing diagrams, the Head of Service (interviewed on 09/12/2015) states that this is acceptable as it does not exceed

driver fatigue levels. Moreover, many research publications have confirmed that monotonous and repetitive jobs have a detrimental impact on performance (Jay et al. 2008, Jap, Lal and Fischer 2011, Othman, Gouw and Bhuiyan 2012). Since the new schedule needs to be created each time, it will have a positive influence on the drivers' performance as well as allowing them to drive on different routes. This would prevent them losing the knowledge of various routes and hence less training will be required.

From the perspective of the scheduling staff, the new system entails two possible alterations in their job functions. First of all, it involves the elimination of geographical barriers, when each scheduler was attached to a particular region and now they might need to work with different areas. Secondly, the nature of the job would shift from generation to verification of the diagrams. In the opinion of the Head of Service (interviewed on 09/12/2015), half of the schedulers would welcome the system, whereas the other half might be a bit sceptical.

11.7 Investment evaluation

This part analyses the impact of the system on an organisation's financial performance and calculates key investment indicators allowing assessment of the profitability and value of the new system acquisition. Typically, the decision to invest or not to invest is taken based on the evaluation of three factors: cost, benefits and risk (Asakiewicz 2011). These factors are calculated for the system investment project and are displayed in Table 28.

Table 28 Automatic Crew Scheduling System Investment Analysis

Cost	
Labour cost	
Business analyst	£24,615.38
Software engineer	£15,268.50
Software developer	£23,342.40
Software tester	£10,383.69
Software cost	
Programming software	£10,000.00
Google API 5-year subscription	£10,000.00
Miscellaneous cost	
Events Attendance	£2,500.00
Travel Cost	£1,360.00
Total Cost	£97,469.98
Benefits	
Cost of deviation	£204,765.00
Workload cost	£926,735.00
Taxi Cost	-£668,315.00
Vans cost	-£140,525.00
Opportunity Cost	£500,000.00
Total	£822,660.00
Investment analysis	
Interest rate	0.50%
Payback period	0.12
ROI (%)	41.20
Profitability Index (PI)	33.13
NPV year 1	£822,660.00
NPV year 2	£822,660.00
NPV year 3	£822,660.00
NPV year 4	£822,660.00
NPV year 5	£822,660.00
NPV	£3,229,653.19
3. Risk (standard deviation of the results)	5,7%

11.7.1 Cost

The labour cost estimate consists of two components: the cost of the initial prototype development (as discussed in section 11.5.1) and the further system enhancement as per schedulers' comments. Table 29 provides the details of both cost components.

Table 29 Labour cost for scheduling system development

	Annual Salary	Prototype development (weeks)	Prototype enhancements (weeks)	Prototype Cost	Enhancement cost	Total Cost
Business Analysis	£40,000	16	16	£12,800	£12,800	£25,600
Solution Design	£61,074	9	4	£10,993	£4,886	£15,879
Build	£39,155	7	24	£5,482	£18,794	£24,276
Test	£25,712	9	12	£4,628	£6,171	£10,799
Total		41	56	£33,903	£42,651	£76,554

In addition to the labour and software cost, there are several miscellaneous costs for travelling and attendance of related conferences and events, which allows for sharing ideas and best practices. However, there were several assumptions when calculating the cost. They are stated below:

1. The role of project manager will be performed by existing staff in the organisation and no additional cost will occur.
2. The system deployment will be conducted by the existing IT manager.
3. No consultancy fees are applied.
4. IT infrastructure is able to accommodate the system enhancement therefore no additional investments in infrastructure are required.
5. The test environment already exists in the organisation.
6. The user training cost is omitted as the final number of users and their competencies are not known.

11.7.2 Benefits

In general, the benefits from application of the software for the train driver scheduling derive from two types of costs: avoidance and opportunity. Avoidance cost is represented in reduction of the excess driver payments for the time exceeding the normal annual contractual hours together with the losses associated with the imbalanced workload distribution amongst the depots.

In addition, the comparison in Table 27 illustrates that the automatically produced software engages four drivers fewer than the manually produced software. There are two potential courses of action the company might want to take to respond to this situation. In the first scenario the organisation can make redundancies in order to save the cost. In this case, based on the annual contractual hours, the

total savings a year would be equal to £160,000 (Head of Service interviewed on 09/12/2015). However, this might lead to trade union actions and the payment of compensation packages. Furthermore, it would also increase the risk of having a shortage of drivers should the demand increase.

The second scenario suggests keeping all the drivers, but to start accepting more customer orders. According to the Head of Service (interviewed on 09/12/2015), there is sufficient additional demand for the freight transportation services and having four available drivers can drive the revenue by up to £500,000 a year.

Therefore, the total benefit from the implementation of the software can hypothetically rise to £822,660 a year. To compute the return on investment and profitability index, it will be assumed that the demand will remain constant and the interest rate will be taken as 0.5% (Bank of England 2016). Given that, it is expected that the initial investments should recoup within 44 days while return on investment will reach 41.2% in the first year and the profitability index is 33. Because the lifecycle for the product is estimated to be five years (the frequency of contract and regulations changes which need to be reflected in the algorithm), the NPV was calculated for the next five years reaching £ 3,229,653 by the end of year 5. Based on the given analysis, the investment project is regarded as attractive.

11.7.3 Risk

The last parameter which needs to be considered when making an investment decision is the risk which might affect the performance and hamper the operations.

As discussed in section 11.6.3, the risk associated with system failure is perceived as relatively low. This is because of two factors. First of all, the company has its own IT department which has expertise in the industry and in the current IT system. This enables the company to rectify any issues relatively quickly. The second factor is that the automatic scheduler will not completely replace the existing system and thus the automatically produced solution can be either partially or entirely overridden by the users.

The risk associated with the algorithm itself is its inability to deliver a high-quality solution. To estimate the level of solution fluctuation a standard deviation of the

final results of the ten runs has been computed and achieved 6%, which is quite reasonable for such an investment project.

11.8 Operational perspective

This section examines how the existing process of constructing a full driver schedule would be transformed if the system were implemented in the company. The radar chart in Figure 131 illustrates the potential changes in the process performance based on the five key performance indicators.

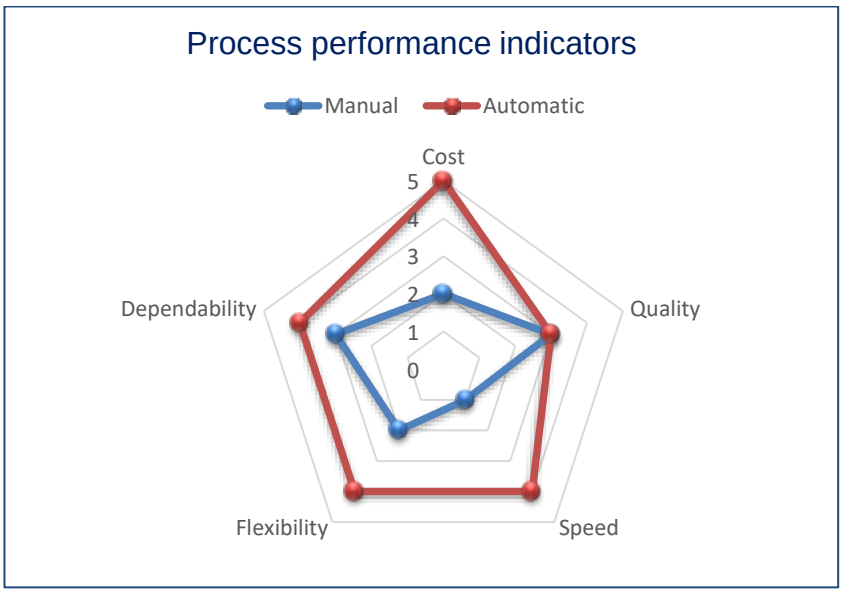


Figure 131 Key process performance indicators

Speed The developed EA is expected to significantly accelerate the current scheduling process. At the moment, diagram construction operations take three days and several departments participate in the process. With the automatic scheduler, it is possible to obtain a schedule within 24 hours, which also can be done during the weekend. So on Monday the schedulers would only need to revise the schedule and perhaps make minor amendments. In this case the automatic system would reduce not only the time needed to produce the schedule, but also the number of staff involved in that process.

Flexibility is expected to increase as well. The current practices are relatively inflexible in terms of insertion or removal of trips into and from existing diagrams, either of which is carried out without the consideration of the full schedule. Moreover, in most cases, this change entails the creation of new diagrams for the last-minute trains, which requires the involvement of additional drivers. The

designed algorithm allows effective incorporation of the new trips into the existing schedule by the timely re-running of the algorithm. The only limitation of the current version is that the algorithm does not have the capability of partial re-scheduling and might change a whole schedule rather than only the affected parts of it.

Cost From the cost perspective, the EA can produce a more cost-effective schedule and decrease the total cost of the schedule by £7,484 a day. The detailed analysis is presented in the Table 27.

Quality The impact of the EA on the quality of the scheduling process is twofold. On the one hand, the algorithm is able to produce a more balanced schedule in terms of the driver utilisation and work distribution. On the other hand, the evaluation reported in Section 11.6.1 revealed that it lacks the expert knowledge of certain locations and particulars of the operations. For this reason, the quality score remains unchanged.

Dependability Quality inspections of the diagrams generated with the EA did not identify any missing trains or incorrect departure or arrival times. The devised algorithm is more reliable than manual practices as it is able to deliver a schedule by the specified time, whereas the exact completion time of the manually created diagrams can vary. The only possible reason for the algorithm's failure is faulty equipment or a power outage, which are very rare events.

11.9 Strategy

No matter how modern the IT system is or what algorithms it is built from, there would be no value for it if it does not help to achieve, or worse, conflicts with the strategic objectives of the organisation. This section investigates the extent to which the proposed system would be aligned to the company's strategy. In order to do this, DB-Schenker's mission and strategic principles will be outlined first and then the role and possible effect of the Automatic Crew Scheduling System on the strategic performance will be analysed.

11.9.1 Description of DB-Schenker's Strategy

DB-Schenker's strategy rests on three aspects of their business: Social, Ecology and Economy (Figure 132). The social part stands for the creation of the

comfortable working environment, retention of current staff and being attractive for potential employees. Corporate culture and employee satisfaction form one of the main strategic priorities for DB-Schenker (DB-Schenker n/d). Furthermore, the company constantly calculates the job attractiveness indicator and strives to achieve higher and higher scores (DB-Schenker n/d) .

The Ecology side of the business focuses on the reduction of CO_2 emissions and aims to make the trains less disruptive for communities and less harmful for the environment.

The Economy area consists of two parts: Customer & Quality and Profitable Growth. The first part concerns the provision of exceptional service to customers while the second part ensures that the business remains profitable. With regard to the latter, the company puts “optimisation of the business processes”, “competitive cost structures”, “innovative products” and “market growth” in the centre of their DB2020 plan (DB-Schenker n/d). Besides, DB-Schenker aims at achieving high capacity utilisation and high productivity (DB-Schenker n/d).

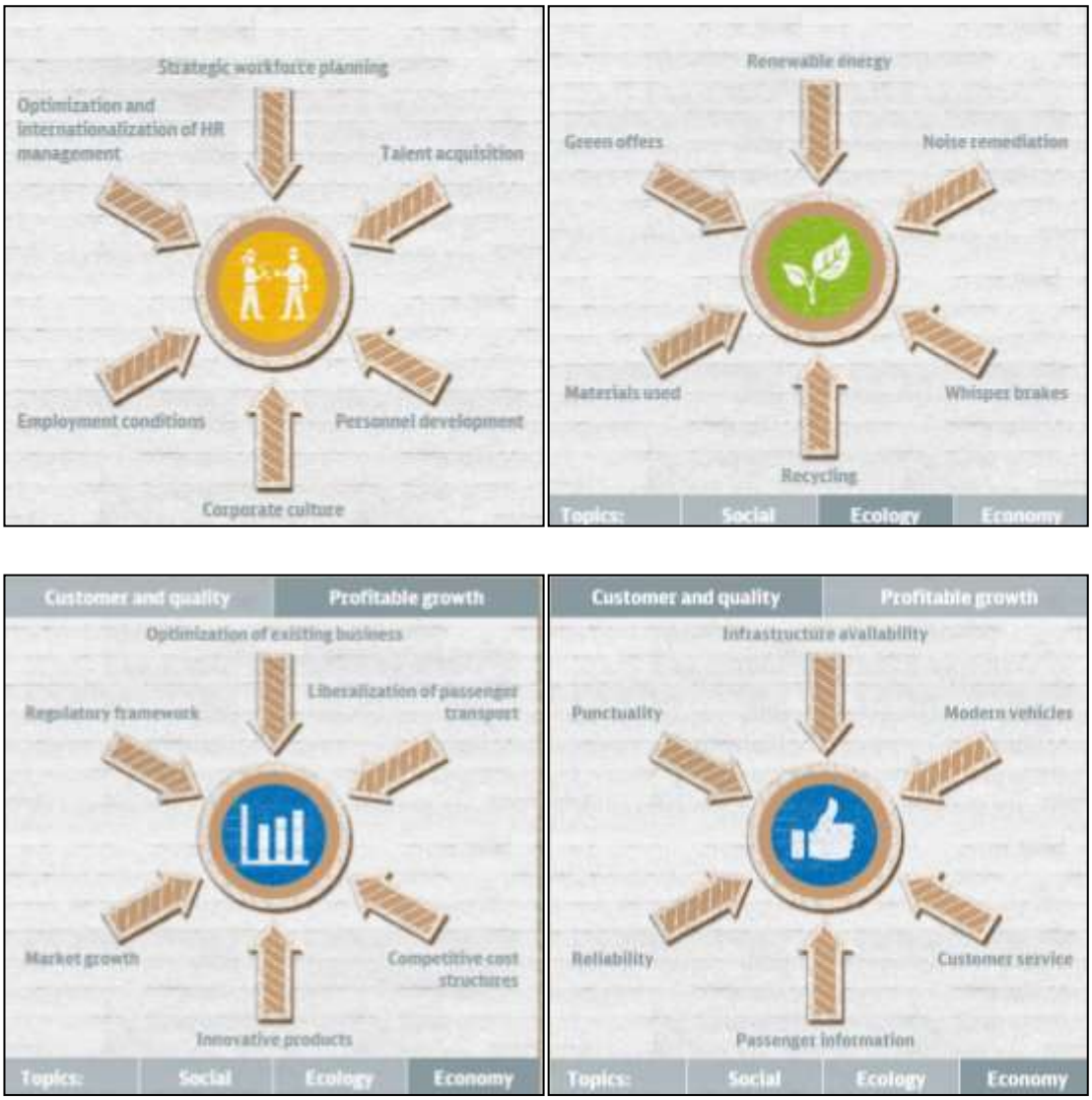


Figure 132 Aspects of DB-Schenker's Strategy

(Source: DB-Schenker n/d)

11.9.2 Alignment of the Automatic Scheduling System with the strategic goals

The developed automatic scheduling system contributes to the achievement of two out of three strategic priorities: Economy and Social. Table 30 demonstrates how the designed algorithm addresses a number of strategic objectives and contributes to their attainment.

Table 30 Automatic Scheduler alignment with the organisational strategy

Strategy part	What aspects the IS address	How IS helps achieve strategic objectives
Social	Employment condition	By implementing the automated system and providing a schedule faster, the schedulers would have more time to review, edit and make amendments if necessary. This should make the job less stressful as they will not be operating to extremely tight deadlines. Struebing (1996) states that realistic deadlines are one of the ways to reduce stress which could lead to poor performance and decreased productivity. In addition, after the implementation of the new system the staff can be involved in making more strategic decisions instead of performing routine tasks. Morris and Venkatesh (2010) observe that the significance of the task given to staff correlates with job satisfaction, while job satisfaction directly correlates with staff retention (Johnson and Yanson 2015). In addition, Limbu, Jayachandran and Babin (2014) showed that technology not only positively influences staff performance, but can also increase job satisfaction as well.
	Personnel development	Implementation of the new IT system implies the training of staff with the latest technological developments.
	Strategic workforce planning	Ability to quickly run the algorithm allows "What if" analysis to be conducted to identify and plan the number of required drivers on a particular day at a specific depot.
	Staff acquisition	All of the above might result in making the job more appealing to new staff as well as elevating the retention of current DB-Schenker employees.
Profitable Growth	Optimisation of existing business	The automatic crew scheduling system increases the effectiveness of the diagram construction and enables better utilisation of the drivers while satisfying all health and safety measures.

	Innovative growth	The system is built on one of the latest technological advancements. i.e. GAs, which are a rapidly growing area of Artificial Intelligence. Mithas and Rust (2016) showed that the organisations that have technology advancement at the core of their strategy have a higher revenue and lower cost compared to those that undervalue the role of technology.
	Market growth	The analysis showed that the better utilisation of drivers implies acceptance of more customer orders, which boosts the revenue and increases market share.

11.10 Conclusion

This chapter has presented the adaptation of the devised EA to real life settings and discussed the results of a comprehensive evaluation of the system's applicability. The profitability analysis of the spectrum of the algorithms proved that the fully customised algorithm delivers significantly better results than re-usable and transferable algorithms even though the development cost is larger.

Despite the issues with the format incompatibility of the data and consideration of the worst case scenario, the developed customised EA has obtained promising results and managed to outperform the manually generated schedule. The success of the automatically produced solution stems mainly from balancing workload distribution across various regions and reduction in the driver cost. This result was achieved by having a centralized view of the problem and analysis of a vast number of options which a human mind cannot process. In addition, the human schedulers perhaps try to avoid having the long taxi trips because this increases the cost of the diagrams they are responsible for. However, as has been shown, connecting taxi trips might be advantageous for the schedule overall, but the decision of which taxi trip should be included cannot be made without having a picture of the entire schedule.

The appraisal of the diagrams by the expert schedulers highlighted some of the limitations of the current model, which caused a disbelief in the system applicability in their everyday work. Conversely, the managers demonstrated an interest and seemed enthusiastic about the new system. This phenomenon has

also been observed by Strebel (1996), who stated “*Managers at the top of the organizational hierarchy see change as an opportunity to improve the company and advance their careers. For other employees, the change is unwelcome*”.

The investment analysis indicated that the proposed algorithm can bring substantial savings and revenue opportunities to the company and that the investment is worthwhile. Further appraisal of such a system revealed that if the automatic scheduling system were implemented in the company it would increase operational effectiveness and contribute to the achievement of the company's long term strategic objectives.

Chapter 12. Conclusions and future research directions

12.1 Introduction

The design of an optimal and cost-efficient schedule for such real life scheduling problems as crew scheduling and job-shop scheduling is a very challenging task. This is because they not only consist of a large number of jobs, but also because they are very constrained by industrial regulations and contain a large number of technical rules underpinning these operations.

As demonstrated in Chapter 5 and Chapter 7, the exact integer programming methods are not always practical as they require generation of all the schedules, which is a very time consuming process for the real-life scheduling models. On the contrary, the metaheuristic methods described in Chapter 2 can effectively handle a large set of data and provide a reasonable solution within an acceptable time frame.

An evolutionary algorithm has been selected for this research rather than other metaheuristic algorithms for its ability to work with population of the solutions, exploitation and exploration capabilities and capacity to retain a good solution in the population. Other metaheuristic methods such as Simulated Annealing and Tabu Search are single solution based techniques, which work only with one solution and are not able to exchange good properties between two or more solutions to create a superior one. Ant colony optimisation is a population based method, but it is more suitable for the problems which can be presented in the form of a graph (e.g. travelling salesman problem).

Going further, Chapter 9 demonstrated that as developed in Chapter 8, intelligent operators for an EA are more efficient than standard genetic operators. This is because the devised operators take into account the domain-specific information when exchanging and permuting genes between parents.

The conceptual comparison of CSP and JSSP in sections 8.3 and 9.5 proved that while it is possible to design an EA applicable to both problems, the efficiency of such an algorithm appeared to be relatively low. Furthermore, the experimental

results demonstrated that the schedule produced by a generalizable algorithm is on average 48% more expensive than the schedule produced by a customised EA. This is because a permutation based chromosome representation which was suitable for both problems could not accommodate the variation of the assignment of operations in JSSP. The rigid chromosome representation precluded fast construction of the schedule for CSP as it was unable to manipulate the position of the drivers in the chromosome.

Further analysis in Chapter 10 indicated that although development of the customised algorithm required more financial resources, the profitability index was 50% higher for the customised algorithm than the generalizable one. Given the comparatively small cost for the algorithm implementation and significantly larger schedule cost savings, it is recommended that enterprises should give a preference to a customised algorithm rather than an off-the-shelf solution.

In order to assess the impact of the algorithm on the performance of the real-life organisation, the customised configuration of the algorithm has been applied to group the real train trips into diagrams and to produce a schedule compared against the manual one. The empirical investigation conducted in Chapter 11 identified various benefits from using the automated schedule builder, which range from the direct cost savings to staff satisfaction and process improvements. The potential benefit from the algorithm, which has been developed in this research, is summarised in the table below.

Table 31 The summary of the business benefits provided by developed EA

Type	Description
Financial benefits	1. The average cost saving is 3.7% 2. 4 driver FTEs saving can result in the opportunity cost of £500,000 3. In five years the total financial benefits can save more than £3 million
Operational benefits	1. The speed of schedule construction is reduced from 3 days to 2 days. The algorithm can also run outside of working hours to provide even greater time saving 2. Intelligent incorporation of last minute orders enhances the scheduling process flexibility
Strategic benefits	1. Better workforce planning and utilisation 2. Staff satisfaction deriving from stress reduction, performance of more strategic roles and personal growth 3. Knowledge retention 4. Business optimisation and market growth

12.2 The limitations of the algorithm experiments

The following limitations of the algorithm experiments are proposed:

Optimisation of EA parameters**.** The parameters for the algorithms relied on results from separate experiments and might not be optimal for the given problems. In future the parameters might be retested in order to verify their suitability for the given problems. Alternatively, this can be done through incorporation of a fuzzy-logic controller to ensure maximum efficiency, higher adaptability to different data sets and problem structures as suggested by Sumer and Turker (2013), Herrera and Lozano (2003), Yu-Chiun Chiou and Lan (2002), McClintock, Lunney and Hashim (1997).

Driver Evolution**.** The driver evolution experiments were conducted along with powerful heuristic operators, which possibly had a greater impact on directing the optimisation process and made the effect of driver evolution less detectable. In future, it might be interesting to repeat the experiments with conventional operators, such as PMX and Swap or to conduct a factorial analysis to determine the impact of each operator.

Machine Evolution. Because of the substantial conceptual differences in the driver and machine assignment identified in Section 8.3, it was impossible to test the same logic of driver permutation on the JSSP.

Test Problems**.** The research has considered the performance of an EA for two problems: job-shop scheduling and crew scheduling. However, this might not be sufficient in order to draw a general conclusion regarding the effectiveness of the operators. It would be interesting to conduct the experiment with other combinatorial problems in order to investigate whether the results can be repeated across various domains.

12.3 The limitation of business evaluation

This section presents the limitations of the practical algorithm development and evaluation, which were caused by data quality, sample size and research scope.

Evaluation framework. Evaluation of the benefits did not consider soft indicators such as organisational culture, user acceptance, and convenience of using the system. Moreover, due to time constraints and the scope of the research, the full

system has not been implemented and all the results are based on the predicted numbers rather than actual ones.

Data quality. The comparison conducted in section 11.6, was based on the "repaired" data as some of the actual data contained missing attributes. The accuracy of the analysis could be improved by carrying out the evaluation on cleaner data to confirm the results.

Research sample**.** First of all, the evaluation of the system and its usefulness was conducted by a small group of potential users who had worked in the company for a long period of time. Therefore, the results might vary if the staff who had recently joined the company participated in the focus group. Furthermore, the research was conducted in a single organisation, and it would be interesting to get an opinion of experts from another company.

12.4 Future research direction

There are several aspects of this research which can be enhanced in the future. They are presented below.

Operator performance analysis**.** The analysis of operator effectiveness has been conducted based on the empirical results of their average performance. Factorial analysis can be performed in the future in order to separate contributions of crossover and mutation operators and better understand their pure impact. Moreover, Markov chains analysis, similar to the one proposed by Ma, et al. (2011), can be employed to analyse the operators' effectiveness in greater depth and predict the state of the system at the next iteration.

Multiple operator performance. The strategies in this research did not succeed in outperforming the results of an EA with single intelligent crossover and mutation. Additional research could be conducted in order to identify the set and number of operators that should be included in the strategy. The design of more sophisticated procedures for operator selection can also be considered. For instance, an additional high-level EA that regulates low-level genetic operators could be developed as it might enable the selection of an optimal combination of operators for the strategies.

Rules extraction**.** The rules of driver scheduling used in this study were gathered from the numerous documents and interviews with the schedulers. However, in order to develop a fully automated scheduling system, a significant number of additional rules, which are not documented, should be obtained. This can be accomplished in at least two ways. The first way is to perform further interviews with the schedulers and possibly to conduct job shadowing. The second way is to devise a data mining system similar to one proposed by Metan, Sabuncuoglu and Pierreval (2010) which would enable automated rule extraction from existing diagrams.

Decoding procedure. In order to accurately estimate the duration of each activity and to build fully feasible diagrams, the decoding procedure can be enhanced with more rules and information as such traction type, rail terminal infrastructure, commodity types and regulations for their transportation, and clients' orders and preferences. To achieve this, the trips input data set needs to be modified to include commodity type, number of wagons, and tonnage.

Vans optimisation. In section 11.3.4 it was assumed that the vans could connect only the trips starting or ending in the same depot. However, in reality the vans can be used anywhere as long as they are finally returned to the depot. The additional optimisation procedure could be incorporated into the fitness function in order to find the effective schedule of the vans' utilisation.

Strategic perspective. It would be ideal to design a hybrid system which is orientated not only to the daily crew scheduling operations, but also takes into account the long term strategic objectives and performs other operations such as scheduling trains and creating a driver roster. In particular, it would be useful to incorporate three functions:

1. Based on the availability of the crew and information about the passenger trains, select the appropriate time for a freight train departure and route. This is because sometimes moving the departure time of the freight train forward can help to fit in the passenger train trip and therefore reduce the taxi charges significantly.
2. Increase the planning horizon to at least seven days. This would enable construction of an effective roster and more equal workload distribution.

3. Incorporate strategic objectives (such as the revenue and cost of running a train) into the fitness function.

Real impact of automation. The automation of the crew scheduling processes is significantly under-researched. It would be interesting to investigate the user acceptance and the impact on the organisation after the system has been deployed and run for several years.

Wider applicability**.** Since the JSSP was an ancillary problem in this research, which served only for the evaluation of effectiveness of the operators, it has not been tested in real life settings. In future research, the prototype representing the functionality of automated job scheduler for the printing industry can also be developed and evaluated in a similar manner to the automated crew scheduling system.

12.5 Final Remarks

The thesis has achieved the key objectives and has answered the research question presented in Chapter 1. The research has designed two EA algorithms for the solution of JSSP and CSP, and compared their domain and cross-domain effectiveness. The successful configuration of the algorithm has also been applied to produce a real train-driver schedule for a large rail freight carrier in the UK. This has allowed the author to hypothetically examine the effect that this algorithm would have on organisational performance if it were integrated into the existing IT system.

It was estimated that the algorithm would significantly reduce operational cost and increase the speed of constructing the schedule. Furthermore, the financial analysis showed the profitability of investing in the design of the automated crew scheduling system. However, while the managers demonstrated enthusiasm about the new system, the potential users express some degree of concern. This is an important finding as well, because awareness and recognition of user opinions at the early stages of a system development enable not only the incorporation of the correct functionality, but also reduce any resistance with an appropriate change management strategy before the system is live.

Given these findings, the author believes that by leveraging the capabilities of EAs and successfully addressing corresponding implementation issues, distribution and transport organisations can successfully transform their current scheduling systems, enhance operational effectiveness and achieve their core strategic objectives.

Bibliography

ABBINK,E.J.W.; ALBINO,L.; DOLLEVOET,T.; HUISMAN,D.; ROUSSADO,J.; SALDANHA,R.L. (2011). Solving large scale crew scheduling problems in practice. *Public transport,* 3 (2), 149-164.

ABBINK,E.J.W.; FISCHETTI, M.; KROON,L.; TIMMER,G.; VROMANS,M.; (2005). Reinventing Crew Scheduling at Netherlands Railways. *Interfaces,* 35 (5), 393-401.

ABDOUN, O.; ABOUCHABAKA, J. (2012). A Comparative Study of Adaptive Crossover Operators for Genetic Algorithms to Resolve the Traveling Salesman Problem, *International Journal of Computer Applications,* 31(11), 49-57.

ABDULLAH, S.; ABDOLRAZZAGH-NEZHAD, M. (2014). Fuzzy job-shop scheduling problems: A review. *Information sciences,* 278, 380-407.

ADAMS, J. E. B.; ZAWACK, D. (1988). The Shifting Bottleneck Procedure for Job Shop Scheduling. *Management science,* 34 (3), 391-401.

AICKELIN, U. (2002). An Indirect Genetic Algorithm for Set Covering Problems. *The journal of the operational research society,* 53 (10), 1118-1126.

AIEX, R. M.; BINATO, S.; RESENDE, M. G. C. (2003). Parallel GRASP with path-relinking for job shop scheduling. *Parallel computing,* 29 (4), 393-430.

ALABAS-USLU, C.; DENGIZ, B. (2014). A Self-Adaptive Heuristic Algorithm for Combinatorial Optimization Problems. *International journal of computational intelligence systems,* 7 (5), 827-852.

AMIRTHAGADESWARAN, K. S.; ARUNACHALAM, V. P. (2006). Improved solutions for job shop scheduling problems through genetic algorithm with a different method of schedule deduction. *The international journal of advanced manufacturing technology,* 28 (5-6), 532-540.

AMIRTHAGADESWARAN, K. S.; ARUNACHALAM, V. P. (2007). Enhancement of performance of Genetic Algorithm for job shop scheduling problems through inversion operator. The international journal of advanced manufacturing technology, 32 (7), 780-786.

APPOINTY (2016). *Online Scheduler Software.* [online]. http://www.appointy.com/.

ASAKIEWICZ, C. (2011). Business Investments in IT: Managing Integration Risks. *IT professional,* 13 (4), 41-45.

ATRILL, P. (2014). *Financial management for decision makers.* 7th. ed.; Harlow, Pearson.

AVISON, D. E. (2006). *Information systems development: methodologies, techniques and tools.* 4th ed.; Maidenhead, McGraw-Hill Education.

AZADEH, A.; FARAHANI-HOSSEINABADI,M.; EIVAZY, H.; NAZARI-SHIRKOUHI, S.; ASADIPOUR, G. (2013). A hybrid metaheuristic algorithm for optimization of crew scheduling. *Applied soft computing,* 13 (1), 158-164.

BALTZAN, P. (2009). *Business driven information systems.* 2nd ed.; International student ed. London, McGraw-Hill Irwin.

BALTZAN, P. (2015). *Business driven information systems.5th* ed.; New York, McGraw-Hill.

BANK OF ENGLAND (2016). *BOE home.* [online]. Last updated 15 March 2016. http://www.bankofengland.co.uk/Pages/home.aspx.

BARNHART, C.; HATAY, L.; JOHNSON, E. L. (1995). Deadhead Selection for the Long-Haul Crew Pairing Problem. *Operations research,* 43 (3), 491-499.

BARNHART,C.; JOHNSON,E.L.; NEMHAUSER,G.L.; SAVELSBERGH, M.W.P.; VANCE,P.H. (1998). Branch-and-Price: Column Generation for Solving Huge Integer Programs. *Operations research,* 46 (3), 316-329.

BARUCH, Y.; HIND, P. (1999). Perpetual Motion in Organizations: Effective Management and the Impact of the New Psychological Contracts on " Survivor Syndrome". *European journal of work and organizational psychology,* 8 (2), 295-306.

BAUR, C.; WEE, D. (2015). *Manufacturing's next act.* McKinsey&Company.[online].http://www.mckinsey.com/business-functions/operations/our-insights/manufacturings-next-act.

BBC (2015). *DB-Schenker to cut 234 rail freight jobs.* [online]. Last updated 22 June 2015 http://www.bbc.co.uk/news/uk-england-33222140.

BEASLEY, J. E.; CAO, B. (1996). A tree search algorithm for the crew scheduling problem. *European journal of operational research,* 94 (3), 517-526.

BEASLEY, J. E.; CHU, P. C. (1996). A genetic algorithm for the set covering problem. *European journal of operational research,* 94 (2), 392-404.

BECKER, J.; KUGELER, M.; ROSEMANN, M. (2011). *Process management : a guide for the design of business processes.* 2nd ed.; London: Springer.

BHATTACHERJEE, A.; HIKMET, N. (2007). Physicians' resistance toward healthcare information technology: a theoretical model and empirical test. *European journal of information systems,* 16 (6), 725-737.

BIERWIRTH, C. (1995). A generalized permutation approach to job shop scheduling with genetic algorithms. *Operations-research-spektrum,* 17 (2-3), 87-92.

BLUM,C.; PUCHINGER,J.; RAIDL,G.; ROLI,A. (2011). Hybrid metaheuristics in combinatorial optimization: A survey. *Applied soft computing journal,* 11 (6), 4135-4151.

BLUM, C. and ROLI, A. (2003). Metaheuristics in combinatorial optimization: Overview and conceptual comparison. *ACM computing surveys,* 35 (3), 268-308.

BO, Z. W.; HUA, L. Z. and YU, Z. G. (2006). Optimization of process route by Genetic Algorithms. *Robotics and computer-integrated manufacturing,* 22 (2), 180-188.

BOCIJ, P. (2015). *Business information systems: technology, development and management for the e-business.*5th ed.; Harlow, Pearson.

BRANDON, J. (2016). *The epic battle between Google Home and Amazon Echo has just begun.* Framingham, Computerworld.com.

BRUCKER, P.; JURISCH, B.; SIEVERS, B. (1994). A branch and bound algorithm for the job-shop scheduling problem. *Discrete applied mathematics, special volume viewpoints on optimization,* 49 (1), 107-127.

BURKE,E.; GENDREAU,M.; HYDE,M.; KENDALL,G.; OCHOA,G.; OZCAN, E.; QU,R. (2013). Hyper-heuristics: a survey of the state of the art. *The journal of the operational research society,* 64 (12), 1695-1724.

BUSINESS WIRE (2015) *Meet Amelia: IPsoft's New Artificial Intelligence Platform Interacts Like a Human.* New York, Business Wire.

CACCHIANI, V.; CAPRARA, A.; TOTH, P. (2010). Scheduling extra freight trains on railway networks. *Transportation research part B: Methodological,* 44 (2), 215-231.

CAPRARA,A.; FISCHETTI,M.; TOTH,P.; VIGO,D.; GUIDA,P.L (1997). Algorithms for railway crew management. *Mathematical programming,* 79 (1-3), 125-141.

CAPRARA,A.; KROON,L.; MONACI,M.; PEETERS,M.; TOTH,P. (2007). Chapter 3 Passenger Railway Optimization. In: *Handbooks in Operations Research and Management Science.* Elsevier, Volume 14, 129-187.

CARRANO,E.G.; SOARES,L.A.E.; TAKAHASHI,R.H.C.; SALDANHA,R.R.; NETO,O.M. (2006). Electric distribution network multiobjective design using a problem- specific genetic alg orithm. *Power delivery, IEEE transactions on,* 21 (2), 995-1005.

CHAN, K. C.; TANSRI, H. (1994). A study of genetic crossover operations on the facilities layout problem. *Computers & industrial engineering,* 26 (3), 537-550.

CHAUDHRY, I. A. (2012). Job shop scheduling problem with alternative machines using genetic algorithms. *Journal of central south university,* 19 (5), 1322-1333.

CHENG, R.; GEN, M.; TSUJIMURA, Y. (1996). A tutorial survey of job-shop scheduling problems using genetic algorithm-representation. *Computers & industrial engineering,* 30 (4), 983-997.

CHENG, R.; GEN, M.; TSUJIMURA, Y. (1999). A tutorial survey of job-shop scheduling problems using genetic algorithms: Part II. Hybrid genetic search strategies. *Computers & industrial engineering, proceedings of the 24th international conference on computers and industrial engineering,* 37 (1), 51-55.

CHINNASRI, W.; KROOTJOHN, S.; SUREERATTANAN, N. (2012). *Performance comparison of Genetic Algorithm's crossover operators on University Course Timetabling Problem.* Computing Technology and Information Management (ICCM), 2012 8th International Conference on, 2 781-786.

CHRISTOFIDES, N.; MINGOZZI, A.; TOTH, P.; SANDI, C. (1979). *Combinatorial Optimization.* Bath, A Wiley-Interscience Publication.

CHU, H. D.; GELMAN, E.; JOHNSON, E. L. (1997). Solving large scale crew scheduling problems. *European journal of operational research,* 97 (2), 260-268.

CHU, P. C.; BEASLEY, J. E. (1998). Constraint Handling in Genetic Algorithms: The Set Partitioning Problem. *Journal of heuristics,* 4 (4), 323-357.

CHUANJUN, Z.; YURONG, C.; CHAOYONG, Z. (2009). A Modified Genetic Algorithm to Due Date of Job Shop Scheduling Problem. In: *Computer network and multimedia technology, 2009. CNMT 2009. international symposium on,* 1-5.

CLARKE,M.; HINDE, C. J.; WITHALL, M. S.; JACKSON, T. W.; PHILLIPS, I. W.; BROWN, S.; WATSON, R.(2010). Allocating railway platforms using a genetic algorithm. In: *Research and development in intelligent systems XXVI. Proceedings of AI- the twenty-ninth SGAI international conference on innovative techniques and applications of artificial intelligence.* 15-17 December. 421-437.

COELLO, A. C. (2000). An updated survey of GA-based multiobjective optimization techniques. *ACM,* 32 (2).

CONTRERAS-BOLTON, C.; PARADA, V. (2015). Automatic Combination of Operators in a Genetic Algorithm to Solve the Traveling Salesman Problem. *PLoS one,* 10 (9).

COOPER, J.; HINDE, C. (2003). Improving Genetic Algorithms Efficiency Using Intelligent Fitness Functions. In: CHUNG, Paul W H.; HINDE, Chris and ALI, Moonis (eds.). Springer Berlin Heidelberg, 2718, 636-643.

CROCE, T. R.; VOLTA, G. (1995). A genetic algorithm for the job shop problem. *Computer operational research,* 22 (1), 15-24.

CRUZ-CHAVEZ, M. A. (2014). Neighbourhood generation mechanism applied in simulated annealing to job shop scheduling problems. *International journal of systems science,* 1-13.

CURET, N. D. (1993). A primal-dual simplex method for linear programs. *Operations research letters,* 13 (4), 233-237.

DA-CHENG, N.; YAN, F.; JUN-LIN, Z.; ZHEN, L.; ZI-KE, Z.; CHUANG, L. (2014). A personalized recommendation algorithm via biased random walk. *Computer Science and Software Engineering* (JCSSE), 2014 11th International Joint Conference on, 292-296.

DAHAL, K.; TAN, C. K.; COWLING, P. (2007). *Evolutionary Scheduling.* Berlin, Springer.

DANTZING, G.; WOLFE, P. (1974). The decomposition algorithm for linear program. In: DANTZING, G. B.; EAVES, B. C. (eds.). *Studies in Optimization.* US, 10.

DATAMONITOR (2012). Commercial Printing Industry Profile: Global. *Commercial printing industry profile: Global*, 1-32.

DAVIS, L. (ed.) (1991). *Handbook of genetic algorithms.* Van Nostrand, Reinhold.

DAVIS, K. A.; SONGER, A. D. (2009). Resistance to IT Change in the AEC Industry: Are the Stereotypes True? *Journal of construction engineering & management,* 135 (12), 1324-1333.

DB-SCHENKER Application and Projects Manager. (2015). Discussion of the potential impact of the automatic scheduling system on organisation performance. 09 December. Personal communication.

DB-SCHENKER Business Manager (2012). *Scheduling operations in the rail freight operating industry*. Interview with the author, 14 December. Personal communication

DB-SCHENKER Head of Finance (2012). *Key operations and main challenges in the rail freight operating industry*. Interview with the author, 07 November. Personal communication.

DB-SCHENKER Head of Finance (2013). *Driver Contractual Terms and Key Operations Objectives.* Interview with the author, 14 December. Personal communication.

DB-SCHENKER Head of HR (2012). *Key operations and main challenges in the rail freight operating industry*. Interview with the author, 07 November. Personal communication.

DB-SCHENKER Head of Service. (2015). Discussion of the potential impact of the automatic scheduling system on organisation performance. 09 December. Personal communication.

DB-SCHNEKER (2014a). *Coal and biomass rail freight services for the energy-generation sector.* [online]. Last updated 05 February 2014. https://www.rail.dbschenker.co.uk/rail-uk-en/industrysectors/coal_biomass.html.

DB-SCHNEKER (2014b). *Our history.* [online]. Last updated 05 February 2014. https://www.rail.dbschenker.co.uk/rail-uk-en/ourcompany/About_DB_Schenker_Rail_UK/history.html.

DB-SCHENKER (2014) *Profitable market leader.* [online]. Last updated 05 February 2014.http://ib2014.deutschebahn.com/ib2014-en/group-management-report/corporate-strategy/dimensions-and-management-approaches/profitable-market-leader.html.

DB-SCHENKER *Talent Acquisition* [online]. Last updated 05 February 2014. http://ib2014.deutschebahn.com/ib2014-en/group-management-report/group-performance-social-dimension/talent-acquistions.html.

DB-SCHENKER (2014). *Developing the strategies of tomorrows employees.* [online]. Last updated 05 February 2014. http://ib2014.deutschebahn.com/ib2014-en/integrated-thinking/ulrich-weber.html.

DE JONG, K. A. (1975). *An analysis of the behavior of a Class of Genetic Adaptive Systems.* PhD thesis, University of Michigan.

DE SNOO, C.; VAN WEZEL, W.; JORNA, R. J. (2011). An empirical investigation of scheduling performance criteria. *Journal of operations management,* 29 (3), 181-193.

DEB,K.; PRATAP,A.; AGARWAL,S.; MEYARIVAN,T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *Evolutionary computation, IEEE transactions on,* 6 (2), 182-197.

DEFERSHA, F.; CHEN, M. (2010). A parallel genetic algorithm for a flexible job-shop scheduling problem with sequence dependent setups. *The international journal of advanced manufacturing technology,* 49 (1-4), 263-279.

DENG, G. F.; LIN, W. T. (2011). Ant colony optimization-based algorithm for airline crew scheduling problem. *Expert systems with applications,* 38 (5), 5787-5793.

DEPARTMENT FOR TRANSPORT (2015). *National Railways freight moved by commodity.* Department for Transport Office of Rail Regulation. [online] Last updated 10 December 2015. https://www.gov.uk/government/statistical-data-sets/tsgb04-freight.

DERIGS, U.; MALCHEREK, D.; SCHAFER, S. (2010). Supporting strategic crew management at passenger railway model, method and system. *Public transport,* 2 (4), 307-334.

DESAULNIERS,G.; DESROSIERS,J.; DUMAS,Y.; MARC,S.; RIOUX,B.; SOLOMON,M.M.; SOUMIS,F. (1997). Crew pairing at Air France. *European journal of operational research,* 97 (2), 245-259.

DESROCHERS, M.; SOUMIS, F. (1988). *A Generalized Permanent Labelling Algorithm for the Shortest Path Problem with Time Windows.* University of Toronto Press. INFOR, 26 (3), 191-212.

DOCBROWN *Archive Steam* [online]. ArchiveSteam. http://www.docbrown.info/docspics/ArchiveSteam/.

DONG, H. (2012). An improvement genetic algorithm for solving the job-shop scheduling. In: *Computer science and information processing (CSIP), 2012 international conference on,* 1136-1139.

DORIGO, M. (1992). *Optimization, Learning and Natural Algorithms.* PhD, Poliecnico di Milano.

DOS SANTOS, A. G.; MATEUS, G. R. (2009). General hybrid column generation algorithm for crew scheduling problems using genetic algorithm. In: *Evolutionary computation, 2009. CEC '09. IEEE congress on,* 1799-1806.

DREXL, M.; PRESCOTT-GAGNON, E. (2010). Labelling algorithms for the elementary shortest path problem with resource constraints considering EU drivers' rules. *Logistics research,* 2 (2), 79-96.

DUCK, V.; WESSELMANN, F.; SUHL, L. (2011). Implementing a branch and price and cut method for the airline crew pairing optimization problem. *Public transport,* 3 (1), 43-64.

ELAOUD, S.; TEGHEM, J.; LOUKIL, T. (2010). Multiple crossover genetic algorithm for the multiobjective traveling salesman problem. *Electronic notes in discrete mathematics,* 36 , 939-946.

ELHADDAD, Y. R. (2012). Combined Simulated Annealing and Genetic Algorithm to Solve Optimization Problems. In: Aug 2012. Canakkale, Italy, Canakkale, World Academy of Science, Engineering and Technology (WASET), 1508-1510.

ELIZONDO,R.; PARADA,V.; PRADENAS,L.; ARTIGUES,C. (2010). An evolutionary and constructive approach to a crew scheduling problem in underground passenger transport. *Journal of heuristics,* 16 (4), 575-591.

EL-MIHOUB, T. A.; HOPGOOD,A.A.; NOLLE,L.; BATTERSBY,A. (2006). Hybrid Genetic Algorithms: A Review. *Engineering letters,* 13 (2), 124-137

EL-MIHOUB, T. A.; HOPGOOD, A.A and AREF, I.A (2013). Accelerating genetic schema processing through local search. In: *Computer, control, informatics and its applications (IC3INA), 2013 international conference on,* 343-348.

EMDEN-WEINERT, T.; PROKSCH, M. (1999). Best Practice Simulated Annealing for the Airline Crew Scheduling Problem. *Journal of heuristics,* 5 (4), 419-436.

ESSAFI, I.; MATI, Y.; DAUZERE-PERES, S. (2008). A genetic local search algorithm for minimizing total weighted tardiness in the job-shop scheduling problem. *Computers & operations research,* 35 (8), 2599-2616.

ETILER,O.; TOKLU,B.; ATAK,M.; WILSON,J. (2004). A Genetic Algorithm for Flow Shop Scheduling Problems. *The journal of the operational research society,* 55 (8), 830-835.

EUROSTAT (2015). Environmental tax statistics. [online]. Last updated 12 January2015http://ec.europa.eu/eurostat/statistics-explained/index.php/Environmental_tax_statistics.

EUROTUNNELGROUP (2014). *Traffic volumes for the past 10 years.* [online]. http://www.eurotunnelgroup.com/uk/eurotunnel-group/operations/traffic-figures/.

FANG, Z.; ZHANG, L.; CHEN, K. (2016). *A behavior mining based hybrid recommender system*. 1-5.

FISHER, M. L. (1981). The Lagrangian Relaxation Method for Solving Integer Programming Problems. *Management science,* 27 (1), 1-18.

FREEZE, R. D.; SCHMIDT, P. J. (2015). To Use or Not to Use-ERP Resistance is the Question: The Roles of Tacit Knowledge and Complexity. *Decision sciences journal of innovative education,* 13 (2), 247-272.

GARNETT DICKINSON Chief Executive (2012). *Scheduling and logistics issues in the printing company*. Interview with the author, 16 October. Personal communication.

GARNETT DICKINSON Divisional Managing Director (2012). *Scheduling and logistics issues in the printing company*. Interview with the author, 16 October. Personal communication.

GARNETT DICKINSON Estimator (2013). *Scheduling processes in the printing industry*. Interview with the author, 23 April. Personal communication.

GARNETT-DICKINSON *Garnett-Dickinson home.* [online]. http://www.garnett-dickinson.co.uk/index.asp.

GARNETT DICKINSON Machine Operator (2013). *Scheduling processes in the printing industry*. Interview with the author, 23 April. Personal communication.

GARNETT DICKINSON Operations Manager (2013). *Overview of the scheduling operations in the printing industry.* Interview with the author, 23 April. Personal communication.

GARNETT DICKINSON Sales Manager (2013). *Scheduling processes in the printing industry*. Interview with the author, 23 April. Personal communication.

GAO, J.; CHEN, R.; LIU, Y. (2012). A knowledge-based genetic algorithm for permutation flowshop scheduling problems with multiple factories. *International journal of advancements in computing technology,* 4 (7), 121-129.

GAO, J.; SUN, L.; GEN, M. (2008). A hybrid genetic and variable neighborhood descent algorithm for flexible job shop scheduling problems. *Computers & operations research, part special issue: Bio-inspired methods in combinatorial optimization,* 35 (9), 2892-2907.

GAZETTELIVE (2015). *DB Schenker: Hundreds of jobs to go at rail freight company with Thornaby base.* [online]. Last updated 22 June 2015. http://www.gazettelive.co.uk/business/business-news/db-schenker-jobs-lost-thornaby-9504333.

GEN, M.; CHENG, R. (1997). *Genetic Algorithms and Engineering Optimization.* Canada, John Wiley & Sons.

GEN, M.; LIN, L. (2014). Multiobjective evolutionary algorithm for manufacturing scheduling problems: state-of-the-art survey. *Journal of intelligent manufacturing,* 25 (5), 849-866.

GENDREAU, M.; POTVIN, J. Y. (2005). Metaheuristics in Combinatorial Optimization. *Annals of operations research,* 140 (1), 189-213.

GERE, W. S.;Jr. (1966). Heuristics in Job Shop Scheduling. *Management science,* 13 (3), 167-190.

GIFFLER, B.; THOMPSON, G. L. (1960). Algorithms for Solving Production-Scheduling Problems. *Operations research,* 8 (4), 487-503.

GILL, J. (2010). *Research methods for managers.* 4th ed.; London, SAGE.

GLASSDOOR (2016). *Salaries in UK.* [online]. https://www.glassdoor.com/.

GLOVER, F. (1986). Future paths for integer programming and links to artificial intelligence. *Computers and operations research,* 13 (5), 533-549.

GOOGLE (2016). *Google Maps API*. [online]. https://developers.google.com/maps/premium/.

GOGNA, A.; TAYAL, A. (2013). Metaheuristics: review and application. *Journal of experimental & theoretical artificial intelligence, J.exp.theor.artif.intell.;* 25 (4), 503-526.

GONCALVES, J. F.; de Magalhaes Mendes, J. J.; RESENDE, M.G. C. (2005). A hybrid genetic algorithm for the job shop scheduling problem. *European journal of operational research,* 167 (1), 77-95.

GOOGLE (2016). *Google Maps Directions Api.* [online]. Last updated January 20. https://developers.google.com/maps/documentation/directions/.

GOPALAKRISHNAN, B.; JOHNSON, E. L. (2005). Airline Crew Scheduling: State-of-the-Art. *Annals of operations research,* 140 , 305-337.

GOUMOPOULOS, C.; HOUSOS, E. (2004). Efficient trip generation with a rule modeling system for crew scheduling problems. *Journal of systems and software,* 69 (1), 43-56.

HAEREM, T.; RAU, D. (2007). The Influence of Degree of Expertise and Objective Task Complexity on Perceived Task Complexity and Performance. *Journal of applied psychology,* 92 (5), 1320-1331.

HALLIKAINEN, P.; KIVIJARVI, H.; NURMIMAKI, K. (2002). Evaluating strategic IT investments: an assessment of investment alternatives for a web content management system. In: *System sciences, 2002. HICSS. proceedings of the 35th annual hawaii international conference,* 2977-2986.

HAMILTON, S.; CHERVANY, N. L. (1981). Evaluating Information System Effectiveness - Part I: Comparing Evaluation Approaches. *MIS quarterly,* 5 (3), 55-69.

HAN, L.; KENDALL, G. (2003). Guided Operators for a Hyper-Heuristic Genetic Algorithm. In: GEDEON, Domonkos and FUNG, Lance Chun Che (eds.). Springer Berlin Heidelberg, 2903, 807-820.

HANAFI, R.; KOZAN, E. (2014). A hybrid constructive heuristic and simulated annealing for railway crew scheduling. *Computers & industrial engineering,* 70 (0), 11-19.

HART, E.; ROSS, P.; NELSON, J. (1998). Solving a Real-World Problem Using an Evolving Heuristically Driven Schedule Builder. *Evolutionary computation,* 6 (1), 61-80.

HART, E.; ROSS, P.; CORNE, D. (2005). Evolutionary Scheduling: A Review. *Genetic programming and evolvable machines,* 6 (2), 191-220.

HASAN, S. M. K.; SARKER, R.; CORNFORTH, D. (2007). Hybrid Genetic Algorithm for Solving Job-Shop Scheduling Problem. In: *Computer and information science, 2007. ICIS 2007. 6th IEEE/ACIS international conference,* 519-524.

HAUPT, R. L. (1998). *Practical genetic algorithms.Canada,* Wiley.

HE, Y.; WU, Y. (2013). Packing non-identical circles within a rectangle with open length. *Journal of global optimization,* 56 (3), 1187-1215.

HERRERA, F.; LOZANO, M. (2003). Fuzzy adaptive genetic algorithms: design, taxonomy, and future directions. *Soft computing,* 7 (8), 545-562.

HILLIER, F. S. (2005). *Introduction to operations research.* 8th ed.; London, McGraw-Hill.

HM REVENUE&CUSTOMS (2015). *Overseas trade statistics.* [online]. Last updated 9 September 2015. https://www.uktradeinfo.com/Statistics/EUOverseasTrade/Pages/EuOTS.aspx.

HOLLAND, J.; H. (1975). *Adaptation in Natural and Artificial Systems, MIT Press.*

HOLT, J. (2009). *Pragmatic Guide to Business Process Modelling. London,* British Computer Society.

HONG, I.; KAHNG, A. B.; BYUNG R. M. (1995). Exploiting synergies of multiple crossovers: initial studies. In: *Evolutionary computation, 1995.; IEEE international conference,* 245.

HONG, T. P.; WANG, H. S and CHEN, W. C. (2000). Simultaneously Applying Multiple Mutation Operators in Genetic Algorithms. *Journal of heuristics,* 6 (4), 439-455.

HOPGOOD, A. A. (2012). *Intelligent Systems for Engineers and Scientists.* 3rd ed.; Boca Raton, CRC Press.

HUISMAN,D.; KROON,L.G.; LENTINK,R.M.; VROMANS,M.J.C.M.; HUISMAN,D.; KROON,L.G.; LENTINK,R. M.; VROMANS,M.L J.C.M. (2005). Operations Research in passenger railway transportation. *Statistica neerlandica,* 59 (4), 467-497.

I-LIN, W. J.; Ellis L.; SOKOL, J. S. (2005). A Multiple Pairs Shortest Path Algorithm. *Transportation science,* 39 (4), 465-476.

INTERNATIONAL FEDERATION OF ROBOTICS (2015). *World Robotics 2015 Industrial Robots.* [online]. http://www.ifr.org/industrial-robots/statistics/.

ISLAM,D.; JACKSON,R.; ZUNDER,T.; BURGESS,A. (2015). Assessing the impact of the 2011 EU Transport White Paper - a rail freight demand forecast up to 2050 for the EU27. *European transport research review,* 7 (3), 1-9.

JAP, B. T.; LAL, S.; FISCHER, P. (2011). Comparing combinations of EEG activity in train drivers during monotonous driving. *Expert systems with applications,* 38 (1), 996-1003.

JARGEN, B.; GUNTSCH, M. (2005). Ant Colony Optimization. In: Chapman and Hall, Handbook of Bioinspired Algorithms and Applications, 3-41-3-54.

JAVADI, R.; HASANZADEH, M. (2012). A new method for hybridizing metaheuristics for multi-objective flexible job shop scheduling. In: *Computer and knowledge engineering (ICCKE), 2012 2nd international Conference,* 105-110.

ISLAM,D.; JACKSON,R.; ZUNDER,T.; BURGESS,A.(2008). Driver fatigue during extended rail operations. *Applied ergonomics,* 39 (5), 623-629.

JENSEN, P. A. (2003). *Operations research: models and methods.* Hoboken, Wiley.

JERIN L. I.; SARAVANA S. S.; PONNAMBALAM, S. G. (2013). An elitist strategy genetic algorithm using simulated annealing algorithm as local search for facility layout design. *The international journal of advanced manufacturing technology*, 1-13.

JIA, Z.; LU, X.; YANG, J.; JIA, D. (2011). Research on job-shop scheduling problem based on genetic algorithm. *International journal of production research,* 49 (12), 3585-3604.

JIANCHAO T.; GUOJI Z.; BINBIN L.; BIXI Z. (2010). Hybrid Genetic Algorithm for Flow Shop Scheduling Problem. In: *Intelligent computation technology and automation (ICICTA), 2010 international conference,* 449-452.

JOHNSON, R. D.; YANSON, R. (2015). Job Satisfaction and Turnover Intentions during Technology Transition: The Role of User Involvement, Core Self-Evaluations, and Computer Self-Efficacy. *Information resources management journal,* 4 (28), 38-51.

JONES, P. (2012). *Operations management.* Oxford, Oxford University Press.

JUTTE, S.; THONEMANN,U. W. (2011). Optimizing Railway Crew Scheduling at DB Schenker. *Interfaces,* 41 (2), 109-121.

KALITA, Z.; DATTA, D. (2014). Solving the bi-objective corridor allocation problem using a permutation-based genetic algorithm. *Computers and operations research,* 52 , 123-134.

KHAN, N.; SIKES, J. (2014). *IT under pressure.* McKinsey&Company. [online]. Last updated March 2014. http://www.mckinsey.com/business-functions/business-technology/our-insights/it-under-pressure-mckinsey-global-survey-results.

KIM, J.L.; ELLIS, R. D. (2008). Permutation-Based Elitist Genetic Algorithm for Optimization of Large-Sized Resource-Constrained Project Scheduling. *Journal of construction engineering and management,* 134 (11), 904-913.

KIM, K. W.; GEN, M.; YAMAZAKI, G. (2003). Hybrid genetic algorithm with fuzzy logic for resource-constrained project scheduling. *Applied soft computing,* 2 (3), 174-188.

KINCAID, R. K. (2008). Metaheuristics for Discrete Optimization Problems. In: CRC Press, 376-414.

KIRAZ, B.; UYAR, S.; OZKAN, E.(2013). Selection hyper-heuristics in dynamic environments. *Journal of the operational research society,* 64 , 1753-1769.

KIRKPATRICK, S.; GELATT C.D and VECCHI, M. P. (1983). Optimization by Simulated Annealing. *Science,* 220, 671-680.

KLABJAN, D.; JOHNSON, E. L.; NEMHAUSER, G. L. (2000). A parallel primal–dual simplex algorithm. *Operations research letters,* 27 (2), 47-55.

KLABJAN,D.;JOHNSON,E.L.; EMHAUSER,G..; GELMAN,E.; RAMASWAMY,S. (2001). Solving Large Airline Crew Scheduling Problems: Random Pairing Generation and Strong Branching. *Computational optimization and applications,* 20 (1), 73-91.

KLAUS,T.; WINGREEN, S.C.; BLANTON, E.J. (2010). Resistant groups in enterprise system implementations: a Q-methodology examination. *Journal of information technology,* 25 (1), 91.

KLEIN, E. E. (2003). Group support systems and the removal of barriers to creative idea generation within small groups: the inhibition of normative influence. *Virtual education: Cases in learning and teaching technologies,* , 91-112.

KORNILAKIS, H.; STAMATOPOULOS, P. (2002). Crew pairing optimization with genetic algorithms. In: *Methods and Applications of Artificial Intelligence.* Springer, 109-120.

KRAMER, O.; KOCH, P. (2007). Self-adaptive Partially Mapped Crossover. In: *Proceedings of the 9th annual conference on genetic and evolutionary computation,* London, England, ACM, 1523-1523.

KURZWEIL, R. (2006). *The Singularity Is Near.* United States, Viking.

KWAN R.S. (2004). *Bus and train driver scheduling*. In: Leung JY-T (ed.). *Handbook of Scheduling: Algorithms, Models, and Performance Analysis.* 300-400.

KWAN, R. S.; KWAN, A. S. K.; WREN, A. (2001). Evolutionary Driver Scheduling with Relief Chains. *Evolutionary computation,* 9 (4), 445-460.

KWAN, R. S.; WREN, A.; KWAN, A. S.; (2000). Hybrid genetic algorithms for scheduling bus and train drivers. In: P*roceedings of the congress on evolutionary computation,* 285-292 vol.1.

KWAN, R. S. (2011). Case studies of successful train crew scheduling optimisation. *Journal of scheduling,* 14 (5), 423-434.

LAND, A. H.; DOIG, A. G. (1960). An Automatic Method of Solving Discrete Programming Problems. *Econometrica,* 28 (3), 497-520.

LAPOINTE, L.; RIVARD, S.(2005). A Multilevel Model of Resistance to Information Technology Implementation. *MIS quarterly,* 29 (3), 461-491.

LASDON, L. S. (1970). *Optimization theory for large systems.*

LAUMER,S.; MAIER,C.; ECKHARDT,A.; WEITZEL,T. (2016). User personality and resistance to mandatory information systems in organizations: a theoretical model and empirical test of dispositional resistance to change. *Journal of information technology (palgrave macmillan),* 31 (1), 67-82.

LAVOIE, S.; MINOUX, M.and ODIER, E. (1988). A new approach for crew pairing problems by column generation with an application to air transportation. *European journal of operational research,* 35 (1), 45-58.

LEI, D. (2009). Multi-objective production scheduling: a survey. *Internation journal of advance manufacturing technologies,* 43 , 926-938.

LEI, D. (2010). Solving fuzzy job shop scheduling problems using random key genetic algorithm. *The international journal of advanced manufacturing technology,* 49 (1-4), 253-262.

LEVINE, D. (1996). Application of a hybrid genetic algorithm to airline crew scheduling. *Computers & Operations research,* 23 (6), 547-558.

LI, J.; KWAN, R. S. (2003). A fuzzy genetic algorithm for driver scheduling. *European journal of operational research, fuzzy sets in scheduling and planning,* 147 (2), 334-344.

LIANG, M.; YIU-MING, C.; YU-PING, W. (2004). A dynamically switched crossover for genetic algorithms. In: *Machine learning and cybernetics, 2004. proceedings of 2004 international conference,* 3254-3257 vol.5.

LIANG, Y.; LEUNG, K.S. (2011). Genetic Algorithm with adaptive elitist-population strategies for multimodal function optimization. *Applied soft computing journal,* 11 (2), 2017-2034.

LIAW, C. F. (2013). An improved branch-and-bound algorithm for the preemptive open shop total completion time scheduling problem. *Journal of industrial and production engineering,* 30 (5), 327-335.

LIMBU, Y. B.; JAYACHANDRAN, C.; BARRY, J. (2014). Does information and communication technology improve job satisfaction? The moderating role of sales technology orientation. *Industrial marketing management,* 43 (7), 1236-1245.

LIU,M.; SUN,Z.J.; YAN,J.W.; KANG,J.S. (2011). An Adaptive annealing genetic algorithm for the job-shop planning and scheduling problem. *Expert systems with applications,* 38 (8), 9248-9255.

LIU, Y.; SUN, F. (2011). A fast differential evolution algorithm using k-Nearest Neighbour predictor. *Expert systems with applications,* 38 (4), 4254-4258.

LOSHIN, D. (2013). Chapter 12 - Data Quality. In: LOSHIN, D. (ed.). *Business Intelligence (Second Edition).* Morgan Kaufmann, 165-187.

LU, D.; GZARA, F. (2015). The robust crew pairing problem: model and solution methodology. *Journal of global optimization,* 62 (1), 29-54.

LUBBECKE, M. E.; DESROSIERS, J. (2005). Selected Topics in Column Generation. *Operations research,* 53 (6), 1007-1023.

LUNDEN, I. (2013). *Forrester: $2.1 Trillion Will Go Into IT Spend In 2013, Apps And The U.S. Lead The Charge.* [online]. Last updated 15/07/2013. https://techcrunch.com/2013/07/15/forrester-2-1-trillion-will-go-into-it-spend-in-2013-apps-and-the-u-s-lead-the-charge/.

LUZ M.; TORRES P.; OSCAR F.; CASTELLANOS,D.; CLAUDIA, N. J. (2010). Evaluating technology intelligence system efficiency. *Ingeniería e investigación,* 30 (3), 106.

MA,Q.; ZHANG,Y.A,, YAMAMORI,K.; SAKAMOTO,M.; FURUTANI,H. (2011). *Markov chain analysis of genetic algorithms for 3-SAT problem.* Natural Computation, Seventh International Conference, (2) 1101-1105.

MAJUMDAR, J.; BHUNIA, A. K. (2011). Solving a multi-objective interval crew-scheduling problem via Genetic Algorithms. OPSEARCH, 48 (3), 197-216.

MAKRI, A.; KLABJAN, D. (2004). A New Pricing Scheme for Airline Crew Scheduling. *INFORMS,* 16 (1), 56-67.

MARKETLINE (2014). Road & Rail Industry Profile: United Kingdom. *Road & rail industry profile: United kingdom,* , 1-37.

MARKUS, M. L. (1983). Power, Politics, and MIS Implementation. *Communications of the ACM,* 26 (6), 430-444.

MARTIN, M. P. (1995). *Analysis and design of business information systems.* 2nd ed.; Prentice-Hall.

MATTFELD, D. C.; BIERWIRTH, C. (2004). An efficient genetic algorithm for job shop scheduling with tardiness objectives. *European journal of operational research,* 155 (3), 616-630.

MCCLINTOCK, S.; LUNNEY, T.; HASHIM, A. (1997). A fuzzy logic controlled genetic algorithm environment. In: *Systems, man, and cybernetics, 1997. computational cybernetics and simulation, IEEE international conference on,* 2181-2186 vol.3.

MEERAN, S.; MORSHED, M. S. (2012). A hybrid genetic tabu search algorithm for solving job shop scheduling problems: a case study. *Journal of intelligent manufacturing,* 23 (4), 1063-1078.

METAN, G.; SABUNCUOGLU, I.; PIERREVAL, H. (2010). Real time selection of scheduling rules and knowledge extraction via dynamically controlled data mining. *International journal of production research,* 48 (23), 6909-6938.

MICHALEWICZ, Z. (1996). *Genetic algorithms + data structures =, evolution programs.* 3rd ed.; London, Springer.

MIRSANEI,H.S.; ZANDIEH,M.; MOAYED,M.J.; KHABBAZI,M.R. (2011). A simulated annealing algorithm approach to hybrid flow shop scheduling with sequence-dependent setup times. *Journal of intelligent manufacturing,* 22 (6), 965-978.

MISIR,M.; VERBEECK,K.; CAUSMAECKER,P.; BERGHE,G.(2013). A new hyper-heuristic as a general problem solver: an implementation in HyFlex. *Journal of scheduling,* 16 (3), 291-311.

MITCHELL, M. (1996). *An introduction to genetic algorithms.* Cambridge, MIT Press.

MITHAS, S.; RUST, R. T. (2016). How Information Technology Strategy and Investments Influence Firm Performance: Conjecture and Empirical Evidence. *MIS quarterly,* 40 (1), 223-246.

MONTANA, D. (2002). So You Want to Build an Automated Scheduling System. In: *The GECCO-2002 industrial track*.

MONTANA, D.; TALIB, H.; VIDAVER, G. A. (2007) Genetic-Algorithm-Based Reconfigurable Scheduler.

MONTANA, D.; HUSSAIN, T.; VIDAVER, G. (2007) A Genetic-Algorithm-Based Reconfigurable Scheduler. In: DAHAL, Keshavp, TAN, KayChen and COWLING, PeterI (eds.). Springer Berlin Heidelberg, 49, 577-611.

MONTGOMERY, D. C. (2013). *Design and analysis of experiments.* 8th ed.; International student version.. ed.; Hoboken, N.J.: John Wiley and Sons.

MORRIS, M. G.; VENKATESH, V. (2010). Job Characteristics and Job Satisfaction: Understanding the Role of Enterprise Resource Planning System Implementation. *MIS quarterly,* 34 (1), 143-161.

MORRIS, M.; DAVIS, G.; DAVIS, F. (2003). User acceptance of information technology: Towards a unified view. *MIS quarterly,* 27 (3), 425-478.

MU, S.; DESSOUKY, M. (2011). Scheduling freight trains traveling on complex networks. *Transportation research part B: Methodological,* 45 (7), 1103-1123.

NABABAN,E. B.; HAMDAN,A. R.; ABDULLAH,S.; ZAKARIA,M.S. (2008). Branch and bound algorithm in optimizing job shop scheduling problems. In: *Information technology, ITSim 2008. international symposium on,* 1-5.

NARRATIVE SCIENCE (2015) *State of AI and Big Data in the Enterprise.* Narrative Science.

NATIONAL RAIL (2015). *National Rail Enquiries.* [online]. http://www.nationalrail.co.uk/.

NATIONAL RAIL (N/A). *Transparency.* [online]. http://www.nationalrail.co.uk/100752.aspx.

NAZIF, H.; LEE, L. S. (2012). Optimised crossover genetic algorithm for capacitated vehicle routing problem. *Applied mathematical modelling,* 36 (5), 2110-2117.

NEGNEVITSKY, M. (2011). *Artificial intelligence:: a guide to intelligent systems.* 3rd ed.; Harlow , Addison Wesley.

NEMHAUSER, G. L.; WOLSEY, L. A. (1988). *Integer and Combinatorial Optimization.* New York, A Wiley-Interscience Publication.

NETA,B.M.M.; ARAJO, G. H.D.; GUIMARAZES, F. G.; MESQUITA,R.C.; EKEL,P. Y. (2012). A fuzzy genetic algorithm for automatic orthogonal graph drawing. *Applied soft computing,* 12 (4), 1379-1389.

NETWORK RAIL (2010). *Value and Importance of Rail Freight.*

NETWORK RAIL (2014). *The future of the rail freight.*

NISHI, T.; MUROI, Y.; INUIGUCHI, M. (2011). Column generation with dual inequalities for railway crew scheduling problems. *Public transport,* 3 (1), 25-42.

NOLLE,L.; GOODYEAR,A.; HOPGOOD,A.A.; PICTON,P.D.; BRAITHWAITE,N.STJ. (2001). On Step Width Adaptation in Simulated Annealing for Continuous Parameter Optimisation. In: REUSCH, Bernd (ed.). Springer Berlin Heidelberg, 2206, 589-598.

NOURANI, Y.; ANDRESEN, B. (1998). A comparison of simulated annealing cooling strategies. *Journal of physics A-mathematical and general,* 31 (41), 8373-8385.

OFFICE OF RAIL AND ROAD (2015). *Freight operator companies.* [online]. http://orr.gov.uk/about-orr/who-we-work-with/industry-organisations/freight-operator-companies.

ONWUBOLU, G. C.; BABU, B. V. (2004). *New Optimization Techniques in Engineering.* Berlin, Springer.

OREG, S. (2006). Personality, context, and resistance to organizational change. *European journal of work and organizational psychology,* 15 (1), 73-101.

OREG,S.; BAYAZIT, M.; VAKOLA,M.; ARCINIEGA,L.; ARMENAKIS,A.; BARKAUSKIENE,R.; BOZIONELOS,N.; FUJIMOTO,Y.; GONZALEZ, L.; HAN,J.; HREBIKOVA, M.; JIMMIESON,N.; KORDACOVA, J.; MITSUHASHI,H.; MIACIC, B.; FERRIC, I.; TOPIC, M. K.; OHLY,S.; SAKSVIK,P.O.; HETLAND,H.; SAKSVIK,I.; VAN DAM, K. (2008). Dispositional Resistance to Change: Measurement Equivalence and the Link to Personal Values Across 17 Nations. *Journal of applied psychology,* 93 (4), 935-944.

OSMAN, I. H and LAPORTE, G. (1996). Metaheuristics: A bibliography. *Annals of operations research,* 63 (5), 511-623.

OTHMAN, M.; GOUW, G. J.; BHUIYAN, N. (2012). Workforce scheduling: A new model incorporating human factors. *Journal of industrial engineering and management,* 5 (2), 259.

OZDEMIR, H. T.; MOHAN, C. K. (2001). Flight graph based genetic algorithm for crew scheduling in airlines. *Information sciences,* 133 (3–4), 165-173.

PARK, T.; RYU, K. (2006). Crew pairing optimization by a genetic algorithm with unexpressed genes. *Journal of intelligent manufacturing,* 17 (4), 375-383.

PASCUAL,J.; LORIDO-BOTRÁN, T.; MIGUEL-ALONSO, J, LOZANO,J. (2015). Towards a Greener Cloud Infrastructure Management using Optimized Placement Policies. *Journal of grid computing, from grids to cloud federations,* 13 (3), 375-389.

PATEL, N.; PADHIYAR, N. (2015). Modified genetic algorithm using Box Complex method: Application to optimal control problems. *Journal of process control,* 26 , 35-50.

PEZZELLA, F.; MORGANTI, G.; CIASCHETTI, G. (2008). A genetic algorithm for the Flexible Job-shop Scheduling Problem. Computers & operations research, 35 (10), 3202-3212.

PINEDO, M. L. (2009). *Planning and scheduling in manufacturing and services.* 2nd ed.; London, Springer.

PINTO, G.; AINBINDER, I.; RABINOWITZ, G. (2009). A genetic algorithm-based approach for solving the resource-sharing and scheduling problem. *Computers & industrial engineering,* 57 (3), 1131-1143.

POLOVINA, S. (2013). A Transaction-oriented architecture for enterprise systems. *International Journal of Intelligent Information Technologies*, 9 (4), 69-79.

PONNAMBALAM, S. G.; JAWAHAR, N.; ARAVINDAN, P. (1999). A simulated annealing algorithm for job shop scheduling. *Production planning & control,* 10 (8), 767-777.

POON, P. W.; CARTER, J. N. (1995). Genetic algorithm crossover operators for ordering applications. *Computers & operations research, genetic algorithms,* 22 (1), 135-147.

PROBST, T. M. (2003). Exploring employee outcomes of organizational restructuring: a Solomon four- group study.(Longitudinal Processes in Groups and Organizations). *Group & organization management,* 28 (3), 416-439.

PUGLIESE, L. D..P and GUERRIERO, F. (2013). A survey of resource constrained shortest path problems: Exact solution approaches. *Networks,* 62 (3), 183-200.

QING-DAO-ER-JI, R.; WANG, Y. (2012). A new hybrid genetic algorithm for job shop scheduling problem. *Computers & operations research,* 39 (10), 2291-2299.

RAEESI N.M.; KOBTI, Z. (2012). A memetic algorithm for job shop scheduling using a critical-path-based local search heuristic. *Memetic computing,* 4 (3), 231-245.

RAKESH K.; GIRDHAR G.; and RAJESH K.; (2013). Novel Crossover Operator for Genetic Algorithm for Permutation Problems. *International journal of soft computing & engineering,* 3 (2), 252-258.

RAKKIANNAN, T.; PALANISAMY, B. (2012). Hybridization of Genetic Algorithm with Parallel Implementation of Simulated Annealing for Job Shop Scheduling. *American journal of applied sciences,* 9 (10), 1694-1705.

REES, P. (1992). User evaluation of expert systems. *Industrial management & data systems,* 92 (6) (1992), 17-23.

REEVES, C. R. (1993). *Modern heuristic techniques for combinatorial problems.* Oxford, Blackwell.

REMENYI, D.; MONEY, A.; TWITE, A. (1991). *A Guide to measuring and managing IT benefits .* Manchester, Blackwel.

ROBINSON, S. (1994). *Successful simulation: a practical approach to simulation projects.* McGraw-Hill.

ROTH, R. M. (2012). *Systems analysis and design.* 5th ed.; International student version.. ed.; Hoboken, N.J,: Wiley.

SARKER, R. A.; NEWTON, C. S. (2007). The Process of Optimization. In: *Optimization Modelling: A Practical Approach.* CRC Press.

SAHU, A.; TAPADAR, R. (2007). Solving the Assignment problem using Genetic Algorithm and Simulated Annealing. *IAENG international journal of applied mathematics,* 36 (1), 37-40.

SAPSFORD, R.; JUPP, V. (2006). *Data collection and analysis.* 2nd ed.; London: SAGE in association with the Open University.

SHARMA, S.; TAPASWI, N. (2013). Solving TSP Using Advanced Crossover & Mutating Operators of Genetic Algorithm. *International journal of electronics communication and computer engineering,* 4 (4), 1289-1292.

SHEBALOV, S.; KLABJAN, D. (2006). Robust Airline Crew Pairing: Move-up Crews. *Transportation science,* 40 (3), 300-312.

SHEN,Y.; PENG,K.; CHEN,K.; LI,J. (2013). Evolutionary crew scheduling with adaptive chromosomes. *Transportation research part B: Methodological,* 56 (0), 174-185.

SHIFTPLANNING (2016). *Shift Planning-Online Employee Scheduling an Management Software.* [online]. https://www.shiftplanning.com/.

SHL, G. (1997). A genetic algorithm applied to a classic job-shop scheduling problem. *International journal of systems science,* 28 (1), 25-32.

SIVANANDAM, S. N.; DEEPA, S. N. (2008). *Introduction to Genetic Algorithm.* Berlin, Springer.

SLACK, N. (2013). *Operations management.* Seventh ed.; Harlow, England : Pearson.

SPANOS, A. C.; PONIS, S.T.; TATSIOPOULOS, I. P.; CHRISTOU, I.T. ; ROKOU, E. (2014). A new hybrid parallel genetic algorithm for the job-shop scheduling problem. *International transactions in operational research,* 21 (3), 479-499.

SPEARS, W.; M.; DE JONG, K.; A (1991). An analysis of multi-point crossover. In: RAWLINS, Gregory J. E. (ed.). *Foundations of genetic algorithm.* 301-315.

STEINHAFEL, K.; ALBRECHT, A.; WONG, C. K. (1999). Two simulated annealing-based heuristics for the job shop scheduling problem. *European journal of operational research,* 118 (3), 524-548.

STITTLE, J. (2004). Accounting for UK rail freight track charges: privatisation, politics and the pursuit of private sector vested interests. *Accounting forum,* 28 (4), 403-425.

STREBEL, P. (1996). Why Do Employees Resist Change? *Harvard business review,* 74 (3), 86-92.

STRUEBING, L. (1996). *Eight ways to reduce employee stress.* Quality progress, 29 (7), 14.

SUMER, E.; TURKER, M. (2013). An adaptive fuzzy-genetic algorithm approach for building detection using high-resolution satellite images. *Computers, environment and urban systems,* 39 (0), 48-62.

TAGAWA,K.; KANZAKI,Y.; OKADA,D.; INOUE,K.; HANEDA,H. (1998). *A new metric function and harmonic crossover for symmetric and asymmetric traveling salesman problems.* Evolutionary Computation Proceedings, IEEE World Congress on Computational Intelligence, 822-827.

TALLON, P. P.; KRAEMER, K. L.; GURBAXANI, V. (2000). Executives' Perceptions of the Business Value of Information Technology: A Process-Oriented Approach. *Journal of management information systems,* 16 (4), 145-173.

THAMILSELVAN, R.; BALASUBRAMANIE, P. (2012). Integrating Genetic Algorithm, Tabu Search and Simulated Annealing For Job Shop Scheduling. *International journal of computer applications,* 48 (5), 42-54.

TING, C.K, SU, C.H.; LEE, C.N. (2010). Multi-parent extension of partially mapped crossover for combinatorial optimization problems. *Expert systems with applications,* 37 (3), 1879-1886.

VAN RENSBURG, A. (2011). *Principles for modelling business processes.* Industrial Engineering and Engineering Management,1710-1714.

VARNAMKHASTI,M. J.; LEE,L.S.; BAKAR,M.R.A.; LEONG, W. J. (2012). A Genetic Algorithm with Fuzzy Crossover Operator and Probability. *Advances in operations research,* 2012 .

VELA, C. R.; VARELA, R.; GONZAILEZ, M. A. (2010). Local search and genetic algorithm for the job shop scheduling problem with sequence dependent setup times. *Journal of heuristics,* 16 (2), 139-165.

VILCOT, G.; BILLAUT, J. C. (2008). A tabu search and a genetic algorithm for solving a bicriteria general job shop scheduling problem. *European journal of operational research,* 190 (2), 398-411.

VON ROSING, M.; POLOVINA, S. (2015). Business process trends. In: *The complete business process handbook*. Elsevier, 187-216.

WANG, L.; ZHENG, D. Z. (2001). An effective hybrid optimization strategy for job-shop scheduling problems. *Computers & Operations research,* 28 (6), 585-596.

WANG, S.; WU, L. (2010). *A novel hybrid genetic algorithm for global optimization.;* Sixth International Conference on Natural Computation, 2, 1058-1061.

WANNER, E. F. (2007). Hybrid genetic algorithms using quadratic local search operators. *COMPEL: The international journal for computation and mathematics in electrical and electronic engineering,* 26 (3), 773-787.

WHENTOWORK (2016). *Online Employee Shift Scheduling.* [online]. http://whentowork.com/.

WIESE, K. C.; GLEN, E. (2003). A permutation-based genetic algorithm for the RNA folding problem: a critical look at selection strategies, crossover operators, and representation issues. *Biosystems,* 72 (1–2), 29-41.

WITHALL, M.S.; HINDE, C.J.; JACKSON, T.; PHILLIPS, I. W.; PHILLIPS, I. W.; BROWN, S.; WATSON, R. (2011). Automating rolling stock diagramming and platform allocation. In: *World congress on railway research SNCF.*

WOOD, M. (1965). Parman Economic Programming Model. *Management science (pre-1986),* 11 (7), 619.

WORLD ENERGY COUNCIL, IBM Corporation and Paul Scherrer Institute (2012). *Global Transport Scenarios 2050.* World Energy Council,.

WREN, A.; WREN, D. O. (1995). A genetic algorithm for public transport driver scheduling. *Computers and operations research,* 22 , 101-110.

WREN,A.; FORES,S.; KWAN,A.; KWAN,R.; PARKER,M.; PROLL,L. (2003). A flexible system for scheduling drivers. *Journal of scheduling, J.sched.;* 6 (5), 437-455.

WTO (2014). *Trade and development: recent trends and the role of the WTO.*

XU, H. Y.; VUKOVICH, G. (1993). A fuzzy genetic algorithm with effective search and optimization. In: *Proceedings of international joint conference on Neural networks,* 2967-2970 vol.3.

XU, X.; XU, Z.; GU, X. (2011). An asynchronous genetic local search algorithm for the permutation flowshop scheduling problem with total flowtime minimization. *Expert systems with applications,* 38 (7), 7970-7979.

YAN, S.; CHANG, J. C. (2002). Airline cockpit crew scheduling. *European journal of operational research,* 136 (3), 501-511.

YANG,H.A, SUN,Q.F, SAYGIN,C.; SUN,S.D. (2012). Job shop scheduling based on earliness and tardiness penalties with due dates and deadlines: an enhanced genetic algorithm. *The international journal of advanced manufacturing technology,* 61 (5-8), 657-666.

YOU-LIAN, Z.; YUAN-XIANG, L.; DE-MING LEI, C.X. (2010). Scheduling fuzzy job shop using random key genetic algorithm. In *international conference on Machine learning and cybernetics,* 1887-1892.

YUAN, C.; FAN, Z. P. (2006). *A Multiobjective Optimization Model for Packing Proposals in Large-Scale R&D Project Review.* International Conference on Management Science and Engineering, 359-363.

YU-CHIUN, C.; LAN, L. W. (2002). Genetic fuzzy logic controllers. In: P*roceedings of the 5th international conference on Intelligent transportation systems,* 200-205.

ZEREN, B.; OZKOL, I. (2012). An Improved Genetic Algorithm for Crew Pairing Optimization. *Journal of intelligent learning systems and applications,* 4 (1), 70-80.

ZHANG, C.; RAO, Y.; LI, P. (2008). An effective hybrid genetic algorithm for the job shop scheduling problem. *The international journal of advanced manufacturing technology,* 39 (9), 965-974.

ZHANG, L.; ZHENG, W. (1995). *Genetic computing for the film copy deliverer problem. In Proceedings of* International Conference on Systems, Man and Cybernetics, 1067-1073.

ZHANG, L.; WANG, L.; ZHENG, D.Z. (2006). An adaptive genetic algorithm with multiple operators for flowshop scheduling. *The international journal of advanced manufacturing technology,* 27 (5), 580-587.

ZHANG, L.; GAO, L.; LI, X. (2013). A hybrid genetic algorithm and tabu search for a multi-objective dynamic job shop scheduling problem. *International journal of production research,* 51 (12), 3516-3531.

ZHANG, R.; WU, C. (2011). A simulated annealing algorithm based on block properties for the job shop scheduling problem with total weighted tardinessobjective. *Computers & operations research,* 38 (5), 854-867.

ZHANG, X.; SONG, Q. (2015). A Multi-Label Learning Based Kernel Automatic Recommendation Method for Support Vector Machine. *PLoS one*, 10 (4).

ZHOU, H. CHEUNG, W.; LEUNG, L. C. (2009). Minimizing weighted tardiness of job-shop scheduling using a hybrid genetic algorithm. *European journal of operational research,* 194 (3), 637-649.

Appendix 1. Focus group discussion

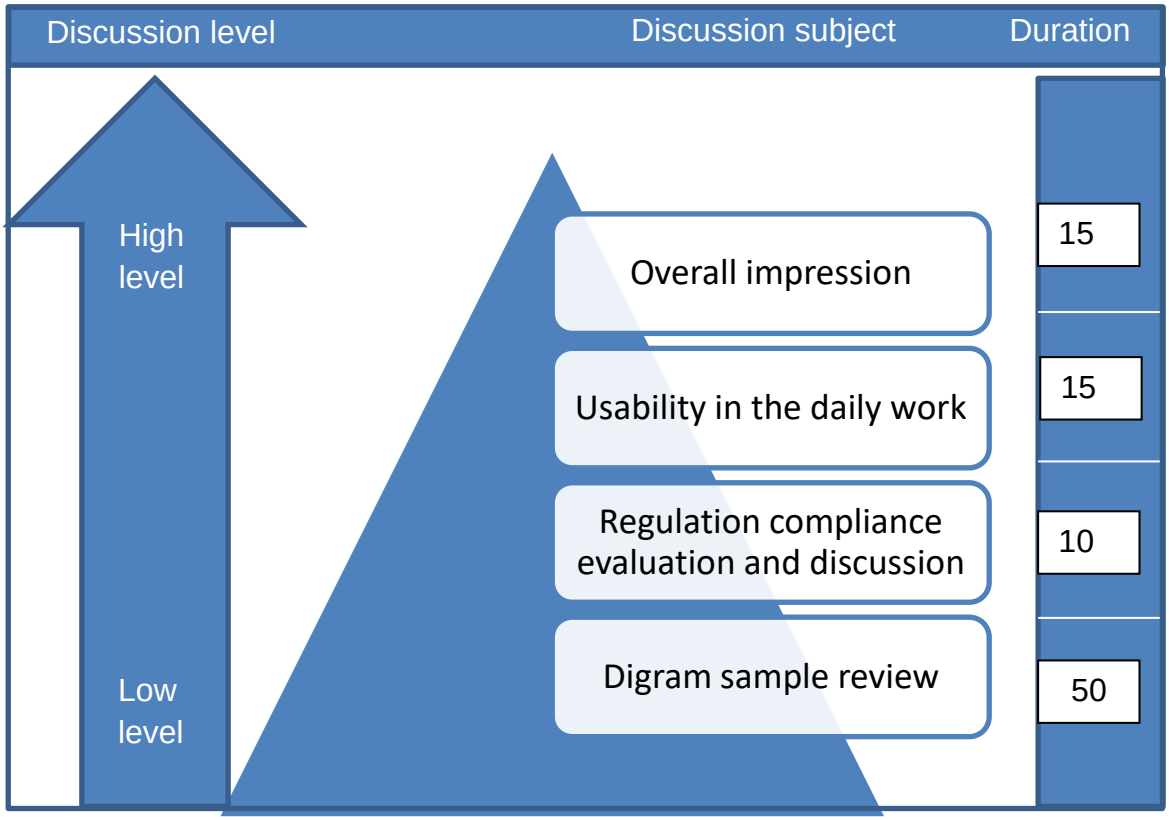


Figure 133 Focus group discussion plan

Appendix 2. Printing Job Schedule

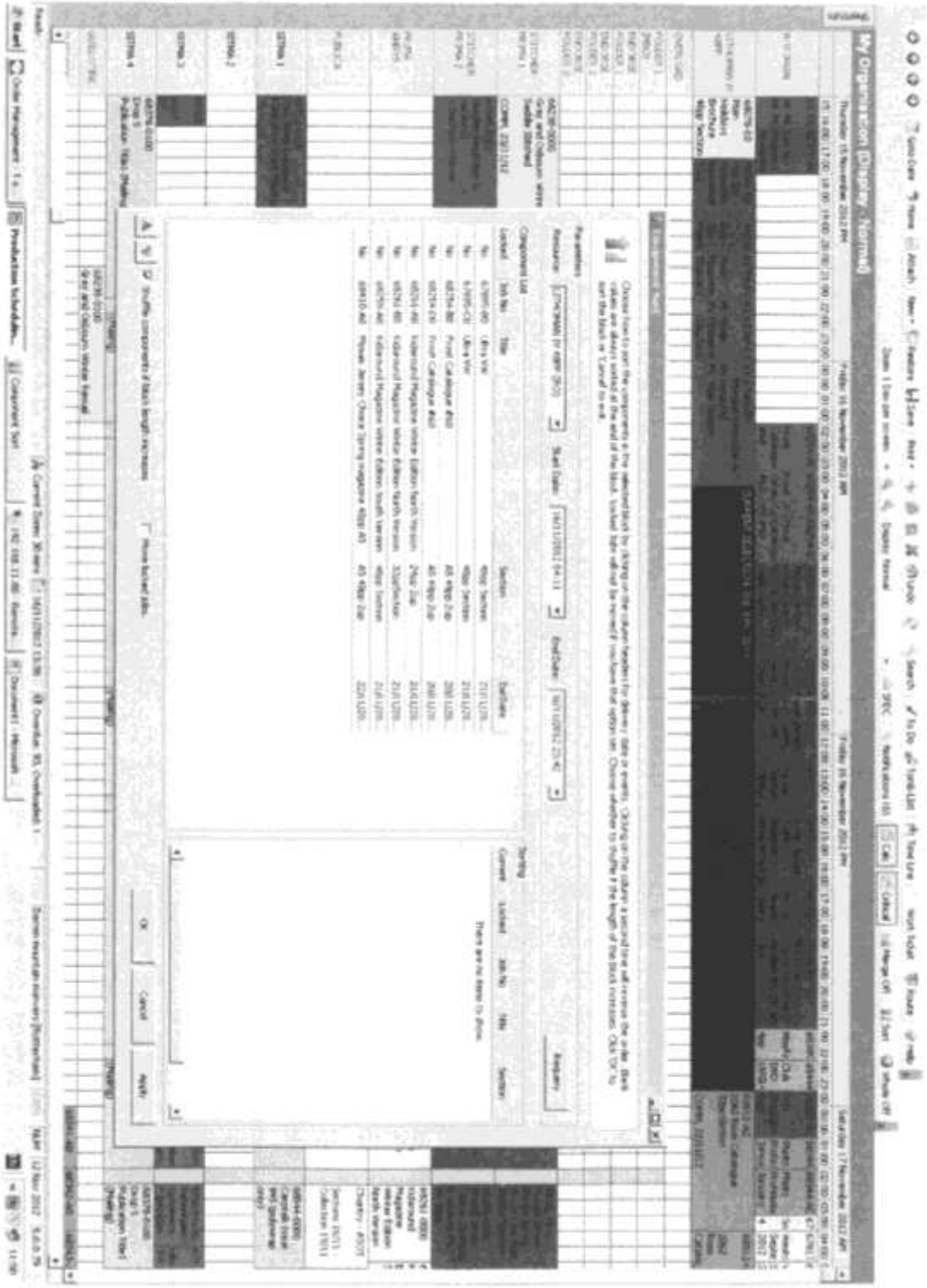


Figure 134 Gantt chart illustrating printing job schedule

Appendix 3. Printing job floor




Figure 135 Printing job floor in Garnett-Dickinson

Appendix 4. Traction types

Class	Description
Class 59/2 	Producer: General Motors Power type: Diesel-electric Max speed: 60–75 mph Tractive effort: 508 kN (114 000 lbf)
Class 60 	Producer: Brush Traction Power type: Diesel Max speed: 62/60 mph Tractive effort: 500 kN (106 500 lbf)
Class 66 	Producer: General Motors/EMD. Power type: diesel powered Max. speed :60/75 mph Tractive effort 409 kN (92 000 lbf)
Class 67 	Producer: Alstom ,General Motors. Power type:: Diesel-Electric Max. speed: 125 mph Tractive effort: 141 kN (31 770 lbf)
Class 90 	Producer: BREL Power type: Electric maximum 110 mph. Maximum tractive effort:258 kN (58000 lbf).
Class 92 	Producer: Brush Traction Power type: Electric Maxium speed of 145 km/h(90 mph) Maximum tractive effort 400 kN (90000 lbf).

Adapted from: Docbrown (2016)

Appendix 5. Data Instances for JSSP

Lawrence 20x10 instance (Table 8, instance 1), also called (setc1) or (C1)

20 10
8 52 7 26 6 71 9 16 2 34 1 21 5 95 4 21 0 53 3 55
4 55 5 98 3 39 9 79 0 12 8 77 6 77 7 66 2 31 1 42
5 37 4 92 2 64 6 54 1 19 7 43 0 83 3 34 9 79 8 62
1 87 5 77 0 93 3 69 2 87 7 38 8 24 6 41 9 83 4 60
2 98 5 25 6 75 9 77 1 49 3 17 8 79 0 44 7 43 4 96
1 7 4 61 0 95 2 35 9 10 8 35 5 28 3 76 7 95 6 9
5 59 9 43 0 46 4 28 6 52 3 16 2 59 1 91 8 50 7 27
5 9 9 43 8 14 7 71 4 20 6 54 3 41 0 87 1 45 2 39
1 28 8 66 0 78 2 37 9 42 3 26 5 33 6 89 4 33 7 8
4 96 3 27 6 78 5 84 2 94 8 69 1 74 9 81 7 45 0 69
4 24 7 32 9 25 2 17 3 87 8 81 5 76 6 18 1 31 0 20
8 90 5 28 1 72 7 86 2 23 3 99 6 76 9 97 4 45 0 58
2 17 4 98 3 48 1 46 8 27 6 67 7 62 0 42 9 48 5 27
0 80 8 50 3 19 7 98 5 28 2 50 4 94 6 63 1 12 9 80
9 72 0 75 4 61 8 79 6 37 2 50 5 14 3 55 7 18 1 41
3 96 2 14 5 57 0 47 7 65 4 75 8 79 1 71 6 60 9 22
1 31 7 47 8 58 3 32 4 44 5 58 6 34 0 33 2 69 9 51
1 44 7 40 2 17 0 62 8 66 6 15 3 29 9 38 5 8 4 97
2 58 3 50 4 63 9 87 0 57 6 21 7 57 8 32 1 39 5 20
1 85 0 84 5 56 3 61 9 15 7 70 8 30 2 90 6 67 4 20

Lawrence 30x10 instance, also called (setd1) or (D1) BKS 1888 (Aiex, Binato and Resende 2003)

30 10
4 21 7 26 9 16 2 34 3 55 8 52 5 95 6 71 1 21 0 53
8 77 5 98 1 42 7 66 2 31 3 39 6 77 9 79 4 55 0 12
2 64 4 92 3 34 1 19 8 62 6 54 7 43 0 83 9 79 5 37
0 93 8 24 3 69 7 38 5 77 2 87 4 60 6 41 1 87 9 83
9 77 0 44 4 96 8 79 6 75 2 98 5 25 3 17 7 43 1 49
3 76 2 35 5 28 0 95 7 95 4 61 8 35 1 7 6 9 9 10
1 91 7 27 8 50 3 16 4 28 5 59 6 52 0 46 2 59 9 43
1 45 7 71 2 39 0 87 8 14 6 54 3 41 9 43 5 9 4 20
2 37 3 26 4 33 9 42 0 78 6 89 7 8 8 66 1 28 5 33
1 74 0 69 5 84 3 27 9 81 7 45 8 69 2 94 6 78 4 96
5 76 7 32 6 18 0 20 3 87 2 17 9 25 4 24 1 31 8 81
9 97 8 90 5 28 7 86 0 58 1 72 2 23 6 76 3 99 4 45
9 48 5 27 6 67 7 62 4 98 0 42 1 46 8 27 3 48 2 17
9 80 3 19 5 28 1 12 4 94 6 63 7 98 8 50 0 80 2 50
2 50 1 41 4 61 8 79 5 14 9 72 7 18 3 55 6 37 0 75
9 22 5 57 4 75 2 14 7 65 3 96 1 71 0 47 8 79 6 60
3 32 2 69 4 44 1 31 9 51 0 33 6 34 5 58 7 47 8 58
8 66 7 40 2 17 0 62 9 38 5 8 6 15 3 29 1 44 4 97
3 50 2 58 6 21 4 63 7 57 8 32 5 20 9 87 0 57 1 39
4 20 6 67 1 85 2 90 7 70 0 84 8 30 5 56 3 61 9 15
6 29 0 82 4 18 3 38 7 21 8 50 1 23 5 84 2 45 9 41
3 54 9 37 6 62 5 16 0 52 8 57 4 54 2 38 7 74 1 52

4 79 1 61 8 11 0 81 7 89 6 89 5 57 3 68 9 81 2 30
9 24 1 66 4 32 3 33 8 8 2 20 6 84 0 91 7 55 5 20
3 54 2 64 6 83 9 40 7 8 0 7 4 19 5 56 1 39 8 7
1 6 4 74 0 63 2 64 9 15 6 42 7 98 8 61 5 40 3 91
1 80 3 75 0 26 2 87 9 22 7 39 8 24 4 75 6 44 5 6
5 8 3 79 6 61 1 15 0 12 7 43 8 26 9 22 2 20 4 80
1 36 0 63 7 10 4 22 3 96 5 40 9 5 8 18 6 33 2 62
4 8 8 15 2 64 3 95 1 96 6 38 7 18 9 23 5 64 0 89

Storer, Wu, and Vaccari hard 50x10 instance BKS is unknown

50 10
0 92 4 47 3 56 2 91 1 49 5 39 9 63 7 12 6 1 8 37
0 86 2 100 1 75 3 92 4 90 5 11 7 85 8 54 9 100 6 38
1 4 4 94 3 44 2 40 0 92 8 53 6 40 9 5 5 68 7 27
4 87 0 48 1 59 2 92 3 35 6 99 7 46 9 27 8 83 5 91
0 83 1 78 4 76 3 64 2 44 8 12 9 91 6 31 7 98 5 63
3 49 0 15 1 100 4 18 2 24 6 92 9 65 5 26 7 29 8 24
0 28 3 53 4 84 2 47 1 85 7 100 5 34 6 35 8 90 9 88
2 61 4 71 3 54 1 34 0 13 9 47 8 2 6 97 7 27 5 97
0 85 2 75 1 33 4 72 3 49 7 23 5 12 8 90 6 87 9 42
2 24 3 20 1 65 4 33 0 75 9 47 6 84 8 44 7 74 5 29
2 48 3 27 4 1 0 23 1 66 6 35 7 46 9 29 5 63 8 44
2 79 0 4 4 61 3 46 1 69 7 10 8 88 9 19 6 50 5 34
0 16 4 31 3 77 2 3 1 25 8 88 7 97 9 49 6 79 5 22
1 40 0 39 4 15 2 93 3 48 6 63 9 74 8 46 7 91 5 51
4 48 0 93 2 8 3 50 1 5 6 48 7 46 9 35 5 88 8 97
3 70 1 8 2 65 0 32 4 84 8 9 6 43 7 10 5 72 9 60
0 21 2 28 1 26 3 91 4 58 9 90 6 43 8 64 5 39 7 93
1 50 2 60 0 51 4 90 3 93 7 20 9 33 8 27 6 12 5 89
1 21 3 3 2 47 4 34 0 53 9 67 8 8 5 68 7 1 6 71
3 57 4 26 2 36 0 48 1 11 9 44 7 25 5 30 8 92 6 57
1 20 0 20 4 6 3 74 2 48 9 77 8 15 5 80 7 27 6 10
3 71 1 40 0 86 2 23 4 29 7 99 8 56 6 100 9 77 5 28
4 83 0 61 3 27 1 86 2 99 7 31 5 60 8 40 9 84 6 26
4 68 1 94 3 46 2 60 0 33 7 46 5 86 9 63 6 70 8 89
4 33 1 13 2 91 3 27 0 38 8 82 7 31 6 23 9 27 5 87
4 58 3 30 0 24 2 12 1 38 8 2 9 37 5 59 6 37 7 36
2 62 1 47 4 5 3 39 0 75 7 60 9 65 8 61 6 77 5 31
4 100 0 21 1 53 3 74 2 3 8 34 6 6 7 91 9 80 5 28
1 8 0 3 2 88 3 54 4 18 9 4 6 34 5 54 8 59 7 42
3 33 4 72 0 83 2 17 1 23 6 24 8 60 9 96 7 78 5 70
4 63 2 36 3 70 0 97 1 99 6 71 9 92 5 41 8 73 7 97
2 28 1 37 4 24 0 30 3 55 8 38 5 9 9 77 7 17 6 51
3 15 0 46 2 14 4 18 1 99 9 48 6 41 5 10 7 47 8 80
4 89 3 78 2 51 1 63 0 29 7 70 9 7 5 14 8 84 6 32
4 26 1 69 2 92 3 15 0 23 8 42 6 95 5 47 9 83 7 56
1 38 2 44 3 47 4 23 0 10 9 63 7 65 6 21 5 70 8 56
3 42 4 85 1 29 0 35 2 66 9 46 8 25 5 90 7 85 6 75
3 99 0 46 4 74 2 96 1 48 5 52 6 13 7 88 8 4 9 30
1 15 3 80 4 47 2 25 0 8 9 61 7 70 8 23 6 93 5 5
0 90 2 51 3 66 4 5 1 86 5 59 6 97 9 28 7 85 8 9

0 59 1 50 4 40 3 23 2 93 7 61 9 96 8 63 6 34 5 14
1 62 2 72 4 30 0 21 3 15 5 77 6 13 7 2 8 22 9 22
2 20 4 14 3 85 1 4 0 2 9 33 7 90 5 48 8 90 6 62
0 49 3 49 4 46 1 89 2 64 9 72 8 6 5 83 6 13 7 66
4 74 1 55 2 73 0 25 3 16 7 19 9 38 6 22 5 26 8 63
3 13 2 96 1 8 0 15 4 97 6 95 7 2 5 66 8 57 9 46
4 73 1 97 3 39 0 22 2 90 9 64 6 65 8 31 5 98 7 85
3 43 2 67 0 38 1 77 4 11 7 61 5 7 9 95 8 97 6 69
0 35 2 68 1 5 3 46 4 4 7 51 6 44 5 58 9 69 8 98
2 68 1 81 0 2 3 4 4 59 9 53 8 69 5 69 6 14 7 21

Appendix 6. CSP Crossover and mutation

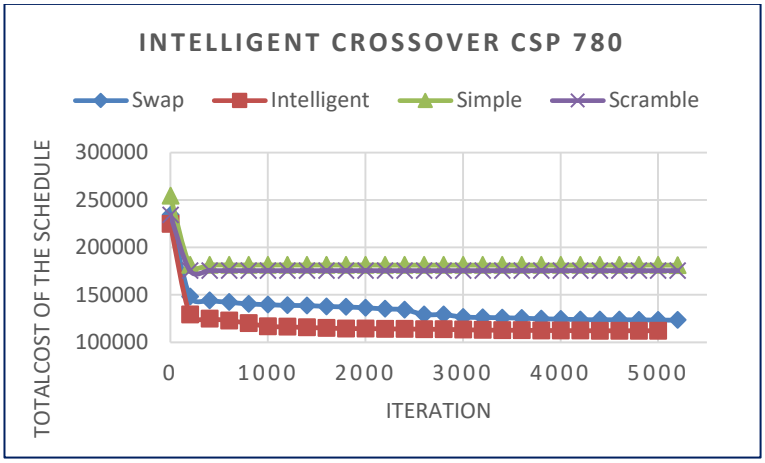


Figure 136 Performance of the intelligent crossover with mutations on the small data set

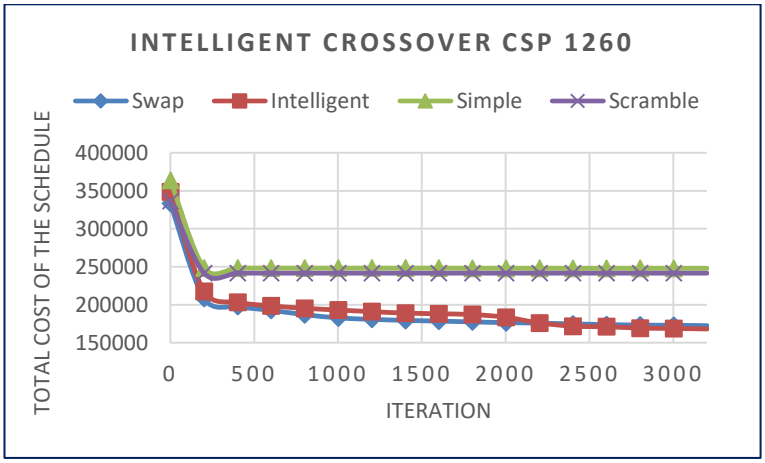


Figure 137 Performance of the intelligent crossover with different mutations on the medium data set

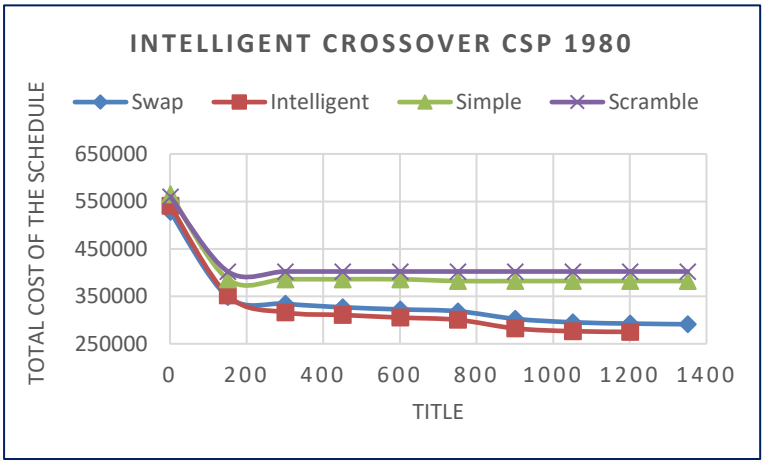


Figure 138 Performance of the intelligent crossover with different mutations on the large data set

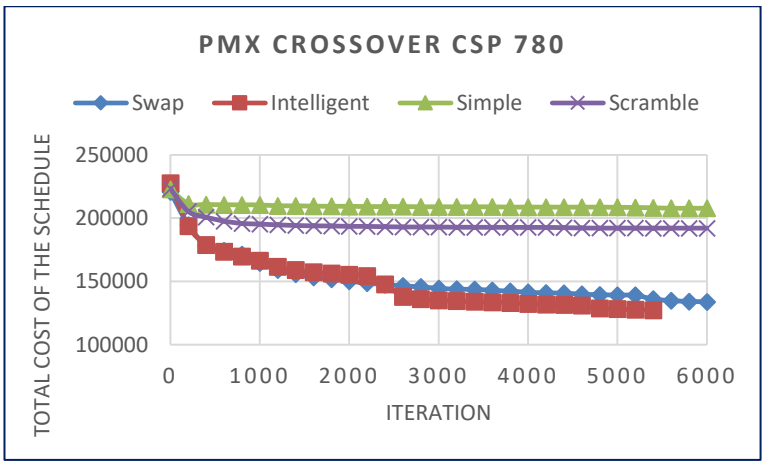


Figure 139 Performance of the PMX crossover with different mutations on the small data set

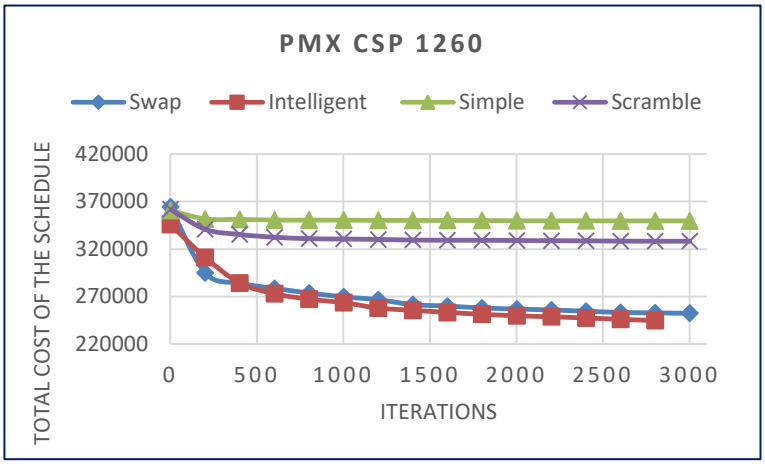


Figure 140 Performance of the PMX crossover with different mutations on the medium data set

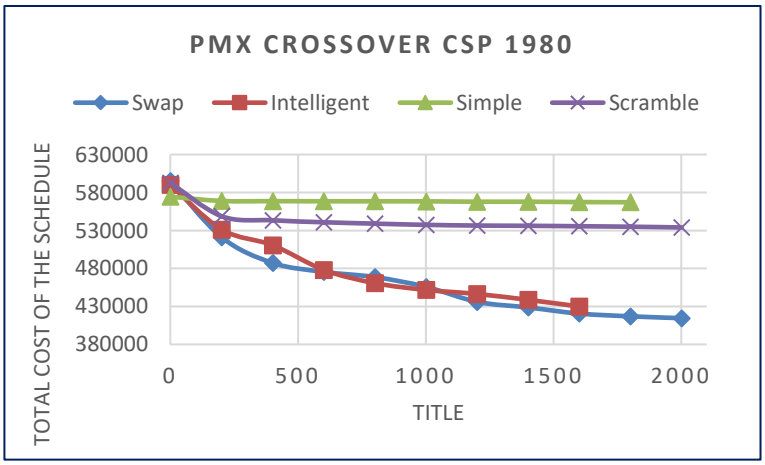


Figure 141 Performance of the PMX crossover with different mutations on the medium data set

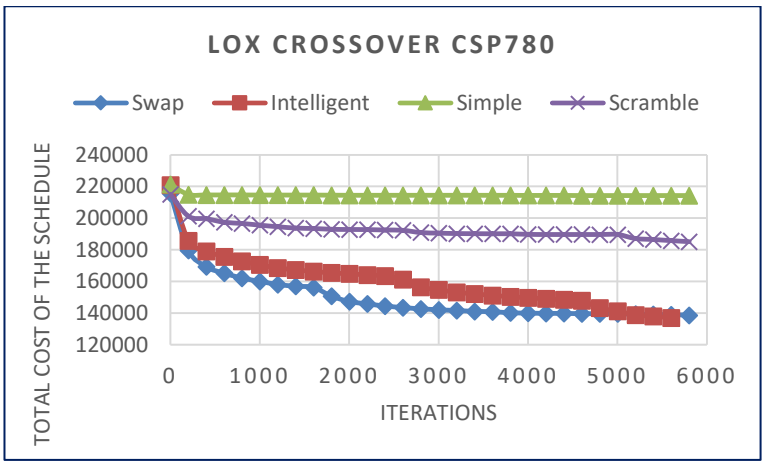


Figure 142 Performance of the LOX crossover with different mutations on the small data set

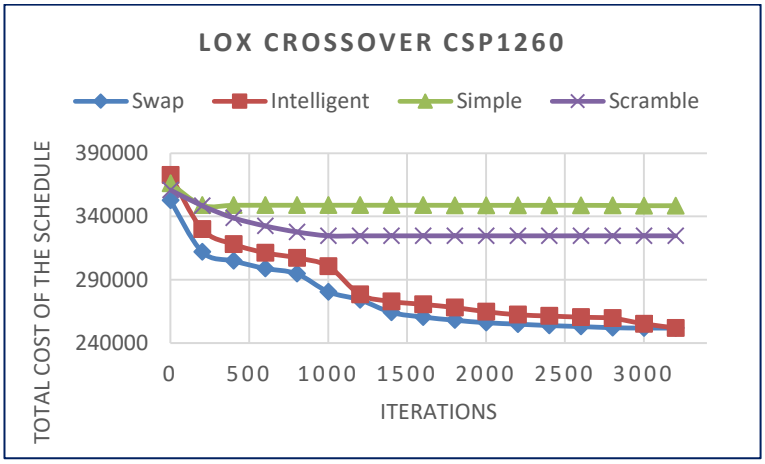


Figure 143 Performance of the LOX crossover with different mutations on the medium data set

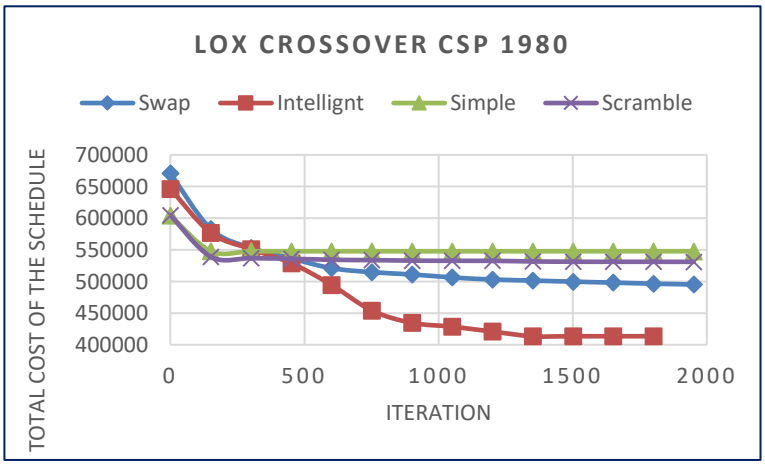


Figure 144 Performance of the LOX crossover with different mutations on the large data set

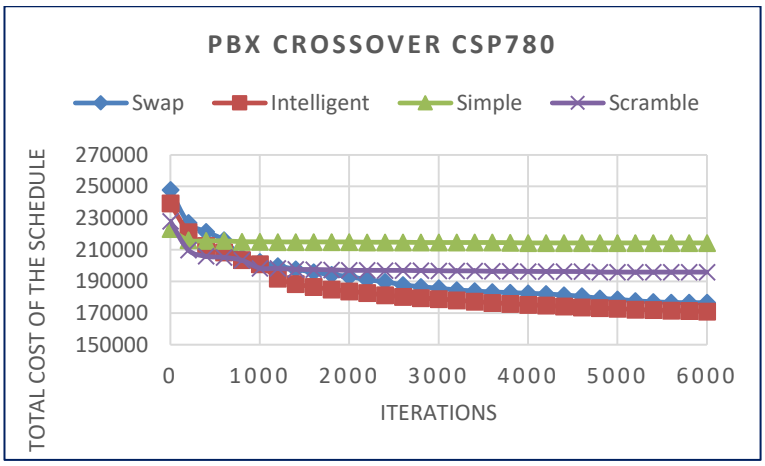


Figure 145 Performance of the PBX crossover with different mutations on the small data set

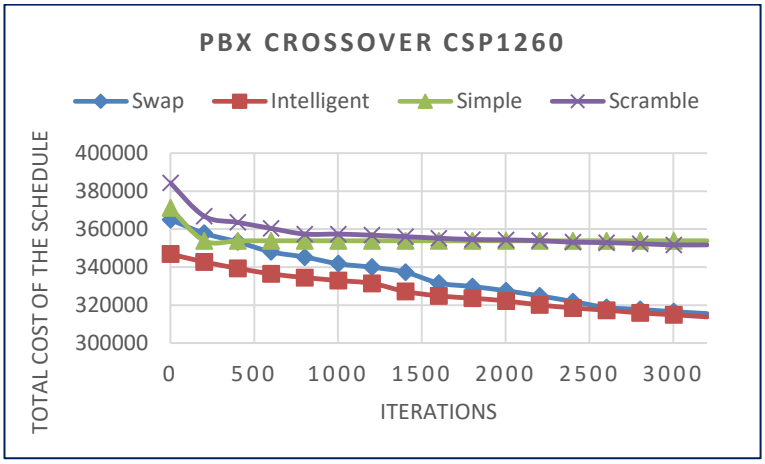


Figure 146 Performance of the PBX crossover with different mutations on the medium data set

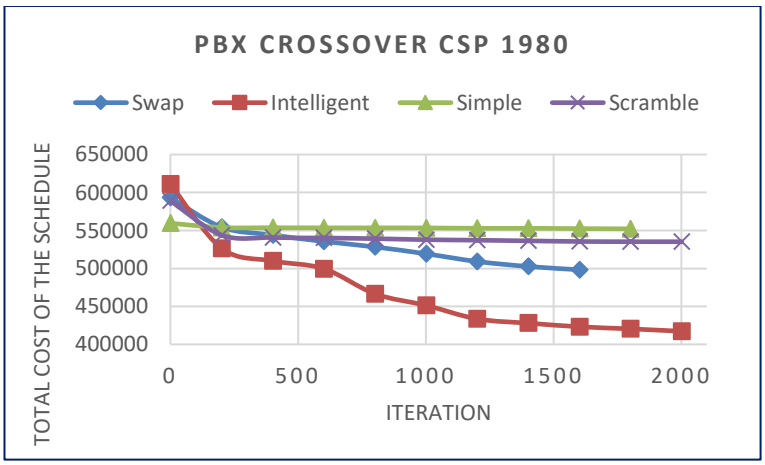


Figure 147 Performance of the PBX crossover with different mutations on the large data set

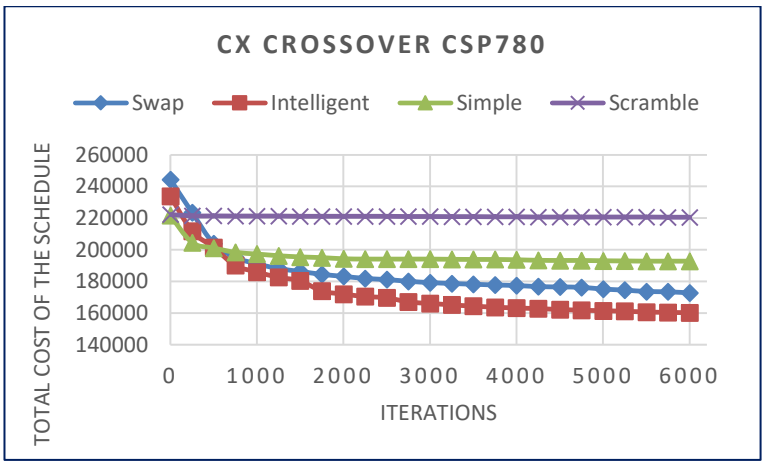


Figure 148 Performance of the CX crossover with different mutations on the small data set

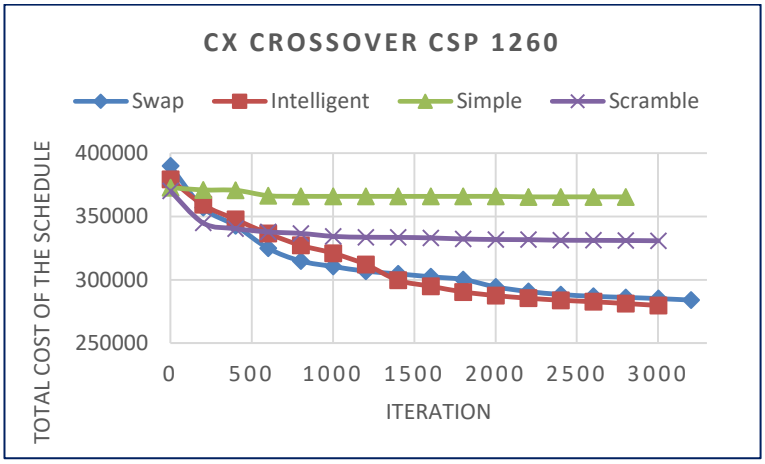


Figure 149 Performance of the CX crossover with different mutations on the medium data set

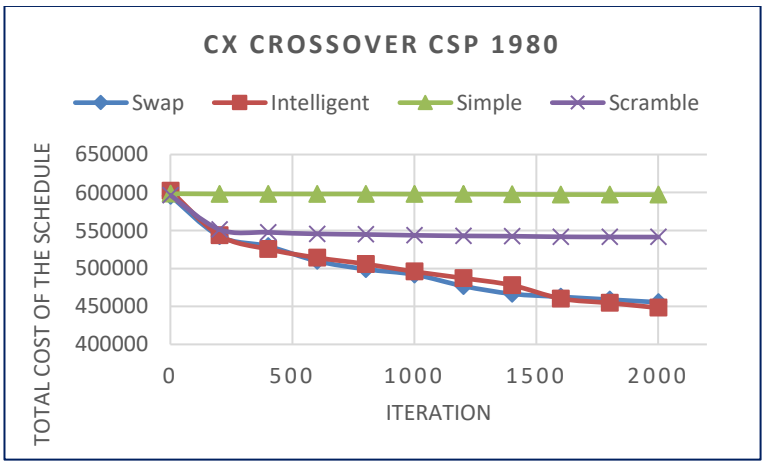


Figure 150 Performance of the CX crossover with different mutations on the large data set

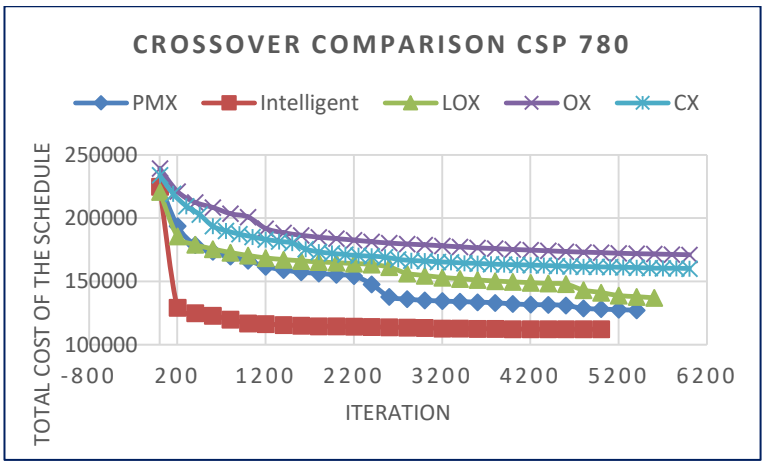


Figure 151 Crossover comparison CSP 780 with Intelligent mutation

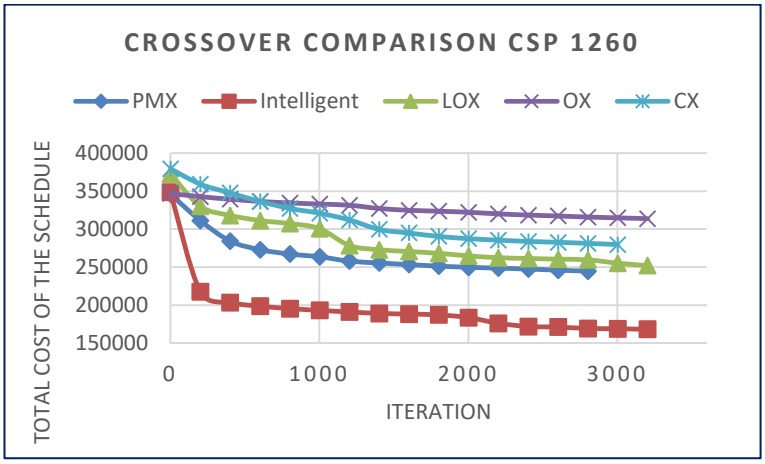


Figure 152 Crossover comparison CSP 1260 with intelligent mutation

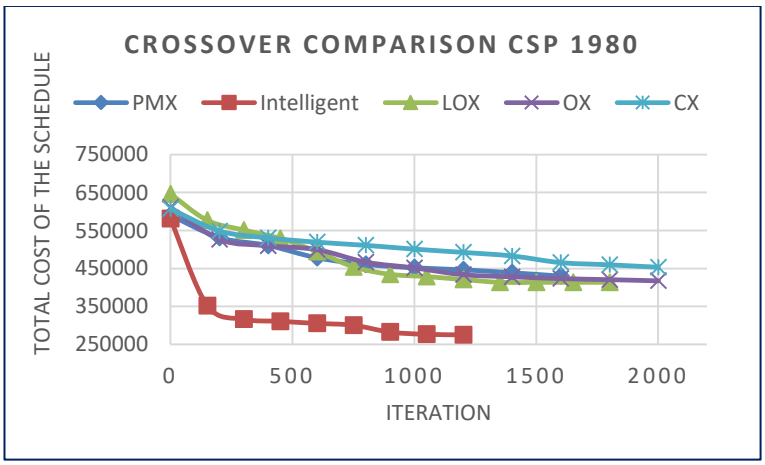


Figure 153 Crossover comparison CSP 1980 with intelligent mutation

Appendix 7. JSSP: Crossover and mutation JSSP

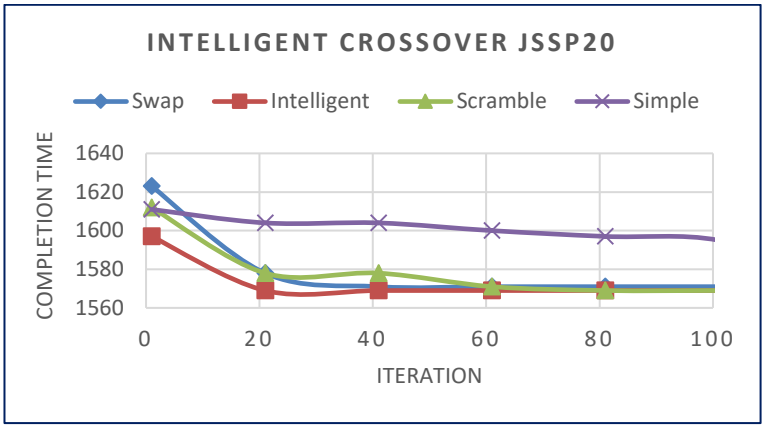


Figure 154 Performance of intelligent crossover on small JSSP data set

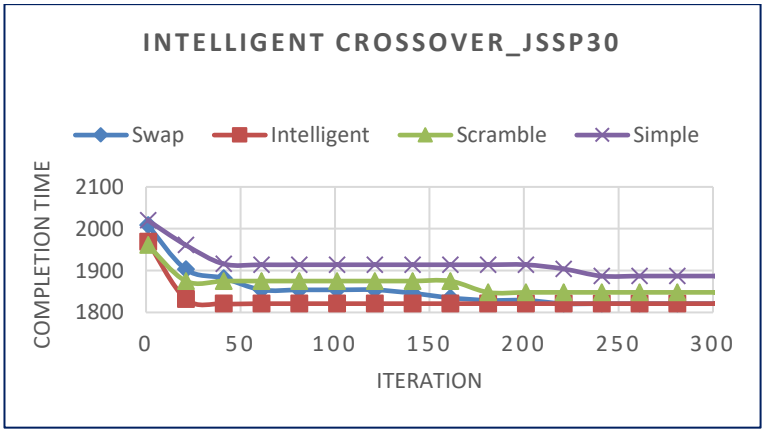


Figure 155 Performance of intelligent crossover of medium JSSP data set

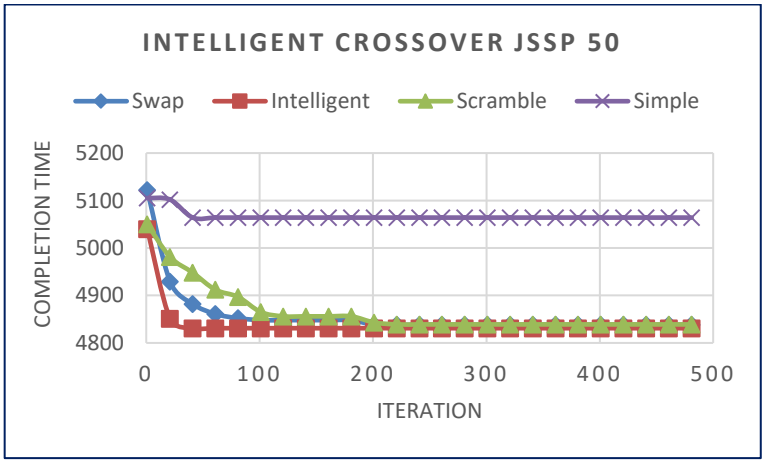


Figure 156 Performance of intelligent crossover on large JSSP data set

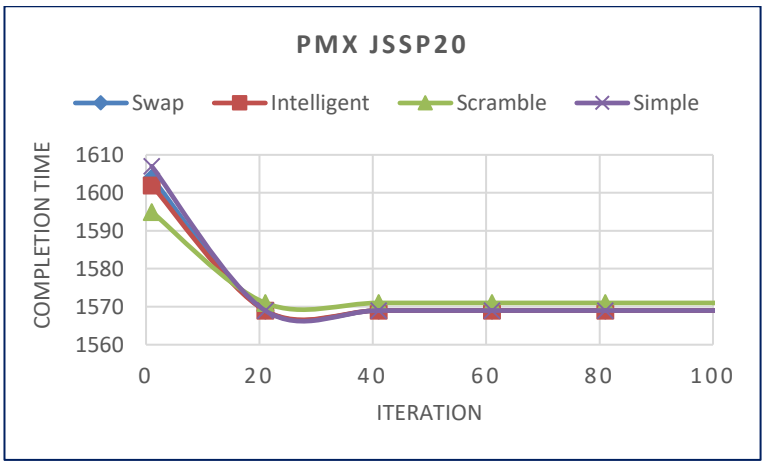


Figure 157 Performance of PMX crossover on small JSSP data set

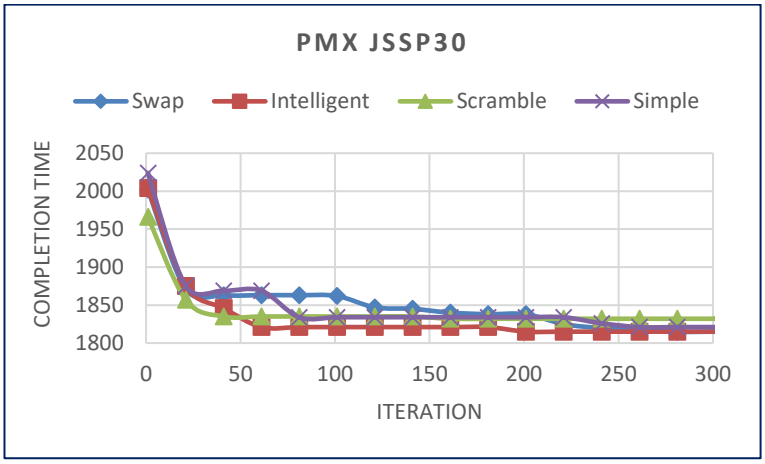


Figure 158 Performance of PMX crossover on medium JSSP data set

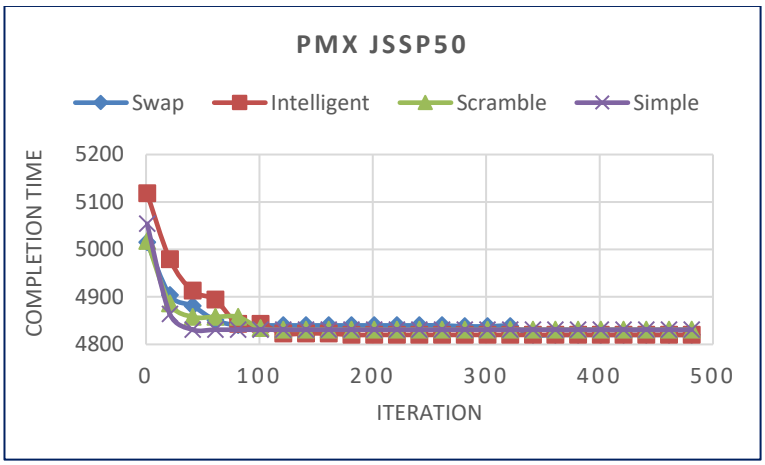


Figure 159 Performance of PMX crossover on large JSSP data set

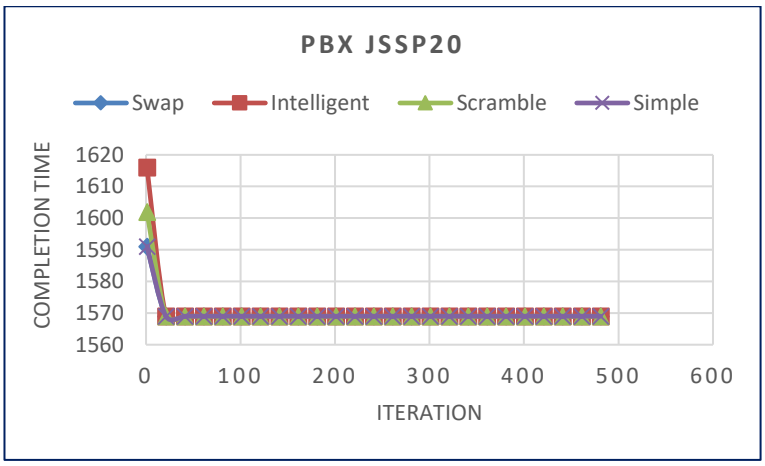


Figure 160 Performance of PBX crossover on a small data set

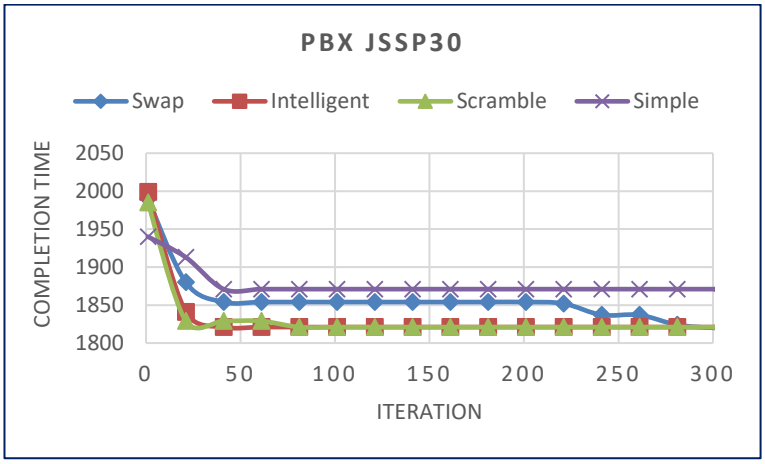


Figure 161 Performance of PBX crossover on a medium data set

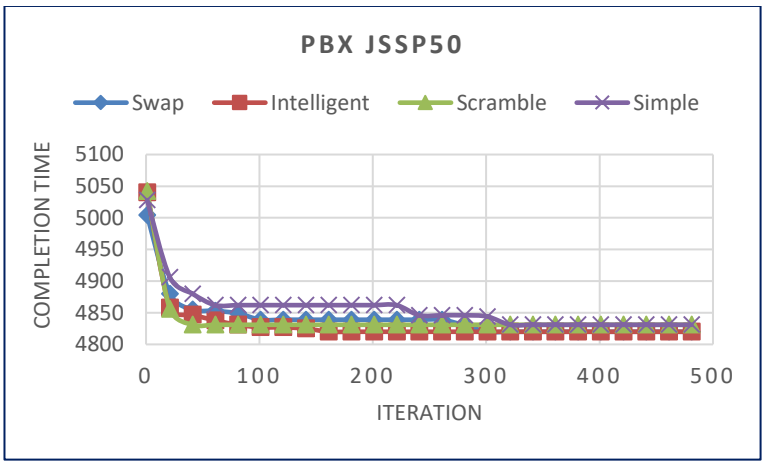


Figure 162 Performance of PBX crossover on a medium data set

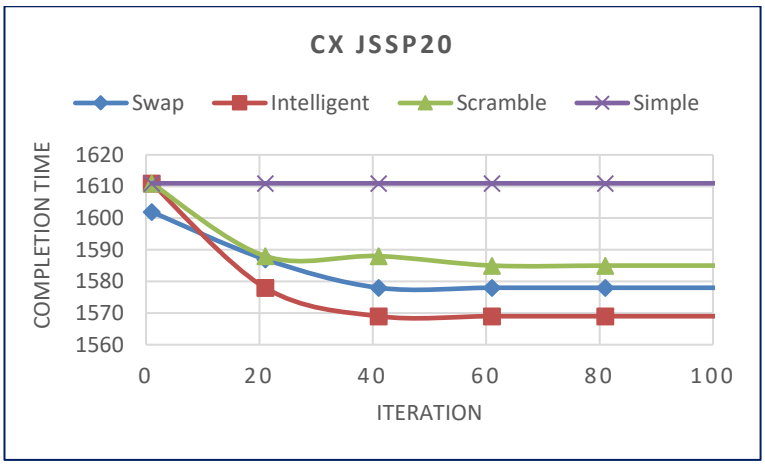


Figure 163 Performance of CX crossover on a small data set

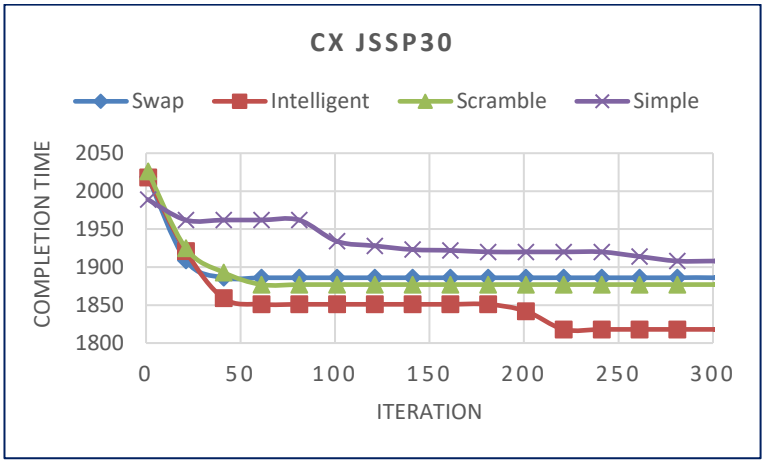


Figure 164 Performance of CX crossover on a medium JSSP data set

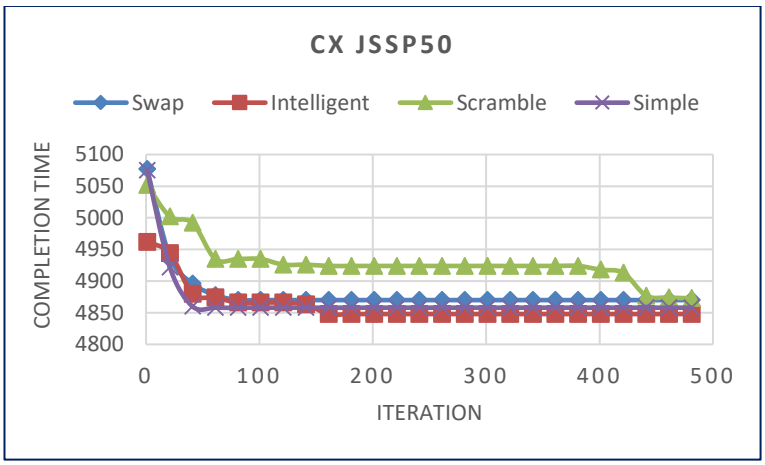


Figure 165 Performance of CX crossover on a medium JSSP data set

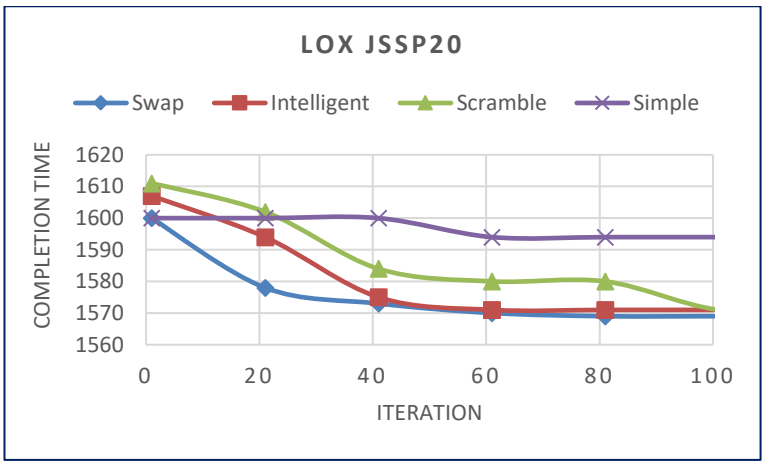


Figure 166 Performance of LOX crossover on a small JSSP data set

Figure 167 Performance of LOX crossover on a medium JSSP data set

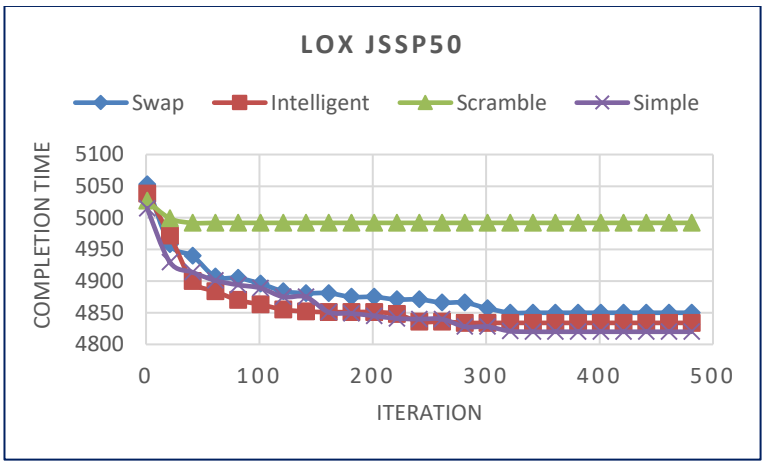


Figure 168 Performance of LOX crossover on a large JSSP data set

Figure 169 Performance of crossovers on the small size of JSSP

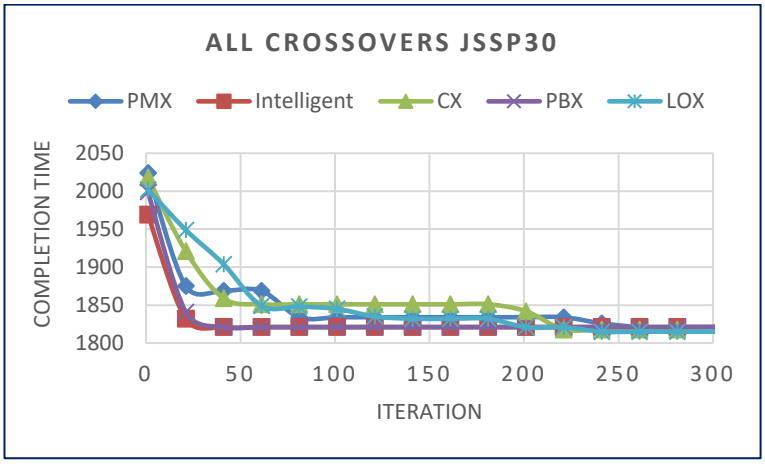


Figure 170 Performance of crossovers on the medium size of JSSP

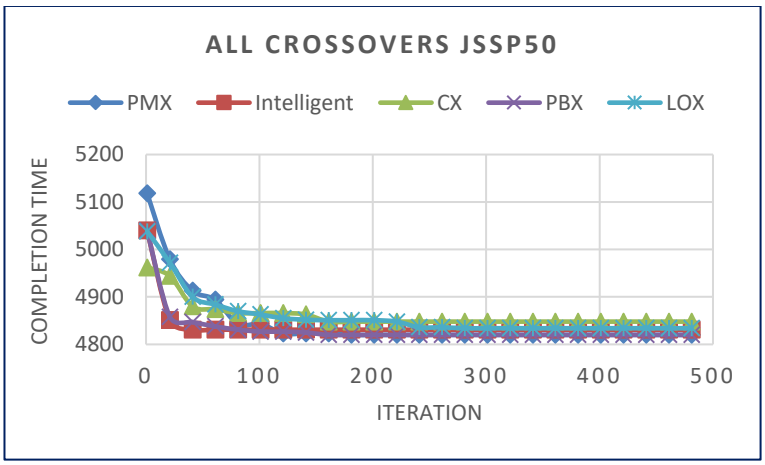


Figure 171 Performance of crossovers on the large size of JSSP

Appendix 8. Driver Evolution

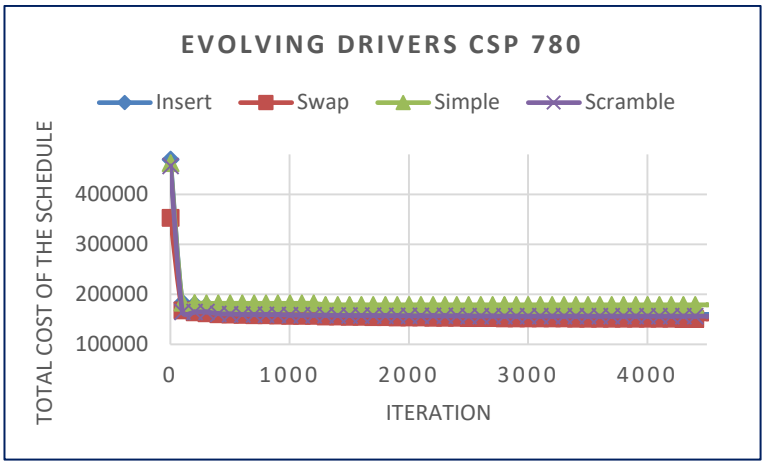


Figure 172 Driver Evolution on a small data set

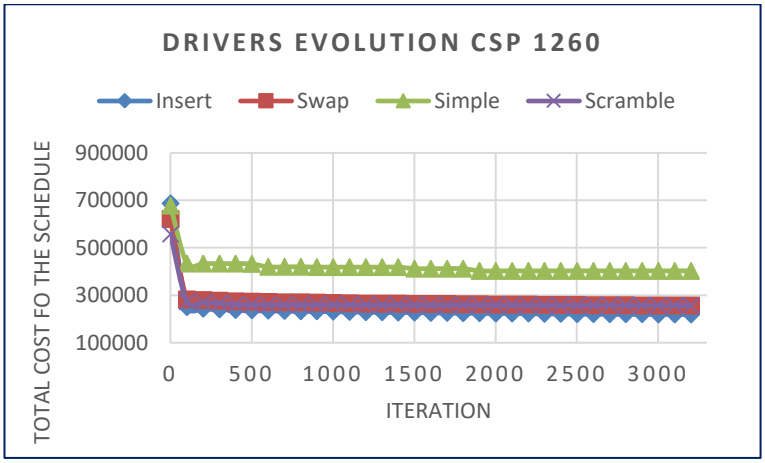


Figure 173 Driver Evolution on a medium data set

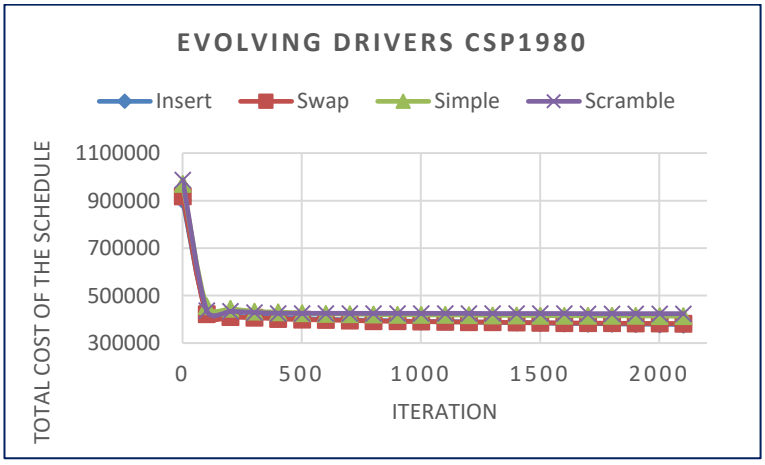


Figure 174 Driver Evolution on the large data sets

Appendix 9. Nearest Driver

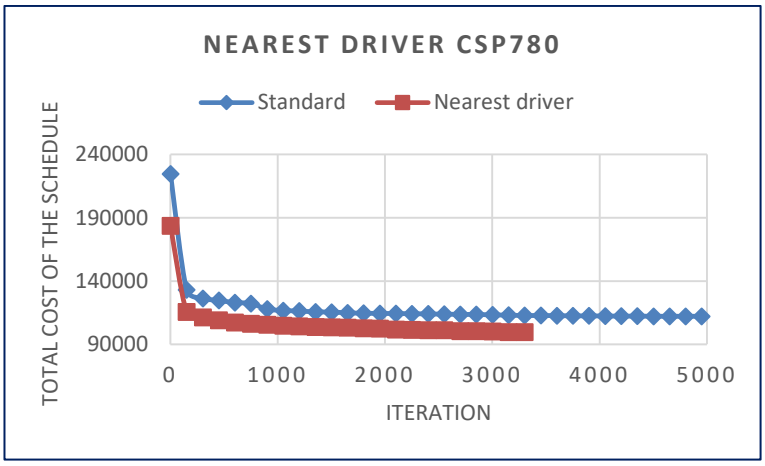


Figure 175 Nearest Driver evolution process on the small data set

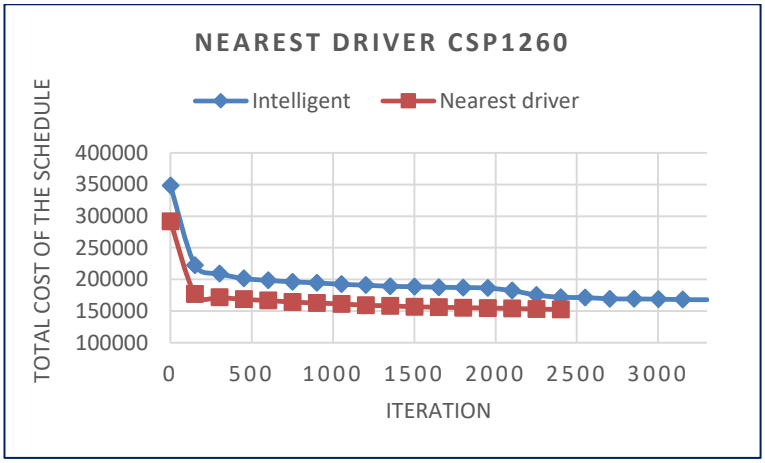


Figure 176 Nearest Driver evolution process on the medium data set

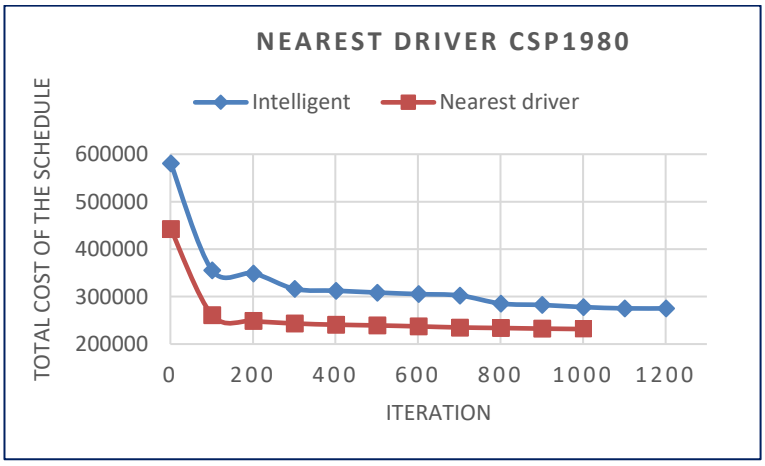


Figure 177 Nearest Driver evolution process on the large data set

Appendix 10. Process of evolution of real data schedule

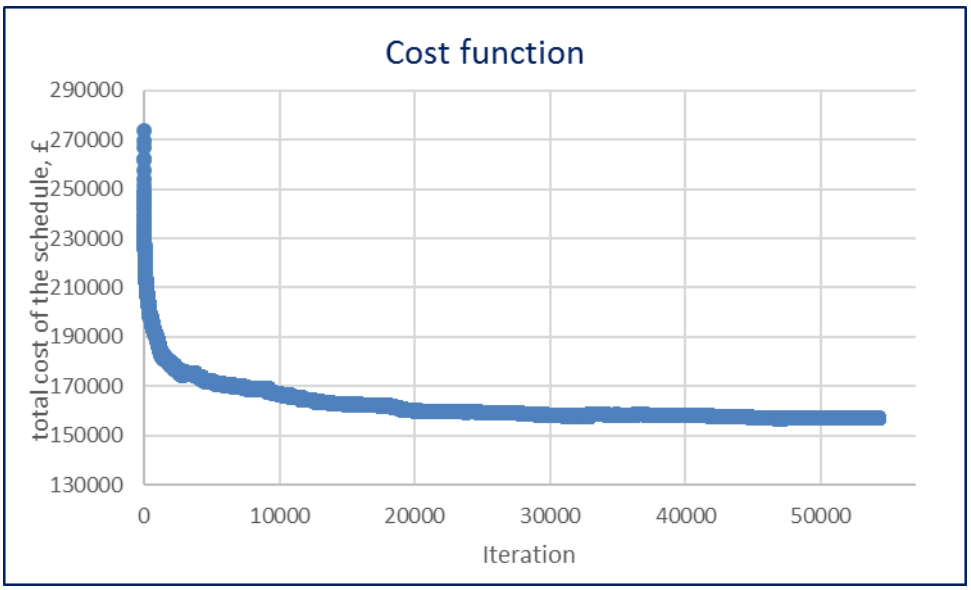


Figure 178 Evolution process: Total Cost of the Schedule

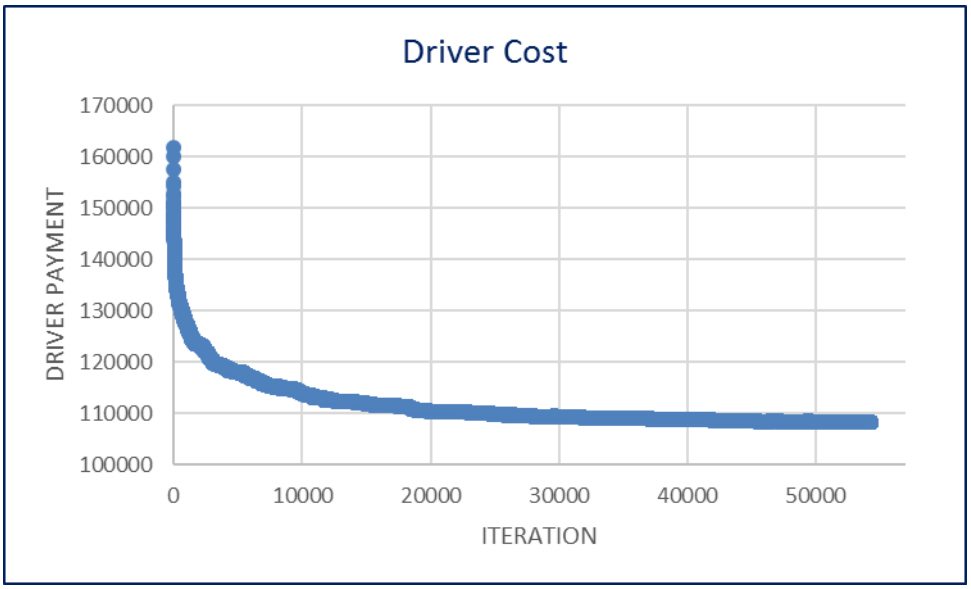


Figure 179 Evolution process: Driver Cost

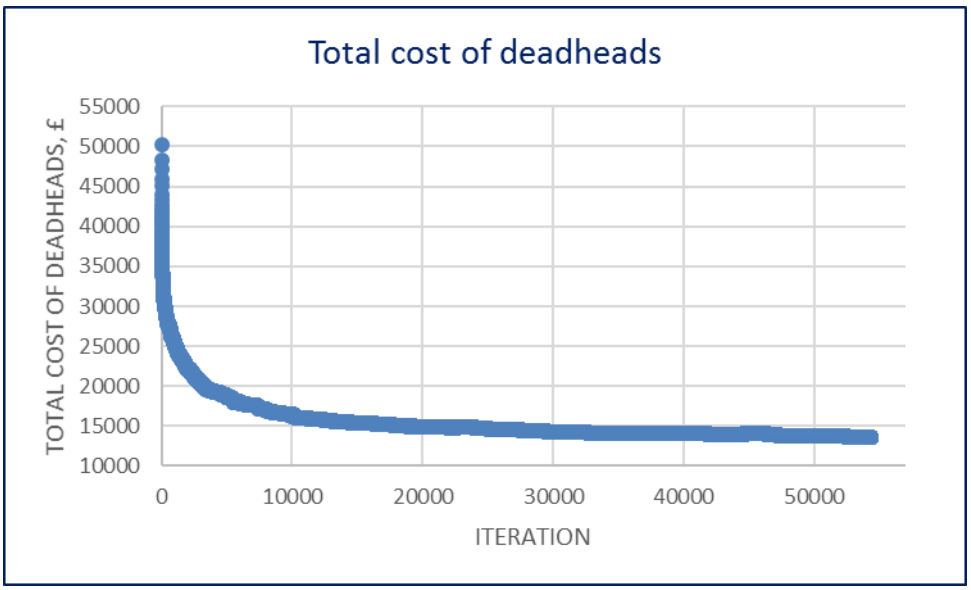


Figure 180 Evolution process: Total Cost of Deadheads

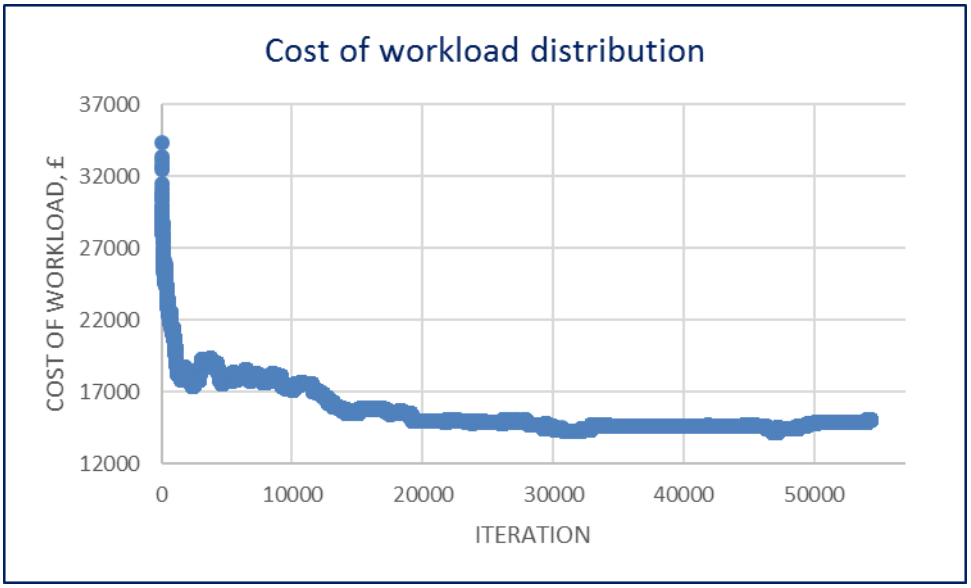


Figure 181 Evolution process: Cost of workload distribution

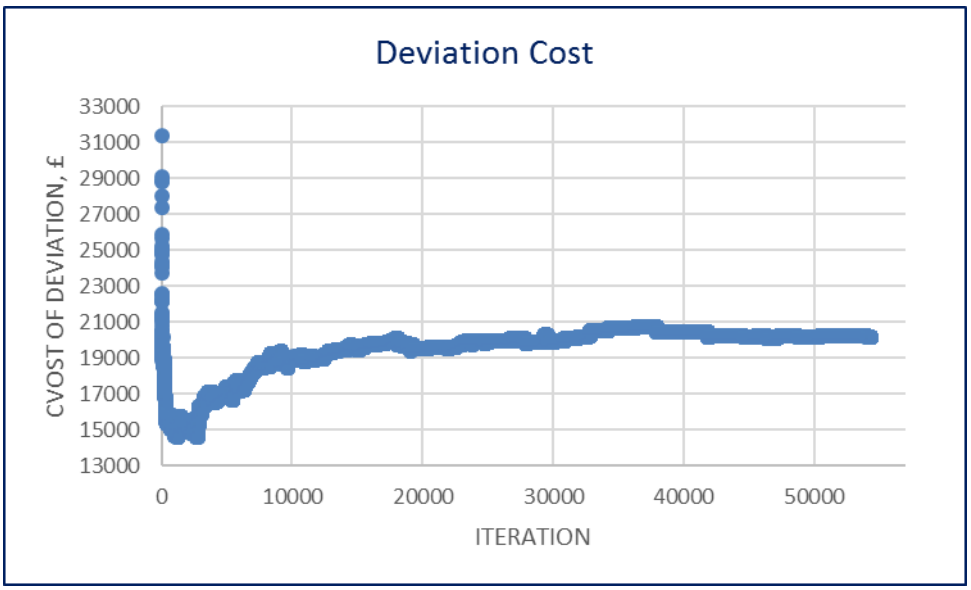


Figure 182 Evolution process: Total Cost of the deviation of the shift length

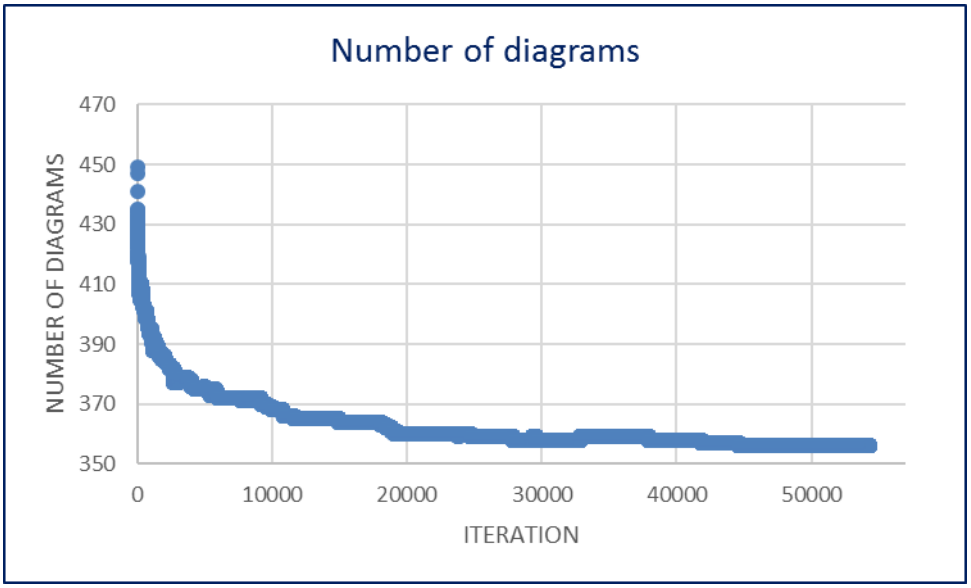


Figure 183 Evolution process: Number of Diagrams

Appendix 11. Evaluated diagrams

Diagram 1

Head Code/Passenger Train Company	Activity	Start	End	Origin	Destination
	book on	11:33	11:43	PeterboroughDepot	PeterboroughDepot
	walk	11:43	11:47	PeterboroughDepot	PeterboroughSigP800
	Relieve driver	14			
	MOB	11:47	12:03	PeterboroughSigP800	PeterboroughSigP800
6L43	driving	12:03	12:30	PeterboroughSigP800	MarchUpRS
	driving	12:39	13:09	MarchUpRS	Ely
	OP	13:10	13:41	Ely	KennettRedlandGF
	FS	13:41	13:45	KennettRedlandGF	KennettLafargeSdgs
	walk	13:45	14:07	KennettLafargeSdgs	KennettTrainStation
Abellio Greater Anglia	PASS	14:07	14:39	KennettTrainStation	Cambridge
Cross Country	PASS	**15:01**	15:14	Cambridge	Ely
	walk	15:14	15:24	Ely	ElyPapworthSidings(PotterGp)
	break	15:24	16:58	ElyPapworthSidings(PotterGp)	ElyPapworthSidings(PotterGp)
	Relieve driver	989			
6M86	DORR	16:58	18:17	ElyPapworthSidings(PotterGp)	PeterboroughSigP451
	Relieved by the driver	756			
	walk	18:17	18:21	PeterboroughSigP451	PeterboroughDepot
	book off	18:21	18:31	PeterboroughDepot	PeterboroughDepot

Diagram2

Head Code	Activity	Start	End	Origin	Destination
	book on	11:20	11:30	ImminghamDepot	ImminghamDepot
	walk	11:30	11:35	ImminghamDepot	ImminghamHumberRefinery
	ATT	11:35	11:40	ImminghamHumberRefinery	ImminghamHumberRefinery
6M00	driving	11:40	13:55	ImminghamHumberRefinery	Nottingham
	Relieved by driver number: 344				
	Relieve: driver number:	802			
	Break	14:00	16:18	Nottingham	Nottingham
6E41	driving	16:18	18:21	Nottingham	ImminghamLindseyRefinery
0D41	DET	18:21	18:35	ImminghamLindseyRefinery	ImminghamLindseyRefinery
	Relieved by driver number: 12				
	walk	18:35	18:42	ImminghamLindseyRefinery	ImminghamHITCoalLdgFacility
	Relieve driver	1001			
6M03	LOAD	18:45	18:55	ImminghamHITCoalLdgFacility	ImminghamHITCoalLdgFacility
	walk	18:55	19:10	ImminghamHITCoalLdgFacility	ImminghamHITCoalLdgFacility
0D41	driving	19:10	19:31	ImminghamLindseyRefinery	ImminghamLindseyRefinery
	walk	19:31	19:34	ImminghamLindseyRefinery	ImminghamSS
	Break	19:34	20:50	ImminghamSS	ImminghamSS
	Relieve driver	921			
0E08	FS	20:50	21:00	ImminghamSS	ImminghamTMD
	DISP	21:00	21:10	ImminghamTMD	ImminghamTMD
	Relieved by driver number:	67			
	walk	21:10	21:15	ImminghamTMD	ImminghamDepot
	book off	21:15	21:25	ImminghamDepot	ImminghamDepot
Duration	10:05				

Diagram3					
Head Code	Activity	Start	End	Origin	Destination
	book on	07:04	07:14	NewportDepot	NewportDepot
	walk	07:14	07:17	NewportDepot	NewportSig.NT1273
6V49	driving	07:17	07:41	NewportSig.NT1273	LeckwithLoopNorthJn
6V49	driving	07:41	08:21	LeckwithLoopNorthJn	MargamKnuckleYard
	taxi	08:21	08:48	MargamKnuckleYard	Port Talbot Parkway
Arriva Trains Wales	PASS	08:48	09:01	Port Talbot Parkway	Bridgend
	taxi	09:01	09:20	Bridgend	AberthawReceptionSdgs
	MOB	09:20	09:25	AberthawReceptionSdgs	AberthawReceptionSdgs
4F69	driving	09:25	10:26	AberthawReceptionSdgs	NewportA.D.JnSdgs
	OP	10:26	10:32	NewportA.D.JnSdgs	NewportA.D.JnSdgs
	DORR	10:32	12:35	NewportA.D.JnSdgs	NewportCourtybellaSdg
	OP	12:35	12:43	NewportCourtybellaSdg	NewportA.D.JnSdgs
	Relieved by driver number:	39			
	walk	12:43	12:46	NewportA.D.JnSdgs	NewportDepot
	book off	12:46	12:56	NewportDepot	NewportDepot
Duration	05:52				

Diagram 4					
Head Code	Activity	Start	End	Origin	Destination
	book on	06:25	06:35	RotherhamDepot	RotherhamDepot
	walk	06:35	06:45	RotherhamDepot	MasboroughFD
	Relieve driver	516			
6E20	RM	06:45	06:50	MasboroughFD	MasboroughSSJn
	PR	06:50	08:05	MasboroughSSJn	ScunthorpeTrentT.C
	Break	08:05	08:42	ScunthorpeTrentT.C	ScunthorpeTrentT.C
	driving	08:42	08:48	ScunthorpeTrentT.C	ScunthorpeSig319
0E20	DET	08:48	08:58	ScunthorpeSig319	ImminghamSS
	Relieved by driver number: 13				
	Break	09:00	10:09	ImminghamSS	ImminghamSS
	taxi	10:09	10:57	ImminghamSS	GooleDocks
	Break	10:57	12:30	GooleDocks	GooleDocks
	MOB	12:30	12:35	GooleDocks	GooleDocks
6J94	FS	12:35	12:59	GooleDocks	GooleUpGoodsLoop
	break	13:00	13:40	GooleUpGoodsLoop	GooleUpGoodsLoop
	RR	13:40	14:23	GooleUpGoodsLoop	HexthorpeJn
	walk	14:23	15:17	HexthorpeJn	MasboroughFD
0J94	DET	15:17	15:22	MasboroughFD	MasboroughLHS
	DISP	15:22	15:32	MasboroughFD	MasboroughFD
	walk	15:32	15:37	MasboroughLHS	RotherhamDepot
	book off	15:37	15:47	RotherhamDepot	RotherhamDepot
Duration:	09:22				

Diagram 5					
Head Code	Activity	Start	End	Origin	Destination
	book on	08:47	08:57	MossendDepot	MossendDepot
	walk	08:57	09:03	MossendDepot	MossendDownYard
6G25	ATT	09:03	09:23	MossendDownYard	MossendDownYard
	driving	09:23	10:48	MossendDownYard	KincardineLC
	Relieved by driver 721				
	taxi	10:48	11:22	KincardineLC	Cumbernauld
ScotRail	PASS	11:22	11:33	Cumbernauld	Whifflet
	taxi	11:33	11:59	Whifflet [WFF]	MossendDepot
	break	11:59	14:35	MossendDepot	MossendDepot
	walk	14:35	14:45	MossendDepot	MossendLHS
	PL	14:45	15:00	MossendLHS	MossendLHS
	PU	15:00	15:15	MossendLHS	MossendLHS
	WAR	15:15	16:00	MossendLHS	MossendLHS
	break	16:00	16:24	MossendLHS	MossendLHS
	PL	16:24	16:39	MossendLHS	MossendLHS
	PU	16:39	16:54	MossendLHS	MossendLHS
	walk	16:54	17:00	MossendLHS	MossendWestYardLHS
	break	17:00	17:43	MossendWestYardLHS	MossendWestYardLHS
	Relieve driver 670				
0S94	PL	17:43	17:58	MossendWestYardLHS	MossendWestYardLHS
0S94	driving	17:58	18:08	MossendWestYardLHS	MossendDownYard
	walk	18:08	18:14	MossendDownYard	MossendDepot
	book off	18:14	18:24	MossendDepot	MossendDepot
Duration	09:37				

Appendix 12. Publication and award

Rail-Freight Crew Scheduling with a Genetic Algorithm

E. Khmeleva[1], A. A. Hopgood[1], L. Tipi[1] and M. Shahidan[1]

Abstract This article presents a novel genetic algorithm designed for the solution of the Crew Scheduling Problem (CSP) in the rail-freight industry. CSP is the task of assigning drivers to a sequence of train trips while ensuring that no driver's schedule exceeds the permitted working hours, that each driver starts and finishes their day's work at the same location, and that no train routes are left without a driver. Real-life CSPs are extremely complex due to the large number of trips, opportunities to use other means of transportation, and numerous government regulations and trade union agreements. CSP is usually modelled as a set-covering problem and solved with linear programming methods. However, the sheer volume of data makes the application of conventional techniques computationally expensive, while existing genetic algorithms often struggle to handle the large number of constraints. A genetic algorithm is presented that overcomes these challenges by using an indirect chromosome representation and decoding procedure. Experiments using real schedules on the UK national rail network show that the algorithm provides an effective solution within a faster timeframe than alternative approaches.

1 Introduction

While international trade continues to expand, businesses are striving to increase reliability and reduce their environmental impact. As a result, demand for rail freight increases every year and rail-freight carriers attempt to maximize their efficiency. The crew cost constitutes 20-25% of the total rail-freight operating cost and is second only to cost of fuel. Therefore even a small improvement in the scheduling processes can save a company millions of pounds a year.

The CSP in the rail-freight industry is the problem of constructing a schedule for a train driver. Each schedule contains instructions for the driver of what he or she should do on a particular day. Within the industry, the driver's schedule is called a *diagram*. Each diagram should cover all the trains driven by a driver in a given day. It must start and end at the same station and obey all labour laws and trade union agreements. These rules regulate the maximum diagram duration, maximum continuous and aggregate driving time in a diagram, and minimum break time.

All drivers are located in *depots* where they start and finish their work. Depots are distributed fairly evenly across the UK. Sometimes in order to connect two trips that finish and start at different locations, a driver has to travel on a passenger train, taxi

[1] Sheffield Business School, Sheffield Hallam University, Howard Street, Sheffield S1 1WB, UK.

e.khmeleva@shu.ac.uk; a.hopgood@shu.ac.uk; l.tipi@shu.ac.uk; m.shahidan@shu.ac.uk

or a freight train driven by another driver. The situation of a driver travelling as a passenger while on duty is called *deadheading*. The cost of deadheading varies and depends on the means of transportation and business agreements between operating companies. Despite the potential cost, deadheading is sometimes inevitable and it can benefit the overall schedule [1].

Due to employment contract terms, the drivers are paid the same hourly rate for any time spent on duty regardless of the number of hours they have actually been driving the train. Moreover, in accordance with collectively bargained contracts, each driver has a fixed number of working hours per year, so the company is obliged to pay for all the stated hours in full even if some of the hours are not utilised. Paid additional overtime hours can be worked at the driver's discretion. Thus it is in the best interests of the company to utilize the agreed driving hours in the most efficient and economical way.

Taking all of this into consideration, the operational objectives for the diagrams are:

1. Minimize a number of unused and excess contract hours at the end of the year with a minimum spread of durations of the diagrams. All diagrams will therefore be of duration close to the average 8.5 hours, i.e. the annual contract hours divided by the number of the working days.

$$T_{diagram} = T_{driving} + T_{deadheading} + T_{break} + T_{idle}$$

$$T_{diagram} \rightarrow T_{average}$$

2. Maximize the throttle time, i.e. the proportion of the work shift that is actually spent driving a train. It excludes time for deadheading and waiting between trips.

$$Throttle\ time = \frac{T_{driving}}{T_{diagram}}$$

2 Approaches to crew scheduling

The CSP is usually solved in two stages. At the first stage, all possible diagrams satisfying the industrial constraints are enumerated. At the second stage, only the set of diagrams that covers the entire schedule in the most cost-effective way is identified. Diagrams are usually modelled as binary vectors (Figure 1) where '1' denotes that the trip *i* is included in the diagram *j*, otherwise '0' is inserted. Each diagram has its own cost. The deadhead journeys are displayed by including the same trip in more than one diagram. In the rest of the article the terms diagram and column will be used interchangeably.

	Diagram1	Diagram2	Diagram3	Diagram4
Trip1	1	0	0	1
Trip2	0	1	1	0
Trip3	0	1	0	1
Trip4	0	1	0	1
Trip5	1	1	0	0

Figure 1 Diagrams

Although the generation of the diagrams can be performed in a simple and relatively straightforward manner using various graph search and label-setting techniques [2], finding an optimal set of diagrams may be highly time-consuming. The problem boils down to the solution of the 0–1 integer combinatorial optimization set covering problem (SCP):

$$\text{minimize} \sum_{i=1}^{n} c_j x_j$$

$$\text{subject to} \sum_{i=1}^{n} a_{ij} x_j \geq 1$$

$$x_i \in \{1,0\}$$

$$i = 1,2 \ldots n \text{ trips}$$

$$j = 1,2 \ldots m \text{ diagrams}$$

where a_{ij} is a decision variable indicating whether a trip *i* is included in the diagram *j*, x_j shows if the diagram is included in the schedule, *cj* is the cost of the diagram.

2.1 Branch-and-price

The complete enumeration of all possible diagrams is likely to be impractical due to the large geographical scope of operations, the number of train services, and industry

regulations. Linear programming methods such as branch-and-price [3, 4] have been popular for the solution of medium-sized CSPs in the passenger train and airline industries [5]. These methods usually rely on a column-generation approach, where the main principle is to generate diagrams in the course of the algorithm, rather than having them all constructed a priori. Despite the ability of the algorithm to work with an incomplete set of columns, the column generation method alone does not guarantee an integer solution of SCP. It is usually utilised in conjunction with various branching techniques that are able to find the nearest integer optimal solution. However, this approach is not quite suitable for the CSP in rail freight, where the possible number of diagrams tend to be considerably higher.

2.2 Genetic algorithms

Linear programming (LP) has been used for CSP since the 1960s [6] but genetic algorithms (GAs) were introduced more recently [7]. GAs have been applied either for the production of additional columns as a part of column generation [6] or for the solution of SCP from the set of columns generated prior to the application of a GA [9-12], but there are not yet any reports of them solving both stages of the problem. Since the diagrams are generated outside the GA in advance, the GA cannot change or add new columns. The GA is therefore confined to finding only good combinations from a pre-determined pool of columns.

For the solution of CSP with a GA, chromosomes are normally represented by integer or binary vectors. Integer vector chromosomes contain only the numbers of the diagrams that constitute the schedule. This approach requires knowledge of the minimum number of diagrams in the schedule and this information is usually obtained from the lower bounds. Lower bounds are usually acquired through the solution of LP relaxation for SCP [13]. Since the number of diagrams tends to be higher than the lower bound, Costa et al [14] have suggested the following approach. In the first population the chromosomes have a length equal to the lower bound. Then, if a solution has not been found within a certain number of iterations, the length of the chromosome increases by one. This process repeats until the termination criteria are met.

In the binary vector representation, each gene stands for one diagram. The figure '1' denotes that the diagram is included in the schedule, otherwise it is '0'. Such chromosomes usually consist of several hundred thousand genes, and only around a hundred of them appear in the final solution. The number of diagrams can be unknown and the algorithm is likely to need a large number of iterations in order to solve the problem.

The application of genetic operators often violates the feasibility of the chromosomes, resulting in certain trips being highly over-covered (i.e. more than one driver assigned to the train) or under-covered (i.e. no drivers assigned to the train). One way of

resolving this difficulty is to penalize the chromosome through the fitness function in accordance with the number of constraints that have been violated. However, the development of the penalty parameters can be problematic as in some cases it is impossible to verify them analytically and they are usually designed experimentally [15]. The penalty parameters are therefore data-dependent and likely to be inapplicable to other industries and companies. Moreover, the feasibility of the entire population is not guaranteed and might be achieved only after a large number of iterations.

Another more straightforward approach to maintaining the feasibility is to design heuristic "repair" operators. These operators are based on the principles "REMOVE" and "INSERT". They scan the schedule and remove certain drivers from the over-covered trips and assign those drivers to under-covered journeys [13, 15]. This procedure might have to be repeated several times, leading to high memory consumption and increased computation time.

3 GA-generated crew schedules

3.1 Initial data

The process starts with a user uploading the freight train and driver data (Figure 2). Each train has the following attributes: place of origin, destination, departure and arrival time, type of train and route code. The last two attributes indicate the knowledge that a driver must have in order to operate a particular train. The system also stores information about the drivers, i.e. where each driver is located and his or her traction and route knowledge. The program also captures all the passenger trains and the taxi times between cities (Figure 3).

After all the necessary data have been uploaded, the GA is applied to construct an efficient schedule. The proposed algorithm overcomes the aforementioned challenges through a novel alternative chromosome representation and special decoding procedure. It allows the feasibility of chromosomes to be preserved at each iteration without the application of repair operators. As a result, the computational burden is considerably reduced.

Figure 2 Freight trains and drivers

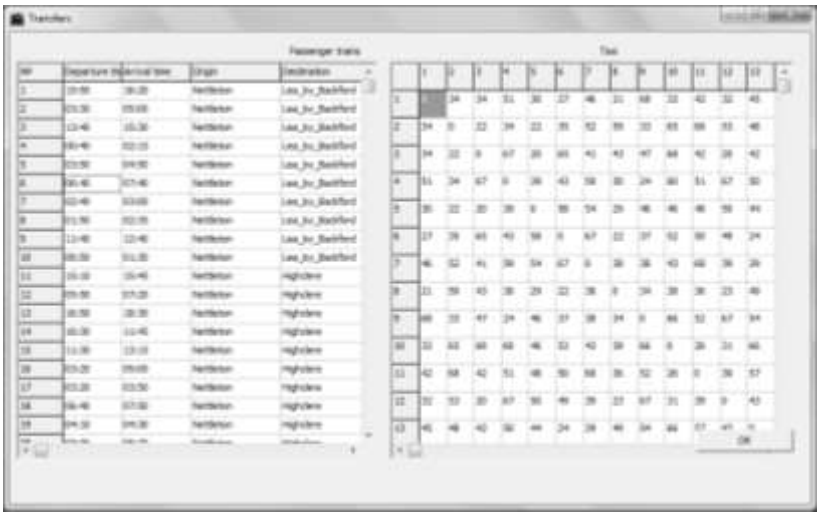


Figure 3 Passenger trains and taxis

3.2 Chromosome representation

The chromosome is represented by a series of integers, where each number stands for the number of the trip (Figure 4). The population of chromosomes is generated at random and then the trips are allocated in series to the diagrams using a specific decoding procedure. This procedure checks if the next trip can be placed after a preexisting one without violation of any restrictions. If it is possible, then it verifies times and locations. If the origin station for the current trip differs from the destination station of the previous trip, then the algorithm first searches for passenger trains and the freight company's own trains that can deliver a driver within the available time slot to the next job location, e.g. Diagram 1, between trips 3 and 8 (Figure 4). If no such trains have been found but there is sufficient interval between the trips, then the algorithm inserts a taxi journey.

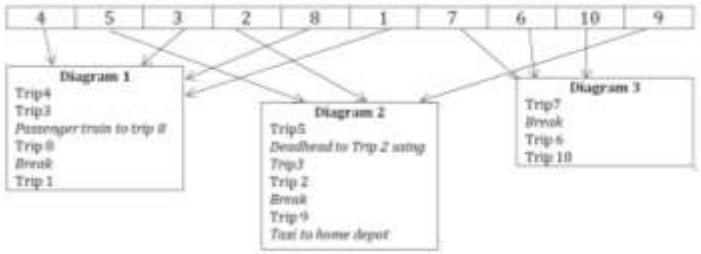


Figure 4 Chromosome representation and decoding procedure

If the trips cannot be placed together, then the procedure checks if one of them can be included into another diagram, otherwise it will create a new diagram. Information regarding driving times and the current duration of the diagrams is stored. Before adding a new trip, the algorithm inserts breaks if necessary. If the time expires and there are no trains to the home depot that a driver can drive, the deadheading activity completes the diagram, as in Diagram2 (Figure 4).

On rare occasions, a few diagrams might be left with only a few trips and a duration that is less than the minimum. This is due to the fact that other drivers are either busy at this time or located at different stations. In order to tackle this problem, a mechanism has been added for finding and assigning a driver from a remote depot with the lowest workload. This approach not only solved the problem of the short diagrams, but also helped in distributing the workload more equally across the depots. After implementation of this procedure, the algorithm has been tested on various data sets including real and randomly generated data. Neither of the chromosomes has been reported to violate the constraint.

The given representation has a visual resemblance to the flight-graph representation suggested by Ozdemir and Mohan [16], but the decoding procedures are different. The flight-graph representation generates trips based on a depth-first graph search, whereas in the proposed GA they are produced at random. Random generation is beneficial since it does not exclude situations where a driver can travel to another part of the country to start working in order to have even workload distribution across the depots, while depth-first search usually places only geographically adjusted trips together.

The advantage of the proposed chromosome representation is that it creates both the diagrams and schedule within the same algorithm, thereby giving the GA greater control over the solution. It also does not require the generation of a large amount of diagrams at the beginning. In addition, this representation does not leave under-covered trips and ensures that no unnecessary over-covering happens. It is possible that at the beginning of the algorithm this chromosome representation might produce schedules with a high number of deadheads. However, due to the specific fitness function and genetic operators, the number of chromosomes containing deadheads decreases rapidly with evolution.

3.3 Fitness function

An adequate solution of the CSP requires the achievement of two conflicting objectives: high throttle time and low deviation from average diagram lengths. It is evident that with the increase in throttle time, the deviation from the average diagram length will be increased towards a minimum diagram length. This is due to the algorithm attempting to allocate a diagram for a single trip in order to achieve 100% throttle time.

Since GAs are a single-objective optimization method, a weighted sum approach has been applied in order to transform all the objectives into scalar fitness information [17]. The advantage of this technique is relative simplicity of implementation as well as high computational efficacy [18].

3.4 Selection

Preference was given to binary tournament selection as it is a comparatively simple and non-time consuming selection mechanism. It is also a popular selection strategy that is utilised in numerous GAs for CSP [9, 15, 16]. Binary tournament selection can be described as follows. Two individuals are selected at random form the population and the fittest among them constitutes the first parent. The same process repeats for the selection of the second parent.

3.5 Crossover and mutation

Since one or two point crossover might produce invalid offspring by removing some trips or copying the same journey several times, a crossover mechanism utilizing domain-specific information has been designed. Firstly, the process detects genes responsible for diagrams with a high throttle time in the first parent. Then these genes are copied to the first child and the rest of the genes are added from the second parent. The same procedure is then used to form the second child. The process is illustrated on the Figure 5.

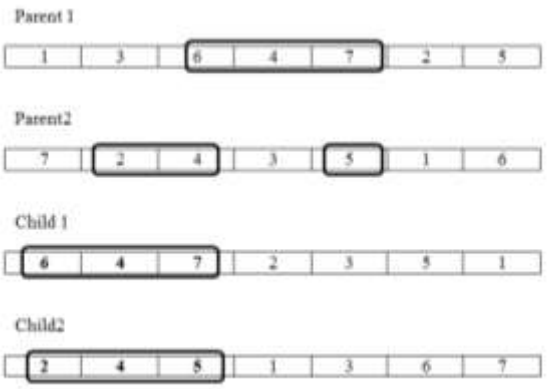


Figure 5 Crossover

In order to maintain diversity in the population, randomly selected genes are mutated with 40% probability. The mutation is performed by swapping two randomly identified genes. The mutation probability was determined through numerous tests and empirical observations.

4 Experimental results

The proposed GA for CSP (referred to as GACSP) has been used to produce diagrams for the freight-train drivers. The GACSP has been tested on a full daily data set obtained from one of the largest rail-freight operators in the UK. The data instances comprise 2000 freight-train legs, 500 cities, 39 depots, 1240 drivers, 500000 passenger-train links, and taxi trips connecting any of the stations at any time. Figures 6 and 7 illustrate a three-hour run of the algorithm and its achievement of the main business objectives, i.e. maximized throttle time and minimized deviation from the average shift duration. Increasing the throttle time indicates a reduction in deadheads and unnecessary waiting, thereby reducing the number of drivers required to operate the given trains. The decrease in deviation of the diagram duration from the average can be translated into equal utilization of the contract hours during the year. A typical resulting diagram is presented in Figure 8.

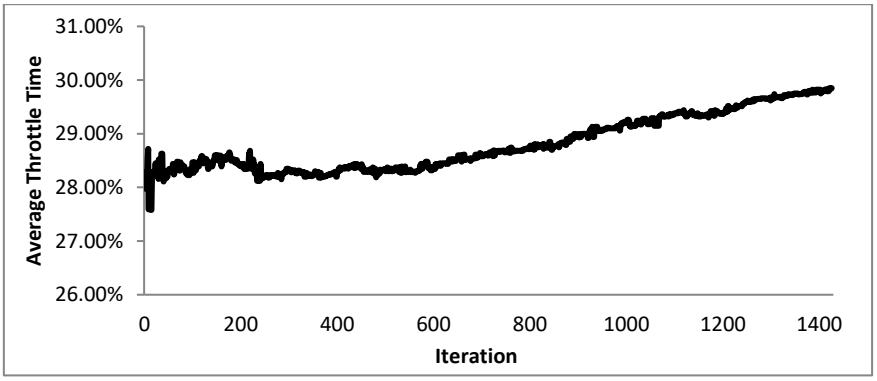


Figure 6 Maximizing average throttle time

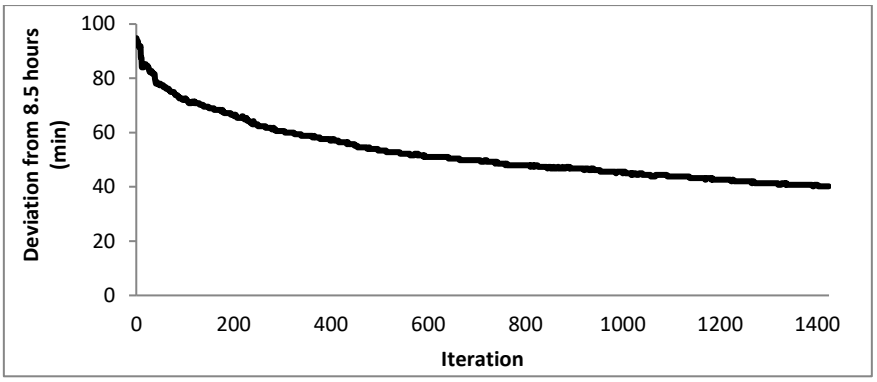


Figure 7 Minimizing deviation from the average shift length of 8.5 hours

Driver Number	Start time	End time	Activity	Departure	Origin	Diagram Length	Throttle Time
24	3:02	3:12	book on	Hougham	Hougham		
24	3:12	4:00	Driving	Hougham	Hampden,Great		
24	4:00	4:10	Idle	Hampden,Great	Hampden,Great		
24	4:10	5:00	Driving	Hampden,Great	Greenshawhill		
24	5:00	5:40	Break	Greenshawhill	Greenshawhill		
24	5:40	6:05	Passenger train	Greenshawhill	Buckland_Tout_Saints		
24	6:05	7:20	Break	Buckland_Tout_Sa	Buckland_Tout_Saints		
24	7:20	8:05	Driving	Buckland_Tout_Sa	Greenshawhill		
24	8:05	8:49	Taxi	Greenshawhill	Hampden,Great		
24	8:49	9:00	Idle	Hampden,Great	Hampden,Great		
24	9:00	9:50	Driving	Hampden,Great	Hougham		
24	9:50	10:20	Break	Hougham	Hougham		
24	10:20	11:00	Driving	Hougham	Greenshawhill		
24	11:00	11:20	Passenger(freight)	Greenshawhill	Hougham		
24	11:20	11:30	book off	Hougham	Hougham	8:28	45.58%

Figure 8 A typical diagram, i.e. driver schedule

In order to evaluate the efficiency of GACSP, it has been compared against two established approaches. The first is B&P, i.e. the combination of column generation and branch and bound methods [4]. The second comparator is Genetic Algorithm Process Optimization (GAPO), a genetic algorithm for CSP enhanced with repair and perturbation operators [9]. Both GAs have been adapted and modified to the current problem and implemented with C++ Builder while B&P was written in CPLEX. They all were run on computer with 4 GB RAM and 3.4 GHz Dual Core Processor. Initially, the intention had been to test all three algorithms on the full data set. However, after twelve hours running of the B&P algorithm, no solution had been reached. For the sake of comparison, the data size was reduced to six cities and 180 train legs, 500 passenger-train links. For the GA the population size was set as 20, crossover rate 90% and mutation probability 40%. As criteria for comparison, real business objectives such as throttle time, number of deadheads, average deviation from the desirable diagram length and computation time have been selected.

Table 1Experimental results using the reduced data set

	B&P			GAPO			GACSP		
Computation time (min)	60	120	228	60	120	228	60	120	228
Number of diagrams	-	-	22	32	28	26	25	23	23
Throttle time (%)	-	-	63	50	56	59	60	62	62
Average Number of deadheads per shift	-	-	1.36	2.21	1.85	1.60	1.66	1.47	1.47
Deviation from the average (min)	-	-	46	51	48	47	62	57	57

The computational results with the reduced data sets are displayed in Table 1. B&P obtained a solution in 228 minutes. Within 10 minutes, B&P had constructed 2000 columns and solved LP relaxation without an integer solution. Further time was required for branching and generation of additional columns. In order to estimate efficiency of GACSP and GAPO, they have been run for the same period of time. GACSP obtained an entire feasible schedule within 10 seconds and after one hour an acceptable schedule had been reached. Although the B&P algorithm ultimately achieved slightly better results, it has been tested on a problem of relatively small size. The computational time for linear programming algorithms usually grows exponentially with the increase in data size, so the B&P algorithm is likely to be impractical in environments where there is a crucial need to make fast decisions from large data sets.

As in other work [9], 3308 columns have been generated for GAPO, which took 30 minutes of computational time. Unlike B&P and GACSP, this approach did not have an embedded ability to generate additional columns, limiting its capability to explore other possible diagrams. It was also observed that 70% of the computational time was consumed by the heuristic and perturbation operators, whose aim was to restore the feasibility of the chromosomes. The repair operations were performed by scanning all available diagrams and selecting the best that could be inserted in the current schedule. GACSP overcomes this challenge by utilising the alternative chromosome representation that does not violate the validity of the chromosomes. Thus GACSP spends less time on each iteration and hence evolves more rapidly.

5 Potential implementation and integration issues

The most common implementation problems with software for scheduling transit systems concern robustness [19], i.e. the ability of the schedule to adapt to different circumstances. An example of such circumstances might be the delay of the previous train, resulting in the driver being unable to catch the planned train. In our system, the

transfer time regulates how much time is allocated for a driver to leave the previous train and start working on the next one. The larger the interval between trips, the lower the risk that the next freight train will be delayed by the late arrival of the previous one. On the other hand, a large transfer time decreases throttle time and requires more drivers to cover the trips. The best way to tackle this situation is to have an effective re-scheduling mechanism that makes changes in as few diagrams as possible.

In addition, the crew scheduling process is extremely complex. It is not always possible to model all the rules, nuances and exceptions of the schedule. For this reason, the system-generated diagrams have to be revised and amended by an experienced human planner until all the knowledge has been fully acquired.

Finally, although GAs are able to find an acceptable solution relatively quickly, they might also converge prematurely around a sub-optimal solution. Convergence can be controlled either by embedding variations in the selection procedure [17] or by changing the mutation rate [13].

6 Conclusions

In this paper, the complexities of CSP in the rail-freight industry in the UK have been described. Due to a high monetary cost of train crew, the profitability and success of the company might rely heavily on the quality of the constructed crew schedule. Given the wide geographical spread, numerous regulations, and severely constrained planning time, an IT system with an effective scheduling algorithm can equip a company with valuable decision-making support.

We have proposed a novel GA for crew scheduling (GACSP). Unlike other GAs for CSP, GACSP works with the entire schedule and does not restrict the algorithm in finding an optimal solution. The special chromosome representation and genetic operators are able to preserve the validity of the chromosomes without the utilization of additional repair operators or penalty functions. This capability enables the algorithm to consume fewer memory resources and to find a solution faster. In addition, the user can to retrieve a feasible schedule at any iteration.

It has been shown that although B&P was capable of finding an optimal solution from the mathematical perspective, time was its main weakness. In real-world operations, the cost for late optimal decision often can be much higher than that of a fast sub-optimal one. In this sense, the GA demonstrated excellent results as it provided a reasonable schedule nearly four times faster when using the reduced rail network. When faced with the scheduling task for the complete UK rail network, B&P had failed to find a solution at all after 12 hours, whereas GACSP was able to find an adequate solution in 2 hours. With further improvements of GACSP and possible hybridization with linear programming methods, its performance maybe further improved.

As future work, more domain specific rules will be incorporated into the chromosome generation process in order to achieve a better initial population. Moreover, it would be worthwhile to investigate a possible hybridization of a GA with the B&P method. The hybridization might seize the advantages of both algorithms to reach a solution that is close to the mathematical optimum in a short computation time.

Acknowledgements

This research has been supported by Sheffield Business School at Sheffield Hallam University. The authors would like to thank DB-Schenker Rail (UK) for the provision of data and information about real-world crew scheduling operations.

References

1. Barnhart, C., Hatay, L., Johnson, E.L.: Deadhead selection for the long-haul crew pairing problem. Oper. Res. Vol.43(3), pp.491-499 (1995).
2. Drexl, M., Prescott-Gagnon., E: Labelling algorithms for the elementary shortest path problem with resource constraints considering EU drivers' rules. Log. Res. Vol. 2(2), pp. 79-96 (2010).
3. Duck, V., Wesselmann, F., Suhl, L.: Implementing a branch and price and cut method for the airline crew pairing optimization problem. Pub. Transp..Vol.3(1), pp.43-64 (2011).
4. Barnhart, C., Johnson, E.L., Nemhauser, G.L., Savelsbergh, M.W., Vance, P.H.: Branch-and-price: Column generation for solving huge integer programs. Oper. Res. Vol.46(3), pp.316-329 (1998).
5. Derigs, U., Malcherek, D., Schafer, S.: Supporting strategic crew management at passenger railway model, method and system. Pub. Transp. Vol.2(4), pp.307-334 (2010).
6. Niederer, M.: Optimization of Swissair's Crew Scheduling by Heuristic Methods Using Integer Linear Programming Models. AGIFORS Symposium (1966).
7. Levine, D.:Application of a hybrid genetic algorithm to airline crew scheduling. Comp. & Oper. Res. Vol. 23(6), pp.547-558 (1996).
8. Santos, A.G., Mateus, G.R.: General hybrid column generation algorithm for crew scheduling problems using genetic algorithm. Proceedings of the 2009 Congress on Evolutionary Computation. pp. 1799-1806 (2009).
9. Zeren, B., Özkol, I.: An improved genetic algorithm for crew pairing optimization. J. Intell. Learn. Sys. & Appl. Vol.4(1), pp. 70-80 (2012).
10. Souai, N., Teghem, J.: Genetic algorithm based approach for the integrated airline crew-pairing and rostering problem. Eur. J. Oper Res. Vol. 199(3), pp. 674-683 (2009).
11. Park, T., Ryu, K.. :Crew pairing optimization by a genetic algorithm with unexpressed genes. J. Intell. Manuf. Vol. 17(4), pp. 375-383 (2006).
12. Kornilakis, H., Stamatopoulos, P.: Crew pairing optimization with genetic algorithms. In: Methods and Applications of Artificial Intelligence, pp. 109-20, Springer (2002).
13. Kwan, R.K., Wren, A., Kwan, A.K.: Hybrid genetic algorithms for scheduling bus and train drivers. Proceedings of the 2000 Congress on Evolutionary Computation.Vol.1, pp.282-292 (2000).
14. Costa, L., Santo, I.E., Oliveira, P.: An adaptive constraint handling technique for evolutionary algorithms. Optimization. Vol.62(2), pp. 241-253 (2013).
15. Chu, P.C., Beasley, J.E.: Constraint handling in genetic algorithms: The set partitioning problem. J. Heuristics. Vol. 4(4), pp.323-57 (1998)
16. Ozdemir, H.T., Mohan, C.K.: Flight graph based genetic algorithm for crew scheduling in airlines. Inf Sci.Vol.133(3-4), pp. 165-173 (2001).
17. Hopgood, A.A.: Intelligent systems for engineers and scientists. 3rd ed. CRC Press, Boca Raton (2012).
18. Coello, A.C.: An updated survey of GA-based multiobjective optimization techniques. ACM. doi:10.1145/358923.358929 (2000).
19. Gopalakrishnan, B., Johnson, E.L.: Airline crew scheduling: State-of-the-art. Ann. of Oper.Res. Vol. 140, pp. 305-337 (2005).

34th Annual International Conference of

BCS The Chartered Institute for IT

Specialist Group on Artificial Intelligence

9th -11th December 2014 Cambridge, UK.

The Specialist Group on Artificial Intelligence (SGAI) recognises the work of:

Elena Khmeleva (Sheffield Hallam University, UK)

as the principal author of the
Best Refereed Paper Written by a Student
In the Application Stream

Rail-Freight Crew Scheduling with a Genetic Algorithm

M.A Bramer

Max Bramer

Chairman