The MACB Problem: Definitions, Variants, and a PDDL+ Approach

PAPAMAXIMOU, Dani, GUO, Rongge, PERCASSI, Francesco and VALLATI, Mauro

The MACB Problem: Definitions, Variants, and a PDDL+ Approach

Dani G. Papamaximou[1], Rongge Guo[2], Francesco Percassi[1][a] and Mauro Vallati[1][b]

[1]*School of Computing and Engineering, University of Huddersfield, Huddersfield, United Kingdom*

[2]*School of Traffic and Transportation, Beijing Jiaotong University, Beijing, China*

{*Danai.Papamaximou, f.percassi, m.vallati*}*@hud.ac.uk, rgguo@bjtu.edu.cn*

Keywords: Modular Autonomous Customised Bus Systems, Automated Planning, Hybrid Planning.

Abstract: Modular Autonomous Customised Bus Systems (MACB) promise to significantly enhance public transport attractiveness and accessibility, leading to improved quality of life and reduced emissions. Compared to traditional bus systems, MACB provides a problem that poses a new set of challenges, ranging from the allocation of vehicles to the optimisation of routes and recharges. The MACB problem is attracting increasing interest within the transport research community, but it presents characteristics and dimensions that lend themselves well to approaches based on planning and combinatorial search. In this paper, with the aim of bridging the gap between different research communities, we provide a crisp definition of the MACB problem and present a planning-based approach to solve a specific variant of the problem, together with a set of benchmarks to foster research on this topic.

1 INTRODUCTION

Customised bus systems are considered an alternative to traditional public transit and private cars, as they enhance the accessibility, punctuality, and flexibility of public transport (Liu and Ceder, 2015). Unlike traditional transport services, customised bus systems do not rely on fixed routes and timetables, but instead cater to passengers with similar travel requirements in terms of time and space, offering flexible routes and schedules that can be adjusted to meet actual travel demands (Guo et al., 2022). According to demand, routes are generated and served by dedicated vehicles. This adaptability allows these systems to provide a more tailored and efficient transportation solution, responding to the dynamic requirements of passengers.

A major area of interest in the field of customised bus service is that of Modular Autonomous Customised Bus Systems (MACB), where modular autonomous vehicles are in operation to serve passengers requests (Wu et al., 2021; Guo et al., 2023a; Guo et al., 2023b). This class of vehicles further increases the flexibility of the service, by removing constraints based on drivers' shifts, and by supporting assembling and disassembling with other modules to form different vehicle formations, to better serve demand. These modules can also improve energy efficiency and charging operations when integrated with electric vehicles (Rezgui et al., 2019).

However, the flexibility of MACB systems comes with a set of significant challenges, ranging from the allocation of vehicles to the optimisation of routes and recharges. A large number of decisions need to be made, such as routing, scheduling, charging, and vehicle assembling or disassembling, while taking into account the distribution of passengers and their requirements in terms of pickup and drop-off times. For its complexity and its expected benefits, the MACB problem and its variants are attracting increasing interest, particularly from the intelligent transport systems research community – that leverages on traditional mathematical methods to address the complexity of the underlying problem. However, a general formal definition of this problem is missing. In formalising the MACB problem, it is easy to notice that it presents characteristics that suggest that model-based AI approaches, such as automated planning and combinatorial search, could provide a significant boost to the research in the area. Further, the use of model-based AI supports a goal-driven definition of the objectives to be achieved, and the explainability and transparency of the reasoning processes (Vallati and Chrpa, 2023; Smith, 2020; Percassi and Vallati, 2024).

In this paper, with the aim of bridging the gap between different research communities, and to ad-

[a] https://orcid.org/0000-0001-7332-0636

[b] https://orcid.org/0000-0002-8429-3570

vance the state of the art in the area, we provide a crisp definition of a major variant of the MACB problem. We then present an action-based formalisation, and we introduce an automated planning-based approach to solve it. Finally, we introduce a set of benchmarks to foster research on this topic, and we demonstrate that the proposed approach is capable of addressing small-sized instances using off-the-shelf domain-independent reasoners.

The remainder of this paper is organised as follows. First, we formalise the Modular Autonomous Customised Bus problem, discussing its variants. Then, in Section 3, we briefly introduce the PDDL+ language, and we present an approach to address the MACB problem by means of PDDL+. Section 4 provides an empirical evaluation of the capabilities of the introduced approach, considering two different maps and a range of passengers' demands. Finally, conclusion are given in Section 5.

2 FORMALISING THE MACB PROBLEM

Let us first contextualise the MACB problem. It shares some aspects with the Vehicle Routing Problem (VRP) (Toth and Vigo, 2002): more specifically, it relates to VRP with Time Windows (VRPTW) (Kallehauge, 2008), in which a fleet of vehicles must serve a set of customers within predefined time intervals, and to the VRP with Pickup and Delivery (VRPPD) (Desaulniers et al., 2002), where each request involves transporting passengers or goods from a pickup location to a delivery location.

The MACB problem is also related to the Dial-a-Ride Problem (DARP) (Ho et al., 2018), where small groups of passengers need to be moved between locations within a given time window. However, DARP focuses on few passengers and a different kind of service, and does not incorporate notions typical of modular electric vehicles. Finally, it also possible to draw a parallel with the School Bus Routing Problem (Ellegood et al., 2020), where buses operations need to be optimised to ensure all students can get to classes on time. The main differences with MACB lie in the fact that in MACB arrival times are not the same for all passengers, and that stops are fixed a-priori.

We are now in the position to define the considered MACB problem; please note that a range of variants can be considered – we will discuss them at the end of this section.

Formally, the MACB problem is defined as the tuple

$$P = \langle G, K, P \rangle,$$

where G is a graph representing the service network, K is a set of modular autonomous vehicles, each with a defined passenger capacity, and P is a set of passenger groups requesting transportation within the network. The service network is modelled as a directed graph $G = \langle V, E \rangle$, where V represents the locations and E denotes the routes connecting them. Each route $e \in E$ is characterised by a travel distance d_e. Locations V are partitioned into depots (D), charging locations (C), and service locations (S), with depots serving as the initial locations for all modules in K. Service locations (stations), on the other hand, are designated for passengers, who can be picked up and dropped off there.

$K = \{m_1, \ldots, m_{|K|}\}$ is the fleet of homogeneous modular vehicles. Each module has a maximum passenger capacity, denoted by $C_{\max}$, and moves at a constant velocity v_e. Consequently, for each route $e \in E$ with a given distance d_e, a travel time τ_e is assigned. This time represents the travel time required to move a module m from v to v' along the route $e = \langle v, v' \rangle$. Given the electric nature of the module, a maximum energy capacity denoted by $B_{\max}$. A constant energy discharge rate per unit distance, denoted by h, and a constant recharge rate per unit time, denoted by g, is also considered. These parameters define the energy expended during travel and replenished at a recharging facility.

The set of passenger groups requesting service is represented by $P = \{p_1, \ldots, p_n\}$. Each group $p_i \in P$ is characterised by its numerosity $q(P_i)$, indicating the number of passengers in the group, as well as a source location $src(P_i) \in S$, where the passengers must be picked up, and a target location $dst(P_i) \in S$, where they must be dropped off. Additionally, each passenger group has preferred time windows indicating the desired pickup and drop-off periods. Given $p_i \in P$, these time windows are denoted as $TW_{src}(p_i) = [l_i^{src}, u_i^{src}]$ for the pickup location and $TW_{dst}(p_i) = [l_i^{dst}, u_i^{dst}]$ for the drop-off.

Modules located at $v \in S$, where v corresponds to the source location of a group of people p_i (i.e., $v = src(p_i)$), can pick up p_i for transport. The pickup operation is feasible provided that loading the passengers does not exceed the module's capacity. Similarly, modules located at $v \in S$, where $v = dst(p_i)$, can drop off p_i. Both actions have a constant duration, denoted as τ_P, which accounts for the time required for passenger boarding and alighting. The same passenger group can be served by one or more modules.

Each of these activities has a duration, and their execution results in a cumulative increase in the total time required to complete them. Specifically, τ_e is added for each route traversed by a module, and τ_P is added for each pick-up and drop-off operation.

The position, energy, and load of each vehicle $m \in K$ are tracked over time using the functions $\text{pos}_m(t)$, $B_m(t)$, and $L_m(t)$, respectively. At the initial time t_0, each vehicle starts with full battery capacity, empty load and a position within the depot set, i.e., for each $m \in K$, $B_m(t_0) = B_{\max}$, $L_m(t_0) = 0$, and $\text{pos}_m(t_0) \in D$.

A solution π for a MACB problem Π is a function that associates to each vehicle $m \in K$ a sequence of actions

$$\pi(m) = \langle a_1, a_2, \ldots a_{k_m} \rangle,$$

where each action a_i is one of the following:

- $\text{drive}(n, n')$, for driving along the edge $e = (n, n') \in E$;
- $\text{pickup}(p)$, for picking-up passengers group $p \in P$ at $src(p) \in V$;
- $\text{dropoff}(p)$, for dropping-off passenger group $p \in P$ at $dst(p) \in V$;
- $\text{recharge}(n)$, for recharging the vehicle m battery at $n \in C$;
- $\text{wait}(n, \tau)$, an idle action that makes vehicle wait for τ time units at $n \in V$.

Observe that for actions drive, pickup, and dropoff, the duration is fixed, i.e., τ_e and τ_P. In contrast, for recharge, since we are considering the variant in which the vehicle must be fully recharged, the duration τ depends on the available energy of vehicle m when the action is executed. Finally, wait is the only action with a controllable duration τ, and its duration must be explicitly specified.

The function $\pi(m)$ accumulates time according to the prescribed durations of each action, resulting in a time trace $H(m) = \langle t_0, t_1, \ldots, t_k, t_{k_m+1} \rangle$, where t_i, with $i \in \{0, \ldots, k_m\}$, is the timing of action a_{i+1}. For each action a_{i+1} applied at time t_i, the values $\text{pos}_m(t_i)$, $E_m(t_i)$, $L_m(t_i)$, and the next timestamp t_{i+1} are updated accordingly. Specifically:

- if $a_{i+1} = \text{drive}(n, n')$, with $e = (n, n')$ then $\text{pos}_m(t_{i+1}) = n'$ and $B_m(t_{i+1}) = B(t) - h \cdot d_e$ where $t_{i+1} = t_i + \tau_e$;
- if $a_{i+1} = \text{pickup}(p)$ then $L_m(t_{i+1}) = L_m(t) + q(p)$ where $t_{i+1} = t_i + \tau_P$;
- if $a_{i+1} = \text{dropoff}(p)$ then $L_m(t_{i+1}) = L_m(t) - q(p)$ where $t_{i+1} = t_i + \tau_P$;
- if $a_{i+1} = \text{recharge}(n)$ then $B_m(t_{i+1}) = B_m(t_i) + g \cdot \tau$, where $\tau = \frac{B_{\max} - B_m(t_i)}{g}$ and $t_{i+1} = t_i + \tau$;
- if $a_{i+1} = \text{wait}(n, \tau)$ then $\text{pos}_m(t_{i+1}) = \text{pos}_m(t_i)$, $B_m(t_{i+1}) = B_m(t_i)$, $L_m(t_{i+1}) = L_m(t_i)$ where $t_{i+1} = t + \tau$.

The solution π is feasible for Π iff, for each $m \in K$, the following constraints are satisfied:

1. (*Passengers' requests constraints*)
   Each passenger group $p \in P$ is uniquely served by a vehicle; formally
   $$\exists!\ m \in K : \text{pickup}(p), \text{dropoff}(p) \in \pi(m)$$
   Each passenger group $p \in P$ is served at the requested locations and within the preferred time windows; formally, for each $m \in K$
   $$\forall\ \text{pickup}(p) \in \pi(m) \text{ at } t \Rightarrow$$
   $$\Rightarrow t \in TW_{src}(p) \wedge pos_m(t) = src(p).$$
   $$\forall\ \text{dropoff}(p) \in \pi(m) \text{ at } t \Rightarrow$$
   $$\Rightarrow t \in TW_{dst}(p) \wedge pos_m(t) = dst(p).$$
2. (*Energy and load constraints*)
   The energy and load of each vehicle never fall below $B_{\min}$ and never exceed the maximum capacity $B_{\max}$; formally, for each $m \in K$
   $$\forall t \in H(m) \Rightarrow B_m(t) \geq B_{\min} \wedge L_m(t) \leq B_{\max}$$
   Each charging action over a vehicle occurs with an empty load; formally:
   $$\forall\ \text{recharge}(n) \in \pi \text{ at } t \Rightarrow L_m(t) = 0$$
3. (*Locations consistency*). Each action can only be executed if the module is physically located at the corresponding node at that time; formally:
   $$\forall\ \text{drive}(n, n') \in \pi(m), \text{recharge}(n), \text{wait}(n, \tau) \in \pi(m), \text{at } t \Rightarrow pos_m(t) = n$$

Additionally, each sequence associated with a vehicle must start and end at a depot; therefore, $a_1 = \text{drive}(n, n')$ with $n \in D$, and $a_{k_m} = \text{drive}(n, n')$ with $n' \in D$.

A feasible solution can be characterised in metric terms, and several studies in the literature have considered different performance indicators, such as schedule duration, operational costs, and profits. In this work, we focus on the schedule duration.

Given a vehicle m, its associated plan $\pi(m)$, and the timing trace $H(m) = \langle t_0, \ldots, t_{k_m+1} \rangle$, the value t_{k_m+1} denotes the completion time of the action sequence $\pi(m)$. Therefore, the cost of π can be simply defined as:

$$makespan(\pi) = \max_{m \in K} t_{k_m+1}.$$

Alternatively, the cost of a solution could also be evaluated in terms of operational expenses, provided that each action is associated with a specific cost. Moreover, in scenarios with high demand, it may be reasonable to relax the notion of feasibility by allowing some passenger groups to remain unserved, at the expense of incurring a penalty cost for each unmet request.

2.1 Variants of the MACB Problem

Let us take a look at some potential alternative characterisations of the MACB problem, according to different perspectives. While the list of characterisations is by no means complete, it aims at providing an overview of the dimensions that can be explored within the MACB problem remit.

2.1.1 Energy

The dynamics governing energy consumption and regeneration can influence the model's accuracy and characteristics. In this formalisation, we consider constant discharge and charge rates. Other formulations can consider more complex non-linear discharge dynamics, that take into account the characteristics of the network as well as the number of passengers in the vehicle, as done for example in the context of VRPs with energy constraints (Mavrovouniotis et al., 2020).

2.1.2 Fleet of Vehicles

In the proposed formulation of the MACB problem, vehicles are assumed to be *homogeneous* in terms of size and capacity, meaning that all modular units have identical dimensions and passenger-carrying capabilities. However, in certain application contexts, the fleet could consist of heterogeneous vehicles with varying capacities, operating costs, and functionalities to provide different services, such as deploying high-capacity vehicles on high-demand routes and smaller, more agile vehicles for low-demand or flexible services. In such cases, utilising a heterogeneous fleet could enable more cost-effective solutions. For example, in some work (Li et al., 2024), the fleet consists of two kinds of vehicles with different capacities and operational costs. In that study, energy-related constraints are not discussed; however, vehicles could also potentially be differentiated based on their energy profile.

Moreover, the study incorporates a holding control strategy, allowing vehicles to dwell at stops for limited periods in order to synchronise with passenger requests. By allowing controlled waiting times at stops, the model becomes more expressive, providing greater flexibility in routing decisions and potentially improving resource utilisation.

2.1.3 Assembling and Disassembling

A significant variant introduced in the literature to leverage the modular structure of vehicles considers the integration of *assembly* and *disassembly* operations among modules, enabling the dynamic configuration of the fleet in response to the demand (Guo et al., 2022; Guo et al., 2023b; Zhang et al., 2020). Within this framework, modules can be combined to create composite vehicles with increased capacity, typically configured before deployment. Conversely, disassembly allows a composite unit to separate into individual modules, which may either operate independently or be redirected to depots or charging facilities (Wu et al., 2021).

3 A PDDL+ APPROACH FOR THE MACB PROBLEM

In this section, we present a PDDL+ encoding of the described MACB problem. To this end, we first briefly recall the basic concepts of PDDL+ and some well-known applications in the transport field, before introducing the designed models.

3.1 PDDL+

In this work, we adopt a discrete formalisation of PDDL+, which abstracts continuous dynamics into discrete transitions to tame the complexity of PDDL+ problems. This approach is commonly used by planners (Scala et al., 2020) and applied methods (Alon et al., 2024), and is consistent with the existing approach for the MACB problem domain (Guo et al., 2023b).

A *discrete-time* PDDL+ *planning task*, denoted by Π, is defined as a tuple $\langle F, X, I, G, A, E, P \rangle$, accompanied by a *time discretisation step* $\delta \in \mathbb{Q}^+$, which determines the temporal granularity of the model. F is a finite set of Boolean variables, each assigned a value in $\{\bot, \top\}$, while X is a finite set of numeric variables with values in $\mathbb{Q}$. A *state* s is a complete assignment to all variables in $F \cup X$, and for each variable v, the notation $s[v]$ indicates its value in state s. The initial state I defines the system's configuration at time zero, and the goal G specifies the conditions required in the final state. Boolean conditions are of the form $\langle v = b \rangle$ for $v \in F$ and $b \in \{\bot, \top\}$, while numeric conditions are expressed as $\langle \varphi \bowtie 0 \rangle$, where φ is an arithmetic expression over X and $\bowtie \in \{>, \geq, =, \leq, <\}$. The sets A, E, and P contain the actions, events, and processes, respectively. Each element $z \in A \cup E \cup P$ is described by a pair $\langle pre(z), eff(z) \rangle$, where $pre(z)$ denotes the preconditions and $eff(z)$ the effects. Preconditions are given as propositional formulae involving Boolean and numeric constraints. For actions and events, effects are expressed as assignments of the form $\langle v := b \rangle$ or $\langle v := \varphi \rangle$. Processes behave differently: although they also have preconditions, their effects are defined as discrete-time numeric updates $\langle v, \varphi \rangle$, meaning that at each time step of size δ, the value of v evolves according to $v(t+\delta) = v(t) + \varphi(t) \cdot \delta$.

A PDDL+ plan π_t is given by the pair $\langle \pi, t_e \rangle$, where π is a sequence of timestamped actions $\langle \langle a_1, t_1 \rangle, \ldots, \langle a_n, t_n \rangle \rangle$, and $t_e \in \mathbb{Q}_0^+$ denotes the makespan or overall duration.

The validity of π_t is assessed by simulating its execution over discrete timepoints $\{0, \delta, 2 \cdot \delta, \ldots, t_e\}$. At each time step, the state is updated by accounting for the effects of actions, triggered events, and active processes. Each action applied at time t_i causes an immediate transition based on its effects. Following the application of each action and each time increment δ, events are evaluated and applied if their preconditions are satisfied. Processes continuously affect the state at every time step by applying their incremental effects. The simulation proceeds until time t_e, and the plan π_t is considered valid for Π if every action is executable and the final state satisfies the goal condition G.

A complete description of the full semantics of PDDL+ is provided in the work of (Fox and Long, 2006), while further semantic details can be found in (Fox et al., 2012) and (Percassi et al., 2025; Percassi et al., 2023).

Notably, PDDL+ planning has been exploited in the transport domain for supporting deployable traffic signal optimisation (Kouaiti et al., 2024; Doria et al., 2026), for simulating traffic evolutions in urban areas (Bhatnagar et al., 2023), and for some preliminary explorations of traffic distribution (Vallati et al., 2021).

3.2 Components of the PDDL+ Model

This section outlines the key elements of the PDDL+ encoding, establishing a correspondence with the variables and actions specified in the MACB target formulation.

The most relevant numeric variables include each vehicle's current energy level, the number of passengers on board, and the residual distance to a destination during movement. Boolean predicates track the vehicle's current location, its charging status, the current itinerary, and the fulfilment status of each passenger service request. An additional numeric variable explicitly represents time. This variable is incremented by one unit whenever the agent decides to let time progress. This discretisation step semantically corresponds to one minute, as commonly adopted in related approaches (Guo et al., 2023b). Passenger requests are modelled through Boolean invariants representing the existence of a service itinerary, while associated numeric constants define the number of passengers to be transported and the preferred time windows for pickup and drop-off.

Each MACB activity involving a vehicle, namely drive, recharge, pickup, dropoff, and wait, takes place over a time interval and is modelled using the standard PDDL+ pattern based on a triplet: an action that initiates the activity, a process that governs its progression, and a terminating event that marks its completion. Intuitively, the action tests the feasibility of the activity, the process controls its duration and the dynamics of the involved numeric variables, and the event updates the state (e.g., tracking that a vehicle has moved to a new location). The following describes how each MACB activity is encoded in PDDL+ using this triplet-based structure.

Movement Between Locations. The movement of a vehicle m located at n along an edge $e = (n, n')$, is modelled by executing an initiating driving action that sets the intended itinerary, the average speed, and the initial distance to the destination n' (which corresponds to d_e). This action is applicable only if the vehicle has sufficient energy to traverse the edge. Once executed, it activates a driving process that updates the vehicle's state over time, decreasing the remaining distance and energy level proportionally to the assigned speed. The process continues until the distance to the destination reaches zero, triggering the corresponding event that terminates the driving activity and completes the movement.

The definition of the terminating event ensures that the activity lasts exactly τ_e, matching the travel duration associated with edge e.

The initiating actions and terminating events are expressed in a grounded form for each pair of nodes among depots and service locations (excluding recharging stations), while the driving process is defined in a lifted form. Figure 1 provides an example of the action `drive_p1_p2` and event `arrived_from_p1_p2`, referring to a generic vehicle navigating from node `p1` to `p2`. Each such triplet provides a PDDL+ encoding of the drive(n, n') action described in the MACB problem.

Recharging. Recharging is modelled analogously to driving activities. The initiating action can be executed when the vehicle is located at a recharging station. This action sets a charging flag, which enables a process responsible for restoring the vehicle's energy level over time. The activity concludes automatically when an event is triggered upon reaching full battery capacity.

Differently from the driving activity, this is specified in a fully lifted fashion. Each such triplet provides a PDDL+ encoding of the recharge(n) action

```
(:action drive_p1_p2
        :parameters (?b -car)
        :precondition (and
            (location ?b p1)
            (>= (energy ?b) N))
        :effect (and
            (not (location ?b p1))
            (itinerary ?b p1 p2)
            (assign (bus-distance ?b
                p2) (distance p1 p2
               ))
            (assign (speed ?b) (
                ave_speed))))

    (:event arrived_from_p1_to_p2
        :parameters (?b -car)
        :precondition (and
            (<= (bus-distance ?b p2)
                0)
            (itinerary ?c p1 p2))
        :effect (and
            (location ?c p2)
            (not (itinerary ?c p1 p2
               ))))
```

Figure 1: Example ground action and event used to model the movement of a vehicle between locations `p1` and `p2`. `N` is a numeric constant calculated in pre-processing.

described in the MACB problem, with a duration constrained by the residual energy.

Navigation towards a recharging station is modelled using the same triplet structure employed for regular driving but with a dedicated initiating action. This action differs from standard driving actions in that it is only executable when the vehicle is not carrying passengers.

Boarding and Alighting of Passengers. Each service request p specifies the number of passengers $q(p)$ to be transported from a given origin $src(p)$ and destination $dst(p)$, and is associated with preferred time windows $TW_{src}(p)$ and $TW_{dst}(p)$ for both pickup and drop-off. The initiating pickup action can be executed when four conditions are met: the vehicle is located at the source of a passenger group, the group has not yet been served, there is sufficient space on board to accommodate all passengers, and the current time falls within the allowed pickup time window. Executing this enables the boarding process during which passengers are transferred from the station to the vehicle. After τ_P time units have elapsed, the process is concluded by triggering an event that transfers the passengers from the location to the vehicle and enables departure. A symmetrical structure is used for alighting, where a dedicated action and process move passengers from the vehicle to their destination station, completing the requested service. Passengers may only disembark at their assigned destination. These constructs provide a PDDL+ encoding of the pickup(p) and dropoff(p) actions defined in the MACB model.

Waiting. In PDDL+, it is not necessary to explicitly define an idle action when a vehicle is not performing any operation. The agent is allowed to wait by default, and if a vehicle is not engaged in any activity, the frame axioms implicitly capture the wait action as described in the MACB model.

A problem is described in terms of the network structure G, the energy discharge rate h, the maximum and initial battery capacities, the number of available vehicles, their average speed, and their initial location. For each passenger group, the corresponding origin and destination locations and the associated time windows are specified. The goal is to have all passengers at their destinations within the given time windows, and the vehicles at a depot location.

3.3 Pre-processing

To reduce the complexity of the problem at hand, we designed a pre-processing method involving 3 steps.

First, let us highlight that amongst the (potentially) large number of locations of the service network, in any given instance of the problem only a subset of them is needed. On the basis of this observation, the first step is to reduce the size of the network. Specifically, a pre-processing step is used to calculate the most efficient itinerary for each request, and to prune parts of the network that are never visited.

Second, given that a vehicle must recharge when its energy drops below a certain level, and it must not recharge when it is transporting people, the following additional conditions on vehicles' movements are used to restrain the preconditions of driving actions: *Condition 1:* When the vehicle transports passengers, its energy should be greater than or equal to the energy required to complete the trip (i.e., reach the drop-off station and return to the closest depot/recharge facility). *Condition 2:* When the vehicle does not transport passengers, the minimum charge that the vehicle must have should be equal to or greater than the energy required to move to the next station, plus the minimum energy needed to move from the next station to the closest recharging point.

Third, after the 2 steps above, it is possible to calculate the shortest paths between every pair of relevant locations in the network. The shortest itineraries from the depot to the pickup locations, and from the drop-off locations to the depot are calculated using Dijkstra's algorithm. The same process is then repeated for couples pickup-dropoff, for drop-off points and recharging stations, and from recharging stations to pickups. All the shortest paths are then joined by means of series composition (Eppstein, 1992), and intermediate locations where stops are not needed are abstracted away.

4 EMPIRICAL ANALYSIS

This experimental analysis aims at showing the capabilities of the introduced PDDL+-based approach for the MACB formulation with energy.

As planning engine we use ENHSP (Scala et al., 2020) with default parameters' configuration. All plans were validated with VAL (Howey et al., 2004). Experiments were ran on a machine equipped with AMD Ryzen 7 pro 5850u CPU, 16GB of RAM, and running on Ubuntu 20.04.5 LTS.

In this analysis we consider two network structures.[1] *Network 1:* A synthetically designed grid comprising eight stations, a depot and a re-charging station. All edges are assumed to be bidirectional. In this network, we consider at most two vehicles to satisfy the passengers requests. Initial charges are set so that vehicles need to recharge in order to complete the travel requests for instances 2 and 3. Instance *N1-1* aims to examine the model using one vehicle serving two services: collecting passengers from both stations 2 and 3, and dropping both groups off at station 6. The vehicle has enough capacity to serve both requests together. Instance *N1-2* includes the a passenger group from station 1 to station 5, in addition to the ones described in the previous instance. Two vehicles are assigned in this instance, and should both be used to satisfy the time windows. Instance *N1-3* includes an additional travel request from station 4 to station 7, and two vehicles to be used to satisfy the required time windows.

Network 2 is instead derived from real-world data. We consider a subset of the travel demand collected in an area of Beijing (Guo et al., 2019) and widely used for assessing public transport optimisation approaches (see, e.g., (Ma et al., 2023; Guo et al., 2023b)). The network comprises 14 stations with one starting depot and one ending depot. The recharging option was not examined in this network. In instance *N2-1*, two vehicles need to serve four requests. In instance *N2-2* five passengers requests have to be served. It is worth noting that for this instance the graph reduction was limited due to the proximity of the boarding stations and the overall grid configuration.

[1]Models and benchmarks are available at: https://github.com/danipmax/icaart-26

Table 1: Performance of ENHSP on the considered instances in terms of runtime (CPU-time seconds) and makespan (minutes in the simulation).

Instance	Vehicles	Runtime	Makespan
N1-1	1	4.91	132
N1-2	2	29.30	144
N1-3	2	44.20	147
N2-1	2	47.10	136
N2-2	2	194.78	185

Results are presented in Table 1 in terms of runtime and makespan of the corresponding solution. The results indicate that the proposed approach allows a domain-independent planning system to tackle complex MACB instances. Instances are solved in a matter of seconds, and are guaranteed to generate valid solutions by construction. However, we acknowledge there is room for improvement, as the state of the art in the field of intelligent transport systems based on traditional operational search approaches can handle larger number of vehicles (in the 20s). On the other hand, the state of the art involved domain/instance-specific models and approaches, while in this paper we introduce a planning model that can be handled by a domain-independent PDDL+ engine. Further, the use of PDDL+ allows to generate precise solution plans that can be used to monitor real-time execution, while other approaches do not allow this level of detail.

It is worth noting that none of the considered instances could be solved without the introduced pre-processing. Further, to contextualise the importance of pruning the network graph of the nodes that are not needed we tested instance *N1-2* without such operation: in that case, the runtime significantly increased to 74.64 seconds, with a negative impact also in terms of makespan of the solution (165). These results confirm the importance of pre-processing.

5 CONCLUSION

In this paper we characterised a version of the MACB problem and provided a corresponding PDDL+ encoding together with an effective pre-processing procedure. Our experimental results demonstrate that even with a domain-independent PDDL+ engine, the proposed approach can efficiently solve realistic, albeit small-scale, MACB instances within seconds, generating valid and detailed execution plans suitable for real-time monitoring. This highlights the potential of automated planning for addressing this class of problems.

Future work will focus on enhancing performance through the development of domain-specific heuristics and exploring a wider range of MACB variants within the PDDL+ framework. We also aim to expand the complexity of modelled instances and network topologies, to extend the set of available benchmarks.

ACKNOWLEDGEMENTS

D. Papamaximou thanks the Communications Workers Union for supporting the completion of this work. F. Percassi and M. Vallati were supported by a UKRI Future Leaders Fellowship [grant number MR/Z00005X/1].

REFERENCES

Alon, L., Weitman, H., Shleyfman, A., and Kaminka, G. A. (2024). Planning to be healthy: Towards personalized medication planning. In *ECAI*, volume 392 of *Frontiers in Artificial Intelligence and Applications*, pages 4232–4239. IOS Press.

Bhatnagar, S., Guo, R., McCabe, K., McCluskey, T. L., Percassi, F., and Vallati, M. (2023). Automated planning for generating and simulating traffic signal strategies. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence, IJCAI*, pages 7119–7122.

Desaulniers, G., Desrosiers, J., Erdmann, A., Solomon, M. M., and Soumis, F. (2002). Vrp with pickup and delivery. *The vehicle routing problem*, 9:225–242.

Doria, F., Percassi, F., Maratea, M., and Vallati, M. (2026). A Domain-specific Heuristic for PDDL+-based Traffic Signal Optimisation. In *AAAI*.

Ellegood, W. A., Solomon, S., North, J., and Campbell, J. F. (2020). School bus routing problem: Contemporary trends and research directions. *Omega*, 95:102056.

Eppstein, D. (1992). Parallel recognition of series-parallel graphs. *Information & Computation*, 98(1):41–55.

Fox, M. and Long, D. (2006). Modelling Mixed Discrete-Continuous Domains for Planning. *Journal of Artificial Intelligence Research*, pages 235–297.

Fox, M., Long, D., and Magazzeni, D. (2012). Plan-based policies for efficient multiple battery load management. *J. Artif. Intell. Res.*, 44:335–382.

Guo, R., Bhatnagar, S., Guan, W., Vallati, M., and Azadeh, S. S. (2023a). Operationalizing modular autonomous customised buses based on different demand prediction scenarios. *Transportmetrica A: Transport Science*, page 2296498.

Guo, R., Guan, W., Bhatnagar, S., and Vallati, M. (2022). A two-phase optimization model for autonomous electric customized bus service design. In *ITSC*, pages 383–388. IEEE.

Guo, R., Guan, W., Huang, A., and Zhang, W. (2019). Exploring potential travel demand of customized bus using smartcard data. In *Proc. of ITSC*, pages 2645–2650.

Guo, R., Guan, W., Vallati, M., and Zhang, W. (2023b). Modular autonomous electric vehicle scheduling for customized on-demand bus services. *IEEE Trans. Intell. Transp. Syst.*, pages 10055–10066.

Ho, S. C., Szeto, W. Y., Kuo, Y.-H., Leung, J. M., Petering, M., and Tou, T. W. (2018). A survey of dial-a-ride problems: Literature review and recent developments. *Transportation Research Part B: Methodological*, 111:395–421.

Howey, R., Long, D., and Fox, M. (2004). Val: Automatic plan validation, continuous effects and mixed initiative planning using pddl. In *Proc. of ICTAI*, pages 294–301.

Kallehauge, B. (2008). Formulations and exact algorithms for the vehicle routing problem with time windows. *Computers & Operations Research*, 35(7):2307–2330.

Kouaiti, A. E., Percassi, F., Saetti, A., McCluskey, T. L., and Vallati, M. (2024). PDDL+ Models for Deployable yet Effective Traffic Signal Optimisation. In *ICAPS*, pages 168–177.

Li, X., Zhao, Y., and Feng, Z. (2024). Customized bus service design with holding control and heterogeneous fleet: A column-generation-based decomposition algorithm. *IEEE Trans. Intell. Transp. Syst.*, 25(12):19563–19580.

Liu, T. and Ceder, A. A. (2015). Analysis of a new public-transport-service concept: Customized bus in china. *Transport Policy*, pages 63–76.

Ma, D., Fang, B., Ma, W., Wu, X., and Jin, S. (2023). Potential routes extraction for urban customized bus based on vehicle trajectory clustering. *IEEE Transactions on Intelligent Transportation Systems*, 24(11):11878–11888.

Mavrovouniotis, M., Menelaou, C., Timotheou, S., Ellinas, G., Panayiotou, C., and Polycarpou, M. M. (2020). A benchmark test suite for the electric capacitated vehicle routing problem. In *CEC*, pages 1–8. IEEE.

Percassi, F., Scala, E., and Vallati, M. (2023). A practical approach to discretised PDDL+ problems by translation to numeric planning. *J. Artif. Intell. Res.*, 76:115–162.

Percassi, F., Scala, E., and Vallati, M. (2025). On the notion of plan quality for pddl+. In *ICAPS*, pages 102–111.

Percassi, F. and Vallati, M. (2024). Leveraging ai planning in a what-if analysis framework for assessing traffic signal strategies. In *27th IEEE International Conference on Intelligent Transportation Systems*. IEEE.

Rezgui, D., Siala, J. C., Aggoune-Mtalaa, W., and Bouziri, H. (2019). Application of a variable neighborhood search algorithm to a fleet size and mix vehicle routing problem with electric modular vehicles. *Computers & Industrial Engineering*, 130:537–550.

Scala, E., Haslum, P., Thiébaux, S., and Ramirez, M. (2020). Subgoaling Techniques for Satisficing and

Optimal NumericPlanning. *J. Artif. Intell. Res.*, 68:691–752.

Smith, S. F. (2020). Smart infrastructure for future urban mobility. *AI Mag.*, 41(1):5–18.

Toth, P. and Vigo, D. (2002). *The vehicle routing problem*. SIAM.

Vallati, M. and Chrpa, L. (2023). In defence of good old-fashioned artificial intelligence approaches in intelligent transportation systems. In *2023 IEEE 26th International Conference on Intelligent Transportation Systems (ITSC)*, pages 4913–4918. IEEE.

Vallati, M., Scala, E., and Chrpa, L. (2021). A hybrid automated planning approach for urban real-time routing of connected vehicles. In *24th IEEE International Intelligent Transportation Systems Conference, ITSC*, pages 3821–3826. IEEE.

Wu, J., Kulcsár, B., Qu, X., et al. (2021). A modular, adaptive, and autonomous transit system (maats): An in-motion transfer strategy and performance evaluation in urban grid transit networks. *TRA: Policy and Practice*, 151:81–98.

Zhang, Z., Tafreshian, A., and Masoud, N. (2020). Modular transit: Using autonomy and modularity to improve performance in public transportation. *TRE: Logistics and Transportation Review*, 141:102033.