Multi robot cooperative area coverage, case study: spraying

JANANI, Ali-Reza, ALBOUL, Lyuba <http://orcid.org/0000-0001-9605-7228>
and PENDERS, Jacques <http://orcid.org/0000-0002-6049-508X>

Published version

JANANI, Ali-Reza, ALBOUL, Lyuba and PENDERS, Jacques (2016). Multi robot cooperative area coverage, case study: spraying. In: ALBOUL, Lyuba, DAMIAN, Dana and AITKEN, Jonathan M., (eds.) Towards autonomous robotic systems : 17th annual conference, TAROS 2016, Sheffield, UK, June 26-July 1, 2016, Proceedings. Lecture notes in Computer Science (2716). Springer, 165-176. [Book Section]

Multi Robot Cooperative Area Coverage, Case Study: Spraying

Alireza Janani, Lyuba Alboul, and Jacques Penders

Sheffield Hallam University,
Sheffield UK
ajanani@my.shu.ac.uk
l.alboul@shu.ac.uk
j.penders@shu.ac.uk

Abstract. Area coverage is a well-known problem in multi robotic systems, and it is a typical requirement in various real-world applications. A common and popular approach in the robotic community is to use explicit forms of communication for task allocation and coordination. These approaches are susceptible to the loss of communication signal, and costly with high computational complexity. There are very few approaches which are focused on implicit forms of communication. In these approaches, robots rely only on their local information for task allocation and coordination. In this paper, a cooperative strategy is proposed by which a team of robots perform spraying a large field. The focus of this paper is to achieve task allocation and coordination using only the robots' local information.

Keywords: Multi Robotic System, Cooperative Behaviour, Cooperative Area Coverage

1 Introduction

In area coverage, a team of robots is cooperatively trying sweep an entire area, possibly containing obstacles. The goal is to achieve coverage with efficient paths for each robot which jointly ensure that every single point in the environment is visited by at least one of the robots while performing the task [6]. Many real world applications require systematic area coverage including search in forested areas, demining, distribution of beacons and line searching. In this paper we focus on application of agricultural robotics.

Recently, there has been an increase of interest in performing agricultural tasks by a team of autonomous robots. One of the main reasons is shortage of labor force. Over the years, various approaches have been suggested to reduce the need of labour force. A conventional approach is to use larger machineries to process larger portions of the field at a time. However, deploying heavy and large machineries results in soil compaction [14]. Soil compaction has devastating outcomes on the crop, and it costs up to 90% of the cultivation cost to recover [2].

Another trend in reducing input labour force is automation. However, single robotic approaches are expensive, and still require occasional human supervision [12]. Deploying a team of smaller and lighter agricultural machineries prevents soil compaction, reduces the cost of cultivation, and increases the fault-tolerance of the overall system [11]. Furthermore, multi robotic approach has the promise to reduce dependency on labour force [12].

If a team of robots is applied, the main question is how the robots should execute the task to cooperatively achieve the global goal? To answer this question, the task of spraying has to be studied in detail.

1.1 Spraying

Spraying is the process of dispensing Plant Protection Products (PPP) on the crop at different stages during cultivation. Conventionally, a tractor with spraying unit is driven throughout the field and the PPPs are gradually dispensed on the crop.

Spraying is distinguished from other agricultural tasks in that of redundancy of processing. In other agricultural tasks (e.g. ploughing, seeding, and harvesting), even though redundancy in processing (that is processing a point in the field more than once) increases the cost of execution, the final result is still acceptable. In spraying, any location in the field has to be processed only once, since excessive PPPs dispensing will destroy the crop. Moreover, in spraying, the direction of field processing is fixed. The sprayer unit is allowed to navigate through the field via gaps inbetween the crop rows. Any other motions or manoeuvres are prohibited.

In a single robotic approach, the sprayer unit simply starts the task from any location of the field and processes one track after another. However, if multiple robots are deployed to execute spraying, the field has to be divided among robots so that no two robots spray the same region in the field. According to the taxonomy presented in [4], this requirement expects the team to be strongly coordinated.

1.2 Problem Statement

As in any multi robotic system, there are various parameters that have to be set. For example: team architecture, communication structure (centralised vs decentralised), control structure (centralised vs distributed), task partitioning, task allocation (self-organising versus market-based), and coordination (not coordinated, weakly coordinated, strongly coordinated). In order to identify the appropriate choices, the scenario has to be described in more details.

Assuming that robots are capable of carrying out multiple tasks, it is possible that, by the time of spraying, not all robots are positioned near each other. Therefore, the initial position of the robots are unknown at the initiation of spraying. As a result, robots cannot rely on direct forms of communication (for example using Bluetooth, WiFi, etc) to perform task allocation since robots could be out of range of the communication signal. Since no direct form of communication is possible, central-based approaches become inapplicable.

The aim of this paper is to develop a self-organising robotic system that would require little computational effort, no central-based control and communication, which at the same time is efficient and adaptable to various fields. The main focus of the proposed strategy is to perform task allocation using only local information of the robots.

In this paper, we are not considering the problem of localization and electro-mechanical aspect of the spraying. The problem of localization has been addressed in many researches in single robotic approaches in precision farming.

1.3 Assumptions and Initial Conditions

In summary, we assume the following:

Assumption 1 Robots are not capable of communicating with each other using explicit forms of communication.
Assumption 2 Robots are initially scattered around the field, hence field accessing is asynchronous.
Assumption 3 Robots are not allowed to perform any manoeuvring motion in the middle of the field, since they will run over the crop.

In addition, the following are considered as simplifying conditions:
Condition 1 All robots are equipped with an accurate localization system.
Condition 2 The robots have limited 180 degrees field of view.
Condition 3 The environment is a large known 2-dimension rectangle. Prior to execution, four coordinates of four corners of the field along with distance between two consecutive tracks are given to robots.

1.4 Overview of the Paper

We first look into related works in cooperative area coverage. Next, we describe and analyse the proposed strategy for task partitioning, task allocation and coordination. And finally, we look at the impact it has on the team performance.

2 Related Works

Cooperative area coverage has been studied for the past decades. This resulted in suggestion of numerous strategies for different applications. We classify the reviewed approaches into three categories: *Static approaches*, *Central-based approaches* (from control point of view), and *Explicit Communication-based approaches*. We provide examples for each category and describe why these approaches are inapplicable in cooperative spraying.

In static approaches, task allocation is carried out manually prior to execution [5]. In these approaches, the initial positions of the robots are known, and the field is divided among robots in a way to minimize the cost of execution using various known theories (e.g. graph theory). Static approaches are not appropriate for cooperative spraying as the initial positions of the robots are unknown by which robots attend their share of task at different instances of time. With this, robots, which are processing two adjacent regions, could fall into congestion in the middle of the field. A logical conclusion is that task allocation has to be performed in real-time which requires some form of interaction among robots.

One common method is to use a central-based unit. The central unit could be fixed in a place somewhere around the field or it could be one of the robots participating in the task. In this paper, we classified central-based approaches into two categories: decision-maker, and data-pool.

In decision-maker approaches, the central-base unit collects necessary information from individuals in the team, and based on the preprogrammed algorithms it organises the robots so that the global task is achieved. [3], [9], [10], and [15] are few examples of this type of approach in which the central unit based on the collected information performs task allocation and coordination if necessary.

In data-pool approaches, the central-unit is used as a shared computational space in which robots exchange information to make their decisions. In most cases, the central unit does not perform any coordination or task allocation. Anil et al [1] demonstrated a low cost, multi-functional team of robots which are capable to perform various agricultural tasks. In this example, first few robots are sent to observe the environment. Once returned, the robots share their findings with others through a central unit via the attached ZigBee modules.

In communication based approaches, the success of the team is susceptible to the loss of the central unit. Besides, robots have to be within range of the central-unit communicating signal. Moreover, the cost and complexity of the system grows as the number of participating robots increases. In spraying, robots are initially scattered around a large field, therefore no central unit could guarantee task allocation and coordination throughout the execution. This requires a task allocation which is only based on local information of the robots. In [8], this form of task allocation is referred to as threshold-based methods. Robots could achieve the required information from two sources: (I) the sharing environment (stigmergy), (II) local interaction between robots.

Ranjbar et al [13] present a cooperative area coverage model based on stigmergy. In this example, each robot dispenses pheromone-like material in the environment to mark its territory. Upon detection of another robot's trail, robots manoeuvre to a predefined direction. In [7], it is demonstrated how a team of robots could perform ploughing cooperatively on a large field using the state of the soil. The result of spraying is detectable only for a short period of time and stigmergic approach are not reliable for spraying. Therefore, the proposed solution has to be based on local interaction among robots.

3 System Description

In any team of robots, the main problem to be addressed is task allocation. In spraying, task allocation has to guarantee that each location in the field is visited only once. In addition, task allocation has to be carried out in real-time, and only the local information of the robots has to be taken into consideration. Therefore, a mechanism is required by which robots are informed about the state of the task. Note that this information cannot be conveyed with explicit forms of communication.

The proposed strategy is that the field is divided into regions (each region consists of few tracks), with the aim that each region is processed by only one robot ($reg_i \rightarrow r_i$). Each robot claims a region by occupying a particular location, which is referred to as checkpoint, outside the region. Checkpoints are set to be the last track of each region. Spraying a region starts from this location and tracks are processed consecutively to the first track in the region. Robots have to check each checkpoint to see if the region is occupied. If an unoccupied location is found, the robot proceeds and occupies the location (see Fig. 1).

.

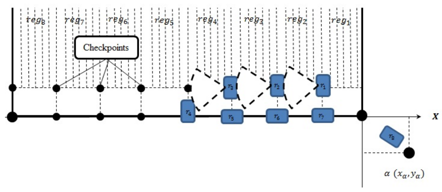

Fig. 1. A team of 8 robots are performing task allocation. r_1, r_2, and r_3 have found unoccupied checkpoints. In the meantime, other robots are examining every checkpoint.

Once the last robot, r_n, occupied the last region, other robots have to be informed otherwise they will never start spraying the field. To solve this, when a robot occupies a checkpoint, it poses itself in a way that it continuously monitors next checkpoint. The last robot does not need to comply with this, and instead it starts spraying the field right after it occupies the last checkpoint. Then, r_{n-1} starts spraying once r_n is no longer detected. With this, robots starts spraying one after another, and the first robot will be the last that starts the spraying(see Fig. 2).

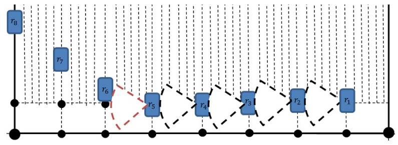

Fig. 2. Task initiation stage for a team of 8 robots; Spraying task starts whenever r_8 reaches checkpoint on the last region. r_7 starts spraying once it perceives r_8 decision.

3.1 Task Partitioning

In spraying, the field can be processed from multiple locations independently. To take advantage of this capability, the field is divided into regions and divided among robots. For this, the number of regions has to be equal to the number of robots in the team. Each region consists of few tracks. If there are n robots and K tracks in the field, the number of tracks in each region, k_i, can be determined as follows:

$$k_i = \begin{cases} \lceil \frac{K}{n} \rceil & \text{if } i \leq K \mod n \\ \\ \lfloor \frac{K}{n} \rfloor & \text{if } i > K \mod n \end{cases} \tag{1}$$

In here, $\lceil \frac{K}{n} \rceil$ represents the smallest following integer, and $\lfloor \frac{K}{n} \rfloor$ represents the largest previous integer.

Next, it is important to identify the track numbers that are within each region. Robots require this to locate the checkpoints. Let's assume that TR is a set of track numbers in the field, $TR = \{tr_l | l \in \{1, 2, ..., K\}\}$, and R is set of robots, $R = \{r_i | i \in \{1, 2, ..., n\}\}$, then track numbers allocated to r_i, can be obtained as follows:

$$G_i = \{tr_j | j \in \{m + 1, m + 2, ..., m + k_i\}\} \tag{2}$$

Where m is the last track number assigned to previous robots, and it can be calculated as follows:

$$m = \sum_{j=1}^{i-1} k_j \tag{3}$$

3.2 Task Allocation

Since robots do not communicate with each other, robots cannot share their understanding of the environment with others; hence robots have to perform redundant checkpoint analysis. As the number of robots increases, the number of locations that the robot has to check also increases. For time analysis, let's first define duration of task allocation for a robot.

Definition 1. *Task allocation duration for a robot is the period from initial position until the time that the robot detects an unoccupied checkpoint.*

With this definition, time analysis becomes complex since robots are initially scattered around the field. To simplify calculations, let's assume that robots have formed a queue behind a location outside of the field. This location is referred to as *alpha* (see Fig. 3). Before robots access the first checkpoint, they first have to access *alpha*. However, distance that a robot has to travel to reach *alpha* depends on the robot's position in the queue. The length of the queue for each robot is $(\lambda + \epsilon)(i + 1)$ meters. In here, λ is the length of a robot, ϵ(*epsilon*) is the minimum distance between two consecutive robots.

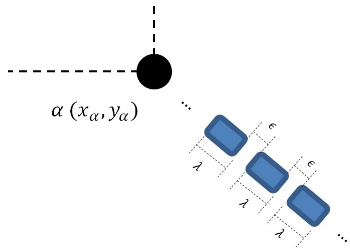

Fig. 3. Illustration of robots queueing for accessing *alpha*. Robots have the same length, λ, and the distances between robots, *epsilon*, are equal.

Once a robot reaches *alpha*, it has to analyse the first checkpoint, and hence all robots have to travel a fixed distance between *alpha* and the first checkpoint, d_{α,l_1}. Also, except the first robot, other robots have to travel to other checkpoints, d_{l_{j-1},l_j}. For example, r_2 has to travel distance between the first and second checkpoint to reach its destination.

Once a robot reaches a checkpoint, it takes a period of time to draw conclusion on the status of the checkpoint. This period is denoted by τ and it will be propagated in the queue since robots have to wait for the path to be cleared. The total delay for a robot is $(2i - 1)\tau$.

This is easy to see. For example in a team of three robots, r_1 will analyse only the first checkpoint, hence spends only a τ of time. Whereas r_2 will analyse checkpoint one and checkpoint two, but r_2 is already affected by the delay of the first robot. Therefore, the queue effect and the checkpoint analysis effect for r_2 sum to 3τ. Also, r_3 will analyse three checkpoints, plus the delay propagated by r_1. In addition another delay will be propagated as a result of analysis of r_2 on the first checkpoint. Any further checkpoint analysis of r_2 does not affect r_3 since both robots are performing their analysis simultaneously. Therefore, the overall queue effect on r_3 is 5τ.

If the velocities of all robots are equal and robots move with constant speed, task allocation duration for a robot (r_i) can be evaluated as follows:

$$t_{ta_i} = \frac{d_{\alpha l_1}}{v} + (2i - 1)\tau + \frac{(\lambda + \epsilon)(i - 1)}{v} + \sum_{j=2}^{i} \frac{d_{l_{j-1} l_j}}{v} \tag{4}$$

3.3 Task Initiation

Task allocation for a robot is completed when it identifies the first unoccupied checkpoint, but the task is not yet initiated since the robot has to assure that all other robots are informed about the status of the occupied region.

Definition 2. *Task initiation duration for a robot is the period that a robot has to remain at a checkpoint until the initiation signal is detected.*

Since spraying will not start for all robots until the last robot has occupied its last region, the task initiation time for a robot will be affected by the last robots task allocation time. However, this impact is partially compensated by the robot's task allocation period. With this, if τ_{init} is the constant time to perceive task initiation, the standby period for a robot before it starts its task is evaluated from the following:

$$t_{st_i} = t_{ta_n} - t_{ta_i} + \tau_{init}(n-1) \tag{5}$$

With this strategy, r_1 is the first robot that completes task allocation, but it is the last robot to initiate spraying. Therefore, (5) can also be expressed as follows:

$$t_{st} = t_{st_1} = t_{ta_n} - t_{ta_1} + \tau_{init}(n-1)$$

In here, t_{st} refers to total spraying execution time, and t_{st_1} is the first robot spraying time. Right after the task is initiated, robots start spraying. Spraying is a two-dimensional navigation task. If the velocity of a robot is denoted v, and length of a track is denoted l_p, then spraying time for a robot after task initiation can be evaluated as follows:

$$t_{s_i} = \frac{1}{v}(k_i l_p + (k_i - 1)d_f) \tag{6}$$

In here, d_f corresponds to the distance between two consecutive tracks. Bear in mind that a robot has to repeat spraying as many as k_i times, obtained from (1). In addition, as part of spraying, a robot also has to switch between tracks for $k_i - 1$ times.

The total execution time including all three steps (task allocation, task initiation and spraying) is the maximum of sum of these times for all robots. However, since task partitioning starts from the first robot, and since r_1 is the last robot that initiates its task, total task execution time can be expressed as follows:

$$t = t_{s_1} + t_{ta_1} + t_{ts_1} \tag{7}$$

The proposed strategy promises that a group of robots can spray a large field. However, robots can perceive only a limited range. In other words if the distance between two consecutive checkpoints becomes larger than the perception range of robots, the task can never be initiated. Furthermore, as number of participating robots increases, the distance between checkpoints decreases. However, there is a limit on how close two checkpoints could be placed from one another as robots require space for transiting from one track to another. In other words, there is a limit on maximum number of robots that could be applied with this strategy.

$$\lfloor \frac{W}{\Gamma} \rfloor \leq n \leq \lfloor \frac{W}{\lambda + \epsilon} \rfloor \tag{8}$$

In here, Γ is the range of perception of robots, W is the width of the field, $W = Kd_f$, λ is the length of the robots, and *epsilon* is the threshold distance between two robots.

4 Simulation Results and Discussion

In this section, we present both numerical results based on mathematical consideration and the results obtained from Stage simulation environment. Since the allowed number of participating robots depends on the dimension of the field, first the boundaries of the team sizes have to be identified. In here, few parameters about the environment have to be fixed.

We assumed that there are 50 to 250 tracks are available in the field and the distance between two consecutive tracks is 20(cm), and the length of each track (l_p) is 20(m). Robots have equal dimensions with length (λ) equal to 50(cm). The threshold distance ($\epsilon(epsilon)$) between two robots is set to be 30(cm), and all robots are assumed to move in constant velocity equal to 0.5(m/s). The checkpoint analysis duration (τ) and robot behaviour analysis duration (τ_{init}) is fixed to 5(s).

From (8), it is possible to identify the maximum and the minimum number of robots allowed in various field sizes, since $W = Kd_f$ (see Fig. 4(a)). The definition of a team in a multi-robotic system fixes the minimum number of robots to be greater than three ($n_{min} \geq 3$). For a field with hundred tracks ($K = 100$), the number of participating robots could vary between four and forty ($n \in [4, 40]$). Consequently, task allocation and task initiation time can be predicted (see Fig 4(b)).

As number of participating robots increases, the durations for both task allocation and task initiation increase by which the total execution time increases. In a field with 100 tracks, $K = 100$, if appropriate team sizes are applied ($n \in [4, 40]$), the total execution time opposes with what is expected from the nature of a multi-robotic system (see Fig.4(b)). This could be concluded that the applied strategy is not efficient enough to receive positive effects from an increase in number of robots.

In (7), there are two increasing linear functions and one decreasing hyperbolic function. Therefore the resulting function will have a global minimum. This is the optimal team size that could be deployed to the given field and it can be obtained as follows:

$$n_{opt} = \sqrt{\frac{K(l_p + d_f)}{2\tau v + \tau_{init} v + \lambda + \epsilon}} \tag{9}$$

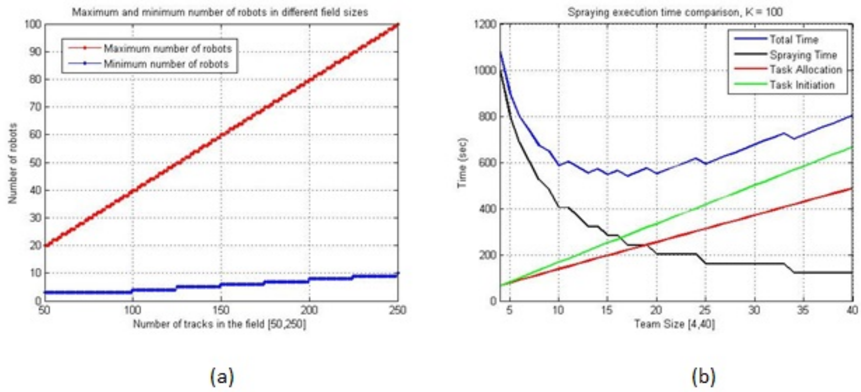

Fig. 4. Numerical results: (a) Maximum and minimum number of robots in different field sizes. (b) Time analysis results.

The optimal number of robots corresponding to the minimum time from the start of time allocation and the completion of spraying depends on the parameters in the expression, and from the proposed ones, is 15.6. So the optimised number of robots is either 15 or 16.

On the other hand, series of simulations were conducted in Stage simulation environment in collaboration with ROS (Robot Operating System). In simulation, each robot is equipped with a model of Hokuyo Laser Range Finder, and a fixed camera which both are placed in front of the robot. In addition, at any given time, a robot could obtain a two dimensional coordinate corresponds to its current location in the global frame. However, during the trials no robot is aware of the position of other robots in the team.

Behaviours on each robot are controlled by collaboration between three C++ modules (see Fig. 5(a)): (1) Task Handler, (2) Reach Point, and (3) Camera Analyser. Task Handler is responsible to triggers specific behaviours in the robot: setting a new target, activating Camera Analyser, analysing the field, and executing the cooperative procedures. Reach Point is responsible to guide the robot to the requested coordinate in collision free manner. Camera Analyser is responsible to analyse the behaviour of other robots, using colour detection as robots are homogeneous, and it signals Task Handler. However, as behaviour monitoring is only necessary at particular locations and moments in the field, Task Handler will trigger the procedure whenever it is required.

Trials conducted for a field with 51 tracks (K = 51), and various team sizes were deployed ($n \in [3, 10]$). In all team sizes, robots performed task allocation and task initiation successfully. During simulation, position and time of robots are recorded. Data recording for a robot initiates when the robot passes , and it stops as soon as the robot exits the margin of the field. The maximum execution time then is plotted and compared with the equivalent corresponding numerical prediction (see Fig. 5(b)). It can be seen that there is a slight difference between

Stage simulation results and the numerical results. This difference is due to the field exiting duration which has not been considered in mathematical description.

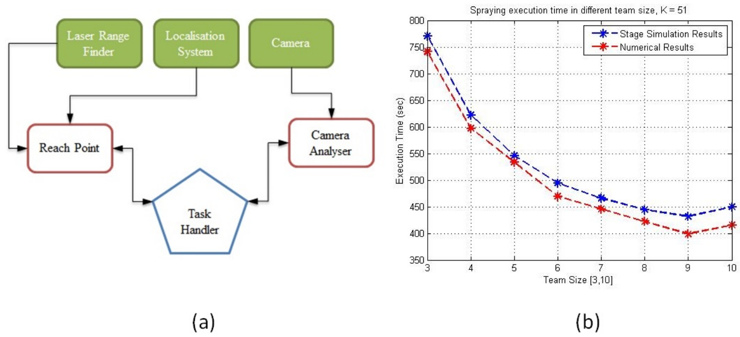

Fig. 5. Stage Simulation Results. (a) Individual behaviour on single robot. (b) Comparison between results collected during simulation and numerical visualisation.

5 Conclusion and Future Works

In this paper, we described a strategy by which a team of robots could cooperatively perform area coverage related task in a known environment. The task demands a delicate solution since each point in the field has to be sprayed only once and it has large dimensions.

In the proposed strategy, robots rely only on their local information to obtain an order and perform task allocation. The mathematical analysis of the strategy is described and numerical validation and conducted simulation in ROS and Stage suggest that this strategy can be successfully applied on real robots.

However, since local perceptions of the robots are limited, the distance between two consecutive checkpoint locations cannot exceeds beyond the robots range of detection. This makes the proposed strategy to be partially scalable. In addition, the proposed method requires that all robots participate all at once. In the future, we aim to improve the method in a way that robots can execute their share of task in more robust way.

References

1. H. Anil, K. Nikhil, V. Charitra, and B. Gurusharan. Revolutionizing farming using swarm robotics. *6th International Conference on Intelligent Systems, Modelling and Simulation*, 19(3):141–147, 2015.
2. S. Blackmore. Precision farming and agricultural engineering for the 21st century: How to make production agriculture more efficient. *Presentation*, 2010.

3. D. Drenjanac, S. Tomic, L. Klausner, and E. Kuhn. Harnessing coherence of area decomposition and semantic shared spaces for task allocation. *In Information Processing in Agriculture*, pages 23–34, 2014.
4. A. Farinelli, L. Iocchi, and D. Nardi. A classification focused on coordination. *Mult-Agent Robotic Systems*, 34(5):2015–2028, 2004.
5. P. Fazeli, A. Davoodi, and A. Pasquier. Fault-tolerant multi-robot area coverage with limited visibility. *In Proceeding of the IEEE International Conference on Robotics Automation (ICRA)*, 2010.
6. P. Fazli. On multi-robot area coverage. *In proc.of 9th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2010)*, pages 1669–1670, 2010.
7. A. Janani, L. Alboul, and J. Penders. Multi-agent cooperative area coverage: Case study ploughing. *International Conference of Autonomous Agents and Multi-Agent Systems (AAMAS) 2016*, 2016.
8. N. Karla and A. Martinoli. A comparative study of market-based and threshold-based task allocation. *8th Symposium on Distributed Autonomous Robotic Systems*, 10(1):91–101, 2006.
9. N. Li, C. Remeikas, Y. Xu, S. Jayasuriya, and R. Ehsani. Task assignment and trajectory planning algorithm for a class of cooperative agricultural robots. *Journal of Dynamic Systems, Measurement and Control*, 137(5):1–9, 2015.
10. C. Liu and A. Kroll. Memetic algorithms for optimal task allocation in multi-robot systems for inspection problems with cooperative tasks. *journal of Soft Computing*, 19(3):567–584, 2014.
11. J. Liu and J. Wu. *Mult-Agent Robotic Systems. In:"Mult-Agent Robotic Systems.* CRC Press, New York, 2001.
12. N. Noguchi. Robot farming system using multiple robot tractors in japan agriculture. *In Preprints of the 18th IFAC World Congress Milano.*, pages 633–637, 2006.
13. B. Rangbar-Sahaei, G. Weiss, and A. Nakisaee. Stigmergic coverage algorithm for multi-robot systems. *In Proceeding of AAMAS12*, 2012.
14. R. L. Raper and J. M. Kirby. Soil compaction: How to do it, undo it, or avoid it. *American Society of Agriculture and Biological Engineers, ASABE Distinguished Lecture Series*, 2006.
15. A. Tsalatsanis, A. Yalcin, and K. Valavanis. Dynamic task allocation in cooperative robotic systems. *International journal of Advanced Robotic Systems*, 6(4):309–318, 2004.