CISTER
Research Center in
Real-Time & Embedded
Computing Systems

Technical Report

Move and Improve: A Distributed Multi-Robot Coordination Approach for Multiple Depots Multiple Travelling Salesmen Problem

Omar Cheikhrouhoua
Anis Koubâa
Hachemi Bennaceur

CISTER-TR-140411
Version:
Date: 5/14/2014

Move and Improve: A Distributed Multi-Robot Coordination Approach for Multiple Depots Multiple Travelling Salesmen Problem

Omar Cheikhrouhoua, Anis Koubâa, Hachemi Bennaceur

CISTER Research Unit

Polytechnic Institute of Porto (ISEP-IPP)

Rua Dr. António Bernardino de Almeida, 431

4200-072 Porto

Portugal

Tel.: +351.22.8340509, Fax: +351.22.8340509

E-mail: , aska@isep.ipp.pt,

http://www.cister.isep.ipp.pt

Abstract

The multiple depot multiple travel salesman problem (MD-MTSP) is a common research problem in several mobile robots applications. The problem is known to be NP-hard and centralized solutions are computation intensive and are not suitable for mobile robots applications, in particular for large instances. In this paper, we propose a new market-based multi-robot coordination technique, called move and improve. The concept is simple: in each step of our algorithm, a robot moves and attempts to improve its solution by coordination with its neighbor robots. Our approach consists of four main steps: (1) initial target allocation, (2) tour construction, (3) negotiation of conflicting targets, (4) solution improvement. Several extensive simulations are conducted and our approach is compared against a centralized Genetic Algorithm (GA) MD-MTSP solver in terms of optimality. We demonstrate that our solution outperforms the centralized GA approach.

Move and Improve: A Distributed Multi-Robot Coordination Approach for Multiple Depots Multiple Travelling Salesmen Problem

Omar Cheikhrouhou †§, Anis Koubâa*‡, Hachemi Bennaceur ¶

† Computer and Embedded Systems Lab, University of Sfax, Tunisia.
§ ISIMA, University of Monastir, Tunisia,
* Prince Sultan University, Saudi Arabia.
‡ CISTER/INESC-TEC, ISEP, Polytechnic Institute of Porto, Porto, Portugal.
¶ Al-Imam Mohamed bin Saud University, Saudi Arabia.

Emails: omar.cheikhrouhou@ceslab.org, aska@isep.ipp.pt, hachemi@ccis.imamu.edu.sa

***Abstract*—The multiple depot multiple travel salesman problem (MD-MTSP) is a common research problem in several mobile robots applications. The problem is known to be NP-hard and centralized solutions are computation intensive and are not suitable for mobile robots applications, in particular for large instances. In this paper, we propose a new market-based multi-robot coordination technique, called move and improve. The concept is simple: in each step of our algorithm, a robot moves and attempts to improve its solution by coordination with its neighbor robots. Our approach consists of four main steps: (1) initial target allocation, (2) tour construction, (3) negotiation of conflicting targets, (4) solution improvement. Several extensive simulations are conducted and our approach is compared against a centralized Genetic Algorithm (GA) MD-MTSP solver in terms of optimality. We demonstrate that our solution outperforms the centralized GA approach.**

I. Introduction

Multi-robot systems are very effective for several applications as compared to single-robot systems, considering the collaborative effect between robots that leads to accomplishing their missions more efficiently. For example, in cleaning applications [1, 2], the use of multiple robots would contribute to clean large areas in much less time than if using a single robot. Data collection from wireless sensor networks [3, 4] and target tracking [5] are other examples where the use of multiple robots would be more effective than relying on one robot. In the PLANET project [6], for instance, a swarm of Unmanned Aerial Vehicles (UAVs) was used for real-life wildlife monitoring application. However, coordinating a team of robots is a challenging and complex problem in multi-robot systems. Indeed, it is important to assign tasks efficiently to robots to optimize the execution of their missions. This type of problems is typically cast to the multi-robot task allocation (MRTA) problem.

In its general form, the MRTA problem is formulated as follows: Given n robots and m tasks, the objective is to assign robots to tasks in order to optimize the performance and reduce the cost. Several approaches were proposed: (*i.*) *centralized approaches*: a central agent is assumed to have a global knowledge about the system, and is able to find optimal solutions to the problem using optimization algorithms. (*ii.*) *distributed approaches*: the decision is made by the robots in distributed fashion based on their partial knowledge about the system. This approach is flexible but leads to non optimal solutions, as only partial information is used to generate solutions in the robots. (*iii.*) *market-based approaches*: these approaches are based on an auctioning-bidding process and provide a good trade-off between centralized and distributed strategies. It eliminates the need for global information maintenance at a central agent, while it provides better solutions qualities as compared to the distributed approach. The idea consists in collecting information using a auctioning-bidding process during which a central agent (auctioneer) transfers task's offers for robots, which respond with their abilities and costs to perform the task. The central agent will then allocate tasks based on the information collected from the robots.

In this paper, we investigate a particular complex instance of the MRTA problem where a team of robots has to visit a set of locations and then return to their original locations, at the minimum cost. This problem is typically modeled as multiple-depot multiple travel salesman problem (MD-MTSP), which is known to be NP-Hard problem. The MD-MTSP problem is applied in a broad range of mobile robots applications, including for example, environment monitoring using a swarm of UAVs, which has to collect data from pre-specified regions and them come back to their original depots [6]. In this work, we propose a market-based approach to solve the MD-MTSP problem called *Move-and-Improve*. The idea of the Move-and-Improve mechanism is based on incrementally improving the solution quality through an iterative process of coordination between the robots during their missions.

The remainder of this paper is as follows. Section 2 surveys the most relevant works on MD-MTSP and its applications.

Section 3 describes the system model of the problem. In Section 4, we present the Move-and-Improve market-based approach and algorithms. Section 5 presents a comprehensive MATLAB simulation study and the performance evaluation results. Section 6 concludes the paper and discusses future works.

II. Related Works

Multiple depots multiple traveling salesmen problem has applications in several mobile robots areas, such as unmanned ground and aerial vehicles having motion and fuel (or energy) constraints [7]. In surveillance or environment monitoring applications with multiple UAVs, the aerial robots have to repeatedly visit a sequence of target regions and then must return to the corresponding depot to recharge fuel. In the literature, several works have proposed different solutions to the MD-MTSP problem and applied it in different contexts. In [8], the authors modeled the multi-robot repeated area coverage as MTSP problem and proposed three distributed cluster-based algorithms, namely Uninformed Clustering Coverage, Edge-based Clustering Coverage, Node-based Clustering Coverage. The performance of these algorithms were evaluated, under different environment representations and robot visual range, in terms of total path length, total average visiting period, total worst visiting period, and the balance in workload distribution. The authors concluded that these results can be used for choosing an appropriate combination of a repeated coverage algorithm, an environment representation, and the robots visual range for a particular scenario considering the metric to be optimized.
In [9], the authors gave the first theoretical complexity analysis of the performance of auction-based methods for multi-robot routing for different objective functions. For this purpose, they provided and analyzed six bidding rules for three team objectives, namely minimizing the total cost, the maximum cost, and the average service cost.

In [10], the authors proposed a distributed and dynamic algorithm for the multiple traveling robots problem (MTRP). Each robot selects its own target location in an incremental and distributed manner. The algorithm works as follows. First, each robot selects the appropriate targets using a shortest distance as cost function. Then, the robot announces its target visiting plan by a single-item auction. Selection of the best robot for a task is performed by using the Contract Net Protocol (CNP), which is an auction-based protocol that allows sharing tasks among multiple robots. Webots simulation results were conducted and revealed the efficiency of the proposed approach in terms of scalability, total path length and communication message overhead. In our work, we also consider an incremental target allocation during the mission execution of the robots, but our approach differs in several perspectives, as we consider four different phases for improving the solution quality, with different objective functions.

Being inspired from the Consensus-Based Bundle Algorithm (CBBA), [11] and the Market-based Approach with Look-ahead Agents (MALA) [12], the authors of [13] proposed an market-based algorithm to solve the MTSP problem. The algorithm is an iterative market process, and in each iteration each robot performs the following sequence of steps: market auction, agent-to-agent trade, agent Switch and agent relinquish step. In the market auction step, each robot selects the best tasks according the objective cost. In the agent-to-agent trade step, robots randomly consider the tasks of other robots and check whether it can perform any of these tasks at a lower cost. In the Complete Takeover or Agent Switch step, the algorithm tries to explore solutions that are not in the local minima. The algorithm terminates after a number of iteration with no improvement in performance.

In [14], the authors proposed a solution to the MDMTSP using probability collectives, where the traveling vehicles (robots) are considered as autonomous agents and every route assigned to vehicles as a strategy. In [15], the MDMTSP was transformed into a Single, Asymmetric Traveling Salesman Problem. This permits to apply the standard TSP solution to the problem.

In this paper, we present a new market-based approach to solve the MD-MTSP problem based on incremental improvement of the solution quality through four phases of coordination between the robots. Our objective is to provide a flexible mechanism that is adaptive to changing conditions during the execution of the robots' mission, while still producing a good solution quality. Indeed, centralized approaches are static and are not able to cope with the dynamicity of mobile robots applications.

III. System Model

We consider the multiple depot multiple traveling salesman problem, where a set of m robots, located at different depots, must visit a set of n target locations and come back to their depots.

The main objective is to find an efficient assignment of the target locations to the team of robots such that all the targets are covered by *exactly* one robot, and the cost is minimal. Typically, the objective function is defined as minimizing the total traveled distance by all the robots or minimizing the maximum tour length of all robots. This problem can be also cast to the repeated coverage problem where the robots need to repeatedly visit the set of allocated target locations after determining their tours.

Formally, we consider a set of n targets' locations $\{T_1,...,T_n\}$ that must be repeatedly visited by m robots $\{R_1,...,R_m\}$, which are initially located at m depots locations $\{D_1,...,D_m\}$. Each robot R_i starts from its depot D_i, then visits the list of ri allocated targets $\{T_{i_1},...,T_{i_{ri}}\}$ in that order, and finally returns back to its depot. The cost to travel from target T_i to target T_j is denoted as $C(T_i,T_j)$. We define the cost $C(Tour_{R_i})$ as the cost of the tour performed by the robot Ri and is expressed as:

$$C(Tour_{R_i}) = C(D_i,T_{i_1}) + \sum_{k=1}^{ri-1} C(T_{i_k},T_{i_{k+1}}) + C(T_{i_{ri}},D_i) \quad (1)$$

We consider two metrics to measure the quality of the solution:

1) *The total cost of all robots' tours*, which refers to the sum of the costs of all tours assigned to the robots, i.e. the sum of the total traveled distance by all the robots. We refer to this objective function as *MinSum* as it consists in minimizing the sum of costs of all robots. Formally, the objective function is expressed as:

$$\begin{aligned} & \underset{Tour_{R_i} \in TOURS}{minimize} \left(\sum_{i=1}^{m} C(Tour_{R_i})\right) \\ \text{subject to : } & Tour_{R_i} \cap Tour_{R_j} = \emptyset, \forall i \neq j, \\ & \bigcup_{i=1}^{i=m} Tour_{R_i} = \{T_j, 1 \leq j \leq n\}. \end{aligned} \tag{2}$$

where $C(Tour_{R_i})$ is the cost of the tour assigned to the robot R_i and $TOURS$ is the set of all possible Tours.

2) *The maximum tour cost among all robots' tours*, which refers to the maximum cost among all tours of the robots. We refer to this objective function as *MinMax* as it consists in minimizing the maximum cost of all robots missions. Formally, the objective function is expressed as:

$$\begin{aligned} & \underset{Tour_{R_i} \in TOURS}{\arg\min} \left(\max_{j \in 1...m} C(Tour_{R_j})\right) \\ \text{subject to : } & Tour_{R_i} \cap Tour_{R_j} = \emptyset, \forall i \neq j, \\ & \bigcup_{i=1}^{i=m} Tour_{R_i} = \{T_j, 1 \leq j \leq n\}. \end{aligned} \tag{3}$$

The *MinSum* case is mostly used when one wants to minimize the fuel consumption of the group of robots, and the *MinMax* case is mostly used in time-critical missions, when there is a need to perform the latest task as early as possible. Other cost functions can be derived as linear combinations of the ones above [16].

We assume that each robot has a global knowledge of the set of targets that must be visited and their locations, but does not know about other robots and their locations unless it is able to communicate with them when they are in the same communication range. Moreover, each robot can estimate the distance between its current location and each target, as the euclidian distance or using a path planning algorithm.

IV. Move and Improve Mechanism

The Move and Improve mechanism comprises four phases: (1) the allocation phase, (2) the tour construction phase, (3) the overlapped targets elimination phase, and (4) the solution improvement phase (see Figure 1). A video demonstration of the Move-and-Improve mechanism using Webots simulator is available in [17].

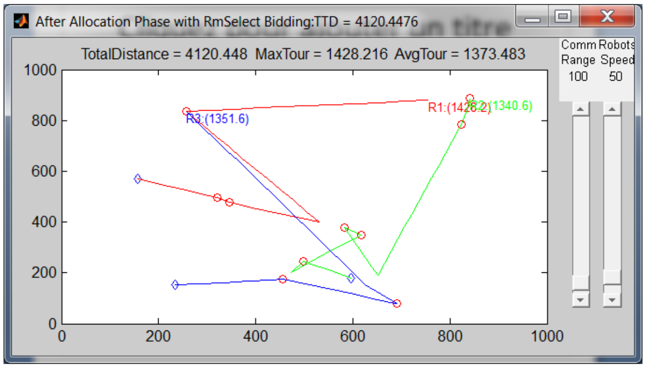
In the allocation phase, each robot receives the list of available targets to be visited. This list of targets can be sent, for example, by a control station. Each robot R_i maintains two lists: (*i.*) the list of *available* targets that are not already allocated, and (*ii.*) the list of *allocated* targets, which contains the targets visited and allocated to the robot R_i itself. At the start of the mission, all targets are marked as *available* in each robot, and the list of allocated targets is empty. Then, each allocated target will be removed from the available targets list and the robot informs its neighbors to also remove it from their available targets list. We note that the target becomes allocated to the robot R_i when it reaches the target. We also refer to an allocated target as a *visited* target. Then, each robot starts by computing the cost (e.g. the traveled distance) of visiting each available target, and finally selects the target with the lowest cost (e.g. the nearest target). Then, the robot R_i starts moving towards the selected target $T_{Selected}$, and in the meanwhile, it keeps discovering other robots in its neighborhood within its communication range, and exchanging information about allocated targets. Indeed, when a robot discovers another robot in its vicinity, they both exchange their lists of available targets that are not yet visited nor allocated. As such, each robot can update its own list of available targets by discarding those allocated/visited by other robots. In addition, if the currently selected target is found to be no longer available (i.e. it was already allocated to another robot), the robot dismisses that target and looks for another one . If the selected target $T_{Selected}$ is still not allocated to other robots, the robot bids on this target with its neighbor robots. The neighbor robots will send their costs to the robot R_i, which played the auctioneer role for target $T_{Selected}$, in response to the bidding request. Finally, the robot R_i will assign the target $T_{Selected}$ to the robot with the lowest cost, including itself .

This process of moving towards the selected target $T_{Selected}$ and exchanging available targets is continuously repeated until the robot R_i dismisses the selected target when it discovers that it is allocated to another robot, or until it reaches the selected target. In the latter case, it adds this target to its own list of allocated targets and removes it from its own list of available targets. The robot R_i repeats the process of selecting a new target and moving to it until the list of available targets becomes empty.

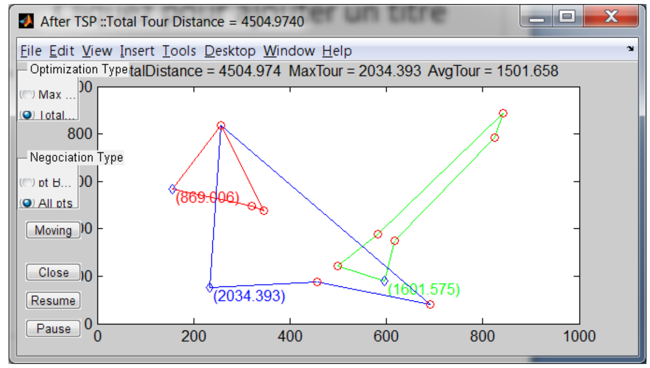
Upon the completion of the allocation process, each robot constructs an optimal tour across its allocated targets using a TSP solver (Phase 3). Once its list of available targets becomes empty, each robot R_i constructs a tour going through all targets in its allocated targets list. The robots can use any TSP tour construction algorithm proposed in the literature such as LKH [18] or [19]. In our MATLAB simulation, we used an existing implementation of a TSP solver based on Genetic Algorithms (GA) [20] for tour construction.

After the tour construction process, each robot generates a first solution to the MD-MTSP problem, and obtain a proper tour to follow through its allocated targets. However, it may happen that some targets would have been allocated to or visited by several robots during the allocation phase. This is possible as robots have limited communication ranges and may not have the opportunity to exchange their lists of allocated targets, either directly or indirectly through other neighbors.

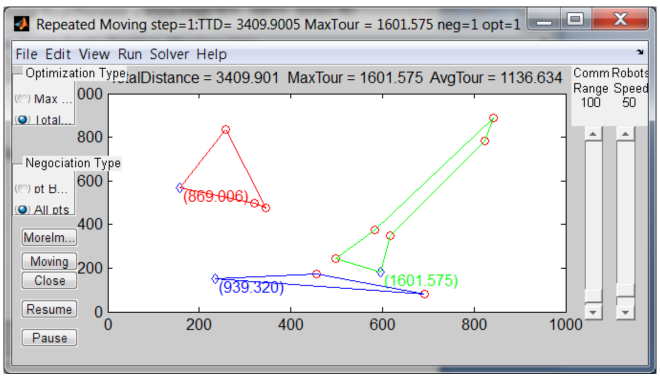
The objective of Phase 3 consists in improving the MD-MTSP solution by eliminating common targets allocated to more than one robot, through a distributed market-based ap-



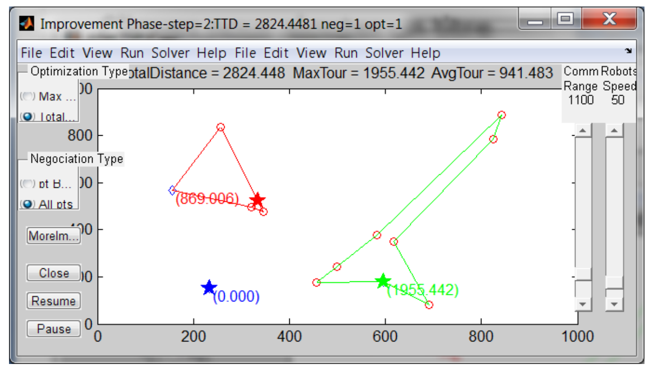
a) Targets Allocation



b) Tour Construction



c) Overlapped Targets Elimination



d) Solution Improvement

Figure 1: Move and Improve

proach, while the robots are moving through their constructed tours. As such, during their tour missions, when two robots are able to communicate, they exchange their lists of allocated targets, and in case there are one or more overlapped targets, the robots will bid on these targets. More precisely, considering a robot R_i and a robot R_j both have the same allocated target T_k. If the gain resulting from eliminating T_k from the tour of R_i is greater than the gain resulting from eliminating T_k from the tour of R_j, then T_k will be eliminated from the tour of R_i, and R_i computes a new tour based on its remaining allocated targets. The gain is defined as the difference between the old tour cost before removing the common target and the new tour cost after target removal. The gain differs by either we wish to minimize the total cost (the sum of traveled distance) or to minimize the maximal individual cost (the maximum tour). Moreover, in case there are several overlapped targets, the robots can bid on them separately (one by one) or all together. Returning to the previous example, where we wish to minimize the total cost, we define :

$$gain_i = C(Tour_{R_i}) - C(Tour_{R_i} \backslash T_k)$$
$$gain_j = C(Tour_{R_j}) - C(Tour_{R_j} \backslash T_k)$$

where $C(Tour_{R_i})$ is the cost of $Tour_{R_i}$ containing the target T_k and $C(Tour_{R_i} \backslash T_k)$ is the cost of $Tour_{R_i}$ after removing the target T_k. $C(Tour_{R_i} \backslash T_k)$ is computed using the TSP algorithm applied on the targets allocated to R_i, except the target T_k. $gain_i$ represents the gain obtained when we remove target T_k from the tour of robot R_i and $gain_j$ represents the gain obtained when we remove target T_k from the tour of robot R_j. If $gain_i > gain_j$, it is more beneficial to remove T_k from the tour of robot R_i and keep it in the tour of robot R_j. If the objective function was to minimize the maximum cost, the gain will be defined as follows:

$$\begin{aligned} gain_i =& \max(C(Tour_{R_i}), C(Tour_{R_j})) \\ &- \max(C(Tour_{R_i} \backslash T_k), C(Tour_{R_j})) \\ gain_j =& \max(C(Tour_{R_i}), C(Tour_{R_j})) \\ &- \max(C(Tour_{R_i}), C(Tour_{R_j} \backslash T_k)) \end{aligned}$$

The last phase (the solution improvement phase) consists in looking for possible additional optimization of the tours resulting from Phase 3. For that purpose, when a robot R_i enters in the communication range of a robot R_j, it sends to R_j a bidding request on the target T_k that induces the biggest cost to visit in its tour. In other words, T_k is the target that when we remove from the tour of robot R_i we get the minimal cost, compared to other allocated targets. Formally speaking,

T_k can be computed as:

$$T_k = \arg\min_{T_j} C(Tour_{Ri} \backslash T_j)$$

The robot R_i computes the gain obtained when it removes this target T_k from its tour and the neighbor robot R_j calculates the lost obtained when it adds this target T_k to its tour. In case of *MinSum*, the gain is computed as follows:

$$\begin{aligned} gain = & [C(Tour_{R_i}) + C(Tour_{R_j})] \\ & - [C(Tour_{R_i} \backslash T_k) + C(Tour_{R_j} \cup T_k)] \end{aligned}$$

$C(Tour_{R_j} \cup T_k)$ is computed using the TSP algorithm applied on the targets allocated to R_j union T_k. If the gain obtained by removing T_k from the R_i's tour is greater than the lost obtained when adding T_k to R_j's tour, the neighbor robot R_j wins this target T_k, and therefore, T_k will be removed from robot R_i's allocated targets, and added to the list of R_j's allocated targets. In case of *MinMax*, the gain is computed as follows:

$$\begin{aligned} gain = & \max(C(Tour_{R_i}), C(Tour_{R_j})) \\ & - \max(C(Tour_{R_i} \backslash T_k), C(Tour_{R_j} \cup T_k)) \end{aligned}$$

V. Performance Evaluation

A. Simulation Model

In this section, we present a comprehensive simulation study using MATLAB that we conducted to evaluate the performance of the Move-and-Improve market-based approach for solving the MD-MTSP problem in the context of mobile robots task allocation. We designed and implemented a complete MATLAB simulator of multi-robot task allocation for different problems including the MD-MTSP and the assignment problems. More details about the simulation tool are available in this link [21]. We evaluated the Move-and-Improve mechanism under several scenarios with different configurations of the number of robots, number of targets and communication ranges. Each scenario is characterized by a specified fixed number of robots, a fixed number of targets and a robot communication range. Each scenario is repeated 30 times where in each run, robots and targets locations are randomly generated. The average values of the metrics are then calculated with 95% of confidence interval.

We considered two evaluation metrics: The Total Traveled Distance (TTD) also known as *MinSum* metric (see Eq. 2), and the Maximum Tour (MT), also known as *MinMax* (see Eq. 3).

By default, we considered a full communication range between all the robots unless it is specified otherwise.

B. Simulation Results

1) Impact of the number of robots: Figure 2 shows the Total Traveled Distance (TTD) and Max Tour (MT) as a function of the number of robots, for a fixed number of targets. The targets and robots are randomly deployed in a 1000x1000 area in each scenario, and robots are assumed to have full communication range with each other. In Figure 2, TTD was presented with the solid line (Figure 2a) and Max Tour was presented in dotted line (Figure 2b). We observe that the Max Tour declines exponentially when the number of robots increases, since the targets become shared by a greater number of robots. On the other hand, the TTD slightly declines when increasing the number of robots. These two figures show the benefit of sharing the tasks among higher number of robots to significantly reduce the individual load (i.e. Maximum Tour) on each robot.

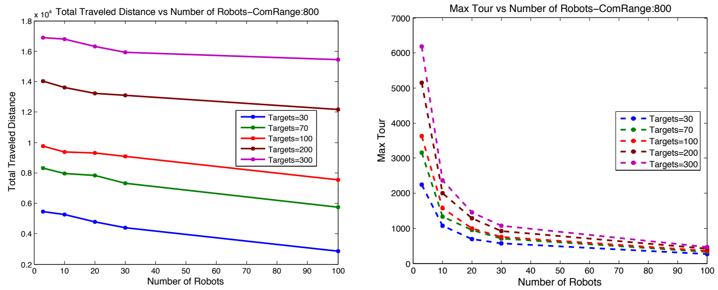
We also observe that when the number of robots increases, the number of targets has less effect on the maximum tour, whereas it significantly increases the TTD.

2) Impact of the number of targets: Figure 3 shows the Total Traveled Distance (TTD) and Max Tour (MT) as a function of the number of targets, for a fixed number of robots. The number of targets varies such that $Number\ of\ Targets = Number\ of\ Robots * [3, 5, 7, 10]$. The targets and robots are randomly deployed in an area of 1000x1000. Plots are shown for a full communication range equal to 800.

In Figure 3, we observe that the TTD (Figure 3a) and MT (Figure 3b) linearly increases with the number of targets and decreases with the number of robots. The results confirms those of Figure 2 as the TTD is more sensitive to the increase of the number of targets than to the increase of the number of robots.

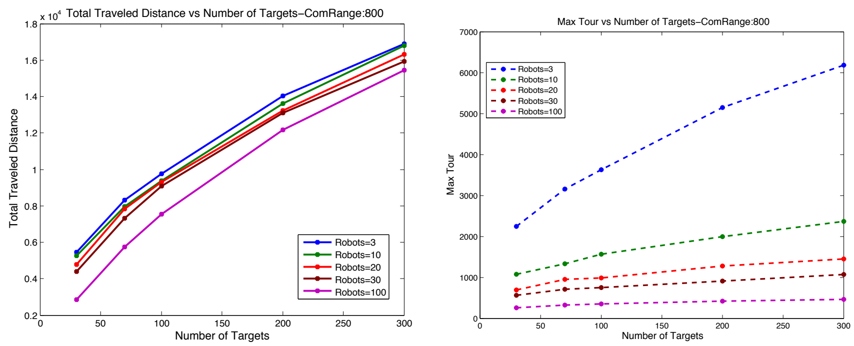
3) Impact of the communication range: The number of target locations was fixed to 300 and the number of robots varies in the interval [3, 10, 20, 30, 100]. From Figure 4, a first observation is that the general trend of TTD and Max Tour is to decrease when the communication range increases. This is expected as a higher communication range allows the robots to avoid overlapped targets and thus reduces the distance they will have to move. We also observe that for very sparse density of robots (3 robots for 300 target locations), the impact of communication range on the TTD is very small, but is significant on the maximum tour. This means that a bad communication leads to increasing the load of some robots over the other while the overall distance traveled by all the robots remains stable. For high density of robots (100 for 300 target) the communication range has no impact on the maximum tour and a small impact of the TTD. This is explained by the fact that robots are close to each other and they coordinate with several neighbors even if with a limited communication range, helping to avoid overlapping targets as in case of full communication range. For medium density of robots (10, 20 and 30 for 300 targets), it is clear that the small communication ranges lead to increasing the TTD and the maximum tour. We conclude that the Move-and-Improve mechanism allows to achieve good performance in terms of reducing the TTD and Max Tour when the coordination between the robots is effective.

4) Comparison with the GA centralized approach: In order to evaluate the performance of our solution in terms of solution quality and execution time (i.e. simulation time), we compared it against a central approach, namely MDMTSP-GA [16], which is based on Genetic Algorithms. The MDMTSP-GA solution was simulated in MATLAB, with a population size equal to 240 and number of iteration equal to 1e4. Note that, these



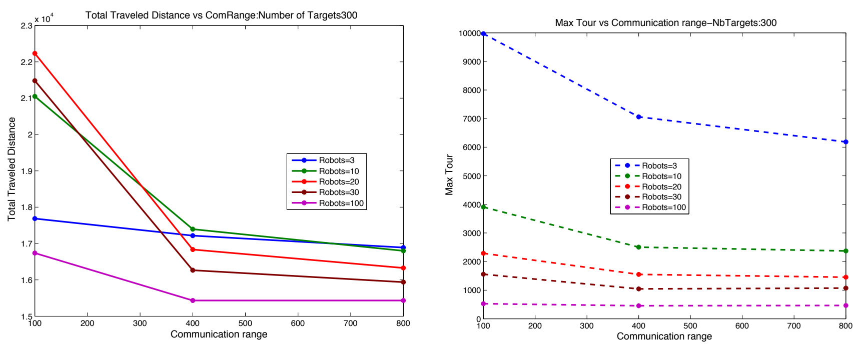
(a) Total Traveled Distance (b) Max Tour

Figure 2: Impact of the number of robots on TTD and MT (with fixed number of targets)



(a) Total Traveled Distance (b) Max Tour

Figure 3: Impact of number of targets on TTD and MT



(a) Total Traveled Distance (b) Max Tour

Figure 4: Impact of communication range on TTD and MT

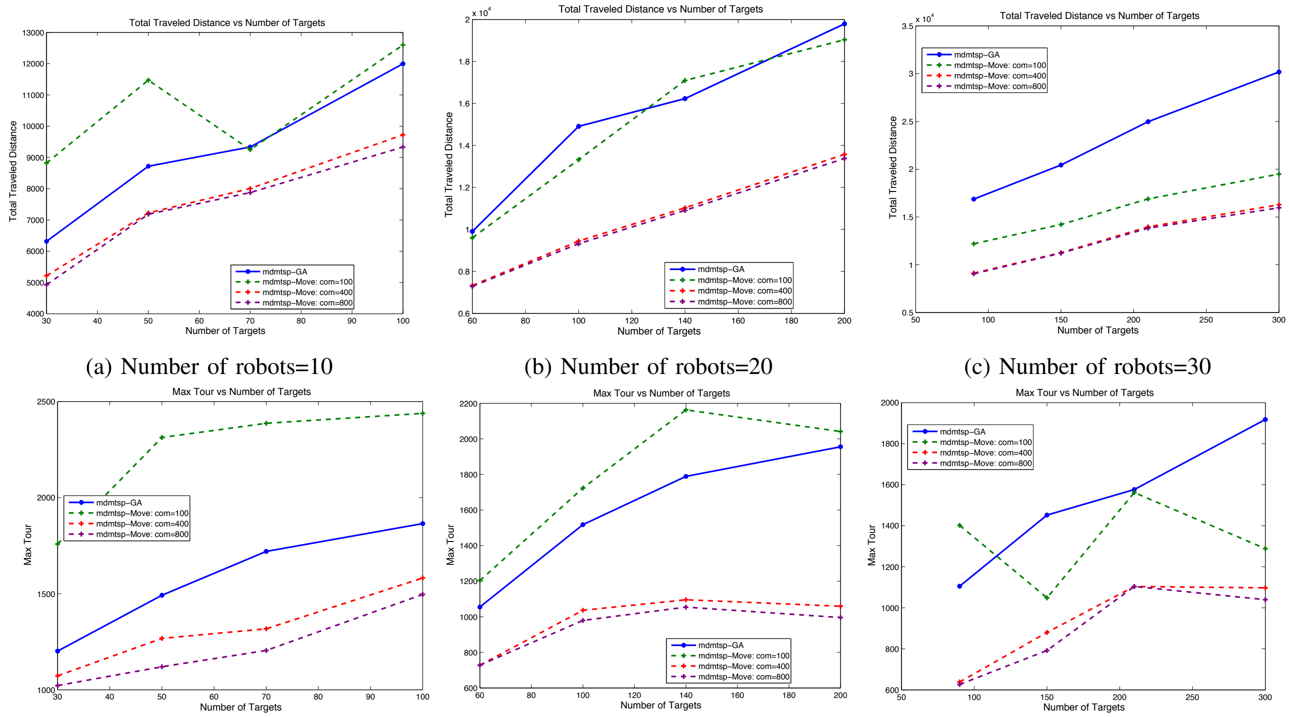
parameters are enough large for the GA solution to provide relatively good results. In Figure 5 and Figure 6, we observe that Move-and-Improve distributed approach outperforms the centralized GA approach in terms of solution quality and execution time, in particular for large instances. Indeed, the MD-MTSP problem becomes intractable with large number of robots and targets, and the solutions space is quite huge, which renders the centralized search more complex. However, the heuristics used in the Move-and-Improve mechanism allows to converge faster to better solutions than those of GA.

The gap between Move-and-Improve and the GA approach reduces with decreasing the communication range, and by decreasing the size of the problem (smaller number of targets and robots). Indeed, with limited communication ranges, robots can only have partial information, which leads to more overlapped targets, which increases the TTD of the Move-and-Improve mechanism. This is confirmed by the results of Figure 4.

In conclusion, the results show that the Move-and-Improve mechanism provides a good performance in terms of solution quality and execution time.

VI. Conclusions and Future Work

In this paper, we have addressed the multiple depot multiple traveling salesman problem. We presented a market-based distributed approach, where robot progressively select targets



(a) Number of robots=10

(b) Number of robots=20

(c) Number of robots=30

(d) Number of robots=10

(e) Number of robots=20

(f) Number of robots=30

Figure 5: Comparison between our solution and a Genetic Algorithm based central approach

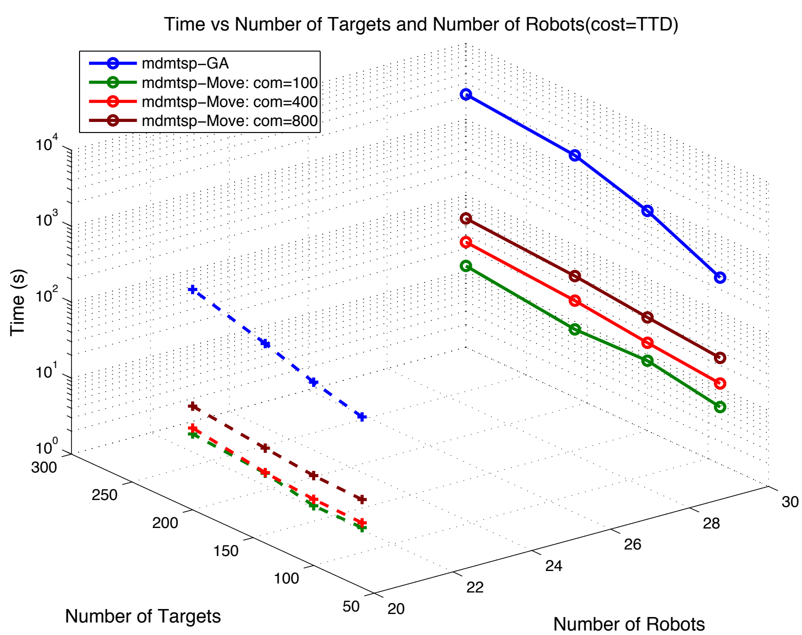


Figure 6: Time Comparison between MDMTSP-GA and MDMTSP-Move and Improve

and then bid on the selected target while moving to it. Our solution is responsive as robots start visiting targets as they are initiated. Comparison with a centralized approach based on Genetic Algorithm shows that our solution gives better results with shorter execution time.

We are currently working towards the completion of our implementation of the Move-and-Improve mechanism using the Robotic Operating System (ROS) to demonstrate the effectiveness of our approach in real-world deployment. We intend to use Move-and-Improve for sensor data collection using a team of robots. This will allow us to evaluate the performance of our approach in terms of communication efficiency and understand real-world challenges of the deployment. Moreover, we have simulated the Move-and-Improve mechanism using the Webots simulator and some scenarios are report in [17].

ACKNOWLEDGEMENTS

This work is supported by the iroboapp project "Design and Analysis of Intelligent Algorithms for Robotic Problems and Applications"[22] under the grant of the National Plan for Sciences, Technology and Innovation (NPSTI), managed by the Science and Technology Unit of Al-Imam Mohamed bin Saud University and by King AbdulAziz Center for Science and Technology (KACST). This work is partially supported by Prince Sultan University.

REFERENCES

[1] D. Jung, G. Cheng, and A. Zelinsky, "Robot cleaning: An application of distributed planning and real-time vision," in *Field and Service Robotics* (A. Zelinsky, ed.), pp. 187–194, Springer London, 1998.

[2] C. Luo and S. X. Yang, "A real-time cooperative sweeping strategy for multiple cleaning robots," in *Proceedings of the 2002 IEEE International Symposium on Intelligent Control, 2002.*, pp. 660–665, 2002.

[3] M. Di Francesco, S. K. Das, and G. Anastasi, "Data collection in wireless sensor networks with mobile elements: A survey," *ACM Transactions on Sensor Networks (TOSN)*, vol. 8, no. 1, p. 7, 2011.

[4] S. Trigui, O. Cheikhrouhou, A. Koubaa, and H. Youssef, "Distributed market-based algorithm for multi-robot assignment problem," in *The International Workshop on Cooperative Robots and Sensor Networks*, June 2 - 5 2014.

[5] S. Trigui, A. Koubaa, M. Ben Jamaa, I. Chaari, and K. Al-Shalfan, "Coordination in a multi-robot surveillance application using wireless sensor networks," in *16th IEEE Mediterranean Electrotechnical Conference (MELECON), 2012*, pp. 989–992, March 2012.

[6] "Planet project: Platform for the deployment and operation of heterogeneous networked cooperating objects." http://www.planet-ict.eu/, 2014.

[7] P. Oberlin, S. Rathinam, and S. Darbha, "A transformation for a multiple depot, multiple traveling salesman problem," in *Proceedings of the 2009 Conference on American Control Conference*, ACC'09, (Piscataway, NJ, USA), pp. 2636–2641, IEEE Press, 2009.

[8] P. Fazli, A. Davoodi, and A. K. Mackworth, "Multi-robot repeated area coverage," *Auton. Robots*, vol. 34, pp. 251–276, May 2013.

[9] M. G. Lagoudakis, E. Markakis, D. Kempe, P. Keskinocak, A. J. Kleywegt, S. Koenig, C. A. Tovey, A. Meyerson, and S. Jain, "Auction-based multi-robot routing.," in *Robotics: Science and Systems*, vol. 5, 2005.

[10] S. Sariel, N. Erdogan, and T. Balch, "An integrated approach to solving the real-world multiple traveling robot problem," in *5th International Conference on Electrical and Electronics Engineering*, 2007. http://www2.itu.edu.tr/s̃ariel/publications.php.

[11] H.-L. Choi, L. Brunet, and J. P. How, "Consensus-based decentralized auctions for robust task allocation," *Robotics, IEEE Transactions on*, vol. 25, no. 4, pp. 912–926, 2009.

[12] R. K. Karmani, T. Latvala, and G. Agha, "On scaling multi-agent task reallocation using market-based approach," in *Self-Adaptive and Self-Organizing Systems, 2007. SASO'07. First International Conference on*, pp. 173–182, IEEE, 2007.

[13] E. Kivelevitch, K. Cohen, and M. Kumar, "A market-based solution to the multiple traveling salesmen problem," *Journal of Intelligent & Robotic Systems*, pp. 1–20, 2013.

[14] A. J. Kulkarni and K. Tai, "Probability collectives: a multi-agent approach for solving combinatorial optimization problems," *Applied Soft Computing*, vol. 10, no. 3, pp. 759–771, 2010.

[15] P. Oberlin, S. Rathinam, and S. Darbha, "A transformation for a multiple depot, multiple traveling salesman problem," in *American Control Conference, 2009. ACC'09.*, pp. 2636–2641, IEEE, 2009.

[16] E. Kivelevitch, K. Cohen, and M. Kumar, "Comparing the robustness of market-based task assignment to genetic algorithm," in *Proceedings of the 2012 AIAA Infotech@ Aerospace Conference. AIAA, AIAA. AIAA-2012-2451*, 2012.

[17] "Webots simulation scenarios,." http://www.iroboapp.org/index.php?title=Videos, 2014.

[18] S. Lin and B. W. Kernighan, "An effective heuristic algorithm for the traveling-salesman problem," *Operations research*, vol. 21, no. 2, pp. 498–516, 1973.

[19] H. Braun, "On solving travelling salesman problems by genetic algorithms," in *Parallel Problem Solving from Nature*, pp. 129–133, Springer, 1991.

[20] J. Kirk, "traveling-salesman-problem-genetic-algorithm." http://www.mathworks.com/matlabcentral/fileexchange/13680-traveling-salesman-problem-genetic-algorithm, 11 2011.

[21] "MRS: The multi-robot simulator." http://www.iroboapp.org/index.php?title=MRTAsim, 2014.

[22] "iroboapp project." http://www.iroboapp.org, 2014.