CISTER
Research Centre in
Real-Time & Embedded
Computing Systems

Conference Paper

A Clustering Market-Based Approach for Multi-robot Emergency Response Applications

Sahar Trigui
Anis Koubâa
Omar Cheikhrouhou
Basit Qureshi
Habib Youssef

CISTER-TR-180106

A Clustering Market-Based Approach for Multi-robot Emergency Response Applications

Sahar Trigui, Anis Koubâa, Omar Cheikhrouhou, Basit Qureshi, Habib Youssef

*CISTER Research Centre

Polytechnic Institute of Porto (ISEP-IPP)

Rua Dr. António Bernardino de Almeida, 431

4200-072 Porto

Portugal

Tel.: +351.22.8340509, Fax: +351.22.8321159

E-mail:

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

Abstract

In this paper, we address the problem of multi-robot systems in emergency response applications, where a team of robots/drones has to visit affected locations to provide rescue services. In the literature, the most common approach is to assign target locations individually to robots using centralized or distributed techniques. The problem is that the computation complexity increases significantly with the number of robots and target locations. In addition, target locations may not be assigned uniformly among the robots. In this paper, we propose, CMMTSP, a clustering market-based approach that first groups locations into clusters, then assigns clusters to robots using a market-based approach. We formulate the problem as multiple-depot MTSP and address the multi-objective optimization of three objectives namely, the total traveled distance, the maximum traveled distance and the mission time. Simulations show that CM-MTSP provides a better balance among the three objectives as compared to a single objective optimization, in particular an enhancement of the mission time, and reduces the execution time to at least 80% as compared to a greedy approach.

A clustering market-based approach for multi-robot emergency response applications

Sahar Trigui *,Anis Koubâa§‡, Omar Cheikhrouhou‖¶, Basit Qureshi§‡, Habib Youssef**

* University of Manouba, ENSI, Tunisia.
§ Prince Sultan University, Saudi Arabia.
‡ CISTER/INESC-TEC, ISEP, Polytechnic Institute of Porto, Porto, Portugal.
‖ Taif University, Saudi Arabia.
¶ ISIMA, University of Monastir, Tunisia.
** University of Sousse, PRINCE Research Unit, Tunisia.

Emails: sahar.trigui@coins-lab.org, aska@isep.ipp.pt, o.cheikhrouhou@tu.edu.sa
qureshi@psu.edu.sa, Habib.youssef@fsm.rnu.tn

***Abstract*—In this paper, we address the problem of multi-robot systems in emergency response applications, where a team of robots/drones has to visit affected locations to provide rescue services. In the literature, the most common approach is to assign target locations individually to robots using centralized or distributed techniques. The problem is that the computation complexity increases significantly with the number of robots and target locations. In addition, target locations may not be assigned uniformly among the robots. In this paper, we propose, CM-MTSP, a clustering market-based approach that first groups locations into clusters, then assigns clusters to robots using a market-based approach. We formulate the problem as multiple-depot MTSP and address the multi-objective optimization of three objectives namely, the total traveled distance, the maximum traveled distance and the mission time. Simulations show that CM-MTSP provides a better balance among the three objectives as compared to a single objective optimization, in particular an enhancement of the mission time, and reduces the execution time to at least 80% as compared to a greedy approach.**

I. INTRODUCTION

Multi-robot coordination has been a major challenge in robotics with a vast array of applications. Emergency response represents one attractive application of multiple robot coordination, where a team of robots coordinates to visit locations affected by the disaster ([1], [2]). Several papers addressed this problem [3], [4] and proposed solutions, which can be classified from two perspectives: (1) Algorithmic approaches, which can be either centralized or distributed (2) optimization problem type, which can be either single objective or multi-objective. Centralized approaches typically rely on evolutionary algorithms [5], [6], which have the advantage of converging to good solutions, but at the cost of intensive computation requirements and long execution times. On the other hand, distributed approaches, including market-based techniques [7], [8], provide lower-quality solutions in general, but executes much faster. The complexity of these approaches significantly increases when the problem switches from single-objective to multi-objective optimization, where the goal is to optimize several metrics that can be conflicting in nature. This class of problems usually has more than one optimal solution that provides a balanced optimization of the different metrics, also known as Pareto-optimal solutions ([9]). Evolutionary algorithms are the typical approaches to solve this type of multi-objective optimization problems to find the Pareto-optimal solutions. However, the execution time is too long when the problem scales, making it non appropriate for situations where an efficient solution is needed in (near) real-time. This work addresses this gap and investigates the problem of assigning locations to robots in emergency response applications while meeting three requirements (*i.*) optimizes multiple objectives, including total traveled distance, maximum traveled distance and mission time for a heterogeneous team of robots, (*ii.*) provides an efficient solution (*iii.*) ensures low execution times and reduces the complexity of the problem even when it scales. Our solution relies on clustering the target locations and then the robots perform bid on clusters rather than on individual targets. The motivation behind this strategy is two-folded: first, we reduce the problem size from a large number of targets to a much smaller number of clusters to be assigned to robots, which intuitively leads to reducing the execution time to find an appropriate solution; second, grouping targets into clusters will improve uniformity of number of targets assigned to each robot, as individual target assignment may lead to assigning more target locations to some robots and preventing others [7]. Our approach uses a market-based approach, allowing robots to bid on the clusters, and be assigned to the cluster that optimizes the global mission objective. Finally, the proposed approach, includes an improvement phase to optimize the solution by switching targets between robots if necessary.

The remainder of this paper is as follows. Section II presents an overview on related works and discusses the contribution of this paper as compared to previous works. Section III describes the system model. Section IV presents the CM-MTSP algorithm and an illustrative example. Section V presents the performance evaluation study and discusses the results. Finally, Section VI concludes the paper.

978-1-5090-2255-7/16 $31.00 © 2016 IEEE
DOI 10.1109/ICARSC.2016.14

II. Related Works

Multiple traveling salesmen problem (MTSP) [10] is the problem of visiting a set of cities by a set of salesmen who all start and end at the same city such that each city must be visited exactly once with the objective of minimizing the total cost of visiting all cities. [10] provided a comprehensive survey on the MTSP and its applications. Also, it provided some formulations of the MTSP and described exact and heuristic solution procedures proposed for solving this problem.
In [8], the authors proposed a market-based approach to solve the multiple depot MTSP. The algorithm consists of four steps: market auction, agent-to-agent trade, agent Switch and agent relinquish step. The performance of the algorithm was measured in terms of the quality of the solution, the number of iterations required to get a solution and the execution time. It was shown that the solution gives good result in comparison with other sub-optimal solution. Also, in [7], the authors proposed a market based solution called *move and improve* to solve the multiple depot MTSP. The solution consists of four steps: initial target allocation, tour construction, elimination of conflicting targets and solution improvement. From the simulation study, it was shown that the move and improve algorithm gives good results compared with the results generated by a centralized approach. However, the solutions proposed in [7] and [8] seek to optimize a single objective at a time (i.e. even the MinMax or the MinSum). As several real world applications need to optimize multiple objectives, in our work, we addressed the multi objective problem and we propose a solution that optimizes several objectives simultaneously, including the mission time, the total traveled distance and the maximum traveled distance.

Clustering methods are recently used to deal with optimization problems. In [11], the authors tackled the multi-robot task allocation problem considering the objective of minimizing the distance traveled by all the robots and balancing the workload between the robots equally. They proposed a solution using K-means clustering method and an auction process. First, N tasks are decomposed into n groups in such a way the distance inside each cluster is minimized. Then, the cost for each robot to visit the n clusters is computed and finally, each robot is assigned to a cluster using an auction mechanism. However, the complexity of the algorithm is relatively high because there is a need to bid on all possible combinations of clusters for robots and thus the complexity of the algorithm increases with the increase of the number of clusters. For the performance evaluation, the authors proposed a scenario of 2 robots and 32 tasks. They used the benchmark VRP data set A-n32-K5.vrp [12]. The total cost used to assign a cluster is equal to the sum of the cost of visiting the tasks in the cluster and the idle cost (i.e. sum of the difference in cost of travel between any two robots). They performed two analysis: one with 2 clusters and the other with 3 clusters. The results presented do not signify the efficiency of the method as it was applied in a small scenario.

Another auction algorithm using a clustering technique has been presented in [13]. The authors aim to achieve two objectives: minimizing both the maximum traveled distance of each robot and the sum of distance traveled by all robots in visiting their assigned locations. They assumed that the robot are homogeneous. Initially, all robots have a list of allocated tasks. When a robot reaches the position of a first task, it sends a signal for all the other robots to start their auction. Upon the completion of all auctions, the robot re-plan its path and move to the next task. When a robot receives a message to start an auction, it forms a new set of clusters of its assigned tasks. Then each robot makes an auction for the new clusters except the cluster that contains its currently initialized task. If a robot receives an auction for a cluster, it bids for that cluster. In the winner-determination stage, the robots with the lowest bid wins the cluster. For the performance evaluation, the authors only have shown the percentage of improvement of the initial assignment as compared to the final assignment.

Most of the earlier research works have proposed solutions to solve the MTSP problem aiming to minimize even the total traveled distance by all agents or the maximum traveled distance by any agent ([7], [8]). In our work, we considered a multi-objective problem where we need to simultaneously optimize several objectives (Section III). We seek to find solutions that keep up the trade-off between the objectives. We propose a Clustering Market-based approach (CM-MTSP) to solve the multiple depot MTSP with the objective of optimizing simultaneously three performance criteria, namely the total traveled distance, the maximum traveled distance and the mission time. The approach incrementally improves the assignment and provides a solution that optimizes the conflicting objectives.

III. System Model

In this paper, we are interested to solve the multi-objective multiple depot multiple traveling salesman problem. Considering a set of n robots located at different positions and a set of m target locations. Each robot has to visit its assigned target locations and then returns to its initial position. The objective is to find an effective assignment of robots to the set of locations such that each target is only visited by one robot. In addition, targets must be assigned in a uniformed way such that the number of allocated targets for each robot is equal or close.
The problem can be formulated as follows. Consider a set of n robots $\{r_1, \ldots, r_n\}$ responsible to reach a set of $m > n$ target locations $\{t_1, \ldots, t_m\}$. The robots are initially located at different positions $\{p_{r_1}, \ldots, p_{r_n}\}$. We define $tour_{r_i}$ as the tour of robot i starting from and ending at its initial position p_i and going through its list of k allocated targets $\{t_{i_1}, \ldots, t_{i_k}\}$. The tour length of the robot r_i is expressed as:

$$Length(tour_{r_i}) = distance(p_{r_i}, t_{i_1}) + \sum_{j=1}^{k} distance(t_{i_j}, t_{i_{j+1}}) + distance(t_{i_k}, p_{r_i}) \quad (1)$$

k is the number of target locations assigned to robot r_i. $distance(t_{i_j}, t_{i_{j+1}})$ represents the Euclidean distance between target location j and target location $j+1$ for robot r_i.

t_{i_1} and t_{i_k} represent the first and the last target locations respectively for robot r_i. $distance(p_{r_i}, t_{i_1})$ represents the Euclidean distance between r_i and t_{i_1}, and $distance(t_{i_k}, p_{r_i})$ represents the Euclidean distance between r_i and t_{i_k}.
In the context of multi objective optimization problem, several objectives must be minimized. We are interested to optimize the following objectives:

1) The total traveled distance: We define TTD as the sum of all tour length performed by all the robots. The tour length of a robot is calculated by summing up the traveled distance of all edges included in that tour. The TTD is given according to Equation 2.

$$TTD = \sum_{i=1}^{n} Length(tour_{r_i}) \quad (2)$$

2) The maximum traveled distance MTD: it is the maximum distance traveled by any robot after the schedule mission is completed. The MTD is expressed as:

$$\begin{array}{c} MTD = max(Length(tour_{r_i})) \\ 1 \leq i \leq n \\ s.t. \;\; tour_{r_i} \neq tour_{r_j} \\ 1 \leq j \leq n, \;\; i \neq j \end{array} \quad (3)$$

3) The mission time of each robot T_{r_i} is the time necessary for the tour completion of each robot.

$$T_{r_i} = \frac{Length(tour_{r_i})}{S_{r_i}} \quad (4)$$

$Length(tour_{r_i})$ represents the tour length of robot r_i as expressed in Equation 1 and S_{r_i} represents the speed of robot r_i.

We assume that robots have different capabilities, including different speeds.

IV. CM-MTSP: Clustering Marked-Based Coordination

The CM-MTSP solution consists in a hybrid approach for solving the multiple depot MTSP problem that combines a clustering technique with a market-based approach with the objective of minimizing the TTD, MTD and the mission time as mentioned in Section III. We define two main roles for the agents (i.e. the server and the robots): *auctioneer* and *bidders*. The auctioneer agent is responsible for announcing tasks and assigning each task to the agent with the best bid. In our work, a central server acts as an auctioneer and the robots as the bidders. We assume that the server initially identifies m target locations to be visited by the robots.

A. CM-MTSP algorithm steps

The CM-MTSP algorithm includes three steps: the clustering step, the auction-based step and the improvement step as illustrated in Algorithm 1.

Algorithm 1. The CM-MTSP Algorithm

1: **Inputs:** Robots r_i $(1 < i < n)$, Targets t_j $(1 < j < m)$, Robots speed S_{r_i} $(1 < i < n)$
2: Clustering step
3: Auction-based step
4: Improvement step
5: **Outputs:** Assignment $(r_i$,$c_j)$ $1 < i, j < n$, TTD, MTD, mission time

1) Clustering step: The server first provides n clusters of locations to be visited such that the number of targets in each cluster is equal as much as possible (Algorithm 2). We used the K-means technique which is one of the most popular methods used to solve clustering problems [14]. K-means provides a partition in which elements in the same cluster are as close as possible and as far as possible from the elements in the other clusters.

Algorithm 2. Clustering step

1: **Inputs:** Robots r_i $(1 < i < n)$, Targets t_j $(1 < j < m)$
2: Build clusters using k-means clustering method
3: **Outputs:** Clusters c_i $(i < 1 < n)$

2) Auction-based step: The market process begins with an announcement phase. After forming n clusters of targets, the server announces the clusters, one by one. Note that the auction for the second cluster does not begin until the auction for the first cluster is completed. Each robot computes its bid for the announced cluster and submits this bid for the server. The cost of a robot to bid for a cluster is defined as the time necessary for that robot to visit all locations in that cluster following Equation 4 and return to its initial location. In the case where a robot has a previously assigned cluster, it can ask to exchange its cluster if it discovers that the new cluster being auctioned has a lower cost to what was assigned to it. In the winner-determination stage, the server evaluates the received bids and allocates the cluster to the robot which leads to get the lowest total cost. Suppose that the server receives an exchange message from a robot which provides the best bid. In this case, the server assigns the announced cluster to that robot and add the old cluster to its non-allocated list of clusters. This process is repeated until all clusters are allocated to the robots (Algorithm 3).

3) Improvement step: The improvement step consists in the permutation of clusters between robots in order to provide a good assignment solution that simultaneously optimizes the TTD, MTD and the mission time (Algorithm 4). This step includes two sub-steps. The first consists in minimizing the mission time that results from the auction-based step. The server selects the robot that provides the maximum time and search for the permutation that leads to decrease that time. This step is repeated while there is an improvement in terms of mission time. The second sub-step leads to optimize the MTD while conserving or even minimizing the mission time. The server selects the robot that provides the maximum traveled distance and search for the permutation that leads to decrease

Algorithm 3. Auction-based step

Inputs: Robots r_i $(1 < i < n)$, Targets t_j $(1 < j < m)$, Robots speed V_{r_i} $(1 < i < n)$, Clusters c_i $(1 < i < n)$
For each cluster c_i **do**
  For each robot r_i **do**
    Robot r_i computes $cost(r_i, c_i)$
    If $list_{targets}(r_i) = \emptyset$ **do**
      robot r_i sends a $bidding\_msg(c_i)$
    else
      If $(cost(r_i, c_i) < cost(r_i, old\_cluster(c_j)))$ **do**
        robot r_i sends a $exchange\_msg(c_j)$
        robot r_i sends a $bidding\_msg(c_i)$
      else
        robot $r_i \leftarrow c_j$
      end
    end
  end
  while $received\_msg = true$ **do**
    If $exchange\_msg(c_j)$ **do**
      If $TTD_{r_i}(c_i) < TTD_{r_i}(c_j)$ **do**
        $r_i \leftarrow c_i$, $list_{unassigned\_clusters} \leftarrow c_j$
      end
    else
      $best\_robot \leftarrow c_i$
    end
  end
end
Outputs: Assignment $(r_i$,$c_j)$ $(1 < i, j < n)$, TTD, MTD, mission time

Algorithm 4. Improvement step

Function improve-time-cost
Inputs: Robots r_i $(1 < i < n)$, Targets t_j $(1 < j < m)$, Assignment $(r_i$,$c_j)$ $(1 < i, j < n)$
While $improve\_time = true$ **do**
  Select c_{max} of r_{max} with T_{max}
  For $r_i \neq r_{max}$ **do**
    If $(cost(r_i, c_{max}) < T_{max})$&$(cost(r_{max}, c_k) < T_{max})$ **do**
      $r_i \leftarrow c_{max}$, $r_{max} \leftarrow c_k$
    end
  end
end
Outputs: Assignment $(r_i$,$c_k)$ $(1 < i, k < n)$, TTD, MTD, mission time

Function improve-distance-cost
Inputs: Robots r_i $(1 < i < n)$, Targets t_j $(1 < j < m)$, Assignment $(r_i$,$c_k)$ $(1 < i, k < n)$
While $improve\_distance = true$ **do**
  Select c_{max} of r_{max} with MTD
  For $r_i \neq r_{max}$ **do**
    If $(cost(r_i, c_{max}) < T_{max})$&$(cost(r_{max}, c_k) < T_{max})$ $(distance(r_i, c_{max}) < MTD)$& $(distance(r_{max}, c_k) < MTD)$& **do**
      $r_i \leftarrow c_{max}$, $r_{max} \leftarrow c_k$
    end
  end
end
Outputs: Assignment $(r_i$,$c_k)$ $(1 < i, k < n)$, TTD, MTD, mission time

that distance with respect to the mission time. This step is repeated while there is improvement into the MTD.

B. Illustrative example

Considering a system with two robots (r_1 and r_2) and five targets (t_1, t_2, t_3, t_4, and t_5) as shown in Figure 1. We define the speed of robot r_i as S_{r_i}. For each robot, the speed is generated randomly in the range [0, 10]. Note that in this example $S_{r_1} > S_{r_2}$. The set of targets is decomposed into two clusters: c_1 (t_2 and t_5) and c_2 (t_1, t_3 and t_4). The bidding cost for each cluster is calculated using Equation 4. Table I shows the cost calculation obtained for each robot. The server starts an auction for the cluster c_1 and both r_1 and r_2 send their bids. The server assigns c_1 to r_1 ($T_{r_1}(c_1) < T_{r_2}(c_1)$), then makes an auction for c_2. As $T_{r_1}(c_1) < T_{r_1}(c_2)$, the robot r_1 keeps his cluster c_1 and c_2 will be assigned to the robot r_2 (Figure 1a). In the improvement step, the server tries to improve the assignment. It is clear that the permutation of clusters between robots r_1 and r_2 minimizes the time cost, so c_1 will be assigned to r_2 and c_2 will be assigned to r_1 (Figure 1b) .

V. Performance Evaluation

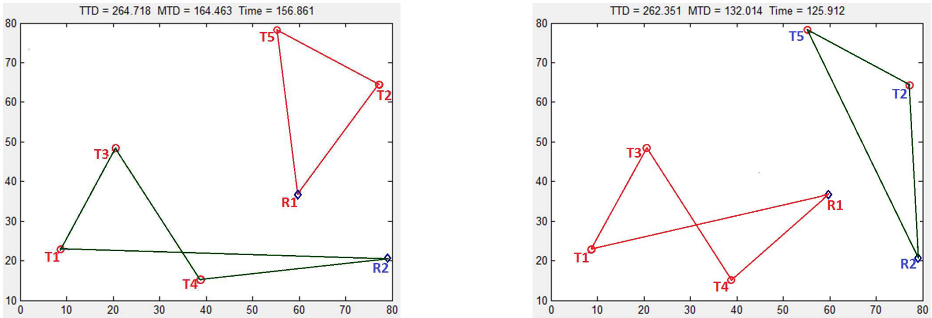
In this section, we present the performance evaluation of the clustering market-based approach for solving the multiple depot MTSP. We build our simulation using MATLAB. We adopted the test problems where the number of target locations varies in [20, 50, 100] and the number of robot varies in [3, 6, 12]. Robots and targets positions are placed in the range of [0, 100]. The LKH-TSP solver [15] is used to find the least distance for the robot to travel from a fixed starting point while visiting the target locations exactly once. The LKH-TSP solver has shown its ability to produce optimal solutions to most problems. Also, the LKH-TSP is tractable for large-scale problems and can generate solutions within a small execution time [16], [17]. For each scenario, we performed 30 different runs of the algorithm, and each run with different clusters. We evaluated the TTD, the MTD and the mission time objectives. We explored the performance of the proposed algorithm with varying the number of robots and targets.

TABLE I: Bids on clusters c_1 and c_2 in terms of time.

	$T(c_1)$	$T(c_2)$
Robot r_1	24.7352	32.1570
Robot r_2	125.9122	156.8610

A. Comparison of the CM-MTSP with a single objective algorithm

The efficiency of our solution is validated through comparison with a clustering single objective market-based algorithm (CSM-MTSP). In CSM-MTSP, the process is the same as in the CM-MTSP, but the server uses the TTD as a unique cost metric to assign clusters to robots. Figures 2, 3 and 4 show the total traveled distance, the maximum traveled distance and the mission time,respectively, for both the CM-MTSP and CSM-MTSP algorithms. As shown in Figures 2, 3, the gap between the CM-MTSP and CSM-MTSP in terms of TTD and MTD is in the range of [**1%, 10%**] for all scenarios in favor of the single objective. On the other hand, the gap between CM-MTSP and CSM-MTSP in terms of mission



(a) After auction-based step

(b) After improvement step

Fig. 1: Illustrative example. 2 robots (blue squares) and 5 target locations (red circles).

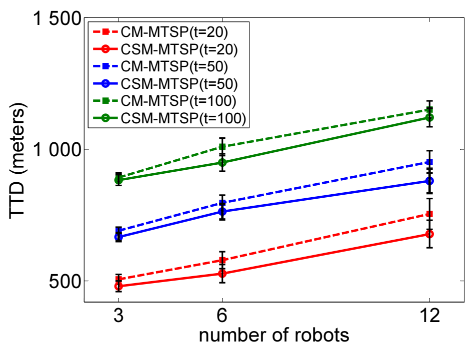


Fig. 2: The TTD of CM_MTSP and CSM_MTSP solutions.

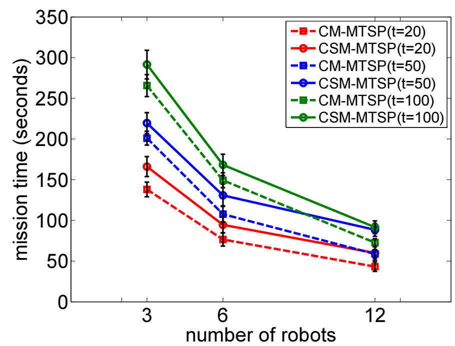


Fig. 4: mission time of CM_MTSP and CSM_MTSP.

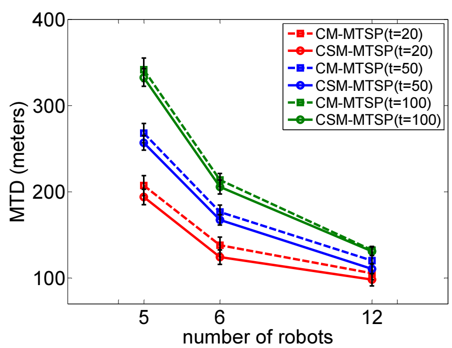


Fig. 3: The MTD of CM_MTSP and CSM_MTSP solutions.

time is in the range of [**8%, 35%**], with a more significant gain for the multi-objective CM-MTSP. This demonstrates that CM-MTSP provides a better tradeoff in satisfying the different objectives as compared to CSM-MTSP. The reason of having reduced mission time is that CM-MTSP form clusters with the objective to group target locations close to each other so that to reduce the maximum tour of the robot, and also assigns clusters based on the mission time, so that slower robots are assigned to clusters with shortest tours. In addition, the improvement phase of the CM-MTSP further optimizes the mission time. This feature is more interesting for applications with real-time constraints such as emergency response applications.

In addition, the decrease of the maximum traveled distance with the increase of the number of robots indicates that there is a uniformity of assignment that means that the number of allocated targets for each robot is equal or close. Figure 5 shows an example of distribution of targets between robots.

B. Comparison of the CM-MTSP with a greedy algorithm

To prove the efficiency of the clustering approach, we compare its performance against a greedy market-based algorithm that allocates targets to robots one by one without prior clustering. The concept of the greedy market-based algorithm is similar to the CM-MTSP algorithm, but we consider individual target allocations instead of cluster allocations.

The algorithm works as follows. The server starts an auction for each target t_i. In the bidding phase, each robot can either bid for a new target t_i, or try to exchange its worst allocated target that leads to increase the mission time of its tour with the announced task t_i. At the end, the server assigns the target

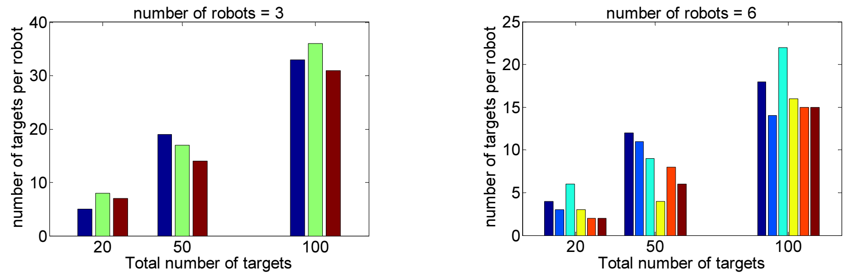
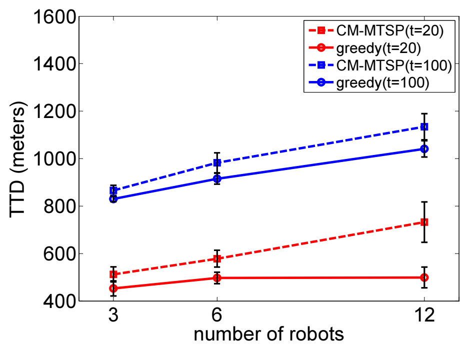
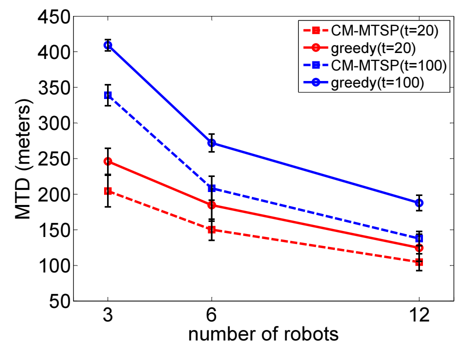


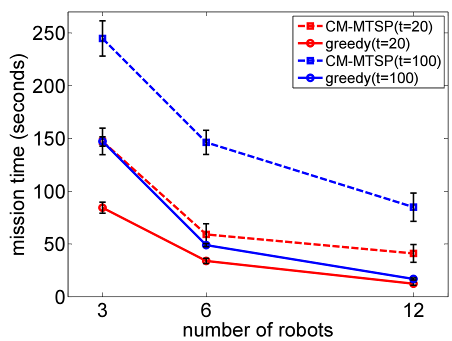
Fig. 5: Distribution of targets in the case of 3 and 6 robots.



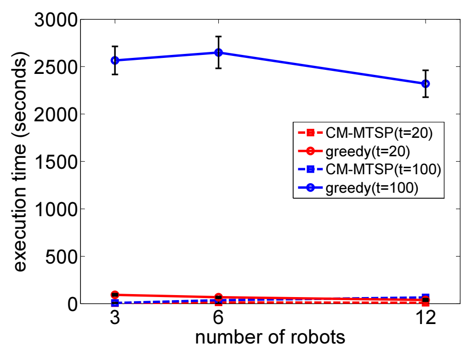
(a) Total Traveled Distance



(b) Maximum Traveled Distance



(c) mission time



(d) execution time

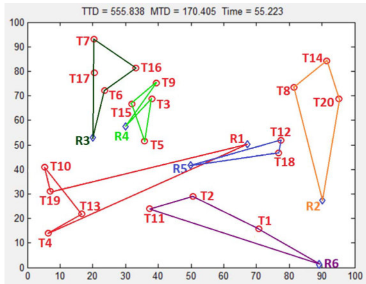
Fig. 6: Comparison results of the CM-MTSP with a greedy algorithm.

t_i to the robot with the least cost. This process is repeated until all targets are assigned to robots.

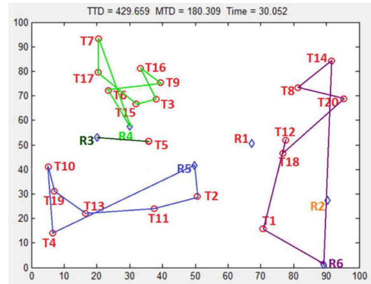
We considered different scenarios, where the number of targets varied in [20, 100], and the number of robots varied in [3, 6, 12]. We performed 10 runs for each scenario to ensure 95% confidence interval. Figure 6 shows the comparative results.

In Figure 6d, we observe that the CM-MTSP significantly reduces the execution time as compared to the greedy algorithm. For example, in the case of 12 robots and 100 targets, the reduction exceeds 95%. This is due to the fact that we used a clustering technique to group targets. So, instead of assigning m targets, we only search to assign $n << m$ clusters. This will significantly reduce the execution time for large instances, which demonstrates that CM-MTSP scale much better than traditional non-clustered market-based approaches.

In addition, we observe that the MTD of the CM-MTSP algorithm was decreased in comparison with the greedy algorithm (Figure 6b). For example, in the case of 12 robots and 100 targets, the MTD is reduced by around 30%. This means that the number of targets assigned to each robot is not the same for both the algorithms. For the greedy algorithm, not all robots are assigned to targets and so, in this case the MTD will increase. Figure 7 shows an illustrative example, between the greedy solution and the CM-MTSP algorithm using a scenario with 6 robots and 20 targets. The increase of the TTD of the CM-MTSP in comparison with the greedy algorithm as shown in Figure 6a is attributed to the fact that the



(a) CM-MTSP algorithm



(b) Greedy algorithm

Fig. 7: Simulation example of the CM-MTSP and the greedy algorithm

targets in the same cluster can be far from each other or even when the number of robots is high with respect to the number of targets. For example, it is clear from figure 6a that the gap between the CM-MTSP algorithm and the greedy algorithm, in the scenario with 6 robots and 20 targets, is reduced in comparison with the scenario with 12 robots and 20 targets.

The result of mission time (Figure 6c) the greedy algorithm conforms the results obtained for the MTD. In the case where a small number of robots is assigned to targets, the TTD increases and the mission time decreases in contrast to the CM-MTSP algorithm.

VI. Conclusion

In this paper, we considered the task assignment problem in multi-robot systems. We addressed the multiple depot MTSP where a set of robots is charged to monitor a specified area by visiting a set of target locations. Our objective is to simultaneously optimize several performance criteria. Also, we aim to uniformly distribute targets between robots such that the number of allocated targets for each robot is equal or close. Toward these objectives, we presented a solution based on the use of a clustering method with an auction process. We compared our algorithm against a single objective and a greedy algorithms. We concluded that the our approach provides a good trade-off between the objectives, as compared to a single objective algorithm with an improvement of the mission time and reduces the execution time as compared to the greedy algorithm.

Acknowledgment

This work is support by the Research and Translation Center (RTC) at Prince Sultan University via Grant Number GP-CCIS-2013-11-10. This work is also supported by a King AbdulAziz City for Science and Technology (KACST) under grant number SP35-157.

References

[1] J. Kubalík, J. Kléma, and M. Kulich, "Application of soft computing techniques to rescue operation planning," in *Artificial Intelligence and Soft Computing-ICAISC 2004*. Springer, 2004, pp. 897–902.

[2] J. Faigl, O. Simonin, and F. Charpillet, "Comparison of task-allocation algorithms in frontier-based multi-robot exploration," in *Multi-Agent Systems*. Springer, 2014, pp. 101–110.

[3] I. Maza, A. Viguria, and A. Ollero, "Networked aerial-ground robot system with distributed task allocation for disaster management," in *Proceedings of the IEEE International Workshop on Safety, Security and Rescue Robotics*, 2006.

[4] I. Maza, F. Caballero, J. Capitán, J. R. Martínez-de Dios, and A. Ollero, "Experimental results in multi-uav coordination for disaster management and civil security applications," *Journal of intelligent & robotic systems*, vol. 61, no. 1-4, pp. 563–585, 2011.

[5] V. A. Shim, K. Tan, and C. Cheong, "A hybrid estimation of distribution algorithm with decomposition for solving the multiobjective multiple traveling salesman problem," *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, vol. 42, no. 5, pp. 682–691, 2012.

[6] L. Ke, Q. Zhang, and R. Battiti, "Moea/d-aco: a multiobjective evolutionary algorithm using decomposition and antcolony." *IEEE T. Cybernetics*, vol. 43, no. 6, pp. 1845–1859, 2013.

[7] O. Cheikhrouhou, A. Koubâa, and H. Bennaceur, "Move and improve: A distributed multi-robot coordination approach for multiple depots multiple travelling salesmen problem," in *Autonomous Robot Systems and Competitions (ICARSC), 2014 IEEE International Conference on*. IEEE, 2014, pp. 28–35.

[8] E. Kivelevitch, K. Cohen, and M. Kumar, "A market-based solution to the multiple traveling salesmen problem," *Journal of Intelligent & Robotic Systems*, vol. 72, no. 1, pp. 21–40, 2013.

[9] K. Miettinen, *Nonlinear Multiobjective Optimization*. Springer US, 1999.

[10] T. Bektas, "The multiple traveling salesman problem: an overview of formulations and solution procedures," *Omega*, vol. 34, no. 3, pp. 209–219, 2006.

[11] M. Elango, S. Nachiappan, and M. K. Tiwari, "Balancing task allocation in multi-robot systems using k-means clustering and auction based mechanisms," *Expert Systems with Applications*, vol. 38, no. 6, pp. 6486–6491, 2011.

[12] "TSPLIB95," www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95/.

[13] B. G. J. Heap and M. Pagnucco, "Repeated sequential auctions with dynamic task clusters." in *AAAI*, 2012.

[14] Z. S. Chan, L. Collins, and N. Kasabov, "An efficient greedy k-means algorithm for global gene trajectory clustering," *Expert Systems with Applications*, vol. 30, no. 1, pp. 137–141, 2006.

[15] K. Helsgaun, "Lkh," http://www.akira.ruc.dk/~keld/research/LKH/, 2012.

[16] K. Alexis, G. Darivianakis, M. Burri, and R. Siegwart, "Aerial robotic contact-based inspection: planning and control," *Autonomous Robots*, pp. 1–25, 2015.

[17] W. Yong, "Hybrid max–min ant system with four vertices and three lines inequality for traveling salesman problem," *Soft Computing*, vol. 19, no. 3, pp. 585–596, 2015.