University of Groningen

Task assignment for robots with limited communication

Bai, Xiaoshan; Yan, Weisheng; Cao, Ming; Huang, Jie

Published in:
Proceedings of the 36th Chinese Control Conference, CCC 2017

DOI:
[10.23919/ChiCC.2017.8028451](https://doi.org/10.23919/ChiCC.2017.8028451)

IMPORTANT NOTE: You are advised to consult the publisher's version (publisher's PDF) if you wish to cite from it. Please check the document version below.

Document Version
Publisher's PDF, also known as Version of record

Publication date:
2017

[Link to publication in University of Groningen/UMCG research database](#)

Citation for published version (APA):

Bai, X., Yan, W., Cao, M., & Huang, J. (2017). Task assignment for robots with limited communication. In T. Liu, & Q. Zhao (Eds.), *Proceedings of the 36th Chinese Control Conference, CCC 2017* (pp. 6934-6939). [8028451] (Chinese Control Conference, CCC). IEEE Computer Society.
<https://doi.org/10.23919/ChiCC.2017.8028451>

Copyright

Other than for strictly personal use, it is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license (like Creative Commons).

The publication may also be distributed here under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license. More information can be found on the University of Groningen website: <https://www.rug.nl/library/open-access/self-archiving-pure/taverne-amendment>.

Take-down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Downloaded from the University of Groningen/UMCG research database (Pure): <http://www.rug.nl/research/portal>. For technical reasons the number of authors shown on this cover page is limited to 10 maximum.

Task assignment for robots with limited communication

Xiaoshan Bai^{1,2}, Weisheng Yan¹, Ming Cao², Jie Huang²

1. School of Marine Science and Technology, Northwestern Polytechnical University, Xi'an 710072, PR China

2. Faculty of Science and Engineering, University of Groningen, Groningen 9747 AG, The Netherlands

E-mail: xiaoshan.bai@rug.nl

Abstract: This paper investigates the task assignment problem for multiple dispersed robots constrained by limited communication range. The robots are initially randomly distributed and need to visit several target locations while trying to minimize the total travel distance. A centralized rendezvous-based algorithm is proposed, under which all the robots first move towards a rendezvous position until communication paths are established between every pair of robots either directly or through intermediate peers, and then one robot is chosen as the leader to make a centralized task assignment for the other robots. Furthermore, we propose a decentralized algorithm based on a single-traveling-salesman tour, which does not require all the robots to be connected through communication. We investigate the variation of the quality of the assignment solutions as the communication range grows. The proposed algorithms are compared with a centralized algorithm with shared global information and a decentralized greedy algorithm respectively. Monte Carlo simulation results show the satisfying performance of the proposed algorithms.

Key Words: Multi-robot task assignment; Traveling salesman problem; Limited communication range

1 Introduction

The last decade has witnessed growing demands in applying robotic networks to complete various tasks, such as terrain mapping, environmental monitoring, and disaster rescue [1]. The task assignment problem is on how to assign a number of robots to efficiently perform a set of tasks, which is generally managed by either centralized or decentralized algorithms [2]. Many centralized algorithms have been proposed to solve the task assignment problem. Considering unmanned aerial vehicles' turning radius constraint, genetic algorithms were integrated with a Dubins car model in [3] for multi-vehicle target assignment. In [4], clustering-based algorithms were proposed for the task assignment of multiple vehicles. Centralized algorithms can obtain optimal or near-optimal solutions for the task assignment problem; however, they require global information [5]. Consequently, centralized algorithms cannot solve task assignment problems in which robots only have local, or possibly outdated, information due to the robots' limited communication capabilities.

On the other hand, decentralized algorithms enable each robot to plan its own route based on available local information [6]. For example, a consensus routine relying on local communication was designed for task allocation in [5]. Regarding aerial vehicles with limited communication range, distributed approaches were employed for surveillance mission assignment [7]. In [8], monotonic task assignment algorithms were proposed to minimize the time until every target is occupied by one robot with limited communication range. Several decentralized algorithms were proposed in [9] to minimize the robots' total travel distance until each target position is occupied by one robot constrained by limited communication and sensing ranges. However, the numbers of robots and targets in [8, 9] are equal, such that a robot stops moving as soon as it reaches its target.

Motivated by the existing literature just mentioned, our research focuses on the more realistic situation, in which the number of targets is greater than the number of robots and

the robots are constrained by limited communication range. To be more precise, a fleet of initially randomly dispersed robots with limited communication range need to visit several target locations while trying to minimize the total travel distance. Each robot is assumed to have knowledge of all the target positions as [8, 9], and has the positions of its communication-connected (CC) robots. The main contribution of this paper is exploration of the fluctuation of the assignment quality with an increase in the robots' communication range. Firstly, we propose a centralized rendezvous-based algorithm (RBA), and a decentralized algorithm which does not require all the robots to be CC. Based on the robots' local information, the two algorithms enable all the target points to be visited in finite travel distance irrespective of the robots' communication range. Secondly, we illustrate that the quality of the solution resulting from the decentralized algorithm does not monotonically increase as the robots' communication range grows, which holds for some other decentralized task assignment algorithms.

The rest of this paper is organized as follows. Some preliminaries are given in Section 2. In Section 3, the formulation of the task assignment problem is presented. Section 4 studies the centralized task assignment algorithm RBA, while in Section 5 the decentralized algorithm is introduced. Monte Carlo simulations are shown in Section 6. Finally, the conclusions of this work are presented.

2 Preliminaries

During the robots' movement, the neighborhood of one robot dynamically changes due to its limited communication range. We use the variable $c_{jk}(t)$ to denote whether robot j can directly communicate with robot k at time t , namely

$$c_{jk}(t) = \begin{cases} 1, & \text{if } |p_j(t) - p_k(t)| \leq r, \\ 0, & \text{if } |p_j(t) - p_k(t)| > r, \end{cases} \quad (1)$$

where $p_j(t)$ and $p_k(t)$ in $\mathbb{R}^2$ are the positions of the robots j and k at time t respectively, and $r > 0$ is the robots' communication range. The following theorem shows the CC probability of a randomly distributed robotic network with respect to the limited communication range of the robots.

This work is supported by National Natural Science Foundation of China (grant No. 61633002 and 61603094), and the China Scholarship Council (grant No. 201506290081).

Theorem 1. (Connectivity of Random Geometric Graphs, [10]) Consider the random geometric graph $\mathcal{G}$ obtained by uniformly randomly distributing m points with communication range r over the unit square. Then as $m \rightarrow \infty$, for any $\varepsilon > 0$

$$P(\mathcal{G} \text{ is connect} | m\pi r^2 - \log m \leq \varepsilon) = e^{-e^{-\varepsilon}}. \quad (2)$$

Now we are ready to formulate the problem to be studied.

3 Problem Statement

Consider in a square area with edge length E_l , a set of initially randomly distributed robots, $\mathcal{R} = \{1, \dots, m\}$, is employed to visit a set of dispersed target points in $\mathcal{T} = \{1, \dots, n\}$. Each robot initially has the position information of the targets through a digital map of the environment and that of the robots within its limited communication range r .

The task assignment problem is to minimize the total travel distance for the robots to visit all the target points. The binary variable y_{ij} is applied to represent whether target i is visited by robot j . Each robot stops moving when knowing all of its assigned targets have been visited, and starts to move again once new assignment arrives through communicating with other moving robots. We assume that the robots move with unit speed. Then, the objective is to minimize

$$f = \sum_{j \in \mathcal{R}} d_j, \quad (3)$$

subject to

$$\sum_{j \in \mathcal{R}} y_{ij} \geq 1, \quad \forall i \in \mathcal{T}, \quad (4)$$

where d_j is the total travel distance of robot j .

4 Centralized task assignment algorithm

If the multi-robot system is initially CC, the task assignment problem is in fact a variant of the NP-hard multi-depot vehicle routing problem (MVRP) [11] where a fleet of vehicles need to deliver products from several depots to a set of scattered customers. In this case, centralized algorithms are usually adopted by choosing one leader robot to make decisions for the other robots based on the global information.

Under a certain communication range, a large number of robots can make the randomly distributed multi-robot system initially CC, for which Theorem 1 can be applied to estimate the number of needed robots. The resulting number of robots makes the robots CC with the same probability during the whole operation time. However, in the task assignment problem, the robotic network is not necessarily always connected, since being fully CC for one time is sufficient to undergo a centralized task assignment in a static environment.

One alternative method is to let the robots initially intentionally move towards a rendezvous position until being CC. Based on this idea, we design the assignment algorithm RBA. Inspired by the location problem [12], the center-of-gravity of the target points is chosen as the rendezvous position. Guided by RBA, if one robot reaches the rendezvous position first, it stops moving and waits for the other robots

Fig. 1: Chromosome structure with 12 target points being assigned to 3 robots, where genes 13 and 14 are marker genes used to separate the target points into three groups.

until all the robots are CC. Then, the robot having the most 1-hop neighbours is chosen to be the leader to make a centralized assignment for all the robots. If there are several robots with the same number of the largest number of 1-hop neighbours, the leader is randomly chosen from these robots.

The co-evolutionary multi-population genetic algorithm (CMGA) of [13] is employed for the leader robot to make the centralized assignment. The CMGA encodes each target as a gene and inserts $m - 1$ marker genes into the target genes. Then, each chromosome represents a candidate solution to the task assignment problem. An example of the chromosome structure is presented in Fig. 1, which contains 12 target points and 2 marker genes. The routes of the 3 robots shown in Fig. 1 are: $p_1(0) \rightarrow 12 \rightarrow 7 \rightarrow 9 \rightarrow 6$, $p_2(0) \rightarrow 8 \rightarrow 5 \rightarrow 4 \rightarrow 11 \rightarrow 2$, $p_3(0) \rightarrow 3 \rightarrow 1 \rightarrow 10$, where $p_i(0)$ is the initial position of robot i .

Let f_{MVA} be the total travel distance of the assignment solution resulting from the multi-vehicle algorithm (MVA) [14], and let f_o be the optimal value for (3).

Lemma 1. [14] Assume that each robot initially has the position information of all the other robots, then $f_{MVA} \leq 2f_o$.

We use f_{RBA} to represent (3) of the assignment solution resulting from RBA, and f_{MST} to be the sum of all the edge weights of a minimum spanning tree (MST) of the graph $\mathcal{G}$ whose vertices contain all the robots and the targets. For each pair of nodes of $\mathcal{G}$, its edge weight is zero if the two nodes represent robots, and is otherwise the Euclidean distance between them. We are now able to present a lower bound of (3) on the solution resulting from RBA.

Theorem 2. For the task assignment problem, the total travel distance of the assignment solution resulting from RBA is bounded by $f_{MST} \leq f_{RBA}$.

Proof. The distance matrix that contains the edge weight between each target point and each robot in $\mathcal{G}$ satisfies the triangle inequality. Inspired by the construction of the MVA in [14], we get that Lemma 1 holds as $f_{MST} \leq f_o$ and $f_{MVA} \leq 2f_{MST}$. As $f_o \leq f_{RBA}$, the proof is complete. $\square$

5 Decentralized Algorithm

If the robotic system is not initially CC, the task assignment problem can be solved in a decentralized manner where two important issues need to be considered. They are how to plan routes for the robots which are not CC with any other robot, and how to make CC robots cooperative based on their carried partial, or even outdated, information on which targets have already been visited.

As the robots have only limited communication range, each robot j carries an information tuple $\mathcal{I}_j(t) =$

$\{j, p_j(t), o_j(t), d_j, u_j(t), s_j(t)\}$, where j is the unique identifier for robot j , $p_j(t)$ is j 's current position, $o_j(t)$ initially contains all the targets in $\mathcal{T}$ and keeps the ordered targets on the route of robot j , $u_j(t)$ is an m -tuple where $u_j^k(t) = 1$ means j has ever been CC with robot k , otherwise $u_j^k(t) = 0$, and $s_j(t)$ is an n -array to record the target status, namely

$$s_j^p(t) = \begin{cases} 1, & \text{if robot } j \text{ knows target } p \text{ is visited at } t, \\ 0, & \text{otherwise.} \end{cases} \quad (5)$$

For the task assignment problem, the worst performance of one robot happens when the robot visits all the targets without being able to communicate with any other robot. Thus, we design an algorithm based on a single-traveling-salesman tour (STST), where all the target locations are connected into a traveling salesman problem (TSP) tour TSP_0 . The TSP_0 can be obtained by the Christofides algorithm [15], whose length is at most $3/2$ times of the optimal one. Then, the robots that have never communicated with any other robot travel along the TSP_0 , which guarantees the worst performance of the robots. Thus, the first issue is solved.

The initial route $o_j(0)$ for each robot j is determined by minimizing the projected d_j to travel along the TSP_0 , which will be shown in Theorem 4. When several robots are CC, each robot j within the CC robots first updates $s_j(t)$ as

$$s_j^p(t) = \bigcup_{k \in \mathcal{R}_i(t)} s_k^p(t-1), \quad \forall j \in \mathcal{R}_i(t), \forall p \in \mathcal{T}, \quad (6)$$

where $\mathcal{R}_i(t)$ is the robot subgroup i in which robots are CC at time t .

Then, the robots in each connected robot set $\mathcal{R}_i(t)$ update their remaining information tuples as follows. Based on the $s_j(t)$, $o_j(t)$ is updated by deleting the visited targets that are known by robot j , namely

$$o_j(t) = o_j(t-1) \setminus s_j(t), \quad \forall j \in \mathcal{R}_i(t), \quad (7)$$

and then the planned total travel distance d_j of robot j is updated to the one considering all the targets on $o_j(t)$ to be visited by the robot. Afterwards, $u_j(t)$ is updated as

$$u_j^k(t) = \begin{cases} 1, & \text{if } u_j^k(t-1) = 1 \text{ or } k \in \mathcal{R}_i(t), \\ 0, & \text{otherwise.} \end{cases} \quad (8)$$

If $\mathcal{R}_i(t)$ has no new members compared with $\mathcal{R}_i(t-1)$, the robots within the group do not undergo task reassignment. To make CC robots in $\mathcal{R}_i(t)$ coordinate based on their carried information, a cooperative strategy is designed. We use robotic set $\mathcal{R}_i^1(t) \subseteq \mathcal{R}_i(t)$ to contain the robots in $\mathcal{R}_i(t)$ that have never been CC with any other robot before time t , and $\mathcal{R}_i^2(t)$ to be $\mathcal{R}_i(t) \setminus \mathcal{R}_i^1(t)$. Then the targets to be divided are those in the set

$$\mathcal{T}_i(t) = \begin{cases} o_p(t), & \text{if } \sum_{k \in \mathcal{R}_i^2(t)} d_k \geq d_p \text{ or } \mathcal{R}_i^2(t) = \emptyset, \\ \bigcup_{k \in \mathcal{R}_i^2(t)} o_k(t), & \text{otherwise,} \end{cases} \quad (9)$$

where $p = \arg\min_{q \in \mathcal{R}_i^1(t)} d_q$ if $\mathcal{R}_i^1(t) \neq \emptyset$, otherwise $d_p = +\infty$. $\sum_{k \in \mathcal{R}_i^2(t)} d_k \geq d_p$ means that the total travel distance

Algorithm 1 The decentralized STSTC for each robot $j, \forall j \in \mathcal{R}(t)$.

Require: Positions of the n targets, target information tuple $\mathcal{I}_j(t)$, communication range r , the tour TSP_0 .

Ensure: Every target on $o_j(t)$ is visited.

```

1: while  $o_j(t) \neq \emptyset$  do
2:   if Robot  $j$  is not CC with any other robot then
3:     Robot  $j$  moves along its  $o_j(t)$ .
4:   else
5:     if Robot  $j$  is in  $\mathcal{R}_i(t)$  then
6:       if  $\mathcal{R}_i(t) = \mathcal{R}_i(t-1)$  then
7:         Robot  $j$  exchanges information with the robots
           in  $\mathcal{R}_i(t)$  and continues moving without task reassignment;
8:       else
9:         Go to step 12.
10:      end if
11:    end if
12:    One leader robot in  $\mathcal{R}_i(t)$  is chosen to calculate and
       divide the targets in  $\mathcal{T}_i(t)$  of (9) to the robots in  $\mathcal{R}_i(t)$ .
13:    if The assignments decrease the total travel distance of
       the robots in  $\mathcal{R}_i(t)$  then
14:      Every robot  $j$  in  $\mathcal{R}_i(t)$  updates its route with  $o_j(t)$ .
15:    else
16:      Every robot  $j$  in  $\mathcal{R}_i(t)$  keeps its previous assign-
       ment.
17:    end if
18:  end if
19: end while

```

incurred by visiting all the targets in $\bigcup_{k \in \mathcal{R}_i^2(t)} o_k(t)$ is larger than that incurred by visiting all the targets in $o_p(t)$.

To make the CC robots cooperative, the objective for the leader robot of each $\mathcal{R}_i(t)$ at time t is to minimize

$$f_i(t) = \sum_{j \in \mathcal{R}_i(t)} d_j. \quad (10)$$

The algorithm integrating STST with the cooperative strategy, namely (STSTC) is shown in Algorithm 1.

For each CC robotic network $\mathcal{R}_i(t)$, its leader robot employs the CMGA to make a centralized task assignment to the robots in $\mathcal{R}_i(t)$. If the resulting assignments do not decrease the total travel distance of the robots in $\mathcal{R}_i(t)$, each robot in $\mathcal{R}_i(t)$ keeps its previous target assignment. Otherwise, once a locally centralized task assignment has been completed, each robot j in $\mathcal{R}_i(t)$ updates its route $o_j(t)$.

5.1 Correctness of STSTC

To prove the correctness of STSTC, we first present its properties.

Lemma 2. During the operation of STSTC, the following statements hold.

(i) Each target $w \in \mathcal{T}$ is assigned to at least one robot, the assignment may change, but target w remains assigned to at least one robot until being visited.

(ii) For robot j and target w , if $s_j^w(t_0) = 1$ at some time t_0 , then $s_j^w(t) = 1$ for all $t \geq t_0$.

Proof. Based on the initialization of the target set $o_j(t)$ for each robot j , the target set initially assigned to each robot is the whole target set $\mathcal{T}$. Thus, if one robot has never been

Fig. 2: The routes based on STSTC for 4 robots to visit 5 targets, where $r = 25\text{m}$. The total travel distance is 174m.

Fig. 3: The routes based on STSTC for 4 robots to visit 5 targets, where $r = 30\text{m}$. The total travel distance is 180m.

CC with any other robot, an arbitrary target is on the robot's route unless being visited by the robot.

When several robots are CC, statement (i) is concluded based on the cooperative strategy shown in (9). We first consider the case when $\mathcal{T}_i(t) = o_p(t)$, where $p = \operatorname{argmin}_{q \in \mathcal{R}_i^1(t)} d_q$. As robot p has never been CC with any other robot before time t , an arbitrary target w satisfies $w \in o_p(t)$ if w has not been visited by robot p . If $\mathcal{T}_i(t) = \bigcup_{k \in \mathcal{R}_i^2(t)} o_k(t)$, target w is among the cooperative

targets if $w \in o_k(t)$ for at least one robot $k \in \mathcal{R}_i^2(t)$. Otherwise, w must be on the route of at least one other robot, assumed to be robot s , who has already communicated with at least one of the robots in $\mathcal{R}_i^2(t)$. As $w \in o_s(t)$, w will be among the cooperative targets of a CC network if robot s is CC with other robots based on the analysis when $w \in o_k(t)$ and $\mathcal{T}_i(t) = \bigcup_{k \in \mathcal{R}_i^2(t)} o_k(t)$. If robot s has not been CC with

other robot after visiting target w , w will be on the route of robot s until being assigned to other robot or being visited.

Based on the above analysis, an arbitrary unvisited target w is either among the cooperative targets of a CC robotic group $\mathcal{R}_i(t)$ or on the route of at least one robot. Once $w \in \mathcal{T}_i(t)$, it will be assigned to one robot until being visited. Otherwise, it will be assigned to at least one of the robots whose route contains the target. Thus, statement (i) is proved.

Statement (ii) follows directly from the union operation of the $s_j(t)$ in (6). It can also be explained by the fact that once a target is visited and its status is known by one robot, the robot will keep this information. $\square$

We further investigate the effect of the length the robots' communication range on the performance of STSTC.

Theorem 3. For the investigated task assignment problem, a longer communication range does not necessarily lead to a better performance for STSTC.

When several robots are CC, they communicate to update their information tuples. As some targets can have already been visited by robots that are not CC, the information shared by the CC robots might not truly reflect the targets' current situation. The incomplete information shared by the

robots can lead to inefficient task assignments; for example, one already visited target is reassigned to another robot.

One illustrative case of the task assignment based on STSTC is shown in Fig. 2 and Fig. 3 where 4 robots need to visit 5 targets $\mathcal{T} = \{1, \dots, 5\}$ in a $100 \times 100 \text{ m}^2$ area. In Fig. 2, the robots' communication range is 25m which makes robots 1 and 2 initially CC. The routes of the robots are $p_1(0) \rightarrow 1 \rightarrow 5 \rightarrow 4$, $p_2(0) \rightarrow 2 \rightarrow 3$, $p_3(0) \rightarrow 3 \rightarrow 2 \rightarrow 1 \rightarrow 5 \rightarrow 4$, $p_4(0) \rightarrow 4 \rightarrow 5 \rightarrow 1 \rightarrow 2 \rightarrow 3$, where $p_i(0), i \in \{1, \dots, 4\}$, is the initial position of robot i . With the movement of the robots, target 3 is first visited by robot 3, and then robot 2 stops moving once it can communicate with robot 3 since robot 3 is nearer to their cooperative target 2. Reaching target 2, robot 3 stops moving as $o_3(t)$ is empty. When target 4 is visited by robot 4, robot 1 should not move towards the target. However, robot 1 continues moving as it does not have the latest target status of target 4. When robot 1 can communicate with robot 4, they stop moving as their cooperative target set is empty based on (9). The total travel distance of the robots is 174m, where the travel distances of the four robots are 57m, 24m, 36m and 57m respectively.

In Fig. 3, the robots' communication range is increased to 30m which makes robots 1, 2 and 3 initially CC. After dividing the cooperative targets based on STSTC, the routes of the CC robots are $p_1(0) \rightarrow 1 \rightarrow 5$, $p_2(0) \rightarrow 2 \rightarrow 3$, $p_3(0) \rightarrow 4$, while $p_4(0) \rightarrow 4 \rightarrow 5 \rightarrow 1 \rightarrow 2 \rightarrow 3$. With the movement of the robots, robot 3 stops moving when it can communicate with robot 4 as robot 4 is nearer to their cooperative target 4. Robots 1 and 2 stop moving when their assigned targets are visited. The total distance to visit all the targets is 180m, where the travel distances of the four robots are 50m, 56m, 34m and 40m respectively. The performance of STSTC for robots with a longer communication range in Fig. 3 is worse than that in Fig. 2, thus illustrating the statement. Theorem 3 holds for some other decentralized task assignment algorithms.

5.2 Time complexity for STSTC

For the robots in each $\mathcal{R}_i(t)$, task reassignment is centrally made by the leader robot in the CC robotic group to minimize the total travel distance of the connected robots.

Lemma 3. Each information updating for the CC robots in

$\mathcal{R}_i(t)$ makes the total travel distance f in (3) non-increasing.

Proof. Guided by STSTC, the CC robots in each $\mathcal{R}_i(t)$ will divide the cooperative targets to the robots if the reassignment reduces their total travel distance (10). Otherwise, each robot j among the CC robots in each $\mathcal{R}_i(t)$ only updates its $s_j(t)$, $o_j(t)$, $u_j(t)$, and the corresponding projected d_j based on their shared information, which does not make the total travel distance $f_i(t)$ in (10) worse. As $f_i(t)$ is one component of f in (3), f is nonincreasing for each information updating. $\square$

For the robot that is not initially CC with any other robot, it will visit all the targets through circling the TSP_0 if it cannot communicate with any other robot during its movement. The following lemma gives an upper bound on the minimal travel distance of the robot.

Lemma 4. [16] Let L be the shortest length of the TSP tour which connects the n target points and one robot in a square area with edge length E_l , then there exist $c_1 \in \mathbb{N}$ and $c_2 \in \mathbb{R}^+$ such that $L \leq c_2 \sqrt{(n+1) * E_l}$ for all $n+1 \geq c_1$.

As the robots move with unit speed, the upper bound on the minimal travel distance for one robot to visit the n targets is not larger than $c_2 \sqrt{(n+1) * E_l}$. With this lemma, we are able to give an upper bound on the total travel distance of the robots guided by STSTC.

Theorem 4. STSTC guarantees the n targets to be visited by m robots with the total travel distance at most $m(\sqrt{2E_l} + (3/2)c_2\sqrt{nE_l})$ where c_2 is obtained in Lemma 4.

Proof. We use matrix D as the distance matrix where $D(i, j)$ contains the distance between vertex i and j . $L(\{p_i(0), T_i^1, T_i^2, \dots, T_i^n\})$ is employed to represent the length of the initial route $o_i(0)$ for robot i , where $o_i(0)$ is $p_i(0) \rightarrow T_i^1 \rightarrow T_i^2 \rightarrow \dots \rightarrow T_i^n$ based on the TSP_0 . Then

$$\begin{aligned} L(o_i(0)) &= D(p_i(0), T_i^1) + D(T_i^1, T_i^2) + \dots \\ &\quad + D(T_i^{n-1}, T_i^n) \\ &= D(p_i(0), T_i^1) + D(T_i^1, T_i^2) + \dots \\ &\quad + D(T_i^{n-1}, T_i^n) + D(T_i^n, T_i^1) - D(T_i^n, T_i^1) \\ &= D(p_i(0), T_i^1) + L(\{T_i^1, \dots, T_i^n, T_i^1\}) - \\ &\quad D(T_i^n, T_i^1) \\ &= D(p_i(0), T_i^1) + L(TSP_0) - D(T_i^n, T_i^1), \end{aligned} \quad (11)$$

where $L(TSP_0)$ is the length of the tour TSP_0 . As $D(p_i(0), T_i^1) \leq \sqrt{2E_l}$ and $L(TSP_0) \leq (3/2)c_2\sqrt{nE_l}$ where TSP_0 is calculated by the Christofides algorithm [15], we get $L(o_i(0)) \leq \sqrt{2E_l} + (3/2)c_2\sqrt{nE_l}$. If robot i cannot communicate with any other robot during its movement, it will travel along the $o_i(0)$ until all the targets being visited by itself. Thus, the longest travel distance of one robot guided by STSTC is upper bounded by $\sqrt{2E_l} + (3/2)c_2\sqrt{nE_l}$. Moreover, the first target to be visited, T_i^1 , and the travel direction of the robot i when traveling along the TSP_0 are chosen based on the minimization of $L(o_i(0))$. In other words, $o_i(0)$ is determined by minimizing $L(o_i(0))$.

Based on Lemma 3, the worst performance of STSTC occurs when every robot circles around the tour TSP_0 without

Fig. 4: The mean solution quality for 4 robots with different communication ranges to visit 15 target locations, where $P(\mathcal{G} \text{ is connect} | r_c = 712.8m) = 0.99$.

communicating with any other robot. In this case, each robot stops moving after visiting all the targets on TSP_0 by itself. Thus, the proof is complete. $\square$

6 Monte Carlo study

We implemented two sets of simulations in a 1000×1000 m² area where the numbers of target points and robots are 15 and 4, 30 and 6 respectively. For each set of simulations, Monte Carlo simulations are carried out on 500 scenarios where the positions of the target points and the robots are randomly generated. The proposed algorithms are compared with a greedy algorithm where robots always move towards the nearest target and the CC robots communicate only when two or more robots are moving towards the same target. All the experiments are performed on an Intel Core (TM) i5-4590 CPU 3.30 GHz with 8 GB RAM, with algorithms compiled by Matlab under Windows 7.

The solution quality of each algorithm is defined by

$$q = \frac{f}{f_{MST}}, \quad (12)$$

where f is the value in (3); f_{MST} is a lower bound of the optimal total travel distance f_o based on Theorem 2. Thus, a smaller q of one algorithm means a better performance of the algorithm.

Testing the algorithms on robots constrained by communication range varying from 1m to the one that makes the robotic system CC with probability 0.99, we show the assignment results of the two sets of simulations in Fig. 4 and Fig. 5. The two figures first show that the quality of the assignment solutions resulting from the decentralized STSTC and the greedy algorithm varies greatly as r grows, where a longer r generally leads to a better performance for STSTC but not for the greedy algorithm. The reason is that more environmental information is shared among the CC robots guided by STSTC where cooperative targets are properly divided, while the CC robots guided by the greedy algorithm communicate only when task assignment conflicts occur. However, the greedy algorithm outperforms STSTC when r is approximately smaller than $r_c/6$ where $P(\mathcal{G} \text{ is connect} | r_c) = 0.99$. That is because the robots

Fig. 5: The mean solution quality for 6 robots with different communication ranges to visit 30 target locations, where $P(\mathcal{G} \text{ is connect} | r_c = 600m) = 0.99$.

guided by STSTC cannot cooperate with other robots frequently when r is short, while the robots using the greedy algorithm do not rely on the communication so much and have relatively smaller travel cost by always moving towards the nearest target.

Furthermore, Fig. 4 and Fig. 5 show that the RBA has a better performance which does not vary greatly with an increase in r . The reason is that the longer r only makes all the robots guided by RBA CC at an earlier time, and then they cooperatively visit the remaining targets. Though STSTC does not perform well when r is short, it outperforms RBA when $r \geq r_c/2$, as shown in the two figures. As for STSTC, a longer r leads to more information shared by the CC robots, which generally results in better cooperation for the CC robots. However, a worse performance of STSTC occurs in Fig. 5 when r is increased from 400m to 500m, which is generally the case for the decentralized greedy algorithm as shown in the two figures. Thus, Theorem 3 is again verified.

Finally, the mean solution quality q of RBA and STSTC displayed in Fig. 4 is approximate to the optimal value 1 when $r \geq r_c/2$, which shows the good performance of the algorithms. However, in Fig. 5, q is a little bit larger than 1 when $r \leq r_c/2$. There are two reasons: one being that the f_{MST} in (12) is a lower bound of the optimal value, which can lead to a larger q ; the other one being that the calculation of the f_{MST} is based on the global information, while the assignments resulting from RBA and STSTC consider the limited communication range r among the robots.

7 Conclusions

In this paper, we have studied the task assignment problem where several initially randomly dispersed robots constrained by limited communication range are coordinated to visit a set of dispersed target locations. The centralized algorithm RBA and the decentralized STSTC proposed in the paper guarantee all the target locations to be visited in finite travel distance irrespective of the communication range. For the task assignment problem, we have illustrated that a longer communication range does not necessarily lead to a better performance for STSTC, which usually holds for the other decentralized algorithms, while Monte Carlo simulations have shown that longer communication ranges lead to

better performances for RBA. The proposed algorithms will be further tested by considering environmental disturbances, for example, winds and obstacles. We are also planning to test the algorithms using a robot fish testbed.

References

- [1] B. Chen and H. H. Cheng, "A review of the applications of agent technology in traffic and transportation systems," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 11, no. 2, pp. 485–497, 2010.
- [2] G. A. Korsah, A. Stentz, and M. B. Dias, "A comprehensive taxonomy for multi-robot task allocation," *The International Journal of Robotics Research*, vol. 32, no. 12, pp. 1495–1512, 2013.
- [3] E. Edison and T. Shima, "Integrated task assignment and path optimization for cooperating uninhabited aerial vehicles using genetic algorithms," *Computers & Operations Research*, vol. 38, no. 1, pp. 340–356, 2011.
- [4] X. Bai, W. Yan, and M. Cao, "Clustering-based algorithms for multi-vehicle task assignment in a time-invariant drift field," under review, *IEEE Robotics and Automation Letters*, 2017.
- [5] 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.
- [6] N. Michael, M. M. Zavlanos, V. Kumar, and G. J. Pappas, "Distributed multi-robot task assignment and formation control," in *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*. IEEE, 2008, pp. 128–133.
- [7] J. J. Acevedo, B. C. Arrue, J. M. Diaz-Bañez, I. Ventura, I. Maza, and A. Ollero, "One-to-one coordination algorithm for decentralized area partition in surveillance missions with a team of aerial robots," *Journal of Intelligent & Robotic Systems*, vol. 74, no. 1-2, pp. 269–285, 2014.
- [8] S. L. Smith and F. Bullo, "Monotonic target assignment for robotic networks," *Automatic Control, IEEE Transactions on*, vol. 54, no. 9, pp. 2042–2057, 2009.
- [9] J. Yu, S.-J. Chung, and P. G. Voulgaris, "Target assignment in robotic networks: Distance optimality guarantees and hierarchical strategies," *Automatic Control, IEEE Transactions on*, vol. 60, no. 2, pp. 327–341, 2015.
- [10] M. D. Penrose, "The longest edge of the random minimal spanning tree," *The annals of applied probability*, pp. 340–361, 1997.
- [11] M. Mirabi, S. F. Ghomi, and F. Jolai, "Efficient stochastic hybrid heuristics for the multi-depot vehicle routing problem," *Robotics and Computer-Integrated Manufacturing*, vol. 26, no. 6, pp. 564–569, 2010.
- [12] C. S. ReVelle and H. A. Eiselt, "Location analysis: A synthesis and survey," *European Journal of Operational Research*, vol. 165, no. 1, pp. 1–19, 2005.
- [13] X. Bai, W. Yan, S. Sam Ge, and M. Cao, "An integrated multi-population genetic algorithm for multi-vehicle task assignment in a drift field," under review, *Information Sciences*, 2016.
- [14] S. Rathinam, R. Sengupta, and S. Darbha, "A resource allocation algorithm for multivehicle systems with nonholonomic constraints," *IEEE Transactions on Automation Science and Engineering*, vol. 4, no. 1, pp. 98–104, 2007.
- [15] C. H. Papadimitriou and K. Steiglitz, *Combinatorial optimization: algorithms and complexity*. Courier Corporation, 1982.
- [16] K. J. Supowit, E. M. Reingold, and D. A. Plaisted, "The travelling salesman problem and minimum matching in the unit square," *SIAM Journal on Computing*, vol. 12, no. 1, pp. 144–156, 1983.