Event- and time-triggered dynamic task assignments for multiple vehicles

Bai, Xiaoshan; Cao, Ming; Yan, Weisheng

Published in:
Autonomous Robots

DOI:
10.1007/s10514-020-09912-1

Noname manuscript No.
(will be inserted by the editor)

Event- and time-triggered dynamic task assignments for multiple vehicles

Xiaoshan Bai · Ming Cao · Weisheng Yan

Received: 30 April 2018 / Accepted: 24 January 2020

Abstract We study the dynamic task assignment problem in which multiple dispersed vehicles are employed to visit a set of targets. Some targets' locations are initially known and the others are dynamically randomly generated during a finite time horizon. The objective is to visit all the target locations while trying to minimize the vehicles' total travel time. Based on existing algorithms used to deal with static multi-vehicle task assignment, two types of dynamic task assignments, namely event-triggered and time-triggered, are studied to investigate what the appropriate time instants should be to change in real time the assignment of the target locations in response to the newly generated target locations. Furthermore, for both the event- and time-triggered assignments, we propose several algorithms to investigate how to distribute the newly generated target locations to the vehicles. Extensive numerical simulations are carried out which show better performance of the event-triggered task assignment algorithms over the time-triggered algorithms under different arrival rates of the newly generated target locations.

Keywords Dynamic task assignment · Multiple vehicles · Event-triggered algorithms · Time-triggered algorithms

X. Bai
College of Mechatronics and Control Engineering, Shenzhen University, Shenzhen 518060, China; Faculty of Science and Engineering, University of Groningen, Groningen 9747 AG, the Netherlands; and School of Marine Science and Technology, Northwestern Polytechnical University, 127 West Youyi Road, Xi'an, 710072, China.
E-mail: xiaoshan.bai@rug.nl.

M. Cao
Faculty of Science and Engineering, University of Groningen, Groningen 9747 AG, The Netherlands.
Tel.: +31503635017
E-mail: m.cao@rug.nl

W. Yan
School of Marine Science and Technology, Northwestern Polytechnical University, 127 West Youyi Road, Xi'an, 710072, China.
E-mail: wsyan@nwpu.edu.cn

1 Introduction

The multi-vehicle task assignment problem in which a fleet of vehicles are employed to visit a set of target locations has been increasingly exploited due to its wide applications in logistics, terrain mapping, and environmental monitoring (Toth and Vigo, 2002; Gerkey and Matarić, 2004; Dahl et al, 2009; Chen and Cheng, 2010; Moon et al, 2013; Di Paola et al, 2015). A typical scenario of the multi-vehicle task assignment problem is the vehicle routing problem (VRP) where several vehicles are employed to deliver products to a group of dispersed customers (Laporte, 2009). For the VRP, it is NP-hard to optimally minimize the vehicles' total travel distance to serve all the customers as the numbers of customers and vehicles grow (Lenstra and Kan, 1981). As a result, many heuristic algorithms have been designed to sub-optimally solve the VRP (Prins, 2004; Kuo, 2010; Escobar et al, 2014). The multi-vehicle task assignment problem under certain setups has also been shown to be NP-hard (Korsah et al, 2013). When the matrix, specifying the cost for a vehicle to travel between each pair of locations, satisfies the triangular inequality and is symmetric, the task assignment algorithm proposed in Lagoudakis et al (2004) ensures that the total travel cost for a fleet of robots to visit a set of target locations is within twice of the optimal. In Shima et al (2006), a genetic algorithm (GA) was designed for unmanned aerial vehicles to visit every target location in which the priority for visiting each target location and vehicles' loading capacity are considered. Furthermore, several auction-based algorithms proposed in Choi et al (2009) guarantee that the vehicles' total travel cost to visit a set of target locations is within twice of the optimal under the assumption that the utility of a focus target is non-increasing as other targets are added to a vehicle's path before this focus target. A heuristic distributed algorithm was designed in Zhao et al (2016) for search and rescue task assignment of multiple vehicles. However, most of the discussed algorithms have been developed for the static multi-vehicle task assignment problem in which the information on all the target locations to be visited is initially known and no new target locations dynamically appear.

Some algorithms have been designed for the dynamic task assignment for multiple vehicles in which the position information of some targets or vehicles is not initially known to every vehicle (Fua and Ge, 2005; Smith and Bullo, 2009; Zhu et al, 2013; Yu et al, 2015; Chopra and Egerstedt, 2015). In Fua and Ge (2005), a cooperative backoff adaptive scheme was designed for the task allocation of robots under limited communication range and potential malfunctions, where completing one task might require the cooperation of several robots. When two robots are communication-connected, a task reassignment might be triggered to improve the system's performance. A self-organizing map neural network was designed by Zhu et al (2013) to dynamically assign a set of targets to several underwater vehicles where the targets might move with constant known velocity. The task assignment for robots under limited communication range was also investigated in Smith and Bullo (2009) in which monotone algorithms were designed to minimize the time until the last target location was occupied by a robot. Though each robot does not know how many other robots exist in the environment in Smith and Bullo (2009), every target position is assumed to be initially known to all the dispersed robots and the numbers of robots and targets are equal. To solve the task assignment problem, Smith and Bullo (2009) designed an assignment strategy where each robot first precomputes a TSP tour through all the target locations, and then all the robots move along the tour with the same direction looking for the unoccupied target locations. Whenever two robots are communication-connected, they update their carried local information on which target locations have already been occupied and negotiate the deal on the target locations to be occupied. Later on, several decentralized algorithms were proposed to minimize the robots' total travel distance until every target location was occupied by one robot subject to limited sensing and communication ranges (Yu et al, 2015). The numbers of robots and targets are equal in both Smith and Bullo (2009) and Yu et al (2015), and consequently a robot stops moving upon reaching an unoccupied target. Chopra and Egerstedt (2015) investigated the routing of multiple robots to serve spatially distributed requests at specified time instants by formulating the problem as a pure assignment problem. However, the corresponding set of planar positions that require simultaneous service at each time instant is assumed to be initially known by every robot. As discussed above, few of the research work investigate when the appropriate time instants should be to change the assignment of the targets as well as in what manner to reassign the targets in response to newly dynamical generated target locations.

In our previous work Bai et al (2017a), several clustering-based algorithms have been proposed for a fleet of vehicles to efficiently visit a set of target locations in a time-invariant drift field while trying to minimize the vehicles' total travel time. In addition, we have investigated the task assignment for heterogeneous vehicles with precedence constraints (Bai et al, 2019a), and studied the task assignment for multiple heterogeneous vehicles in a time-invariant drift field with obstacles (Bai et al, 2019b). Furthermore, for vehicles operating in a time-varying drift field, a co-evolutionary multi-population GA was designed in Bai et al (2018) for multiple vehicles to deliver products to a set of target locations. In this paper, we investigate the dynamic task assignment for multiple vehicles to visit a set of target locations where some target locations are initially known and the other target locations are dynamically generated during the vehicles' movement. The objective is to visit every target location while minimizing the vehicles' total travel time. Our main contributions are as follows. Firstly, for the specified dynamic multi-vehicle task assignment problem, we have investigated when the appropriate time instants should be to change the assignment of the target locations in response to the newly generated target locations. Both event- and time-triggered task assignments under different time horizons have been investigated. Secondly, we have studied how to dynamically reassign the targets to minimize the vehicles' total travel time under each target reassignment. For both the event- and time-triggered dynamic task assignments, several algorithms are investigated on how to assign the newly generated target locations based on the existing assignment of the vehicles.

The rest of this paper is organized as follows. In Section 2, the formulation of the task assignment problem is given. Section 3 presents dynamic target assignment algorithms. We present the simulation results in Section 4 and conclude the paper in Section 5.

2 Problem Statement

Consider that m dispersed robots are employed to visit a set of target locations where n target locations are initially known while some other new target locations are to be dynamically randomly generated. Each vehicle will be informed about the position of each newly generated target once it appears. Each vehicle is assumed to move continuously with the unit speed until all the target locations assigned to the vehicle have been visited, and start to move again once being reassigned with some new target locations. The task assignment problem is to minimize the vehicles' total travel time to visit all the target locations.

Let $\mathcal{R}$ denote the set of indices of the m vehicles, $\mathcal{R} = \{1, \cdots, m\}$, and $\mathcal{T}_{ini} = \{m+1, \cdots, m+n\}$ be the set of indices of the n target locations initially known. Let $\mathcal{T} = \mathcal{T}_{ini} \cup \mathcal{T}_{new}$ where the set $\mathcal{T}_{new}$ contains the indices of the newly generated target locations whose position information is initially unknown, and $t_i, i \in \mathcal{T}$, be the time instant when target location i is generated. So $t_i = 0$ for all $i \in \mathcal{T}_{ini}$. The binary variable $\sigma_{ij}(t)$ is used to represent the target-assigning mapping that maps the indices $i \in \mathcal{T}$ and $j \in \mathcal{R}$ of the assignment of target location i to vehicle j at time t, which equals one if and only if it is planned that vehicle j is assigned to visit i at time t. The binary variable y_{ij} is used to represent whether target location i is visited by vehicle j, and let $d(j)$ be the total travel time of vehicle j. Then, the problem is to minimize the vehicles' total travel time to visit all the target locations

$$f = \sum_{j \in \mathcal{R}} d(j), \tag{1}$$

subject to

$$\sum_{j \in \mathcal{R}} y_{ij} = 1, \ \forall\, i \in \mathcal{T}; \tag{2}$$

$$t_i \le t, \ \text{if } \sigma_{ij}(t) = 1, \forall\, i \in \mathcal{T}, \forall\, j \in \mathcal{R}; \tag{3}$$

$$\sum_{j \in \mathcal{R}} \sigma_{ij}(t) \le 1, \forall\, i \in \mathcal{T}, \forall\, t; \tag{4}$$

$$y_{ij} \in \{0, 1\}, \ \forall\, i \in \mathcal{T}, \forall\, j \in \mathcal{R}; \tag{5}$$

$$\sigma_{ij}(t) \in \{0, 1\}, \ \forall\, i \in \mathcal{T}, \forall\, j \in \mathcal{R}, \forall\, t. \tag{6}$$

Constraint (2) ensures that each target location is visited once and only once by one vehicle; (3) ensures that the time that a target location is assigned to a vehicle must be larger than the time when the target location is generated; and (4) guarantees that each target location is assigned to at most one vehicle at any time t.

Remark 1 Optimally minimizing (1) is NP-hard even when no new target locations are dynamically generated; in this case it is then a variant of the NP-hard vehicle routing problem (Lenstra and Kan, 1981).

3 Task assignment algorithms

3.1 Algorithms for assigning the initially known target locations

In this section, to assign the target locations that are initially known to the vehicles, we first present two task assignment algorithms: the extended Voronoi clustering strategy integrated with the smallest marginal cost principle (EVM), and the marginal-cost-based clustering strategy (MC) because of their satisfying performance for the static multi-vehicle task assignment (Bai et al, 2017a).

3.1.1 Task assignment algorithm EVM

The EVM first iteratively clusters the target locations initially known to the vehicles, and then puts the target locations assigned to each vehicle into a sequence to minimize the vehicle's travel time (Bai et al, 2017a). For each $j \in \mathcal{R}$, initialize $\mathcal{T}_j$ such that it contains $p_j(0)$, which is the index of the position where vehicle j is located at $t = 0$. Add in $\mathcal{T}_j$ the indices of those target locations that have already been assigned to j. Let $\mathcal{T}_{ini}$ be the initial choice of $\mathcal{T}^u(0)$ that contains the indices of those unclustered targets at $t = 0$. Then, for the EVM, the first target $k^\star$ in $\mathcal{T}^u(0)$ to be clustered and its assigned vehicle $j^\star$ are determined by

$$(j^\star, k^\star) = \underset{i \in \mathcal{T}_j,\ j \in \mathcal{R},\ k \in \mathcal{T}^u(0)}{\operatorname{argmin}} t(i, k), \tag{7}$$

where $t(i, k)$ is the time for a vehicle to travel from i to k. After clustering target $k^\star$, $\mathcal{T}^u(0)$ is updated to

$$\mathcal{T}^u(0) = \mathcal{T}^u(0) \setminus \{k^\star\}, \tag{8}$$

while the targets assigned to vehicle $j^\star$ are updated to

$$\mathcal{T}_{j^\star} = \mathcal{T}_{j^\star} \cup \{k^\star\}. \tag{9}$$

The target clustering procedure continues until $\mathcal{T}^u(0)$ is empty.

After assigning the target locations to the vehicles, the EVM iteratively determines the sequence for each employed vehicle to visit its assigned target locations. Let $o_j(0)$, initialized as $p_j(0)$, store the indices of the ordered target location for vehicle j for each $j \in \mathcal{R}$ at $t = 0$, and let $\mathcal{T}_j^u$, initialized as $\mathcal{T}_j$, contain the targets in $\mathcal{T}_j$ that have not been inserted into $o_j(0)$. Then, the

EVM determines the first target $k^\star$ in $\mathcal{T}_j^u$ to be inserted and its visiting sequence $q^\star$ for vehicle j by

$$(k^\star, q^\star) = \underset{\substack{1<q\leq|o_j(0)|+1,\\ k\in\mathcal{T}_j^u}}{\operatorname{argmin}} \{t(o_j(0) \oplus_q k) - t(o_j(0))\}, \quad (10)$$

where the operator $o_j(0) \oplus_q k$ inserts the target location k at the qth position of $o_j(0)$ and $t(o_j(0))$ denotes the total travel time for vehicle j to visit all the targets in $o_j(0)$. If $q = |o_j(0)| + 1$, $o_j(0) \oplus_q k$ puts target location k to the end of $o_j(0)$. Then, $\mathcal{T}_j^u$ and $o_j(0)$ are updated to

$$\mathcal{T}_j^u = \mathcal{T}_j^u \setminus \{k^\star\}, \quad o_j(0) = o_j(0) \oplus_{q^\star} k^\star. \quad (11)$$

The target ordering process continues until $\mathcal{T}_j^u$ is empty.

3.1.2 Task assignment algorithm MC

Different from the EVM, the task assignment algorithm MC determines the visiting sequence of a target location during its clustering process (Bai et al, 2017a). Let $o_j(0)$ for each $j \in \mathcal{R}$ and $\mathcal{T}^u(0)$ be defined as those for EVM. Then, the first target $k^\star$ in $\mathcal{T}^u(0)$ to be clustered, its assigned vehicle $j^\star$ and the inserting position $q^\star$ are

$$(j^\star, k^\star, q^\star) = \underset{\substack{j\in\mathcal{T}^u,\ k\in\mathcal{V},\\ 1<q\leq|o_k|+1}}{\operatorname{argmin}} \{t(o_k \oplus_q j) - t(o_k)\}. \quad (12)$$

Then, $\mathcal{T}^u(0)$ and $o_{j^\star}(0)$ are updated to

$$\mathcal{T}^u(0) = \mathcal{T}^u(0) \setminus \{k^\star\}, \quad o_{j^\star}(0) = o_{j^\star}(0) \oplus_{q^\star} k^\star. \quad (13)$$

The target ordering process continues until $\mathcal{T}^u(0)$ is empty.

As the vehicles move with the unit speed, the travel cost matrix, containing the time for a vehicle to move between a set of specified locations, is symmetric and satisfies the triangular inequality. The worst performance of the solutions, resulting from the EVM and the MC for minimizing (1), has been investigated in Bai et al (2017a) as summarized as follows.

Lemma 1 *If no new target locations appear, both the EVM and the MC guarantee that (1) is within twice of the optimal (Bai et al, 2017a).*

Now, we introduce the algorithms to dynamically assign the target locations based on the EVM and MC.

3.2 Event-triggered dynamic task assignment

In this section, we construct several event-triggered task assignment algorithms to carry out dynamically target reassignment whenever a new target appears. Assume that at time t, a new target location r is generated, and then $\mathcal{T}_{new} = \mathcal{T}_{new} \cup \{r\}$ where $\mathcal{T}_{new}$ is initially empty. Let $o_j(t)$ contain the indices of the ordered target locations that have not been visited on the path of vehicle j at time t for each $j \in \mathcal{R}$, and $o_j(t) = [p_j(t)\ o_j(t)]$ where $p_j(t)$ is the index of the vertex where vehicle j is currently located. If all the target locations on j's path have been visited before the generation of the new target location, vehicle j has stopped moving and is located at the last visited target location, and then $o_j(t) = [p_j(t)]$. To determine how the newly generated target location is assigned, we design several event-triggered task assignment algorithms as follows.

3.2.1 Inserting each newly generated target locations into the vehicles' current paths

The first type of the event-triggered task assignment algorithms considers to insert each newly generated target location into one of the current paths of the vehicles. We first present the event-triggered EVM for dynamically assigning each newly generated target location, and the resulting algorithm is named as EEVME.

EEVME first determines the vehicle $j^\star$ that wins the newly generated target location r by

$$j^\star = \underset{i\in o_j(t),\ j\in\mathcal{R}}{\operatorname{argmin}}\ t(i, r). \quad (14)$$

Then, the sequence for vehicle $j^\star$ to visit r is determined by

$$q^\star = \underset{1<q\leq|o_{j^\star}(t)|+1}{\operatorname{argmin}} \{t(o_{j^\star}(t) \oplus_q r) - t(o_{j^\star}(t))\}. \quad (15)$$

Afterwards, $o_{j^\star}(t)$ is updated to

$$o_{j^\star}(t) = o_{j^\star}(t) \oplus_{q^\star} r. \quad (16)$$

We also consider the event-triggered MC which dynamically inserts each newly generated target location into the current paths of the vehicles, which we call EMCE. The algorithm determines that vehicle $j^\star$ to be assigned with the newly generated target r and the corresponding sequence $q^\star$ for visiting r according to

$$(j^\star, q^\star) = \underset{\substack{j\in\mathcal{R},\\ 1<q\leq|o_j(t)|+1}}{\operatorname{argmin}} \{t(o_j(t) \oplus_q r) - t(o_j(t))\}. \quad (17)$$

Then, $o_{j^\star}(t)$ is updated as (16).

3.2.2 Reassigning all the target locations currently unvisited

The second type of the event-triggered task assignment algorithms considers to reassign all the target locations currently unvisited whenever a new target location is generated. We first present the event-triggered EVM for dynamically reassigning all the unvisited target locations whenever a new target location is generated, and the resulting algorithm is named as EEVMA. Similar to the EVM, EEVMA also first clusters all the unvisited target locations to the vehicles, and then orders the target locations assigned to each vehicle in sequence.

Let $\mathcal{T}^u(t)$ contain the indices of those unvisited targets on the paths of all the vehicles at time t, and $o_j^{2:|o_j(t)|}(t)$ save the ordered target vertices located between the second and the $|o_j(t)|$th locations of $o_j(t)$ if $|o_j(t)| > 1$ where $|o_j(t)|$ is the length of $o_j(t)$. Then,

$$\mathcal{T}^u(t) = \bigcup_{j\in\mathcal{R}} o_j^{2:|o_j(t)|}(t). \tag{18}$$

With the generation of the target r at time t, all the target locations that have not been visited are those in $\mathcal{T}^u(t) = \mathcal{T}^u(t) \cup \{r\}$. Let $o_j(t+1)$, initialized as the index of vehicle j's position $p_j(t)$ for each $j \in \mathcal{R}$, store the indices of the ordered target locations for vehicle j for the task reassignment. Then, for EEVMA, the assignment for all the unvisited target locations in $\mathcal{T}^u(t)$ and the vehicles' path $o_j(t+1)$ for each $j \in \mathcal{R}$ are iteratively updated according to (7) to (11) at the time instant t until all the unvisited target locations are ordered.

The other one is the event-triggered MC for dynamically reassigning all the unvisited target locations whenever a new target location is generated, which we call EMCA. Let $o_j(t)$ and $\mathcal{T}^u(t)$ be defined the same as those for EEVMA. Then, for EMCA, the assignment for all the unvisited target locations in $\mathcal{T}^u(t)$ and the vehicles' path $o_j(t+1)$ for each $j \in \mathcal{R}$ are iteratively updated by (12) and (13) at the time instant t until $\mathcal{T}^u(t)$ is empty.

3.3 Time-triggered dynamic task assignment

The event-triggered dynamic task assignment algorithms might require a higher computational effort to change the assignment of the target locations if the time instants when new target locations are generated are too close. Generally speaking, time-triggered task assignment algorithms carry out the task reassignment within fixed time horizon. However, it is unnecessary to reassign the unvisited target locations if no new target locations have not been generated during a time horizon. Thus, we design several time-triggered algorithms to dynamically change the target assignment at the end of each fixed time horizon H if at least one new target location has been generated during the time horizon. Let $\mathcal{T}_{lH}$ store the indices of newly generated target locations during the lth time horizon where $l \in \{1, ..., \lfloor \frac{L}{H} \rfloor\}$ and L is the whole time horizon during which new target locations are generated. For a positive number a, the flooring function $\lfloor a \rfloor$ returns the largest integer that is smaller than or equal to a. Then, $\mathcal{T}_{new} = \mathcal{T}_{new} \cup \mathcal{T}_{lH}$. Let $o_j(lH)$ contain the indices of the ordered target locations that have not been visited on the path of vehicle j for each $j \in \mathcal{R}$ at time lH, and $o_j(lH) = [p_j(lH)\ o_j(lH)]$. If all the target locations on j's path have been visited before lH, vehicle j stops moving and stays at the lastly visited target location, and then $o_j(lH) = [p_j(lH)]$. To determine how the target locations newly generated during each time horizon are assigned, we design several time-triggered task assignment algorithms as follows.

3.3.1 Inserting the newly generated target locations into the vehicles' current paths

The first type of the time-triggered task assignment algorithms considers to insert the target locations newly generated during each time horizon into the current paths of the employed vehicles. We first present the time-triggered EVM for dynamically assigning the newly generated target locations at the end of each time horizon, and the resulting algorithm is called TEVME. Similar to EEVME, TEVME also first clusters all the newly generated target locations to the vehicles, and then inserts the target locations clustered to each vehicle into the vehicle's current path.

Let $\mathcal{T}_j$, initially empty, store the indices of those target locations in $\mathcal{T}_{lH}$ that have already been assigned to vehicle j for each $j \in \mathcal{R}$, and $\mathcal{T}^u(lH)$ contain the indices of those unclustered targets, which is initialized as $\mathcal{T}_{lH}$. Then, the first target $k^\star$ in $\mathcal{T}^u(lH)$ to be clustered and its assigned vehicle $j^\star$ are determined by

$$(j^\star, k^\star) = \operatorname*{argmin}_{i\in\mathcal{T}_j\cup o_j(lH),\ j\in\mathcal{R},\ k\in\mathcal{T}^u(lH)} t(i,k). \tag{19}$$

After clustering target $k^\star$, $\mathcal{T}^u(lH)$ is updated to

$$\mathcal{T}^u(lH) = \mathcal{T}^u(lH) \setminus \{k^\star\}, \tag{20}$$

while the newly generated targets assigned to vehicle $j^\star$ are updated to

$$\mathcal{T}_{j^\star} = \mathcal{T}_{j^\star} \cup \{k^\star\}. \tag{21}$$

The target clustering procedure continues until $\mathcal{T}^u(lH)$ is empty.

Then, if $\mathcal{T}_j \neq \emptyset$, TEVME determines the first target $k^\star$ in $\mathcal{T}_j$ to be inserted and its visiting sequence $q^\star$ for vehicle j by

$$(k^\star, q^\star) = \operatorname*{argmin}_{1<q\leq|o_j(lH)|+1,\ k\in\mathcal{T}_j} \{t(o_j(lH) \oplus_q k) - t(o_j(lH))\}. \tag{22}$$

Afterwards, $\mathcal{T}_j$ and $o_j(lH)$ are updated to

$$\mathcal{T}_j = \mathcal{T}_j \setminus \{k^\star\}, \quad o_j(lH) = o_j(lH) \oplus_{q^\star} k^\star. \tag{23}$$

The target ordering process continues until $\mathcal{T}_j$ is empty.

Table 1 The mechanisms used to construct the dynamic task assignment algorithms.

Trigger mechanism	Origin algorithm	Dynamic task assignment algorithms	
		Only assign newly generated targets	Reassign all the unvisited targets
Event-triggered	G	EGE	EGA
	EVM	EEVME	EEVMA
	MC	EMCE	EMCA
Time-triggered	G	TGE	TGA
	EVM	TEVME	TEVMA
	MC	TMCE	TMCA

The other is the time-triggered MC for dynamically assigning the newly generated target locations into the current paths of the vehicles at the end of each time horizon, which we call TMCE. Let $\mathcal{T}^u(lH)$ contain the indices of those unclustered targets, which is initialized as $\mathcal{T}_{lH}$. Then, for TMCE, the assignment of the target locations in $\mathcal{T}^u(lH)$ and the vehicles' paths $o_j(lH)$ for every $j \in \mathcal{R}$ are iteratively updated by (12) and (13) at each time instant $t = lH$ until $\mathcal{T}^u(lH)$ is empty.

3.3.2 Reassigning all the target locations currently unvisited

The second type of the time-triggered task assignment algorithms is to reassign all the target locations currently unvisited at the end of each time horizon. We first present the time-triggered EVM for dynamically reassigning all the unvisited target locations at the end of each time horizon, and the resulting algorithm is called TEVMA. Similar to the EVM, TEVMA also first clusters all the unvisited target locations to the vehicles, and then orders the target locations assigned to each vehicle in sequence.

Let $\mathcal{T}^u(lH)$ contain the indices of those unvisited targets on the paths of all the vehicles at time lH, and $o_j^{2:|o_j(lH)|}(lH)$ store the indices of the ordered target locations that have not been visited on vehicle j's path if $|o_j(lH)| > 1$. Then,

$$\mathcal{T}^u(lH) = \bigcup_{j\in\mathcal{R}} o_j^{2:|o_j(lH)|}(lH). \tag{24}$$

With the generation of new target locations $\mathcal{T}_{lH}$ during the time horizon from $t = (l-1)H$ to $t = lH$, all the target locations that have not been visited are those in $\mathcal{T}^u(lH) = \mathcal{T}^u(lH) \cup \mathcal{T}_{lH}$. Let $o_j(lH+1)$, initialized as the index of vehicle j's position $p_j(lH)$ for each $j \in \mathcal{R}$, store the indices of the ordered target locations for carrying out the task reassignment at time lH. Then, for TEVMA, the assignment for all the unvisited target locations in $\mathcal{T}^u(lH)$ and the vehicles' path $o_j(lH+1)$ for each $j \in \mathcal{R}$ are iteratively updated by (7) to (11) at the time instant $t = lH$ until all the unvisited target locations are ordered.

The other is the time-triggered MC for dynamically reassigning all the unvisited target locations at the end of each time horizon, which we call TMCA. Let $o_j(lH+1)$ and $\mathcal{T}^u(lH)$ be defined the same as those for TEVMA. Then, for TMCA, the assignment for all the unvisited target locations in $\mathcal{T}^u(lH)$ and the vehicles' path $o_j(lH+1)$ for each $j \in \mathcal{R}$ are iteratively updated by (12) and (13) at the time instant $t = lH$ until $\mathcal{T}^u(lH)$ is empty.

Now we have presented all the design of the algorithms. In the following section, we carry out simulation studies.

4 Simulations

Monte Carlo simulations are carried out to test the proposed algorithms compared with the popular greedy task assignment algorithm (G) where vehicles always move towards the nearest unassigned target location. The mechanisms to construct the dynamic task assignment algorithms are shown in Table 1. All the experiments have been performed on an Intel Core $i5-4590$ CPU 3.30 GHz with 8 GB RAM, and the algorithms are compiled by Matlab under Windows 7. Let f_{MST} to be the sum of all the edge weights in one minimum spanning tree (MST) of the weighted target-vehicle graph $\mathcal{G}$ whose vertices contain the indices of all the vehicles' initial locations in $\mathcal{R}$ and the targets' locations in $\mathcal{T}$. For each pair of nodes of $\mathcal{G}$, if both nodes correspond to some vehicles' initial locations, its edge weight is zero; otherwise, the edge weight is the Euclidean distance between the two nodes. The solution quality of each algorithm is quantified by

$$q = \frac{f}{f_{MST}}, \tag{25}$$

where f is the objective value in (1). Since f_{MST} is a lower bound of the total travel time of an optimal solution (Rathinam et al, 2007; Bai et al, 2017b), a value of the ratio q closer to 1 means a better performance of the solution. An MST of the weighted target-vehicle graph $\mathcal{G}$ can be obtained by Algorithm 1 in Bai et al (2017a).

The algorithms are first tested on the task assignment problem in which $n = 30$ target locations initially

Table 2 The average solution quality q of the algorithms (A) on 100 test instances for the task assignment problem with $n = 30$ target locations and $m = 5$ vehicles under different arrival rates r (Hz) of new target locations where the initial target assignment is from the EVM.

A \ r	0.001	0.002	0.004	0.006	0.008	0.010
EGE	1.3646	1.5005	1.7138	1.8994	2.0708	2.2443
EGA	1.3092	1.3996	1.5479	1.6757	1.7731	1.8687
EEVME	1.3224	1.4151	1.5576	1.6592	1.7359	1.7956
EEVMA	1.3177	1.4072	1.5439	1.6440	1.7219	1.7862
EMCE	1.3180	1.4079	1.5458	1.6508	1.7381	1.8054
EMCA	1.3028	1.3914	1.5341	1.6599	1.7504	1.8421
TGE_1	1.3620	1.4912	1.6861	1.8507	1.9912	2.1290
TGA_1	1.3198	1.4153	1.5656	1.6919	1.7930	1.8861
$TEVME_1$	1.3316	1.4288	1.5757	1.6835	1.7643	1.8243
$TEVMA_1$	1.3275	1.4239	1.5658	1.6756	1.7567	1.8226
$TMCE_1$	1.3257	1.4211	1.5657	1.6694	1.7600	1.8307
$TMCA_1$	1.3149	1.4082	1.5574	1.6777	1.7759	1.8594
TGE_2	1.3638	1.4960	1.7000	1.8747	2.0308	2.1874
TGA_2	1.3143	1.4086	1.5619	1.6897	1.7835	1.8805
$TEVME_2$	1.3255	1.4219	1.5671	1.6715	1.7528	1.8106
$TEVMA_2$	1.3221	1.4168	1.5553	1.6579	1.7372	1.8046
$TMCE_2$	1.3213	1.4145	1.5561	1.6605	1.7501	1.8203
$TMCA_2$	1.3093	1.4006	1.5509	1.6718	1.7639	1.8548

Table 3 The corresponding average computation time (s) for the algorithms (A) to obtain the solutions for the task assignment problem with $n = 30$ target locations and $m = 5$ vehicles under different target arrival rates r (Hz), where the initial assignment of the target locations is from the EVM.

A \ r	0.001	0.002	0.004	0.006	0.008	0.010
EGE	0.0009	0.0016	0.0027	0.0038	0.0049	0.0062
EGA	0.0046	0.0083	0.0165	0.0252	0.0323	0.0444
EEVME	0.0136	0.0219	0.0438	0.0678	0.0924	0.1191
EEVMA	0.0753	0.1634	0.3557	0.5988	0.8988	1.3541
EMCE	0.0126	0.0214	0.0414	0.0662	0.0878	0.1146
EMCA	0.0818	0.1503	0.3324	0.5344	0.7942	1.0851
TGE_1	0.0022	0.0034	0.0052	0.0067	0.0077	0.0087
TGA_1	0.0043	0.0066	0.0103	0.0128	0.0139	0.0159
$TEVME_1$	0.0307	0.0502	0.0812	0.1090	0.1339	0.1605
$TEVMA_1$	0.0684	0.1195	0.1960	0.2777	0.3393	0.4160
$TMCE_1$	0.0191	0.0287	0.0440	0.0571	0.0708	0.0832
$TMCA_1$	0.0426	0.0688	0.1069	0.1389	0.1683	0.1996
TGE_2	0.0026	0.0040	0.0066	0.0089	0.0104	0.0125
TGA_2	0.0052	0.0086	0.0149	0.0199	0.0251	0.0278
$TEVME_2$	0.0370	0.0585	0.0939	0.1266	0.1445	0.1843
$TEVMA_2$	0.0872	0.1576	0.2940	0.4367	0.5707	0.7336
$TMCE_2$	0.0276	0.0411	0.0641	0.0864	0.1086	0.1382
$TMCA_2$	0.0793	0.1360	0.2370	0.3346	0.4286	0.5358

distribute in a square area with edge length 10^3m and the number of dispersed vehicles is $m = 5$. One hundred test instances of the initial positions of the 30 targets and 5 vehicles are randomly generated, where for each instance, the arrival times of new target locations, determined by the Poisson process under different rates $r \in \{0.001, 0.002, 0.004, 0.006, 0.008, 0.010\}$, are investigated. The appearance of the time instants when new targets arrive is generated ten times for each test instance under each arrival rate r, and the positions of the newly generated target locations are randomly generated. The event-triggered dynamic task assignment algorithms will make a task reassignment whenever a new target appears. For each test instance, the whole time horizon L of the time-triggered assignment algorithms is set to be the lower bound of the minimal total travel time for the vehicles to visit all the target locations initially known, which is obtained by solving for an MST of the corresponding weighted target-vehicle graph. The average whole time horizon L of the 100 test instances is 3330.5s, and an average of Lr new target locations appear under each pair of L and r. The time-triggered assignment algorithms are tested under two different time horizons H with $\lfloor \frac{L}{H} \rfloor = 10$ and $\lfloor \frac{L}{H} \rfloor = 20$, respectively. During each time horizon H, the time-triggered assignment algorithms will be activated if at least one new target location arrives during this time horizon. The time-triggered assignment algorithms are marked with subscript 1 if they are triggered with $\lfloor \frac{L}{H} \rfloor = 10$ and otherwise 2.

To invest the impact of the initial assignment of the target locations on the following task assignment, the algorithms are first tested with the assignment of the initially known target locations resulting from the EVM. The average q of the algorithms on all the instances under each arrival rate r is shown in Table 2, and the corresponding average computation time for the algorithms to plan the paths for the vehicles is listed in Table 3. In Table 2, the proposed algorithms generally have a smaller average q compared with the greedy algorithms under different scenarios, and the average q of each proposed task assignment algorithm is at worst around 1.32 times of the optimal under a small arrival rate $r = 0.001$ of new targets, which shows the satisfying performance of the algorithms for the static multi-vehicle task assignment. Secondly, for each assignment algorithm, the average q shown in Table 2 increases with the increase of the arrival rate r. The reason is that the task assignment algorithms only have the positions of those target locations just appeared to make task reassignment to the vehicles at each triggered time instant, where the positions of the target locations that will appear in the coming time horizon are unknown. As a result, the more frequent generations of new target locations from a given time, the higher is their impact on the quality of the current assignment of the target locations. However, the average q of the algorithms is still within twice of the optimal even under the higher arrival rate $r = 0.010$ of the new targets, which shows the robustness of the algorithms. Thirdly, under

Table 4 The average solution quality q of the algorithms (A) on 100 test instances for the task assignment problem with $n = 30$ target locations and $m = 5$ vehicles under different arrival rates r (Hz) of new target locations where the initial assignment of the target locations is from the MC.

A \ r	0.001	0.002	0.004	0.006	0.008	0.010
EGE	1.2940	1.4310	1.6461	1.8385	2.0172	2.1970
EGA	1.2709	1.3788	1.5432	1.6768	1.7819	1.8738
EEVME	1.2705	1.3784	1.5415	1.6599	1.7466	1.8190
EEVMA	1.2756	1.3884	1.5398	1.6505	1.7337	1.8007
EMCE	1.2669	1.3726	1.5351	1.6562	1.7543	1.8347
EMCA	1.2642	1.3726	1.5348	1.6597	1.7589	1.8477
TGE_1	1.2929	1.4240	1.6250	1.7937	1.9414	2.0828
TGA_1	1.2787	1.3899	1.5600	1.6960	1.8047	1.8936
$TEVME_1$	1.2780	1.3917	1.5604	1.6850	1.7806	1.8514
$TEVMA_1$	1.2818	1.3954	1.5575	1.6781	1.7668	1.8378
$TMCE_1$	1.2735	1.3865	1.5507	1.6748	1.7755	1.8555
$TMCA_1$	1.2724	1.3864	1.5518	1.6784	1.7808	1.8652
TGE_2	1.2939	1.4275	1.6362	1.8145	1.9764	2.1387
TGA_2	1.2741	1.3857	1.5567	1.6926	1.7966	1.8909
$TEVME_2$	1.2737	1.3845	1.5537	1.6730	1.7634	1.8356
$TEVMA_2$	1.2788	1.3924	1.5525	1.6633	1.7469	1.8177
$TMCE_2$	1.2698	1.3795	1.5439	1.6642	1.7652	1.8505
$TMCA_2$	1.2686	1.3797	1.5475	1.6694	1.7745	1.8623

Table 5 The corresponding average computation time (s) for the algorithms (A) to obtain the solutions for the task assignment problem with $n = 30$ target locations and $m = 5$ vehicles under different target arrival rates r (Hz), where the initial assignment of the target locations is from the MC.

A \ r	0.001	0.002	0.004	0.006	0.008	0.010
EGE	0.0009	0.0016	0.0027	0.0038	0.0049	0.0061
EGA	0.0041	0.0077	0.0169	0.0243	0.0316	0.0423
EEVME	0.0128	0.0225	0.0426	0.0659	0.0863	0.1242
EEVMA	0.0744	0.1592	0.3673	0.6426	0.9684	1.4286
EMCE	0.0123	0.0216	0.0407	0.0648	0.0860	0.1195
EMCA	0.0418	0.1432	0.3119	0.4965	0.7343	1.1020
TGE_1	0.0021	0.0034	0.0050	0.0065	0.0074	0.0089
TGA_1	0.0038	0.0062	0.0098	0.0120	0.0130	0.0140
$TEVME_1$	0.0266	0.0411	0.0705	0.0852	0.1141	0.1359
$TEVMA_1$	0.0481	0.0887	0.1453	0.2063	0.2704	0.3372
$TMCE_1$	0.0176	0.0264	0.0404	0.0548	0.0640	0.0773
$TMCA_1$	0.0473	0.0800	0.1299	0.1759	0.2291	0.2686
TGE_2	0.0027	0.0039	0.0065	0.0090	0.0112	0.0130
TGA_2	0.0048	0.0082	0.0144	0.0200	0.0247	0.0267
$TEVME_2$	0.0338	0.0543	0.0884	0.1227	0.1532	0.1834
$TEVMA_2$	0.0668	0.1250	0.2349	0.3456	0.4699	0.6131
$TMCE_2$	0.0231	0.0325	0.0489	0.0649	0.0783	0.0953
$TMCA_2$	0.0656	0.1144	0.2014	0.2867	0.3707	0.4665

each arrival rate r, each event-triggered algorithm of EGE, EGA, EEVME, EEVMA, EMCE and EMCA respectively has a smaller q than the corresponding time-triggered algorithm of TGE, TGA, TEVME, TEVMA, TMCE and TMCA under both of the two time horizons, as shown in Table 2. This can be partly explained by the fact that the event-triggered algorithms perform task reassignment whenever a new target appear, which makes them have a faster response to the newly generated targets.

However, Table 3 shows that the better performance of the event-triggered algorithms is generally at the cost of a longer computation time with the exception of EEVME and EMCE compared with TEVME and TMCE. That might be explained by the NP-hardness of the task assignment problem where iteratively inserting each newly generated target of all the targets generated during each time horizon into the current vehicles' paths is less time-consuming than that of inserting a small bundle of the targets generated during the time horizon into the current vehicles' paths. Furthermore, Table 2 shows that EEVMA, and TEVMA have a smaller q in the comparison respectively with EEVME and TEVME which just insert the newly generated target locations into the current paths of the vehicles. We observe that reassigning all the remaining unvisited target locations enables EEVMA and TEVMA to plan a more efficient sequence for the vehicles to visit all the unvisited target locations in response to the newly generated target locations. In addition, the average q of $TEVME_2$, $TEVMA_2$, $TMCE_2$ and $TMCA_2$ is smaller than that of $TEVME_1$, $TEVMA_1$, $TMCE_1$ and $TMCA_1$, respectively. This is because shortening the time horizon H from $\lfloor \frac{L}{H} \rfloor = 10$ to $\lfloor \frac{L}{H} \rfloor = 20$ enables $TEVME_2$, $TEVMA_2$, $TMCE_2$ and $TMCA_2$ to readjust the vehicles' current paths in a faster response to the newly generated target locations compared with $TEVME_1$, $TEVMA_1$, $TMCE_1$ and $TMCA_1$. However, the smaller q of the algorithms is also at the cost of longer computation time as shown in Table 3. What is more, if the time horizon H is short enough, the proposed time-triggered algorithms TEVME, TEVMA, TMCE and TMCA are respectively in essence the event-triggered EEVME, EEVMA, EMCE and EMCA as at the end of each time horizon TEVME, TEVMA, TMCE and TMCA are triggered only if at least one new target locations is generated during the horizon.

Finally, EMCA has the smallest average q among all the algorithms when the arrival rate r is low as in the set $\{0.001, 0.002, 0.004\}$ while EEVMA is the best under a higher $r \in \{0.006, 0.008, 0.010\}$. This is interesting since in Bai et al (2017a) we have both theoretically proved and experimentally shown that the MC is better than the EVM in the static multi-vehicle task assignment problem where no new target locations are dynamically generated. The reason can be that EMCA can better assign the existing unvisited target locations if not so many new target locations appear as in the static multi-vehicle task assignment Bai et al (2017a) while EEVME performs better under a higher r as it assigns a target

Table 6 The average solution quality q of the algorithms (A) on 50 test instances for the task assignment problem with $n = 50$ target locations and $m = 10$ vehicles under different arrival rates r (Hz) of new target locations where the initial assignment is from the EVM.

A \ r	0.001	0.002	0.004	0.006	0.008	0.010
EGE	1.3011	1.3917	1.5506	1.6978	1.8183	1.9325
EGA	1.2709	1.3375	1.4715	1.5946	1.6929	1.7855
EEVME	1.2840	1.3667	1.5120	1.6334	1.7226	1.8045
EEVMA	1.2794	1.3580	1.4941	1.6088	1.6967	1.7823
EMCE	1.2817	1.3615	1.5000	1.6222	1.7098	1.7992
EMCA	1.2646	1.3343	1.4660	1.5822	1.6839	1.7742
TGE	1.3013	1.3906	1.5459	1.6848	1.7985	1.9069
TGA	1.2763	1.3454	1.4829	1.6052	1.7059	1.8032
TEVME	1.2864	1.3708	1.5131	1.6390	1.7293	1.8149
TEVMA	1.2823	1.3611	1.4985	1.6179	1.7144	1.7900
TMCE	1.2843	1.3658	1.5022	1.6254	1.7162	1.8063
TMCA	1.2720	1.3455	1.4772	1.5993	1.6989	1.7900

Table 7 The corresponding average computation time (s) for the algorithms (A) to obtain the solutions for the task assignment problem with $n = 50$ target locations and $m = 10$ vehicles under different target arrival rates r (Hz), where the initial assignment is from the EVM.

A \ r	0.001	0.002	0.004	0.006	0.008	0.010
EGE	0.0015	0.0023	0.0039	0.0056	0.0072	0.0094
EGA	0.0067	0.0121	0.0224	0.0353	0.0427	0.0603
EEVME	0.0168	0.0257	0.0454	0.0630	0.0792	0.0837
EEVMA	0.0821	0.2239	0.2270	0.4740	0.7308	0.9909
EMCE	0.0134	0.0222	0.0456	0.0608	0.0768	0.1011
EMCA	0.0555	0.0718	0.1620	0.2186	0.2791	0.3703
TGE	0.0031	0.0041	0.0065	0.0078	0.0091	0.0105
TGA	0.0055	0.0084	0.0120	0.0140	0.0152	0.0167
TEVME	0.0429	0.0694	0.1025	0.1161	0.1604	0.1769
TEVMA	0.0670	0.1233	0.1770	0.2676	0.2814	0.3845
TMCE	0.0193	0.0411	0.0620	0.0815	0.1272	0.1529
TMCA	0.0597	0.1321	0.2243	0.2476	0.2762	0.3914

based on both the vehicles' current locations and the locations of the targets already assigned to the vehicles and finally inserts the targets using the marginal-cost principle.

Then, the algorithms are tested with the initial assignment of the target locations resulting from the MC where the arrival time instants of each new target location under every arrival rate r and their positions are the same as those for the previous experiment when the algorithms are tested with the assignment of the initially known target locations resulting from the EVM. The average q of the algorithms on the instances under each arrival rate r is shown in Table 4, and the corresponding average computation time for the algorithms to plan the paths for the vehicles is listed in Table 5. Generally speaking, the average q of the algorithms shown in Table 4 has the same changing trend with those shown in Table 2 when increasing r, and EMCA has the best performance when $r \in \{0.001, 0.002, 0.004\}$ while EEVMA is the best under a higher $r \in \{0.006, 0.008, 0.010\}$, which shows the algorithms' robustness.c

However, some new phenomena are noticed. Firstly, under a low arrival rate $r \in \{0.001, 0.002, 0.004\}$, each of the proposed algorithms and the compared greedy algorithms has a better performance than the corresponding same algorithm with the initial assignment of the target locations resulting from the EVM in the comparison of Table 2 and Table 4. The reason can be that the assignment of the initial target locations resulting from the MC is better than those from the EVM as the arrival time as well as the positions of the new generated target locations are the same for the two simulation setups. Secondly, when $r \in \{0.006, 0.008, 0.010\}$, the average q of each proposed algorithm with the initial assignment resulting from MC shown in Table 4 is generally larger than that of the average q of the same algorithm with the initial assignment resulting from the EVM as shown in Table 2. That again shows that the EVM-based algorithms are more robust against a higher arrival rate r of new target locations compared with the MC-based algorithms as previously analyzed, which shows EMCA has the smallest average q among all the algorithms when $r \in \{0.001, 0.002, 0.004\}$ while EEVMA is the best under $r \in \{0.006, 0.008, 0.010\}$.

To further test the scalability of the proposed algorithms, we test the algorithms on the task assignment problem in which $m = 10$ dispersed vehicles and $n = 50$ target locations initially distribute in the same square area. Fifty test instances of the initial positions of the 50 targets and 10 vehicles are randomly generated, where for each instance, the arrival times of new target locations determined by the Poisson process under different rates $r \in \{0.001, 0.002, 0.004, 0.006, 0.008, 0.010\}$ are investigated. Ten appearances of the time instants when new targets arrive are generated for each test instance under each arrival rate r, and the newly generated target locations are randomly distributed. For each test instance, the whole time horizon L of the time-triggered assignment algorithms is also set to be the lower bound of the minimal total travel time for the vehicles to visit all the target locations initially known. The average whole time horizon L of the 50 test instances is 4117.6s, and the time-triggered assignment algorithms are tested under the time horizon H with $\lfloor \frac{L}{H} \rfloor = 20$.

The average q of the algorithms with the initial assignment resulting from both the EVM and the MC are respectively shown in Table 6 and Table 8, and the corresponding average computation time for each algorithm to achieve the solutions are shown in Table 7 and Table 9. Firstly, the average q of each proposed algo-

Table 8 The average solution quality q of the algorithms (A) on 50 test instances for the task assignment problem with $n = 50$ target locations and $m = 10$ vehicles under different arrival rates r (Hz) of new target locations where the initial assignment is from the MC.

A \ r	0.001	0.002	0.004	0.006	0.008	0.010
EGE	1.2376	1.3281	1.4895	1.6386	1.7600	1.8793
EGA	1.2284	1.3085	1.4539	1.5831	1.6863	1.7818
EEVME	1.2261	1.3055	1.4513	1.5799	1.6836	1.7785
EEVMA	1.2344	1.3204	1.4697	1.5902	1.6835	1.7740
EMCE	1.2241	1.3027	1.4437	1.5715	1.6700	1.7680
EMCA	1.2210	1.3006	1.4418	1.5641	1.6652	1.7619
TGE	1.2371	1.3269	1.4830	1.6239	1.7388	1.8508
TGA	1.2301	1.3100	1.4546	1.5803	1.6871	1.7848
TEVME	1.2287	1.3101	1.4566	1.5867	1.6926	1.7847
TEVMA	1.2328	1.3166	1.4599	1.5891	1.6920	1.7817
TMCE	1.2263	1.3066	1.4478	1.5747	1.6770	1.7743
TMCA	1.2230	1.3040	1.4457	1.5714	1.6736	1.7693

Table 9 The corresponding average computation time (s) for the algorithms (A) to obtain the solutions for the task assignment problem with $n = 50$ target locations and $m = 10$ vehicles under different target arrival rates r (Hz), where the initial assignment is from the MC.

A \ r	0.001	0.002	0.004	0.006	0.008	0.010
EGE	0.0015	0.0023	0.0039	0.0055	0.0071	0.0092
EGA	0.0066	0.0108	0.0229	0.0322	0.0427	0.0540
EEVME	0.0131	0.0282	0.0459	0.0637	0.0836	0.1100
EEVMA	0.0795	0.1294	0.2327	0.4112	0.4343	0.8228
EMCE	0.0166	0.0280	0.0367	0.0512	0.0579	0.1094
EMCA	0.0635	0.1203	0.2396	0.3775	0.5255	0.6859
TGE	0.0030	0.0040	0.0060	0.0073	0.0087	0.0101
TGA	0.0049	0.0069	0.0105	0.0119	0.0128	0.0142
TEVME	0.0349	0.0629	0.1309	0.1689	0.2050	0.2053
TEVMA	0.0546	0.1039	0.1735	0.2666	0.4016	0.4453
TMCE	0.0190	0.0396	0.0527	0.0583	0.0947	0.1369
TMCA	0.0730	0.0784	0.1930	0.2198	0.2731	0.4084

rithm shown in Table 6 and Table 8 also increases with a higher arrival rate r of new targets as those in Table 2 and Table 4. However, the average q of the algorithms under different r is within twice of the optimal, which shows the algorithms' satisfying performance. Secondly, the average q of each proposed algorithm shown in Table 8 under each arrival rate r is smaller than that the corresponding one shown in Table 6, which shows the initial assignment of the target locations resulting from the MC leads to a smaller q for the algorithms. The reason is that the new target locations arrive during the movement of the vehicles where a better initial assignment leads to a faster visiting of the existing target locations. Thirdly, Table 6 and Table 8 both show that the average q of EMCA under each r is the smallest among all the proposed algorithms, which verifies the satisfying performance of EMCA. However, the average computation time of EMCE is the smallest among all the proposed algorithms as shown in Table 7 and Table 9.

As discussed above, we first conclude that a shorter time horizon H leads to a better performance for the time-triggered task assignment algorithms according to the average solution quality q shown in Table 2 and Table 4. However, the average computation time of the time-triggered task assignment algorithms with a shorter H is longer according to Table 3 and Table 5. Secondly, the average q listed in Table 2, Table 4, Table 6 and Table 8 shows that each of the proposed event-triggered task assignment algorithms performs better than the corresponding time-triggered task assignment algorithm under each arrival rate r. As an example, EEVME has the smaller q compared with TEVME under each r, and so does EEVMA compared with TEVMA. That is because the event-triggered task assignment algorithms assign targets whenever a new target location appears, which guides to adjust the vehicles' paths in a faster response to the newly generated target locations compared with the time-triggered task assignment algorithms. However, the computation time of the event-triggered task assignment algorithms that reassign all the target locations currently unvisited is longer compared with the corresponding time-triggered task assignment algorithms that make a task reassignment during each fixed time period. Furthermore, among the event-triggered task assignment algorithms, EEVMA and EMCA perform better than EEVME and EMCE, respectively. However, the better performance of them is generally at the cost of a longer computation time as shown in Table 3, Table 5, Table 7 and Table 9. Finally, Table 3, Table 5, Table7 and Table 9 show the scalability of the proposed task assignment algorithms for dealing with the dynamic multi-vehicle task assignment with moderate numbers of targets and vehicles under different arrival rates of new targets. Generally speaking, for the dynamic multi-vehicle task assignment, it is suggested to use EMCA to plan routes for the vehicles if more computation time is allowed as it generally has the best performance among the algorithms, and otherwise one may choose to use EMCE as it can still achieve a satisfying solution under a shorter computation time compared with EMCA.

5 Conclusion

In this paper, we have investigated the dynamic task assignment for multiple vehicles to visit a set of target locations where some target locations are initially known and the others are dynamically randomly generated. The problem is to employ the vehicles to visit all the target locations while minimizing the vehicles' total travel time. Both event-triggered and time-triggered dynamic target assignments have been studied to investigate when the appropriate time instants should be to change the assignment of the target locations in response to the newly generated target locations. In addition, for both the event-triggered and time-triggered task assignments, we have designed several task assignment algorithms to investigate how to assign the newly generated target locations based on the existing assignment of the vehicles. Numerical simulations have shown the satisfying performance of the event-triggered task assignment algorithms compared with their time-triggered counterparts under different arrival rates of the newly generated target locations. Future works will focus on developing more efficient task assignment algorithms to deal with the multi-vehicle task assignment in a more complex dynamic environment in which potential malfunctions of the vehicles will be considered.

References

Bai X, Yan W, Cao M (2017a) Clustering-based algorithms for multivehicle task assignment in a time-invariant drift field. IEEE Robotics and Automation Letters 2(4):2166–2173

Bai X, Yan W, Cao M, Huang J (2017b) Task assignment for robots with limited communication. In: Control Conference (CCC), 2017 36th Chinese, IEEE, pp 6934–6939

Bai X, Yan W, Ge SS, Cao M (2018) An integrated multi-population genetic algorithm for multi-vehicle task assignment in a drift field. Information Sciences 453:227–238

Bai X, Cao M, Yan W, Ge SS (2019a) Efficient routing for precedence-constrained package delivery for heterogeneous vehicles. IEEE Transactions on Automation Science and Engineering 17(1):248–260

Bai X, Yan W, Cao M, Xue D (2019b) Distributed multi-vehicle task assignment in a time-invariant drift field with obstacles. IET Control Theory & Applications 13(17):2886–2893

Chen B, Cheng HH (2010) A review of the applications of agent technology in traffic and transportation systems. IEEE Transactions on Intelligent Transportation Systems 11(2):485–497

Choi HL, Brunet L, How JP (2009) Consensus-based decentralized auctions for robust task allocation. IEEE Transactions on Robotics 25(4):912–926

Chopra S, Egerstedt M (2015) Spatio-temporal multi-robot routing. Automatica 60:173–181

Dahl TS, Matarić M, Sukhatme GS (2009) Multi-robot task allocation through vacancy chain scheduling. Robotics and Autonomous Systems 57(6-7):674–687

Di Paola D, Gasparri A, Naso D, Lewis FL (2015) Decentralized dynamic task planning for heterogeneous robotic networks. Autonomous Robots 38(1):31–48

Escobar JW, Linfati R, Toth P, Baldoquin MG (2014) A hybrid granular tabu search algorithm for the multi-depot vehicle routing problem. Journal of Heuristics 20(5):483–509

Fua CH, Ge SS (2005) Cobos: Cooperative backoff adaptive scheme for multirobot task allocation. IEEE Transactions on Robotics 21(6):1168–1178

Gerkey BP, Matarić MJ (2004) A formal analysis and taxonomy of task allocation in multi-robot systems. The International Journal of Robotics Research 23(9):939–954

Korsah GA, Stentz A, Dias MB (2013) A comprehensive taxonomy for multi-robot task allocation. The International Journal of Robotics Research 32(12):1495–1512

Kuo Y (2010) Using simulated annealing to minimize fuel consumption for the time-dependent vehicle routing problem. Computers & Industrial Engineering 59(1):157–165

Lagoudakis MG, Berhault M, Koenig S, Keskinocak P, Kleywegt AJ (2004) Simple auctions with performance guarantees for multi-robot task allocation. In: 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE, vol 1, pp 698–705

Laporte G (2009) Fifty years of vehicle routing. Transportation Science 43(4):408–416

Lenstra JK, Kan A (1981) Complexity of vehicle routing and scheduling problems. Networks 11(2):221–227

Moon S, Oh E, Shim DH (2013) An integral framework of task assignment and path planning for multiple unmanned aerial vehicles in dynamic environments. Journal of Intelligent & Robotic Systems 70(1-4):303–313

Prins C (2004) A simple and effective evolutionary algorithm for the vehicle routing problem. Computers & Operations Research 31(12):1985–2002

Rathinam S, Sengupta R, Darbha S (2007) A resource allocation algorithm for multivehicle systems with nonholonomic constraints. IEEE Transactions on Automation Science and Engineering 4(1):98–104

Shima T, Rasmussen SJ, Sparks AG, Passino KM (2006) Multiple task assignments for cooperating uninhabited aerial vehicles using genetic algorithms. Computers & Operations Research 33(11):3252–3269

Smith SL, Bullo F (2009) Monotonic target assignment for robotic networks. IEEE Transactions on Automatic Control 54(9):2042–2057

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

Yu J, Chung SJ, Voulgaris PG (2015) Target assignment in robotic networks: Distance optimality guarantees and hierarchical strategies. IEEE Transactions on Automatic Control 60(2):327–341

Zhao W, Meng Q, Chung PW (2016) A heuristic distributed task allocation method for multivehicle multitask problems and its application to search and rescue scenario. IEEE Transactions on Cybernetics 46(4):902–915

Zhu D, Huang H, Yang SX (2013) Dynamic task assignment and path planning of multi-auv system based on an improved self-organizing map and velocity synthesis method in three-dimensional underwater workspace. IEEE Transactions on Cybernetics 43(2):504–514

Xiaoshan Bai received his first Ph.D. degree in Systems Engineering from the University of Groningen, the Netherlands, in 2018, and the second Ph.D. degree in Armament Science and Technology from the Northwestern Polytechnical University, Xi'an, China, in 2020. He was a Research Fellow in the Department of Electrical and Computer Engineering, National University of Singapore, Singapore, from January to July 2015. From November 2018 to July 2019, he was a lecturer with the Faculty of Science and Engineering, University of Groningen, the Netherlands. Since August 2019, he has been a postdoctoral researcher with the Department of Cognitive Robotics, Delft University of Technology, the Netherlands. His main research interests include multi-vehicle/robot task assignment, logistic scheduling, and vehicle path planning.

Ming Cao is currently a Professor of Systems and Control with the Engineering and Technology Institute (ENTEG) at the University of Groningen, the Netherlands, where he started as a tenure-track Assistant Professor in 2008. He received the Bachelor's degree in 1999 and the Master's degree in 2002 from Tsinghua University, Beijing, China, and the Ph.D. in 2007 from Yale University, New Haven, CT, USA, all in Electrical Engineering. From September 2007 to August 2008, he was a postdoctoral research associate with the Department of Mechanical and Aerospace Engineering at Princeton University, Princeton, NJ, USA. He worked as a research intern during the summer of 2006 with the Mathematical Sciences Department at the IBM T. J. Watson Research Center, NY, USA. He is the 2017 and inaugural recipient of the Manfred Thoma Medal from the International Federation of Automatic Control (IFAC) and the 2016 recipient of the European Control Award sponsored by the European Control Association (EUCA). He is an Associate Editor of IEEE Transactions on Automatic Control, and IEEE Transactions on Circuits and Systems, and a senior editor of Systems and Control Letters. He is also a vice chair of the IFAC Technical Committee on Large-Scale Complex Systems. His main research interests are in autonomous agents and multi-agent systems, mobile sensor networks and complex networks.

Weisheng Yan received the B.Sc., M.Sc. and Ph.D. degrees in mechanical engineering and automatic control from Northwestern Polytechnical University, Xi'an, China, in 1990, 1993 and 1999, respectively. Since 1993, he has been with the Department of Mechanical Engineering and Automation, School of Marine Science and Technology, NWPU, Xi'an, China, as Lecturer (1993), Associate Professor (1999) and Full Professor (2003). From 2003 to 2013, he served as the Dean of the Department of Mechanical Engineering and Automation. His main research interests are in advanced control theory and control of underwater vehicles.