Farouq Zitouni
Saad Harous
Ramdane Maamri

TOWARDS A DISTRIBUTED SOLUTION TO MULTI-ROBOT TASK ALLOCATION PROBLEM WITH ENERGETIC AND SPATIOTEMPORAL CONSTRAINTS

Abstract *This paper tackles the Multi-Robot Task Allocation problem. It consists of two distinct sets: a set of tasks (requiring resources), and a set of robots (offering resources). Then, the tasks are allocated to robots while optimizing a certain objective function subject to some constraints; e.g., allocating the maximum number of tasks, minimizing the distances traveled by the robots, etc. Previous works mainly optimized the temporal and spatial constraints, but no work focused on energetic constraints. Our main contribution is the introduction of energetic constraints on multi-robot task allocation problems. In addition, we propose an allocation method based on parallel distributed guided genetic algorithms and compare it to two state-of-the-art algorithms. The performed simulations and obtained results show the effectiveness and scalability of our solution, even in the case of a large number of robots and tasks. We believe that our contribution is applicable in many contemporary areas of research such as smart cities and related topics.*

Keywords multi-robot systems, multi-robot task allocation, energetic constraints, spatial constraints, temporal constraints, objective function, parallel distributed guided genetic algorithms

Citation Computer Science 21(1) 2020: 3–24

1. Introduction

A multi-robot system (MRS) is a set of robots that are designed to communicate and cooperate with each other in order to achieve some common goals [54]. In the last decades, MRSs have been used to solve many real-world problems such as smart security [31], the search and rescue of victims [37], environmental monitoring [17, 35, 46], and health-care [45]. In an MRS, we usually face some challenging problems like task allocation, coalition formation, object detection and tracking, communication relay, and self-organization [23]. In this paper, we deal with the task allocation problem.

The Multi-Robot Task Allocation problem (MRTA) is informally defined as follows: "given two sets of robots and tasks, the purpose is to allocate tasks to robots while optimizing some criteria (i.e., objective function) under several constraints" [30, 36, 49]. The problem is known to be $\mathcal{NP}$-hard; solving it in an optimal way is a great challenge, especially when heterogeneous robots, complex tasks, and dynamic environments should be considered [41]. In order to understand the MRTA problem, the following sections will give some useful definitions.

1.1. Basic definitions

Definition 1.1 *A robot is an autonomous entity that acts in an environment and is capable of performing some actions* [41]. *If an MRTA problem is taken into account, a robot is then typically modeled as a material point; i.e., the physical layer is omitted.*

Definition 1.2 *A robot group is a set of robots working together to achieve a common goal. If a given group is dynamic, then it is called a "coalition"; i.e., formed to perform a task and dissolved just after its accomplishment* [43].

Definition 1.3 *A task is an action to be performed by one or several robots [4, 14].*

Definition 1.4 *If a task is considered, a "time window" is an interval in which the lower and upper values are "the earliest start time" and "the latest finish time," respectively. If the earliest start time is not provided, then the latest finish time is called the "deadline." A time window is closed if both times are given [41].*

Definition 1.5 *Synchronization constraints specify temporal restrictions on the tasks [41]; e.g., "tasks t_1 and t_2 must start at the same time."*

Definition 1.6 *Precedence constraints specify relationships between tasks [41]; e.g., "task t_2 should start after task t_1 is finished."*

Definition 1.7 *A schedule is a table in which each task has a time window [40]. In some cases, each robot has its own schedule; but in other cases, all robots share the same schedule.*

Definition 1.8 *If a schedule is taken into consideration, a makespan is the difference between the finishing time and starting time of its last and first tasks [41].*

Definition 1.9 *Given robot r and task t, if r is capable of performing t, then one can define application $u(r, t)$, which is called a "utility" of r for t [26].*

1.2. Temporal models

Time can be modeled as time points (e.g., 4 pm) or intervals (e.g., [1–5pm]). In practice, time representation as intervals is frequently adopted. Intervals might be used to express temporal constraints on tasks (as can be seen in Figure 1) [3]. Otherwise, temporal constraints can be modeled as graphs called "Simple Time Networks" [15], where nodes represent time points and weighted arcs express temporal constraints.

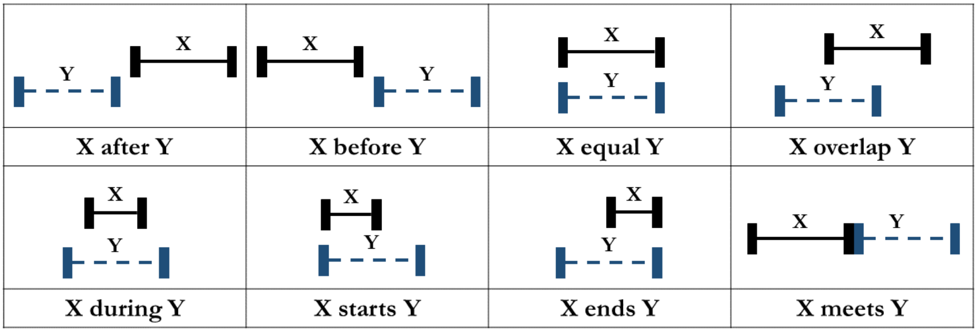

Figure 1. Different types of temporal constraints between tasks X and Y

1.3. Taxonomies of MRTA problems

There are three taxonomies for the categorization of MRTA problems. For simplicity, we call them "Taxonomy 1," "Taxonomy 2," and "Taxonomy 3," respectively, according to their chronological order of appearance.

1.3.1. Taxonomy 1

In 2014, Gerkey and Matarić [19] proposed an elegant taxonomy for the categorization of MRTA problems. It considers the characteristics of robots, tasks, and assignments as follows:

- Single-task robots (ST) vs. multi-task robots (MT):
  1. ST: each robot can only do one task at a time.
  2. MT: some robots can simultaneously do several tasks.
- Single-robot tasks (SR) vs. multi-robot tasks (MR):
  1. SR: each task requires exactly one robot for its accomplishment.
  2. MR: some tasks require the cooperation of several robots for their accomplishment.
- Instantaneous assignments (IA) vs. time-extended assignments (TA):
  1. IA: tasks are allocated to robots considering only current information (i.e., no temporal model is available).
  2. TA: tasks are allocated to robots considering both current and future information (i.e., a temporal model is used).

1.3.2. Taxonomy 2

In 2013, Ayorkor Korsah and his co-authors [26] improved the taxonomy of Gerkey and Matarić by considering dependencies between robots and tasks as follows (it concerns the estimation of utility values):

- No dependencies (ND): the utility of robot r for task t depends on r and t.
- In-schedule dependencies (ID): the utility of robot r for task t depends on the schedule of r.
- Cross-schedule dependencies (XD): the utility of robot r for task t depends on the schedules of all robots in the system; "however, schedules are static."
- Complex dependencies (CD): the utility of robot r for task t depends on the schedules of all robots in the system; "however, schedules are dynamic."

1.3.3. Taxonomy 3

In 2017, Nunes and his co-authors [41] extended the taxonomy of Gerkey and Matarić by developing the "time-extended assignments (TA)" axis in order to include temporal and ordering constraints. This latter now considers two sub-axes as follows:

- TA: TW: temporal constraints are considered and expressed in the form of "time windows."
- TA: SP: ordering constraints are considered and expressed in the form of "synchronization and precedence constraints."

1.4. Contribution and paper organization

We propose a distributed solution for solving strongly constrained MRTA problems; it considers energetic, spatial, and temporal constraints on the robots and tasks. We have two main contributions. First, we use energetic constraints in MRTA problems; i.e., the robots' and tasks' energetic consumptions are not omitted. Actually, previous works have not really addressed this aspect – they were mainly axed on temporal and spatial constraints. Only one paper has dealt with energetic constraints [53], but in a superficial way: the quantities of the consumed energies are given and supposed to be constant. In our work, these quantities are dynamically computed using established physics laws. This constraint is expressed as follows: "each robot has a gauge of energy." Second, we propose some objective functions and modify the MRTA mathematical formulation of [41] by adding two equations expressing energetic constraints.

The remainder of the paper is organized as follows. Section 2 gives an overview of the literature on MRTA problems. Section 3 explains the proposed solution. Section 4 demonstrates the simulations and obtained results as well as their discussions. Finally, a conclusion and some perspectives are given in Section 5.

2. Related work

The MRTA problem shares some common points with the K-traveling repairmen problem (K-TRP) [18] and M-traveling salesmen problem (M-TSP) [7], which are variants of the traveling salesman problem (TSP). The aim of K-TRP is to find tours between the repairmen and customers that minimize the average waiting times for the customers. The aim of M-TSP is to determine tours between salesmen and cities that minimize the average distance that all salesmen should travel; besides, each city should be visited one time. The MRTA problem considered in this paper deals with energetic constraints. In the literature, there are several approaches that solve the TSP and its variants [22, 34, 51, 52].

In combinatorial optimization, MRTA problems are known to be strongly $\mathcal{NP}$-hard [19, 26]. The numbers of tasks and robots are two crucial features in this kind of problem. It is computationally very expensive to consider all possible combinations of tasks and robots in order to find an optimal solution. Hence, approaches based on metaheuristics are used to speed up the task-allocation process and maintain their efficiency and scalability [8, 16, 32, 34, 50].

Centralized approaches dealing with MRTA problems use a central robot that communicates with the other robots and computes optimal allocations between the robots and the tasks. Thus, the objective function is easily optimized as long as all required information is available [9, 20]. Centralized approaches have several advantages and disadvantages. Among their advantages: i) the objective function is easily optimized; ii) the implementation is simple; and iii) the rates of communication are greatly reduced. Among their drawbacks: i) handling real-world and large-scale scenarios is difficult [56]; ii) maintaining a permanent connection between the central robot and other robots is quite hard; (iii) limited range of communications between the central robot and other robots; iv) considerable load of calculations on the central robot; and v) if the central robot fails, then the system will as well. Distributed approaches overcome these disadvantages.

In distributed approaches, all robots execute the same allocation algorithm individually and simultaneously. Usually, distributed methods adopt consensus steps in order to ensure the convergence to a coherent allocation regardless of the used network topology [48]. Consensus steps use additional computations, which can lead robots to take a long time to converge to a coherent allocation – i.e., slowing the convergence of these approaches [2]. Auction-based algorithms [5, 29, 42, 53] are typically distributed methods that have been adopted to solve MRTA problems. These algorithms are efficient and produce suboptimal solutions [19]. In auction-based methods, robots place bids on tasks based on the available information; the tasks are then assigned to robots with the highest bids. The auctioneer (the central robot) may be one of the system's robots or any other central system [27, 44]. In MRTA problems, there is usually a synergy between the tasks and the robots [58]; e.g., "we consider a set of tasks and a robot. If the total cost of simultaneously allocating these tasks to the robot is less than the sum of the individual allocation costs, then we have a positive synergy; conversely, we have a negative synergy." We have three types of auction-based methods. Single-round combinatorial auctions [55] put the tasks into groups and then calculate the bids of each robot. These auctions can produce near-optimal allocations; however, their time complexity increases exponentially with the number of tasks. In sequential single-item auctions [38], only unassigned tasks are allocated during each round. This process is repeated until all of the tasks are assigned. In sequential simultaneous auctions [20], only one task is allocated to a robot at a time. A major limitation of auction-based algorithms is the used network topology, as robots should communicate with the auctioneer.

Market-based approaches [16] have been successfully applied to efficiently solve many MRTA problems and find near-optimal solutions in a distributed manner. Groups of robots cyclically trade tasks to minimize their costs. A cost is considered when a robot visits a task location; this might be its energetic consumption, distance traveled, or time to reach a target [28]. Auctions are commonly used in market-based approaches to allocate tasks to robots [38]. The process is composed of several bidding rounds in which the robots place bids on tasks. A robot that has placed a bid lower than any other robot wins and is allocated to the considered task. The advantage of using market-based approaches is that, when local costs are minimized, global costs are minimized as well [16].

Choi, Brunet, and How [10] propose an algorithm called the Consensus-Based Bundle Algorithm (CBBA), which combines two approaches (consensus and auctions) in order to merge their advantages. In CBBA, the winning bid values are determined using the consensus process. Zhao, Meng, and Chung [58] describe an algorithm called PI, which is an improved extension of CBBA. This algorithm optimizes the objective function of the considered problem and outperforms most of the developed algorithms based on CBBA [6, 11, 13, 21, 24]. Its main idea is this: "a local contribution value is calculated when a task is assigned to a robot, then the overall cost could be decreased if these values satisfy certain constraints." Both CBBA and PI algorithms are robust with respect to the network topology used. However, they suffer from sub-optimality, as greedy-based strategies are used in the task-inclusion phase, and they cannot handle the dynamic rescheduling. Also, they are inefficient when communications are unstable.

There are many papers that have proposed different solutions to MRTA problems [39, 57]. Agarwal, Kumar, and Vig [1] propose an application for solving the problem of multi-objective coalition formation using the Pareto Archived Evolution Strategy algorithm. This method is centralized and does not take dynamic scenarios into account. Luo, Qin, and Lim [34] describe a dynamic rescheduling module for the PI algorithm in order to permit real-time dynamic online rescheduling. This work is quite inefficient, especially when new information arrives frequently. Zitouni and Maamri [60] propose a solution that combines two algorithms: firefly algorithm and Powerset. Zitouni and Maamri [59] describe a solution that combines quantum genetic algorithms and reinforcement learning. Zitouni, Maamri, and Harous [61] present a solution that combines three metaheuristics: firefly algorithms, quantum genetic algorithms, and an artificial bee colony. These three approaches are inefficient when communications are unstable. Lozenguez and his co-authors [33] combine clustering and Markov Decision Processes to allocate a set of exploration tasks to a group of mobile robots. This method is centralized and does not take dynamic scenarios into account.

3. Proposed solution

We propose an efficient solution for solving heavily constrained MRTA problems. Section 3.1 presents a description of the problem. Section 3.2 shows the modified mathematical formulation where energetic constraints are added. Section 3.3 describes the proposed objective functions. Finally, the allocation methodology is explained in Section 3.4.

3.1. Problem description

We deal with MRTA problems where each robot can perform only one task at a time and some tasks require the cooperation of several robots for their accomplishment. Also, we consider time-extended assignments where temporal constraints are expressed in the form of time windows. The found allocations respect the energetic, spatial, and temporal constraints on the robots and tasks as well as optimize a given objective function; e.g., minimizing the traveled distances.

We assume that we have a set of n robots $B = \{b_1, \ldots, b_n\}$ and a set of m tasks $T = \{t_1, \ldots, t_m\}$. When a task is available for allocation, it is announced to the robots. Next, the robots cooperate to compute an allocation for this task. Finally, the considered task is allocated to the appropriate robots. Table 1 summarizes the used notations, variables, and symbols that we will use to explain our solution.

Table 1
Notations used in paper

Notation	Meaning
v_b	Velocity of robot b
m_b	Mass of robot b
(x_b, y_b, z_b)	Coordinates of robot b
a_b	Altitude of robot b relative to ground
E_K^b	Kinetic energy of robot b
E_P^b	Potential energy of robot b
U^b	Battery voltage of robot b
A^b	Battery capacity of robot b
η	Peukert's exponent of robot batteries
R^b	Battery hour-rating of robot b
G^b	Gauge energy of robot battery
(x_t, y_t, z_t)	Coordinates of task t
DUR_t	Duration of task t

Table 1 (cont.)

Notation	Meaning
ES_t	Earliest start time of task t
S_t	Estimated start time of task t
LS_t	Latest start time of task t
EF_t	Earliest finish time of task t
F_t	Estimated finish time of task t
LF_t	Latest finish time of task t
$\|A\|$	Cardinality of set A
$TT^b_{\langle t_1,t_2\rangle}$	Time taken by robot b to move from task t_1 to task t_2
$DE^b_{\langle t_1,t_2\rangle}$	Energy consumed by robot b to move from task t_1 to task t_2
$EE^b_{\langle R,t\rangle}$	Energy consumed by robot b to perform task t. Symbol R expresses relationship between b and t

where:

$$a_b = \sqrt{(z_b)^2} \tag{1}$$

$$E^b_K = 0.5 \times m_b \times (v_b)^2 \tag{2}$$

$$E^b_P = 9.81 \times m_b \times a_b \tag{3}$$

$$TT^b_{\langle t_1,t_2\rangle} = \frac{\sqrt{(x_{t_2} - x_{t_1})^2 + (y_{t_2} - y_{t_1})^2 + (z_{t_2} - z_{t_1})^2}}{v_b} \tag{4}$$

3.2. Mathematical formulation of MRTA problems

If an MRTA problem is taken into account and we wish to allocate task $t \in T$ to robot $b \in B$, then it is quite natural to consider an allocation relationship between them. Intuitively, this allocation relationship is expressed as follows: "b is capable of doing t." Actually, this relationship is directly linked to the chosen problem. In our case, we chose to use sensors as the relationship between tasks and robots; i.e., the tasks need sensors, and the robots offer them.

We suppose that we have a set of k sensors $\Omega = \{\omega_1, \ldots, \omega_k\}$. Set $\wp(B) = \{\{b_1\}, \{b_2\}, \ldots, \{b_1, \ldots, b_n\}\}$ represents all robot groups that can be formed from B. Set $\Omega^b \subseteq \Omega$ represents the sensors of robot b. Set $\Omega^t \subseteq \Omega$ represents the sensors needed by task t. In addition, we define indicator $o^C_{\langle b,\omega\rangle} \in \{0, 1\}$, which means "if robot $b \in C$ offers sensor $\omega \in \Omega^b$ to group $C \in \wp(B) \setminus \emptyset$, then it takes a value of 1; otherwise, it takes a value of 0." Set $\Omega^b_C = \{\omega | o^C_{\langle b,\omega\rangle} = 1\}$ represents all sensors that $b \in C$ offers to $C \in \wp(B) \setminus \emptyset$ ($\Omega^b_C \subseteq \Omega^b$). Therefore, we give the following corollary that expresses the allocation relationship between a task and a robot group.

Corollary 3.1 *Given task $t \in T$ and robot group $C \in \wp(B) \setminus \emptyset$, task t can be allocated to robot group C if the following two conditions (expressed by Equations* (5) *and* (6)*) are simultaneously satisfied.*

$$\bigcap \Omega_C^b = \emptyset \tag{5}$$

$$\bigcup \Omega_C^b = \Omega^t \tag{6}$$

Equation (5) means that a given sensor could not be offered by two distinct robots from the same group to the same task. Equation (6) means that all of the sensors that a task needs should be offered by robots of the same group. Finally, we give the modified mathematical formulation of the MRTA problems [41]. Equations (5) and (6) are our contributions and express the energetic constraints.

We optimize a considered objective function $f(.)$ subject to Equations (7) through (17):

- Equation (7): "each task t is allocated to one robot group C" at most.

$$\forall t \in T : \sum_{C \in \wp(B) \setminus \emptyset} s_t^C \leq 1 \tag{7}$$

- Equation (8): "if a task t is allocated to a robot group C, then all required sensors must be available."

$$\forall t \in T \forall C \in \wp(B) \setminus \emptyset : \sum_{b \in C} \sum_{\omega \in \Omega^b} o_{\langle b, \omega \rangle}^C = |\Omega^t| \times s_t^C \tag{8}$$

- Equation (9): "energy gauge of robot b is valid."

$$\forall b \in B : 0 \leq G^b \leq 100 \tag{9}$$

- Equation (10): "starting time of task t is valid."

$$\forall t \in T : ES_t \leq S_t \leq LS_t \tag{10}$$

- Equation (11): "finishing time of task t is valid."

$$\forall t \in T : EF_t \leq F_t \leq LF_t \tag{11}$$

- Equation (12): "duration of task t is long enough."

$$\forall t \in T : (F_t - S_t) \geq DUR_t \tag{12}$$

- Equation (13): "moving time between two consecutive tasks is long enough."

$$\forall t, t' \in T \forall b \in B : S_t + DUR_t + TT_{\langle t, t' \rangle}^b - M \times (1 - d_{\langle t, t' \rangle}^b) \leq S_{t'} \tag{13}$$

- Equation (14): “robot energy needed to reach a given task and perform it is enough. Symbol N is a threshold.”

$$\forall t, t' \in T \forall b \in B : G^b - DE^b_{\langle t,t' \rangle} - \sum_{\omega \in \Omega^b} (DUR_{t'} \times o^C_{\langle b,\omega \rangle} \times EE^b_{\langle \omega,t' \rangle}) - M' \times (1 - d^b_{\langle t,t' \rangle}) > N \tag{14}$$

- Equation (15): “indicates if robot b offers sensor ω to robot group C.”

$$\forall b \in B \forall C \in \wp(B) \setminus \emptyset : o^C_{\langle b,\omega \rangle} \in \{0, 1\} \tag{15}$$

- Equation (16): “indicates if task t is allocated to robot group C.”

$$\forall t \in T \forall C \in \wp(B) \setminus \emptyset : s^C_t \in \{0, 1\} \tag{16}$$

- Equation (17): “indicates if robot b does task t then task t'.”

$$\forall t, t' \in T \forall b \in B : d^b_{\langle t,t' \rangle} \in \{0, 1\} \tag{17}$$

3.3. Proposed objective functions

First, we present the formal definition of the nine applications that we will use to define the proposed objective functions.

- The application defined by Equation (18) assigns a positive value to each robot sensor, which represents its cost.

$$\begin{array}{rccl} cost: & (B, \Omega) & \to & \mathbb{R}^+ \\ & (b, \omega) & \mapsto & \begin{cases} cost(b, \omega), & \text{if } \omega \in \Omega^b \\ 0, & \text{otherwise} \end{cases} \end{array} \tag{18}$$

- The application defined by Equation (19) assigns a positive value to each robot sensor, which represents its working current.

$$\begin{array}{rccl} workingcurrent: & (B, \Omega) & \to & \mathbb{R}^+ \\ & (b, \omega) & \mapsto & \begin{cases} workingcurrent(b, \omega), & \text{if } \omega \in \Omega^b \\ 0, & \text{otherwise} \end{cases} \end{array} \tag{19}$$

- The application defined by Equation (20) assigns a positive value to each sensor needed by a task, which represents its reward.

$$\begin{array}{rccl} reward: & (T, \Omega) & \to & \mathbb{R}^+ \\ & (t, \omega) & \mapsto & \begin{cases} reward(t, \omega), & \text{if } \omega \in \Omega^t \\ 0, & \text{otherwise} \end{cases} \end{array} \tag{20}$$

- The application defined by Equation (21) assigns a positive value to each sensor needed by a task, which represents its working duration.

$$\begin{array}{llll} workingduration: & (T,\Omega) & \rightarrow & \mathbb{R}^+ \\ & (t,\omega) & \mapsto & \begin{cases} workingduration(t,\omega), & \text{if } \omega \in \Omega^t \\ 0, & \text{otherwise} \end{cases} \end{array} \tag{21}$$

- The application defined by Equation (22) calculates the Euclidean distance separating two tasks.

$$\begin{array}{llll} distance: & (T,T) & \rightarrow & \mathbb{R}^+ \\ & (t,t') & \mapsto & \sqrt{(x_t - x_{t'})^2 + (y_t - y_{t'})^2 + (z_t - z_{t'})^2} \end{array} \tag{22}$$

- The application defined by Equation (23) calculates the gain robot b will get if it is allocated to task t', knowing that b is already allocated to task t (γ is a regularization parameter).

$$\begin{array}{llll} gain: & (B,\wp(B)\setminus\emptyset,T,T) & \rightarrow & \mathbb{R}^+ \\ & (b,C,t,t') & \mapsto & [\sum_{\omega\in\Omega_C^b} reward(t',\omega)] \times e^{-\gamma\times(TT^b_{\langle t,t'\rangle})^2} \end{array} \tag{23}$$

- The application defined by Equation (24) calculates the percentage of consumed energy that robot b spends when it moves from task t to task t'.

$$\begin{array}{llll} moving: & (B,T,T) & \rightarrow & \mathbb{R}^+ \\ & (b,t,t') & \mapsto & 100 \times \frac{TT^b_{\langle t,t'\rangle}\times(\frac{E_K^b+E_P^b}{U^b})^\eta}{R^b\times(\frac{A^b}{R^b})^\eta} \end{array} \tag{24}$$

- The application defined by Equation (25) calculates the percentage of consumed energy that robot b spends when it uses its sensors to achieve task t.

$$\begin{array}{llll} sensor: & (B,T,\Omega) & \rightarrow & \mathbb{R}^+ \\ & (b,t,\omega) & \mapsto & 100 \times \frac{DUR_t\times workingduration(t,\omega)\times(workingcurrent(b,\omega))^\eta}{R^b\times(\frac{A^b}{R^b})^\eta} \end{array} \tag{25}$$

- The application defined by Equation (26) calculates the rate of the offered sensors of robot b to group C.

$$\begin{array}{llll} rate: & (B,\wp(B)\setminus\emptyset) & \rightarrow & \mathbb{R}^+ \\ & (b,C) & \mapsto & \frac{|\Omega_C^b|}{|\Omega^b|} \end{array} \tag{26}$$

Now, we give the formal definitions of the proposed objective functions. Coefficients α and β are used to accentuate the equation terms.

- The objective function defined by Equation (27) is used to minimize the costs of the tasks.

$$\begin{array}{lll} f_1: & (B, \wp(B)\setminus\emptyset, T, T) & \rightarrow \mathbb{R}^+ \\ & (b, C, t, t') & \mapsto (|C|)^{\alpha} \times \sum_{b\in C}[(\frac{gain(b,C,t,t')}{rate(b,C)})^{\beta} \sum_{\omega\in\Omega_C^b} cost(b,\omega)] \end{array} \quad (27)$$

- The objective function defined by Equation (28) is used to maximize the rewards of the robots.

$$\begin{array}{lll} f_2: & (B, \wp(B)\setminus\emptyset, T, T) & \rightarrow \mathbb{R}^+ \\ & (b, C, t, t') & \mapsto (\frac{1}{|C|})^{\alpha} \times \sum_{b\in C}[(\frac{rate(b,C)}{gain(b,C,t,t')})^{\beta} \sum_{\omega\in\Omega_C^{t'}} reward(t',\omega)] \end{array} \quad (28)$$

- The objective function defined by Equation (29) is used to maximize the benefits of the robots.

$$\begin{array}{lll} f_3: & (B, \wp(B)\setminus\emptyset, T, T) & \rightarrow \mathbb{R}^+ \\ & (b, C, t, t') & \mapsto max(f_2(b,C,t,t') - f_1(b,C,t,t'), 0) \end{array} \quad (29)$$

- The objective function defined by Equation (30) is used to minimize the traveled distances of the robots.

$$\begin{array}{lll} f_4: & (B, \wp(B)\setminus\emptyset, T, T) & \rightarrow \mathbb{R}^+ \\ & (b, C, t, t') & \mapsto (|C|)^{\alpha} \times \sum_{b\in C}[(\frac{gain(b,C,t,t')}{rate(b,C)})^{\beta} \times distance(t,t')] \end{array} \quad (30)$$

- The objective function defined by Equation (31) is used to minimize the travel times of the robots.

$$\begin{array}{lll} f_5: & (B, \wp(B)\setminus\emptyset, T, T) & \rightarrow \mathbb{R}^+ \\ & (b, C, t, t') & \mapsto (|C|)^{\alpha} \times \sum_{b\in C}[(\frac{gain(b,C,t,t')}{rate(b,C)})^{\beta} \times TT^b_{\langle t,t'\rangle}] \end{array} \quad (31)$$

- The objective function defined by Equation (32) is used to minimize the consumed energies of the robots.

$$\begin{array}{lll} f_6: & (B, \wp(B)\setminus\emptyset, T, T) & \rightarrow \mathbb{R}^+ \\ & (b, C, t, t') & \mapsto \begin{array}{l} (|C|)^{\alpha} \times \sum_{b\in C}[(\frac{gain(b,C,t,t')}{rate(b,C)})^{\beta} \times moving(b,t,t')] \\ +(|C|)^{\alpha} \times \sum_{b\in C}[(\frac{gain(b,C,t,t')}{rate(b,C)})^{\beta} \sum_{\omega\in\Omega_C^b} sensor(b,t',\omega)] \end{array} \end{array} \quad (32)$$

3.4. Allocation methodology

We suppose that we have an environment that contains some tasks to be allocated. Also, some robots are distributed in the environment and should cooperate to achieve the considered tasks. Each robot b has a list of neighbors Γ^b that contains close robots (i.e., we say that a robot b is close to another robot b' if the distance between them is less than a given threshold). Therefore, the robots' behaviors are summarized as follows.

1. The robots move randomly in an environment and look for tasks to accomplish. "The way of how tasks are discovered" is abstracted, as this is not the focus of this research paper.
2. If a robot discovers a task, then it should determine the information about it; i.e., the position, time window, needed sensors, etc. Next, message "**M1**" is broadcast to all robots to inform them about the availability of a task that needs to be allocated.
3. When message "**M1**" is received, robot b should make sure that it is able to perform considered task t. Therefore, robot b i) verifies whether it has the sensors that are required by task t, ii) confirms whether it can reach the position of task t before its latest start time, and iii) examines whether its energy is enough to move to task t and achieve it. In summary, if conditions i), ii), and iii) are simulataneously satisfied, then we say that robot b is able to perform task t. Eventually, if robot b is able to perform task t, then message "**M2**" is broadcast to its neighbors; otherwise, it broadcasts message "**M3**" to them.
4. Each robot counts the number of received "**M2**" messages and forms corresponding group of robots Γ'^b ($\Gamma'^b \subseteq \Gamma^b$). Group Γ'^b is composed of its neighbors that are able to perform considered task t. It is worth pointing out that each robot b belongs to its list of neighbors Γ^b.
5. When the Γ'^b list is built, each robot b executes a genetic algorithm for task t as follows:

   Encoding scheme and initial population: an individual is composed of one chromosome. Chromosome Φ represents a group of robots for the considered task, and its length is $|\Phi|$; therefore, each gene corresponds to a sensor. Besides, the gene values are strings: if the value of a gene is $\perp$, then it means that the corresponding sensor is not required by the considered task; otherwise, we should find a robot name. This means that this robot offers the corresponding sensor to the considered task. For example, if we consider $\Omega = \{\omega_1, \omega_2, \omega_3, \omega_4, \omega_5\}$, $\Omega^t = \{\omega_2, \omega_3, \omega_5\}$, $\Omega^{b_1} = \{\omega_1, \omega_2\}$, $\Omega^{b_2} = \{\omega_2, \omega_3, \omega_4\}$ and $\Omega^{b_3} = \{\omega_3, \omega_4, \omega_5\}$, then one chromosome could be encoded as $\Phi =[\perp, b_1, b_2, \perp, b_3]$; i.e., sensor ω_1 is not needed by task t, sensor ω_2 is offered by robot b_1, and so on. The initial population of each robot b is composed of N individuals created from the Γ'^b list; i.e., $(b' \in \Phi) \Rightarrow (b' \in \Gamma'^b)$.

Fitness function: fitness function $F(\Phi)$ assigns a numerical value to each chromosome that measures its quality. This value is used to sort and compare the chromosomes. We use Equation (33) to calculate the fitness value of a chromosome.

$$F(\Phi_p) = \frac{f_i(\Phi_p)}{\sum_{q=1}^{N} f_i(\Phi_q)}, \tag{33}$$

where Φ_p is the p^{th} individual, N is the size of the population, and f_i is one of the objective functions defined by Equations (27) through (32). In order to avoid the premature convergence and maintain a fairly constant selective pressure, each objective value is then scaled using Equation (34) "sigma truncation scaling [12]" as follows:

$$f_i(\Phi_p) = \frac{f_i(\Phi_p) - \frac{\sum_{q=1}^{N} f_i(\Phi_q)}{N} - \sigma}{\sigma}, \tag{34}$$

where terms $\frac{\sum_{q=1}^{N} f_i(\Phi_q)}{N}$ and σ are the average and standard deviation, respectively.

Selection: each robot uses tournament selection to create a sub-population, which is called the mating pool. To do this, i) k individuals are randomly selected from the current population (e.g., $k \leq \frac{N}{4}$) and ii) the individual with the best fitness value is taken and inserted into the mating pool. The remaining $(k-1)$ individuals are returned to the current population. This process is repeated until the size of the mating pool reaches a given size N'; e.g., $N' \leq \frac{N}{2}$.

With this method, each robot ensures that bad individuals are not selected and the best ones will not dominate. Actually, the value of k is directly related to the selective pressure; i.e., a reasonable value would ensure a near--optimal solution [47].

Crossover: once the mating pool is created, each robot applies the crossover operator. Its goal is to create individuals for the next population. Initially, the next population contains individuals from the mating pool; the rest of the individuals will be created using the crossover operator.

To do this, each robot b chooses two random robots b' and b'' from its list of neighbors Γ^b and send them message "**M4**" to ask for an individual. When message "**M4**" is received, robots b' and b'' select a random individual from their respective mating pools and answer robot b. It is worth pointing out that the following cases are allowed: $b = b'$, $b = b''$, or $b' = b''$. When the two individuals are received, each robot applies a uniform crossover [47] on them to produce a new one. The principle of uniform crossover is shown in Figure 2.

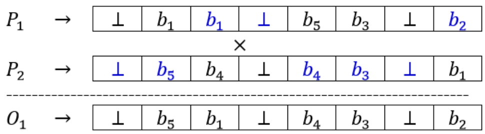

Figure 2. Principle of uniform crossover operator

This process is repeated until the size of the next population becomes the same as the current population. Henceforth, the next population becomes the current one.

Mutation: the random initialization of the first population could sometimes limit the good exploitation of the search space. We can avoid this problem by using the mutation operator.
The mutation is applied as follows: each robot b chooses i) a random individual from its current population and ii) a random robot b' from its list of neighbors Γ^b. Then, robot b sends message "**M5**" to robot b'. When message "**M5**" is received, robot b' also selects a random individual from its current population and swaps it with robot b.

6. When all robots finish the execution of their genetic algorithm, each determines its local best allocation and sends a reply message to the robot that sent message "**M1**."
7. When all local best allocations are received, the robot having initiated the allocation request determines the best global allocation and notifies all robots about its decision.
8. Finally, the robots concerned with the global best allocation should move to reach the position of the considered task.
9. Steps 1 through 9 are repeated for each discovered task.

4. Simulation and result discussion

We evaluate the performance of our solution by comparing it to two state-of-the-art solutions [53]. Wei, Hindriks, and Jonker [53] propose two acceptable solutions to the MRTA problem in a foraging field where some robots' groups are requested to i) search for targets (i.e., tasks) in an environment and ii) retrieve them and take back to a home base.

The first solution of [53] presents an auction-based approach extended from sequential-single-item (SSI) auctions. It has been shown that SSIs can provide a good compromise between computational complexity and solution quality if the set of tasks is initially known [25, 28]. The first solution is abbreviated "AUCTION." The second solution in [53] provides a prediction approach where each robot should predict the decisions of the other robots about task allocation without using auctions or negotiations. The second solution is abbreviated "PREDICTION." Finally, the solution proposed in this paper is abbreviated "DistMRTA."

We carry out an experimental study (simulation) to compare the performance of "AUCTION," "PREDICTION," and "DistMRTA" in terms of completion time. The following parameters are taken into account: i) size of environment; ii) size of robot groups; and iii) initial robot positions. Table 2 shows the different experimental configurations used to perform the comparison.

Table 2

Experimental configurations used for comparison.

Solutions	Size of the environment	Size of robots' groups	Initial robots' positions
AUCTION,PREDICTION,DistMRTA	SMALL,LARGE	1,5,10	CLOSE,DISPERSAL

We use two environments: "SMALL" or "LARGE" (i.e., size of the environment). All solutions were tested with one, five, and ten robots. We use two alternatives for robot deployment: all robots are initially in the same place (i.e., "CLOSE") or are dispersed in the "DISPERSAL" environment. The goal of the simulations is to accomplish ten tasks that are in the environment; their locations are set randomly.

To compare "AUCTION," "PREDICTION," and "DistMRTA" performance, we use completion time as an evaluation criterion. Each configuration ⟨solutions, size of the environment, size of robot groups, Initial robot positions, number of tasks ⟩ was run 50 times in order to reduce the variance and filter noise effects in our experiments. Figure 3 shows the results of the comparative study.

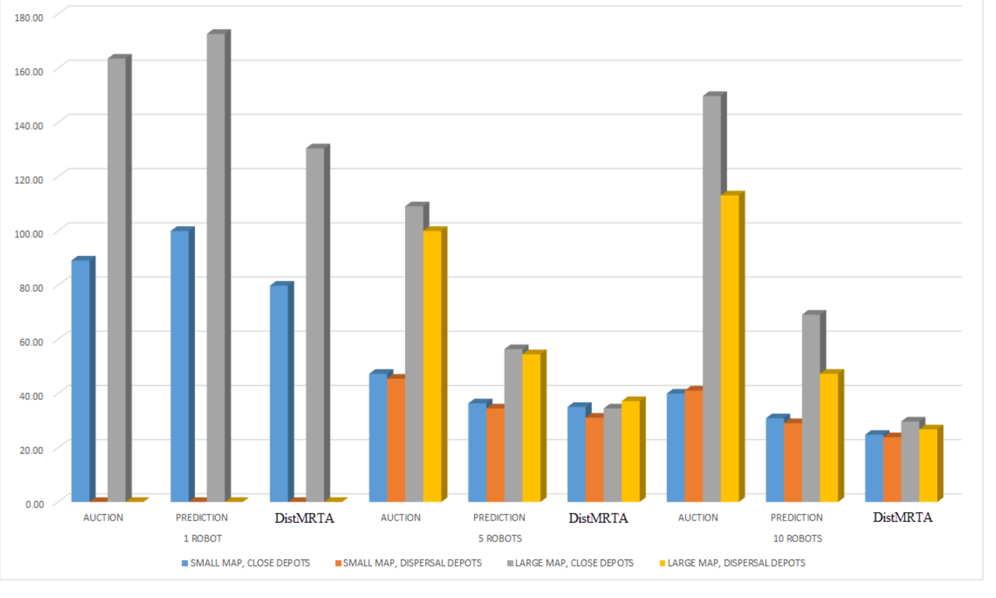

Figure 3. Completion time for comparative study

In the first case (i.e., one robot), we observe that the completion time of all tasks is almost the same because no workload is shared between robots. It is worth mentioning here that the execution time of the robot programs might also influence this criterion. However, it is clear that "DistMRTA" gives the best performance when compared to "AUCTION" and "PREDICTION."

In the remaining cases (i.e., five and ten robots), we observe that "DistMRTA" gives the best completion time, followed by "PREDICTION" and finally by "AUCTION." This is clearly visible when the size of the environment is "LARGE": the completion time of "PREDICTION" and "AUCTION" are nearly twice and thrice as long, respectively. These trends are explained as follows:

1. In "AUCTION," the auctioneer should consider all of the robots' bids in a round then determine the winners. Obviously, the completion time increases with the number of robots. However, the completion time constantly decreases in "DistMRTA" because the number of bids diminishes; e.g., a robot performing a task cannot submit a bid on a new one because its energy is not enough to move to a new task.
2. In "PREDICTION," a robot is prohibited from submitting a bid on a new task if it is currently allocated to another one. Actually, this is not optimal, especially in the case when the tasks are critical. However, a robot can submit a bid on a new task in "DistMRTA" even when it is currently carrying out another one; e.g., a robot performing a task that will finish soon can submit a bid on a new one.

Finally, we find out that the initial robot deployment is directly related to the size of the environment and number of robots (and consequently to the completion time of all tasks). As a conclusion, the initial robot deployment is very important in MRTA problems.

5. Conclusion and perspectives

We dealt with MRTA problems that considered spatial, temporal, and energetic constraints. A distributed solution based on parallel distributed guided genetic algorithms is proposed for allocating tasks to some group of robots. Six objective functions that express spatial temporal and energetic constraints have been proposed and extensively discussed. A well-known mathematical formulation of MRTA problems [41] is modified, and two equations that express energetic constraints have been proposed and explained. Finally, we compared our solution to the two state-of-the-art solutions described in [53]. The simulation result of our solution outperforms this pair of approaches in terms of completion time. In the case when we utilized ten robots, our solution improved the completion time of the two methods by 50 and 67%, respectively. In the future, we plan to use real robots to assess the performance of our solution.

References

[1] Agarwal M., Kumar N., Vig L.: Non-additive multi-objective robot coalition formation, *Expert Systems with Applications*, vol. 41(8), pp. 3736–3747, 2014.

[2] Alighanbari M., How J.P.: Decentralized task assignment for unmanned aerial vehicles. In: *Proceedings of the 44th IEEE Conference on Decision and Control*, pp. 5668–5673, IEEE, 2005.

[3] Allen J.F.: Maintaining knowledge about temporal intervals. In: *Readings in qualitative reasoning about physical systems*, pp. 361–372, Elsevier, 1990. https://doi.org/10.1016/B978-1-4832-1447-4.50033-X.

[4] Balas E., Simonetti N., Vazacopoulos A.: Job shop scheduling with setup times, deadlines and precedence constraints, *Journal of Scheduling*, vol. 11(4), pp. 253–262, 2008.

[5] Bertsekas D.P.: The auction algorithm for assignment and other network flow problems: A tutorial, *Interfaces*, vol. 20(4), pp. 133–149, 1990.

[6] Binetti G., Naso D., Turchiano B.: Decentralized task allocation for surveillance systems with critical tasks, *Robotics and Autonomous Systems*, vol. 61(12), pp. 1653–1664, 2013.

[7] Brown E.C., Ragsdale C.T., Carter A.E.: A grouping genetic algorithm for the multiple traveling salesperson problem, *International Journal of Information Technology & Decision Making*, vol. 6(02), pp. 333–347, 2007.

[8] Butt S.E., Cavalier T.M.: A heuristic for the multiple tour maximum collection problem, *Computers & Operations Research*, vol. 21(1), pp. 101–111, 1994.

[9] Chen J., Sun D.: Coalition-based approach to task allocation of multiple robots with resource constraints, *IEEE Transactions on Automation Science and Engineering*, vol. 9(3), pp. 516–528, 2012.

[10] Choi H.L., Brunet L., How J.P.: Consensus-based decentralized auctions for robust task allocation, *IEEE Transactions on Robotics*, vol. 25(4), pp. 912–926, 2009.

[11] Choi H.L., Whitten A.K., How J.P.: Decentralized task allocation for heterogeneous teams with cooperation constraints. In: *Proceedings of the 2010 American Control Conference*, pp. 3057–3062, IEEE, 2010.

[12] Cordón O., Herrera F., Hoffmann F., Luis M.: *Genetic fuzzy systems: evolutionary tuning and learning of fuzzy knowledge bases*, vol. 19, World Scientific, 2001.

[13] Das G.P., McGinnity T.M., Coleman S.A., Behera L.: A fast distributed auction and consensus process using parallel task allocation and execution. In: *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 4716–4721, IEEE, 2011.

[14] Davis R.I., Burns A.: A survey of hard real-time scheduling for multiprocessor systems, *ACM Computing Surveys (CSUR)*, vol. 43(4), p. 35, 2011.

[15] Dechter R., Meiri I., Pearl J.: Temporal constraint networks, *Artificial intelligence*, vol. 49(1–3), pp. 61–95, 1991.

[16] Dias M.B., Zlot R., Kalra N., Stentz A.: Market-based multirobot coordination: A survey and analysis. In: *Proceedings of the IEEE*, vol. 94(7), pp. 1257–1270, 2006.

[17] Espina M.V., Grech R., De Jager D., Remagnino P., Iocchi L., Marchetti L., Nardi D., Monekosso D., Nicolescu M., King C.: Multi-robot teams for environmental monitoring. In: *Innovations in Defence Support Systems–3*, pp. 183–209. Springer, 2011.

[18] Fakcharoenphol J., Harrelson C., Rao S.: The k-traveling repairmen problem, *ACM Transactions on Algorithms (TALG)*, vol. 3(4), p. 40, 2007.

[19] Gerkey B.P., Matarić M.J.: A formal analysis and taxonomy of task allocation in multi-robot systems, *The International Journal of Robotics Research*, vol. 23(9), pp. 939–954, 2004.

[20] Huang L., Ding Y., Zhou M., Jin Y., Hao K.: Multiple-Solution Optimization Strategy for Multirobot Task Allocation, *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2018.

[21] Johnson L., Ponda S., Choi H.L., How J.: Asynchronous decentralized task allocation for dynamic environments. In: *Infotech@ Aerospace 2011*, p. 1441, 2011.

[22] Junjie P., Dingwei W.: An ant colony optimization algorithm for multiple travelling salesman problem. In: *First International Conference on Innovative Computing, Information and Control-Volume I (ICICIC'06)*, vol. 1, pp. 210–213, IEEE, 2006.

[23] Khamis A., Hussein A., Elmogy A.: Multi-robot task allocation: A review of the state-of-the-art. In: *Cooperative Robots and Sensor Networks 2015*, pp. 31–51, Springer, 2015.

[24] Kivelevitch E., Cohen K., Kumar M.: A market-based solution to the multiple traveling salesmen problem, *Journal of Intelligent & Robotic Systems*, vol. 72(1), pp. 21–40, 2013.

[25] Koenig S., Keskinocak P., Tovey C.: Progress on agent coordination with cooperative auctions. In: *Twenty-fourth AAAI Conference on Artificial Intelligence*. 2010.

[26] Korsah G.A., Dias M.B., Stentz A.: A comprehensive taxonomy for multi-robot task allocation, *The International Journal of Robotics Research*, vol. 32(12), pp. 1495–1512, 2013.

[27] Kwasnica A.M., Ledyard J.O., Porter D.P., DeMartini C.: A new and improved design for multi-object iterative auctions, *Handbook of Spectrum Auction Design*, pp. 391–417. Cambridge University Press, 2017.

[28] Lagoudakis M.G., Markakis E., Kempe D., Keskinocak P., Kleywegt A.J., Koenig S., Tovey C.A., Meyerson A., Jain S.: Auction-Based Multi-Robot Routing, *Robotics: Science and Systems*, vol. 5, pp. 343–350. Rome, Italy, 2005.

[29] Lee D.H.: Resource-based task allocation for multi-robot systems, *Robotics and Autonomous Systems*, vol. 103, pp. 151–161, 2018.

[30] Lerman K., Jones C., Galstyan A., Matarić M.J.: Analysis of dynamic task allocation in multi-robot systems, *The International Journal of Robotics Research*, vol. 25(3), pp. 225–241, 2006.

[31] Liao Y.L., Su K.L.: Multi-robot-based intelligent security system, *Artificial Life and Robotics*, vol. 16(2), p. 137, 2011.

[32] Lin S.W., Vincent F.Y.: A simulated annealing heuristic for the team orienteering problem with time windows, *European Journal of Operational Research*, vol. 217(1), pp. 94–107, 2012. https://doi.org/10.1016/j.ejor.2011.08.024

[33] Lozenguez G., Adouane L., Beynier A., Mouaddib A.I., Martinet P.: Map partitioning to approximate an exploration strategy in mobile robotics, *Multiagent and Grid Systems*, vol. 8(3), pp. 275–288, 2012. https://doi.org/10.3233/MGS-2012-0195

[34] Luo Z., Qin H., Lim A.: Branch-and-price-and-cut for the multiple traveling repairman problem with distance constraints, *European Journal of Operational Research*, vol. 234(1), pp. 49–60, 2014.

[35] Marino A., Parker L.E., Antonelli G., Caccavale F.: A decentralized architecture for multi-robot systems based on the null-space-behavioral control with application to multi-robot border patrolling, *Journal of Intelligent & Robotic Systems*, vol. 71(3–4), pp. 423–444, 2013.

[36] Mosteo A.R., Montano L.: A survey of multi-robot task allocation. In: *Instituto de Investigación en Ingenierıa de Aragón, University of Zaragoza, Zaragoza, Spain, Technical Report No. AMI-009-10-TEC*, 2010.

[37] Nagatani K., Okada Y., Tokunaga N., Kiribayashi S., Yoshida K., Ohno K., Takeuchi E., Tadokoro S., Akiyama H., Noda I., et al.: Multirobot exploration for search and rescue missions: A report on map building in RoboCupRescue 2009, *Journal of Field Robotics*, vol. 28(3), pp. 373–387, 2011.

[38] Nanjanath M., Gini M.: Repeated auctions for robust task execution by a robot team, *Robotics and Autonomous Systems*, vol. 58(7), pp. 900–909, 2010.

[39] Nguyen S., Zhang M., Johnston M., Tan K.C.: Automatic Programming via Iterated Local Search for Dynamic Job Shop Scheduling, *IEEE Transactions on Cybernetics*, vol. 45(1), pp. 1–14, 2015.

[40] Nunes E., Gini M.: Multi-robot auctions for allocation of tasks with temporal constraints. In: *Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015.

[41] Nunes E., Manner M., Mitiche H., Gini M.: A taxonomy for task allocation problems with temporal and ordering constraints, *Robotics and Autonomous Systems*, vol. 90, pp. 55–70, 2017.

[42] Oliver G., Guerrero J.: Auction and swarm multi-robot task allocation algorithms in real time scenarios. In: *Multi-Robot Systems, Trends and Development*, IntechOpen, 2011.

[43] Parker L.E., Tang F.: Building multirobot coalitions through automated task solution synthesis. In: *Proceedings of the IEEE*, vol. 94(7), pp. 1289–1305, 2006.

[44] Sariel S., Balch T.: Real time auction based allocation of tasks for multi-robot exploration problem in dynamic environments. In: *Proceedings of the AAAI-05 Workshop on Integrating Planning into Scheduling*, pp. 27–33, AAAI Palo Alto, CA, 2005.

[45] Shiomi M., Kamei K., Kondo T., Miyashita T., Hagita N.: Robotic service coordination for elderly people and caregivers with ubiquitous network robot platform. In: *2013 IEEE Workshop on Advanced Robotics and its Social Impacts*, pp. 57–62, IEEE, 2013.

[46] Shkurti F., Xu A., Meghjani M., Higuera J.C.G., Girdhar Y., Giguere P., Dey B.B., Li J., Kalmbach A., Prahacs C., et al.: Multi-domain monitoring of marine environments using a heterogeneous robot team. In: *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1747–1753, IEEE, 2012.

[47] Siddique N., Adeli H.: *Computational intelligence: synergies of fuzzy logic, neural networks and evolutionary computing*, John Wiley & Sons, 2013.

[48] Tahbaz-Salehi A., Jadbabaie A.: On consensus over random networks. In: *44th Annual Allerton Conference*, 2006.

[49] Tang F., Parker L.E.: A complete methodology for generating multi-robot task solutions using ASyMTRe-D and market-based task allocation. In: *Proceedings 2007 IEEE International Conference on Robotics and Automation*, pp. 3351–3358, IEEE, 2007.

[50] Tang H., Miller-Hooks E.: A TABU search heuristic for the team orienteering problem, *Computers & Operations Research*, vol. 32(6), pp. 1379–1407, 2005.

[51] Venkatesh P., Singh A.: Two metaheuristic approaches for the multiple traveling salesperson problem, *Applied Soft Computing*, vol. 26, pp. 74–89, 2015.

[52] Wang Y., Chen Y., Lin Y.: Memetic algorithm based on sequential variable neighborhood descent for the minmax multiple traveling salesman problem, *Computers & Industrial Engineering*, vol. 106, pp. 105–122, 2017.

[53] Wei C., Hindriks K.V., Jonker C.M.: Dynamic task allocation for multi-robot search and retrieval tasks, *Applied Intelligence*, vol. 45(2), pp. 383–401, 2016.

[54] Yan Z., Jouandeau N., Cherif A.A.: A survey and analysis of multi-robot coordination, *International Journal of Advanced Robotic Systems*, vol. 10(12), p. 399, 2013.

[55] Zaman S., Grosu D.: A combinatorial auction-based mechanism for dynamic VM provisioning and allocation in clouds, *IEEE Transactions on Cloud Computing*, vol. 1(2), pp. 129–141, 2013.

[56] Zhang K., Collins Jr E.G., Shi D.: Centralized and distributed task allocation in multi-robot teams via a stochastic clustering auction, *ACM Transactions on Autonomous and Adaptive Systems (TAAS)*, vol. 7(2), p. 21, 2012.

[57] Zhang S., Wang S.: Flexible Assembly Job-Shop Scheduling With Sequence-Dependent Setup Times and Part Sharing in a Dynamic Environment: Constraint Programming Model, Mixed-Integer Programming Model, and Dispatching Rules, *IEEE Transactions on Engineering Management*, vol. 65(3), pp. 487–504, 2018.

[58] Zhao W., Meng Q., Chung P.W.: A heuristic distributed task allocation method for multivehicle multitask problems and its application to search and rescue scenario, *IEEE Transactions on Cybernetics*, vol. 46(4), pp. 902–915, 2015.

[59] Zitouni F., Maamri R.: Cooperative learning-agents for task allocation problem, *Interactive Mobile Communication, Technologies and Learning*, pp. 952–968, Springer, 2017.

[60] Zitouni F., Maamri R.: FA-SETPOWER-MRTA: A Solution for Solving the Multi-Robot Task Allocation Problem. In: *IFIP International Conference on Computational Intelligence and Its Applications*, pp. 317–328. Springer, 2018.

[61] Zitouni F., Maamri R., Harous S.: FA–QABC–MRTA: a solution for solving the multi-robot task allocation problem, *Intelligent Service Robotics*, vol. 12, pp. 407–418, 2019.

Affiliations

Farouq Zitouni
Kasdi Merbah University, Department of Computer Science, Ouargla, Abdelhamid Mehri University, LIRE Laboratory, Constantine, Algeria, farouq.zitouni@univ-constantine2.dz, ORCID ID: https://orcid.org/0000-0003-2566-1457

Saad Harous
UAE University, Department of Computer Science and Software Engineering, Abu Dhabi, United Arab Emirates, harous@uaeu.ac.ae,
ORCID ID: https://orcid.org/0000-0001-6524-7352

Ramdane Maamri
Abelhamid Mehri University, Department of Computer Science, Constantine; Abdelhamid Mehri University, LIRE Laboratory, Constantine, Algeria,
ramdane.maamri@univ-constantine2.dz, ORCID ID: https://orcid.org/0000-0001-7962-6775

Received: 07.09.2019
Revised: 12.10.2019
Accepted: 15.10.2019