This article can be cited as F. Zitouni, R. Maamri and A. Zitouni, Multi-Robot Task Allocation with Energetic, Spatial and Temporal Constraints, International Journal of Artificial Intelligence, vol. 17, no. 1, pp. 102-138, 2019.

Multi-Robot Task Allocation with Energetic, Spatial and Temporal Constraints

Farouq Zitouni[1], **Ramdane Maamri**[2] **and Abdelhafid Zitouni**[3]

[1]Department of Computer Science
Kasdi Merbah University
Ouargla – Algeria
farouq.zitouni@univ-constantine2.dz

[2]Department of Computer Science
Constantine 2 – Abdelhamid Mehri University
Constantine – Algeria
ramdane.maamri@univ-constantine2.dz

[3]Department of Computer Science
Constantine 2 – Abdelhamid Mehri University
Constantine – Algeria
abdelhafid.zitouni@univ-constantine2.dz

Abstract

The Multi-Robot Task Allocation problem is the situation where some tasks and robots are given, then assignments between them must be found in order to optimize a certain measure (e.g. allocate the maximum number of tasks, etc.). We propose a generic framework to address heavily constrained MRTA problems. Some objective functions are proposed and extensively tested on ten datasets, which is our main contribution. Also, two allocation methods – exact and heuristic – are implemented, in order to compare values of adopted evaluation metrics. Performed simulations, obtained results, and comparative study show the effectiveness of the solution, even with a large number of robots and tasks.

Keywords: Multi-Robot Systems, Multi-Robot Task Allocation, Energetic Constraints, Spatial Constraints, Temporal Constraints.

2000 Computing Classification System 49K05, 49K15, 49S05.
ACM Subject Classification: Computing methodologies, Artificial intelligence, and Distributed artificial intelligence: Multi-agent systems; Computing methodologies, Artificial intelligence, and Distributed artificial intelligence: Cooperation and coordination; Bio-inspired approaches: Genetic algorithms;

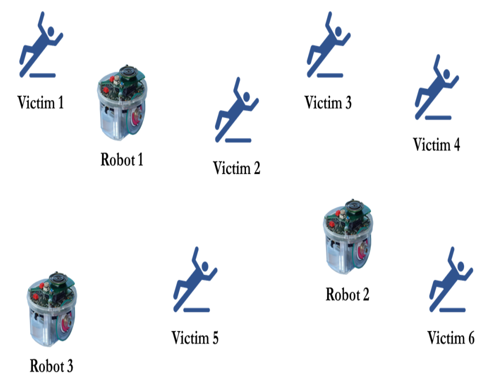
1 Introduction

A multi-robot system is a population of robots designed to communicate and cooperate with each other, in order to achieve some goals (Yan, Jouandeau and Cherif, 2013). This topic has always been present on research agendas for many years, but lately it has become a very hot research topic.

Because of their various advantages, e.g. complex tasks' resolution and designing simplicity (Khamis, Hussein and Elmogy, 2015), multi-robot systems have greatly attracted the attention of many researchers to examine their adequacy on several application areas, such as smart security (Liao and Su, 2011), victim search and rescue (Nagatani, Okada, Tokunaga, Kiribayashi, Yoshida, Ohno, Takeuchi, Tadokoro, Akiyama, Noda, Yoshida and Koyanagi, 2011), environment monitoring (Marino, Parker, Antonelli and Caccavale, 2013; Espina, Grech, De Jager, Remagnino, Iocchi, Marchetti, Nardi, Monekosso, Nicolescu and King, 2011; Shkurti, Xu, Meghjani, Higuera, y. Girdhar, Giguere, Dey, Li, Kalmbach, Prahacs, Turgeon, Rekleitis and Dudek, 2012), and health-care (Shiomi, Kamei, Kondo, Miyashita and Hagita, 2013).

Usually, multi-robot systems are concerned with some classical problems, like task allocation, coalition formation, object detection and tracking, communication relay, and self-organization (Khamis et al., 2015). We have chosen to deal with the first one, i.e. task allocation problem.

The Multi-Robot Task Allocation problem (MRTA) is informally defined as follows "given two sets of robots and tasks, what are the best assignments are between these two sets that optimize some criteria (Lerman, Jones, Galstyan and Matari, 2006; Tang and Parker, 2007; Mosteo and Montano, 2010)". This problem is a great challenge, and hard to resolve! Especially when robots are numerous, heterogeneous, and unreliable. Also, tasks must be accomplished under several constraints. Constrained task allocation – i.e. when, where, and in which order tasks need to be done – is an important key-problem in many real-life applications, e.g. space exploration and warehouse management (Nunes, Manner, Mitiche and Gini, 2017).



(a) the environment.

Victims	Requirements
Victim 1	{T}
Victim 2	{G,S}
Victim 3	{T,S}
Victim 4	{G}
Victim 5	{T,G,S}
Victim 6	{S}

Robots	Sensing devices
Robot 1	{T,S}
Robot 2	{T,G,S}
Robot 3	{G,S}

T	Thermometer
G	Glucometer
S	Sphygmomanometer

(b) the dataset.

Figure 1: An example of MRTA scenario.

Figure 1 depicts an instance of MRTA situation. Given an environment (sub-figure 1a) – with six tasks (i.e. the victims) and three robots – and an example of dataset (sub-figure 1b). Each victim has different requirements for its medical care – sensing devices, such as Thermometer (T), Glucometer (G), or Sphygmomanometer (S) – and each robot has some of these capabilities. For instance, the victim named "Victim 2" needs a Glucometer and Sphygmomanometer $\{G, S\}$. So, it can be assigned to "Robot 2", "Robot 3", "both Robot 1 and Robot 2", "both Robot 1 and Robot 3", or "both Robot 2 and Robot 3". Therefore, five allocations are possible, which one is the best? This is the scope of MRTA search question.

The main objective of the paper is to propose a generic solution to solve strongly constrained MRTA problems, i.e. allocate efficiently tasks to robots with best performance. It considers realistic constraints such as energetic, spatial, and temporal ones. Besides, it takes into account task requirements and robot capabilities in terms of sensing devices. In this work, we propose two main contributions.

The first contribution is the introduction and use of energy constraints in the MRTA problem, i.e. energy consumptions of robots and tasks are not omitted. In other words, previous actions of robots, i.e. displacements and task performing, will also define the quality of current and future allocations. Previous works, to treat this problem, have not really addressed this aspect – they mainly focused on temporal and spatial constraints. One paper has dealt with energetic constraints (Liu, Winfield, Sa, Chen and Dou, 2007), but in a superficial way – quantities of consumed energies are given and supposed to be constant. In our work, these quantities are dynamically calculated using established laws of physics. As we know, the majority of task allocation scenarios involve autonomous robots that generally do their tasks with minimum human, e.g. search and rescue of victims in dangerous areas. Therefore, it is very important to consider energy autonomy of robots in MRTA problems, e.g. the current energy of a robot might not suffice to carry out a certain task: so useless it is considered, even if it optimizes, for instance, the traveled distance. Finally, this constraint is expressed in the form of an energy gauge for each robot, and its value is variable over time, i.e. decreases according to taken actions.

The second contribution is the modification of some objective functions, frequently used in MRTA problems, that consider some usual optimization criteria, i.e. traveled distances, travel times, and spent/obtained costs/rewards. Moreover, a new objective function is proposed. It considers the energy consumptions of robots, i.e. displacements in the environment and execution of tasks. Besides, these objective functions consider also the number of cooperating robots, gains that they obtain, and their contributions in terms of sensing devices. Lastly, the mathematical formulation of MRTA evoked in (Nunes et al., 2017) was modified, and two equations expressing energy constraints are added.

The remainder of the paper is organized as follows. Section 2 gives an overview of needed concepts to understand the MRTA problem. Section 3 shows some already done works to address the different MRTA problems. Section 4 explains the proposed solution. Section 5 demonstrates simulations, obtained results, and their discussions. Finally, a conclusion and some perspectives are given in Section 6.

2 Background

Temporal, spatial, and energetic constraints are crucial in our solution. We show some basic definitions in Section 2.1 and temporal models in Section 2.2. Taxonomies for MRTA problems are presented in Section 2.3.

2.1 Basic definitions

Definition 2.1. a robot is an autonomous entity, immersed in an environment and capable of performing actions (Nunes et al., 2017). If MRTA problem is taken into consideration, robots are typically modeled as material points – physical layer is omitted, i.e. motions, how actions are done, etc.

Definition 2.2. a group is a set of robots working together to achieve a common goal – notice that a group must contain at least one robot. If it is dynamic, a group is commonly called "coalition" – formed to perform a task and dissolved just after its accomplishment (Parker and Tang, 2006).

Definition 2.3. a task is an action to perform. Other names may be adopted: work unit, activity or query. In some cases, tasks are composed of jobs (Davis and Burns, 2011), but in other cases jobs are composed of tasks (Balas, Simonetti and Vazacopoulos, 2008).

Definition 2.4. a time window is an interval, where lower and upper bound values are respectively "the earliest start date" and "the latest finish date" of a certain task. If the earliest start date is not provided, latest finish date will be called "deadline". A time window is close if both dates are given (Nunes et al., 2017).

Definition 2.5. synchronization constraints specify temporal restrictions on tasks, e.g. both tasks t_1 and t_2 must start at the same time (Nunes et al., 2017).

Definition 2.6. precedence constraints specify relationships between tasks, e.g. task t_2 should start after the finishing of task t_1 (Nunes et al., 2017).

Definition 2.7. a schedule is a table in which each task has a starting date, a finishing date, or both. In some cases, each robot has its own schedule (Nunes and Gini, 2015), but in other cases, all robots share the same one.

Definition 2.8. if a schedule is taken into consideration, a makespan represents the difference between the finishing date of last task and the starting date of first task (Nunes et al., 2017).

Definition 2.9. given a group of robots R and a task t, if robots in R are capable of doing t, one can define an application $u(R, t)$ which is called "utility" of R for t (Korsah, Stentz and Dias, 2013).

2.2 Temporal models

Time can be modeled as time points, e.g. 3 pm, or intervals, e.g. [3 pm-5 pm]. In practice, time representation in the form of intervals is frequently adopted. Intervals might be used to express temporal constraints on tasks. Figure 2 (Allen, 1983) depicts all used ones. Also, temporal constraints can be modeled in the form of graphs called "Simple Time Networks" (STN) (Dechter, Meiri and Pearl, 1991). Nodes represent time points and weighted arcs express temporal constraints.

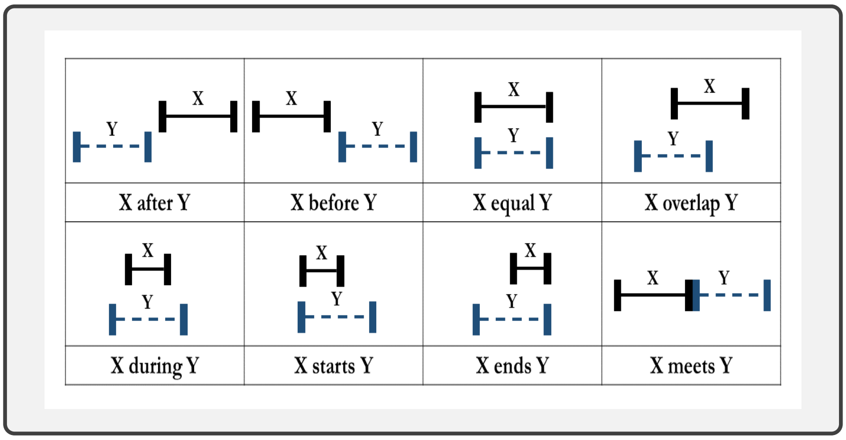


Figure 2: Used relationships to express temporal constraints.

2.3 Taxonomies for MRTA problems

We mainly find three taxonomies for categorization of MRTA problems. For simplicity, they will be called “Taxonomy 1”, “Taxonomy 2”, and “Taxonomy 3”, respectively according to their chronological order of appearance.

2.3.1 Taxonomy 1

Gerkey and Matarić (Gerkey and Matari, 2004) proposed, in 2004, an elegant taxonomy – quickly became widely used – for categorization of MRTA problems. It considers characteristics of robots, tasks, and assignments as follows.

- Single-Task robots (ST) vs. Multi-Task robots (MT)
  - ST: each robot can only do one task at a time.
  - MT: some robots can simultaneously do several tasks.
- Single-Robot tasks (SR) vs. Multi-Robot tasks (MR)
  - SR: each task requires exactly one robot for its accomplishment.
  - MR: some tasks require the cooperation of several robots for their accomplishment.
- Instantaneous Assignments (IA) vs. Time-extended Assignments (TA)
  - IA: tasks are allocated to robots considering only current allocations.
  - TA: tasks are allocated to robots considering both current and future allocations.

2.3.2 Taxonomy 2

Ayorkor Korsah and his co-authors (Korsah et al., 2013) improved, in 2013, the taxonomy of Gerkey and Matarić by adding a new level. This new level considers four dependencies (interrelated utilities and temporal constraints) between robots and tasks as follows.

- **No Dependencies (ND):** no robot utility, for a given task, depends on other robots or tasks.
- **In-Schedule Dependencies (ID):** each robot utility, for a given task, only depends on its own schedule.
- **Cross-Schedule Dependencies (XD):** each robot utility, for a given task, does not solely depend on its own schedule, but it also depends on other robot schedules – construction of schedules is static.
- **Complex Dependencies (CD):** each robot utility, for a given task, does not solely depend on its own schedule, but it also depends on other robot schedules – construction of schedules is dynamic.

2.3.3 Taxonomy 3

Nunes and his co-authors (Nunes et al., 2017) extended, in 2017, the taxonomy of Gerkey and Matarić by developing "Time-extended Assignments (TA)" axis, in order to include temporal and ordering constraints. This latter considers now 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".

3 Related works

Several centralized and distributed approaches have been proposed to solve MRTA problems (Zheng and Koenig, 2008; Shehory and Kraus, 1998; Chapman, Micillo, Kota and Jennings, 2009; Fatima and Wooldridge, 2001; Kong, Zhang and Ye, 2014; Scerri, Farinelli, Okamoto and Tambe, 2005; Petcu and Faltings, 2005; Michalak, Sroka, Rahwan, Wooldridge, McBurney and Jennings, 2010; Ramchurn, Polukarov, Farinelli, Truong and Jennings, 2010; Du, Zhou, Qu, Shi and Yang, 2010; Zitouni and Maamri, 2016). In general, these solutions can be categorized in two families – market and optimization based approaches (Khamis et al., 2015). We discuss some market and optimization based approaches in Sections 3.1 et 3.2. Additionally, we describe the behavioral approaches and give a short synthesis in Section 3.3.

3.1 Market-based approaches

Market-based approaches have been often used to resolve MRTA problems, due to their several advantages (Tang and Parker, 2007; Dias, 2004; Zlot, 2006), like efficiency (Coltin and Veloso, 18-22 Oct. 2010), robustness (Coltin and Veloso, 18-22 Oct. 2010; Zlot, 2006; Dias and Stentz, 2000), scalability (Coltin and Veloso, 18-22 Oct. 2010; Zlot, 2006), online input (Dias and Stentz, 2002; Zlot, 2006), and uncertainty (Zlot, Stentz, Dias and Thayer, 2002).

They provide an effective way to coordinate robot activities, are inspired from the economic theory, and are based on the auctions – assign goods to bidders considering bids and auction criteria (Zlot, 2006).
Market-based approaches use explicit communications. Auctions start when an auctioneer announces the availability of tasks, and robots offer bids. Then, the auctioneer receives bidders' responses, allocates tasks to robots considering an objective function, and notifies concerned ones (Zlot and Stentz, 2006).
Among market-based approaches, First-Price Auctions (Zlot et al., 2002), Dynamic Role Assignment (Chaimowicz, Campos and Kumar, 2002), Traderbots (Dias, 2004), M+ (Botelho and Alami, 1999), MURDOCH (Gerkey and Mataric, 2002), and DEMiR-CF (Sariel and Balch, 2006). Authors of (Parker, Nunes, Godoy and Gini, 2016; Hussein, Adel, Bakr, Shehata and Khamis, 2014; Zitouni and Maamri, 2017) proposed three solutions to the MRTA problem taking into consideration search and rescue missions. Coalitions are also exploited in the works (Zhang and Parker, 2013a; Zhang, Parker and Kambhampati, 2014) to address this problem. Because of their scalability, these methods are well-suited to distributed robots and generally produce optimal allocations – at local level. However, they are not without disadvantages. Robots cooperate through explicit communications, and consume many resources – if communication medium is broken, performance will significantly degrade (Kalra and Martinoli, 2006). They are suitable for small – to medium – scale MRTA problems. Finally, they suffer from the formalization lack of objective functions (Chaimowicz et al., 2002) and the use of negotiation protocols (Dias, 2004).

3.2 Optimization-based approaches

Optimization – branch of applied mathematics – solves problems by finding an optimal solution according to an objective function – describes quantitatively the purpose (Horst, 2002). There are many optimization-based approaches to address MRTA problems (Spall, 2003; Diwekar, 2003; Lenagh, Dasgupta and Munoz-Melendez, 2015; Zhang and Parker, 2013b; Campbell, Johnson and How, 2013; Ding, Zhu, He and Jiang, 2006; Zitouni and Maamri, 2018). Authors of (Atay and Bayazit, 2006; Manathara, Sujit and Beard, 2011) proposed two solutions based on linear integer programming – to cover a region by some heterogeneous robots and solve the MRTA problem in UAV, respectively. Authors of (Mosteo and Montano, 2006; Mosteo, 2010) used traveling salesman problem and simulated annealing to formulate and solve this problem. In (Juedes, Drews, Welch and Fleeman, 2004; Kmiecik, Wojcikowski, Koszalka and Kasprzak, 2010), authors combined simulated annealing and some heuristics to assign tasks to processors. Additionally, two solutions based on genetic algorithms have been proposed in (Shea, Alexander and Peterson, 2003; Jones, Dias and Stentz, 2011) to design an individual monitoring system, capable of simultaneously tracking several targets and manage fire extinguishing scenarios, respectively. Optimization by ant colony was used in (Wang, Gu and Li, 2012; Ding, He and Jiang, 2003; Dong and Zheng-ou, 2004) to solve MRTA problems. Likewise, this problem was solved using tabu-search (Kmiecik et al., 2010; Chen and Lin, 2000). In (Liu and Kulatunga, 2007), simulated annealing and ant colony have been combined to solve path planning and MRTA problems. In (Badreldin, Hussein and Khamis, 2013), some solutions – based on trajectory and population meta-heuristics – have been proposed and extensively tested on some scenarios – with extended tasks and highly heterogeneous robots. Authors of (Nedjah, de Mendona and de Macedo Mourelle, 2015) have proposed a distributed algorithm based on particles swarm optimization for dynamic task allocation with real robots. Authors of (Zhang, Xie, Yu and Wang, 2007; Zhang, Liu, Fu and Wu, 2009; Liu, Zhang, Wu and Liu, 2010; Liu, Sun and cheng Hung, 2011) have used swarm intelligence to adaptive task assignment in large-scale multi-robot systems. The paper (Ghosn, Drouby and Harmanani, 2016) proposed a good solution to the well-known open-shop scheduling problem, where it investigated the use of parallel genetic algorithms. The paper (Gini, 2017) presented a well-structured survey, where the class of scheduling problems involving the allocation of tasks with temporal and ordering constraints is shown. A good introduction to widely-used human operator and slave robot models is presented in the paper (Haidegger, Kovacs, Precup, Benyo, Benyo and Preitl, 2012), along with an overview of the telehealth concept. The works in (Aragues, Cortes and Sagues, 2011; Echegoyen, Villaverde, Moreno, na and d'Anjou, 2010; Borja, Jose Manuel, Ekaitz, Zelmar and Manuel, 2011) exploited the field of Multi-Component Robotic Systems and respectively provided some good solutions to problems involving cooperative robots, such as cooperative mapping of an environment, establishing dynamic communication links, and driving a hose to a goal.

3.3 Behavioral approaches and synthesis

In this category, tasks to be performed are divided into behavioral groups – tasks of the same group are interrelated. Generally, these approaches are robust, fault-tolerant and operate in real-time, but found solutions are optimal at the local level. Among behavioral approaches, we can find Alliance (Parker, 1998), BLE (Werger and Mataric, 2000), and ASyMTRe (Fang and Parker, 2005).

In fact, most real-life MRTA applications manipulate heterogeneous robots and tasks – disparate configurations (Khamis et al., 2015). Thus, it is crucial to consider these differences in proposed solutions. Several features can be considered to handle robot heterogeneities, such as spatial positions, physical properties, and energetic constraints. Similarly, task heterogeneities can be characterized by their spatial positions, needs, and temporal constraints.

4 Proposed solution

We propose an efficient solution to address MRTA problems. Adopted hypotheses and used notations are presented in Sections 4.1 and 4.2, respectively. Mathematical formulation of the problem is shown in Section 4.3. Proposed objective functions are exposed in Section 4.4. Finally, used algorithms and allocation methods are explained in Sections 4.5 and 4.6, respectively.

Table 1: Notations used in the paper.

Notation	Meaning	Mathematical formula
v_b	Velocity of bidder b.	-
m_b	Mass of bidder b.	-
(x_b, y_b, z_b)	Cartesian coordinates of bidder b.	-
a_b	Altitude of bidder b relative to the ground.	$a_b = \sqrt{(z_b)^2}$
E^b_K	Kinetic energy of bidder b – the energy that it possesses due to its motion.	$E^b_K = 0.5 \times m_b \times (v_b)^2$
E^b_P	Potential energy of bidder b – the energy that it possesses due to its altitude relative to the ground.	$E^b_P = 9.81 \times m_b \times a_b$
U^b	Battery voltage of bidder b.	-
A^b	Battery capacity of bidder b.	-
η	Peukert's exponent of bidder battery.	-
R^b	Hour-rating of bidder battery (C-rate).	-
G^b	Gauge energy of bidder battery.	-
(x_t, y_t, z_t)	Cartesian coordinates of task t.	-
DUR_t	Duration of task t.	-
ES_t	Earliest start date of task t.	-
S_t	Estimated start date of task t.	-
LS_t	Latest start date of task t.	-
EF_t	Earliest finish date of task t.	-
F_t	Estimated finish date of task t.	-
LF_t	Latest finish date of task t.	-
$\|A\|$	Denotes cardinality of set A.	-
$TT^b_{\langle t,t' \rangle}$	Travel time between tasks t and t' – taken by bidder b.	$TT^b_{\langle t,t' \rangle} = \frac{\sqrt{(x_t - x_{t'})^2 + (y_t - y_{t'})^2 + (z_t - z_{t'})^2}}{v_b}$
$DE^b_{\langle t,t' \rangle}$	Energy consumed during displacement between tasks t and t' – spent by bidder b.	-
$EE^b_{\langle R,t' \rangle}$	Energy consumed during performing task t' – spent by bidder b. The symbol R expresses a relation between b and t', e.g. b offers a sensing device to t'.	-

4.1 Preliminaries and assumptions

We explain the proposed solution to address a class of MRTA problems. We deal with the problems assuming that each robot can only do one task at a time, some tasks require cooperation of several robots for their accomplishment, tasks are allocated to robots considering both current and future allocations, each robot utility for a given task only depends on its own schedule, and temporal constraints are considered and expressed in the form of time windows. Found allocations do respect energetic, spatial and temporal constraints on robots and tasks, and minimize/maximize an objective function (e.g. minimize traveled distances, maximize obtained gains, minimize consumed energies, etc.).
It is worth pointing out that this paper is the continuation of some previous papers, that we have published to address the MRTA problem. First, the paper (Zitouni and Maamri, 2016) proposed a dynamic protocol to deal with the MRTA problem, and the task allocation stage is done using Ant Colony Optimization. Second, the paper (Zitouni and Maamri, 2017) presented an approach that uses Quantum Genetic Algorithms and Reinforcement Learning to solve this problem. Third, the paper (Zitouni and Maamri, 2018) solved this problem combining the Firefly and Power Set algorithms. Finally, in this paper we propose some objective functions to cover commonly used optimization criteria, such as Energetic, Special, and Temporal. Also, a comparison between an exact and heuristic methods is performed to show limits of each one of these approaches.
Proposed algorithms use a market-based approach, i.e. they adopt auctions consisting of auctioneers, bidders, and goods. In the case of MRTA problem: robots are bidders, goods are tasks, and auctioneers announce tasks' arrival and determine an allocation for each task. Finally, algorithms use "single-good auctions" (Gelenbe, 2009), and adopt "Contract Net Protocol" for messages' exchange (Smith, 1980a).
We assume that we have one auctioneer, a set of bidders $B = \{b_1, b_2, \cdots, b_n\}$, and a set of tasks $T = \{t_1, t_2, \cdots, t_m\}$. Auctioneer role is central, it should announce tasks to bidders, receive their offers, calculate allocations that minimize/maximize a considered objective function, and finally transmit the found results to the concerned bidders. Although the use of an auctioneer can be seen as a bottleneck – i.e. if it fails, the system fails too, this limitation is acceptable since it results in minimizing the amount of exchanged messages, maintaining – at any moment – a global view of the system state, and sharing calculations on bidders.

4.2 Notations used in the paper

In order to avoid unnecessary repetitions in defining different variables and symbols, used notations are summarized in Table 1.

4.3 Mathematical formulation of MRTA problem

If we wish to allocate a task $t \in T$ to a bidder $b \in B$, it is quite natural to define a relationship between them. This relationship is intuitively expressed as follows "b is capable of doing t". Capability factor is intimately linked to the considered domain and varies according to several parameters. In our case, we chose to use sensors as a relationship between tasks and bidders

– i.e. tasks need sensors and bidders have sensors. For instance, a task consists of getting the ambient temperature of a room and a bidder has a thermometer.
We suppose that we have a set of sensors $\Omega = \{\omega_1, \omega_2, \cdots, \omega_p\}$. The set $2^B = \{\emptyset, \{b_1\}, \{b_2\}, \cdots\}$ represents all bidder groups that can be formed from B – all subsets of B or $\wp(B)$. The set $\Omega^b \subseteq \Omega$ represents sensors of bidder b. The set $\Omega^t \subseteq \Omega$ represents sensors needed by task t. We give the following corollary that expresses the primitive allocation relationship between tasks and bidder groups – notice that a group may contain one bidder.

Corollary 4.1. *Given a task $t \in T$ and a bidder group $C \in 2^B$. Task t can be allocated to group C if, and only if, condition (4.1) is satisfied.*

$$\forall b \in C, \bigcup \Omega^b \cap \Omega^t = \Omega^t \tag{4.1}$$

We define the indicator $o^C_{\langle b,\omega \rangle} \in \{0, 1\}$. If $\langle b \in C$ and $\omega \in \Omega^b$ and b offers sensor ω to group $C\rangle$, it takes the value "1". Otherwise, it takes the value "0". The set $\Omega^b_C = \{\omega | o^C_{\langle b,\omega\rangle} = 1\}$ represents all sensors that b offers to C ($\Omega^b_C \subseteq \Omega^b$). We can generalize previous corollary as follows.

Corollary 4.2. *Given a task $t \in T$ and a bidder group $C \in 2^B$. Task t can be allocated to group C if, and only if, following two conditions, i.e. (4.2), are simultaneously satisfied.*

$$\forall b \in C : \begin{cases} \bigcap \Omega^b_C & = & \emptyset \\ \bigcup \Omega^b_C & = & \Omega^t \end{cases} \tag{4.2}$$

Example 4.3. *We present an example to explain previous concepts. Different sets are summarized in Table 2.*

Table 2: Example of dataset.

Sensor	**Meaning**
ω_1	Heat sensor
ω_2	PIR sensor
ω_3	Ultrasonic sensor
ω_4	Humidity sensor
ω_5	Light sensor

Tasks	Ω^t
t_1	$\{\omega_3, \omega_4\}$
t_2	$\{\omega_2\}$
t_3	$\{\omega_5\}$
t_4	$\{\omega_1, \omega_2, \omega_3, \omega_5\}$
t_5	$\{\omega_1, \omega_3, \omega_4\}$
t_6	$\{\omega_3, \omega_5\}$

Bidders	Ω^b
b_1	$\{\omega_3, \omega_4\}$
b_2	$\{\omega_1\}$
b_3	$\{\omega_3\}$
b_4	$\{\omega_1, \omega_2, \omega_3, \omega_4, \omega_5\}$
b_5	$\{\omega_3, \omega_5\}$
b_6	$\{\omega_2, \omega_3\}$
b_7	$\{\omega_1, \omega_3, \omega_4\}$

For instance, we consider task t_4 which can be assigned to the group $C = \{b_4, b_5\}$, where $\Omega^{b_4}_C = \{\omega_1, \omega_2\}$ and $\Omega^{b_5}_C = \{\omega_3, \omega_5\}$ – of course other assignments are also possible.

As it is known, the assignment problem is one of the fundamental combinatorial optimization problems in optimization and operations research. The problem considers some agents and tasks and each agent can be assigned to perform any task, with some cost that varies according to the agent-task assignment. It is required to perform all tasks by assigning exactly one agent to each task and exactly one task to each agent, in such a way that the total cost function of the assignment is minimized. This is the basic definition of the task allocation problem: no constraints are considered. What about the case where agents are with limited abilities, e.g.

an agent cannot perform all tasks, a robot can theoretically perform a task but the distance is very far to reach its location in a given time, etc.? All these constraints make the problem hard to handle. For this reason, in this paper, we propose a solution to the problem as realistic as possible by handling energetic, spatial and temporal constraints. Finally, we give the modified mathematical formulation of MRTA problem (Nunes et al., 2017). Table 3 summarizes different equations and constraints. Equations (4.7) and (4.13) present added energetic constraints.

Table 3: The modified standard mixed-integer linear formulation of MRTA problem.

Minimize or maximize considered objective function $f(.)$
Subject to

$$\forall t \in T : \sum_{C \in 2^B} x_t^C \leq 1 \tag{4.3}$$

$$\forall t \in T, C \in 2^B : \sum_{b \in C} \sum_{\omega \in \Omega^b} o_{\langle b,\omega \rangle}^C = |\Omega^t| \times x_t^C \tag{4.4}$$

$$\forall b \in B : \sum_{t \in T} y_{\langle t^s,t \rangle}^b = 1 \tag{4.5}$$

$$\forall b \in B : \sum_{t \in T} y_{\langle t,t^f \rangle}^b = 1 \tag{4.6}$$

$$\forall b \in B : 0 \leq G^b \leq 100 \tag{4.7}$$

$$\forall t \in T : ES_t \leq S_t \leq LS_t \tag{4.8}$$

$$\forall t \in T : EF_t \leq F_t \leq LF_t \tag{4.9}$$

$$\forall t \in T : (F_t - S_t) \geq DUR_t \tag{4.10}$$

$$\forall t' \in T, b \in B : \sum_{t \in T, t \neq t'} y_{\langle t,t' \rangle}^b - \sum_{t'' \in T, t' \neq t''} y_{\langle t',t'' \rangle}^b = 0 \tag{4.11}$$

$$\forall t, t' \in T, b \in B : S_t + DUR_t + TT_{\langle t,t' \rangle}^b - M \times (1 - y_{\langle t,t' \rangle}^b) \leq S_{t'} \tag{4.12}$$

$$\forall t, t' \in T, b \in B : G^b - DE_{\langle t,t' \rangle}^b - \sum_{\omega \in \Omega^b} (DUR_{t'} \times o_{\langle b,\omega \rangle}^C \times EE_{\langle \omega,t' \rangle}^b) - M' \times (1 - y_{\langle t,t' \rangle}^b) > N \tag{4.13}$$

$$\forall b \in B, C \in 2^B : o_{\langle b,\omega \rangle}^C \in \{0,1\} \tag{4.14}$$

$$\forall t \in T, C \in 2^B : x_t^C \in \{0,1\} \tag{4.15}$$

$$\forall t, t' \in T, b \in B : y^b_{\langle t,t' \rangle} \in \{0, 1\} \tag{4.16}$$

The different constrains of previous formulations are explained as follows. Constraint (4.3): each task t is allocated at most to one group C. Constraint (4.4): if task t is allocated to group C, all required sensors must be available. Constraint (4.5): each bidder b has a fictional task t^s at its starting position. Constraint (4.6): each bidder b has a fictional task t^f at its finishing position. Constraint (4.7): energy gauge of bidder b is valid. Constraint (4.8): starting date of task t is valid. Constraint (4.9): finishing date of task t is valid. Constraint (4.10): task time is long enough. Constraint (4.11): each bidder b must respect execution order of its tasks. Constraint (4.12): travel time between two consecutive tasks is long enough. Constraint (4.13): bidder energy, to reach a given task and perform it, is enough. N is a constant. Indicator (4.14): bidder b offers sensor ω to group C. Indicator (4.15): task t is allocated to group C. Indicator (4.16): bidder b does task t' just after task t.

4.4 Objective functions and presentation of the optimization problem

We present the formal definitions of proposed objective functions. First, we give the formal definition of some elementary applications used to define these objective functions in next sections with their meanings. Table 5 summarizes expressions of objective functions, and their optimality criterion. Coefficients α and β are used to accentuate equation terms. Table 4 summarizes the presentation of the optimization problem.

$$\begin{array}{rccl} cost: & (B,\Omega) & \rightarrow & \mathbb{R}^+ \\ & (b,\omega) & \mapsto & \begin{cases} cost(b,\omega) & \text{, if } \omega \in \Omega^b \\ 0 & \text{, otherwise} \end{cases} \end{array} \tag{4.17}$$

It assigns to each bidder sensor a positive value – representing its cost.

$$\begin{array}{rccl} intensity: & (B,\Omega) & \rightarrow & \mathbb{R}^+ \\ & (b,\omega) & \mapsto & \begin{cases} intensity(b,\omega) & \text{, if } \omega \in \Omega^b \\ 0 & \text{, otherwise} \end{cases} \end{array} \tag{4.18}$$

It assigns to each bidder sensor a positive value – representing its working current.

$$\begin{array}{rccl} reward: & (T,\Omega) & \rightarrow & \mathbb{R}^+ \\ & (t,\omega) & \mapsto & \begin{cases} reward(t,\omega) & \text{, if } \omega \in \Omega^t \\ 0 & \text{, otherwise} \end{cases} \end{array} \tag{4.19}$$

It assigns to each needed task sensor a positive value – representing its reward.

$$\begin{array}{rccl} rate: & (T,\Omega) & \rightarrow & \mathbb{R}^+ \\ & (t,\omega) & \mapsto & \begin{cases} rate(t,\omega) & \text{, if } \omega \in \Omega^t \\ 0 & \text{, otherwise} \end{cases} \end{array} \tag{4.20}$$

It assigns to each needed task sensor a positive value – representing the estimated time it will be exploited.

$$
\begin{array}{rlcl} distance: & (T,T) & \rightarrow & \mathbb{R}^+ \\ & (t,t') & \mapsto & \sqrt{(x_t - x_{t'})^2 + (y_t - y_{t'})^2 + (z_t - z_{t'})^2} \end{array} \tag{4.21}
$$

It calculates the Euclidean distance between two task positions.

$$
\begin{array}{rlcl} gain: & (B, 2^B, T, T) & \rightarrow & \mathbb{R}^+ \\ & (b, C, t, t') & \mapsto & [\sum_{\omega \in \Omega_C^b} reward(t', \omega)] \times e^{-\gamma \times (TT^b_{\langle t,t' \rangle})^2} \end{array} \tag{4.22}
$$

It calculates bidder gain for task t', knowing that bidder b is allocated to task t. γ is a regularization parameter.

$$
\begin{array}{rlcl} displacement: & (B,T,T) & \rightarrow & \mathbb{R}^+ \\ & (b,t,t') & \mapsto & 100 \times \frac{TT^b_{\langle t,t' \rangle} \times (\frac{E^b_K + E^b_P}{U^b})^\eta}{R^b \times (\frac{A^b}{R^b})^\eta} \end{array} \tag{4.23}
$$

It calculates the percentage of bidder consumed energy when displacing from task t to task t'.

$$
\begin{array}{rlcl} sensor: & (B,T,\Omega) & \rightarrow & \mathbb{R}^+ \\ & (b,t,\omega) & \mapsto & 100 \times \frac{DUR_t \times rate(t,\omega) \times (intensity(b,\omega))^\eta}{R^b \times (\frac{A^b}{R^b})^\eta} \end{array} \tag{4.24}
$$

It calculates the percentage of bidder consumed energy when using its sensors to do the considered task.

$$
\begin{array}{rlcl} contribution: & (B, 2^B) & \rightarrow & \mathbb{R}^+ \\ & (b, C) & \mapsto & \frac{|\Omega_C^b|}{|\Omega^b|} \end{array} \tag{4.25}
$$

It calculates the rate of bidder offered sensors in the considered group.

Table 4: Presentation of the optimization problem.

Objective functions	Minimize or maximize one of the objective functions defined in Table 5 or at least a combination of two objective functions

Constraints	**Robots**	**Tasks**
	Constraint (4.5) in Table 3 Constraint (4.6) in Table 3 Constraint (4.7) in Table 3 Constraint (4.11) in Table 3 Constraint (4.12) in Table 3 Constraint (4.13) in Table 3	Constraint (4.3) in Table 3 Constraint (4.4) in Table 3 Constraint (4.8) in Table 3 Constraint (4.9) in Table 3 Constraint (4.10) in Table 3

Variables	**Robots**	**Tasks**
	Indicator (4.14) in Table 3 Indicator (4.16) in Table 3	Indicator (4.15) in Table 3

Table 5: Objective functions.

Formal definition of objective functions	**Optimality**
$f_1: (B, 2^B, T, T) \rightarrow \mathbb{R}^+$ $(b, C, t, t') \mapsto (\lvert C\rvert)^\alpha \times \sum_{b \in C}[(\frac{gain(b,C,t,t')}{contribution(b,C)})^\beta \sum_{\omega \in \Omega_C^b} cost(b, \omega)]$	Minimize tasks' costs.
$f_2: (B, 2^B, T, T) \rightarrow \mathbb{R}^+$ $(b, C, t, t') \mapsto (\frac{1}{\lvert C\rvert})^\alpha \times \sum_{b \in C}[(\frac{contribution(b,C)}{gain(b,C,t,t')})^\beta \sum_{\omega \in \Omega_C^{t'}} reward(t', \omega)]$	Maximize bidders' rewards.
$f_3: (B, 2^B, T, T) \rightarrow \mathbb{R}^+$ $(b, C, t, t') \mapsto max(f_2(b, C, t, t') - f_1(b, C, t, t'), 0)$	Maximize bidders' benefits.
$f_4: (B, 2^B, T, T) \rightarrow \mathbb{R}^+$ $(b, C, t, t') \mapsto (\lvert C\rvert)^\alpha \times \sum_{b \in C}[(\frac{gain(b,C,t,t')}{contribution(b,C)})^\beta \times distance(t, t')]$	Minimize bidders' traveled distances.
$f_5: (B, 2^B, T, T) \rightarrow \mathbb{R}^+$ $(b, C, t, t') \mapsto (\lvert C\rvert)^\alpha \times \sum_{b \in C}[(\frac{gain(b,C,t,t')}{contribution(b,C)})^\beta \times TT^b_{\langle t,t' \rangle}]$	Minimize bidders' travel times.
$f_6: (B, 2^B, T, T) \rightarrow \mathbb{R}^+$ $(b, C, t, t') \mapsto (\lvert C\rvert)^\alpha \times [\sum_{b \in C}[(\frac{gain(b,C,t,t')}{contribution(b,C)})^\beta \times displacement(b, t, t')]$ $+ \sum_{b \in C}[(\frac{gain(b,C,t,t')}{contribution(b,C)})^\beta \sum_{\omega \in \Omega_C^b} sensor(b, t', \omega)]]$	Minimize bidders' consumed energies.

4.5 Proposed algorithms

Algorithms 1 and 2 describe the internal functioning of different system actors, i.e. auctioneer and bidders. Their cooperative behaviors are summarized as follows.

1. Auctioneer announces to bidders the existence of tasks to allocate;
2. Each bidder chooses tasks that it can perform, and replies to the auctioneer;
3. If it is possible, the auctioneer determines an allocation for each task;
4. Auctioneer notifies concerned bidders.

Algorithm 1: behavior of the auctioneer "a".

Input : Set of bidders B.

Output: Allocations between bidders and tasks.

```
while (auctioneer is alive) do
    PeriodicBehavior
        T ← getDiscoveredTasks();
        if (|T| ≠ 0) then
            foreach b ∈ B do
                sendMessageToBidder(b,"REQUEST-CHOOSE-TASKS",T);
            end
        end
    end
    MessageBehavior
        message ← receiveMessagesFromBidder();
        if (message.isEmpty() ≠ true) then
            if (message.getSubject() = "ANSWER-CHOOSE-TASKS") then
                T′ ← message.getContent();
                foreach t′ ∈ T′ do
                    if (t′.canBeAllocated() = true){equation (4.2)} then
                        B′ ← computeAllocationFor(t′);
                        foreach b′ ∈ B′ do
                            sendMessageToBidder(b′,"REQUEST-REACH-TASK",t′);
                        end
                    end
                end
            end
        end
    end
end
```

Algorithm 2: behaviour of the bidder "b".

```
Input  :
Output:
while (bidder is alive) do
    OneBehavior
        t^s ← createFictionalTask();
        STN_b.add(t^s);
    end
    MessageBehavior
        message ← receiveMessageFromAuctioneer();
        if (message.isEmpty() ≠ true) then
            if (message.getSubject() = "REQUEST-CHOOSE-TASKS") then
                T ← message.getContent();
                T' ← createEmtyList();
                foreach t ∈ T do
                    if (b.canDo(t) = true){equation (4.1)} then
                        T'.add(t);
                    end
                end
                sendMessageToAuctioneer(a,"ANSWER-CHOOSE-TASKS",T');
            else if (message.getSubject() = "REQUEST-REACH-TASK") then
                t' ← message.getContent();
                b.moveTowards(t');
                STN_b.add(t');
            end
        end
    end
end
```

As it has been presented, behaviors of auctioneer and bidders are divided into three parallel sub-behaviors, which are “OneBehavior”, “PeriodicBehavior”, and “MessageBehavior”. This notation allows us to classify instructions according to their execution frequency.

1. **OneBehavior**: instructions executed once.
2. **PeriodicBehavior**: instructions executed periodically.
3. **MessageBehavior**: instructions executed when a message is received.

4.5.1 Tasks’ announcement

The auctioneer periodically runs the method “**getDiscoveredTasks()**” – PeriodicBehavior of algorithm 1. This method abstracts the process of task discovering and definition of their attributes, e.g. position, needs, etc. If tasks are found, a message – containing all useful information – is broadcasted to bidders. It is important to note that the way tasks are discovered is not our focus – we just abstract this step.

4.5.2 Tasks’ selection

Each bidder, when a message is received – MessageBehavior of algorithm 2, browses the list of received tasks – the set T, and determines the set T' of tasks it can perform – $T' \subseteq T$. If the following three conditions are simultaneously satisfied, the bidder b is capable of doing the task t.

1. If bidder b can at least offer a sensor to task t, the first condition is satisfied (equation (4.1)).
2. If bidder b can reach task t before its latest start date, the second condition is satisfied (constraint (4.12) in Table 3).
3. If the energy of bidder b allows it to reach and carry out task t, the third condition is satisfied (constraint (4.13) in Table 3).

In summary, if the previous three conditions are jointly fulfilled for a given task $t \in T$, it will be added to the set T'. Finally, the set T' will be sent to the auctioneer – list T' can be seen as the bid of the bidder. Likewise, another condition can be considered with previous ones: if the priority of the last task in STN_b, list of previously allocated tasks to bidder b, is greater than or equal to the priority of the task to be allocated, add it to the set T'. This heuristic implicitly allows us to fairly allocate tasks to bidders.

4.5.3 Tasks’ allocation

The auctioneer, when bidders’ bids are received – MessageBehavior of algorithm 1, sorts tasks in a descending order – according to their priorities – and assigns each one to a group of bidders. Both tasks and bidders’ bids are organized in the form of a matrix M, which has the following format.

Table 6: The general structure of bid matrix.

	Sorted tasks $t \in T$		
Bidders $b \in B$	bid_{11}	$\cdots$	bid_{1m}
	$\vdots$	$\ddots$	$\vdots$
	bid_{n1}	$\cdots$	bid_{nm}

The auctioneer processes the matrix M, column by column, checks whether the corresponding task can be allocated (equation (4.2)) – if yes it calculates its allocation, and finally notifies the concerned bidders. Bidders appearing on each column of M represent the largest group of bidders having selected the corresponding task, and the auctioneer must calculate the minimal one considering an objective function. We propose the following definitions to explain the meaning of allocation relationship between a task and a minimal group of bidders.

Definition 4.1. a task is realizable by a minimal group of bidders if they can offer the required sensors for its accomplishment (constraint (4.4) in Table 3).

Definition 4.2. a task is reachable by a minimal group of bidders if they can reach its position before its latest start date (all bidders in the group satisfy constraint (4.12) in Table 3).

Definition 4.3. a task is feasible by a minimal group of bidders if they have sufficient energy for its accomplishment (all bidders in the group satisfy the constraint (4.13) of Table 3).

Definition 4.4. a task can be done by a minimal group of bidders if, and only if, the task is realizable, reachable, and feasible.

To avoid overlapping groups, if a bidder is allocated to a given task, its bids for the other tasks are removed from the matrix M.

4.5.4 Bidders' notification

If allocated to a given task t, bidder b will receive a notification message from the auctioneer – MessageBehavior of algorithm 2. Consequently, it should move to reach the task location, and insert it in its STN_b. Once task t is performed, energy gauge of bidder b is decreased according to the effort it has put to achieve the task.

4.6 Used allocation methods

As known, the MRTA problem is NP-hard (Sandholm, Larson, Andersson, Shehory and Tohmé, 1999), and cannot effectively be solved by exact methods. Two methods – exact and heuristic – are compared to show advantages and limitations of each one. As an exact method, we have used the Cartesian product algorithm of several sets. As a heuristic method, we have used genetic algorithms (GAs) (Siddique and Adeli, 2013). Details of these methods are described in sections 4.6.1 and 4.6.2.

4.6.1 Exact method

Given a task t and its largest group of bidders C – $C \in 2^B$. We must determine all minimal groups which verify equation (4.2). Example 4.4 shows how to perform and choose the optimal one.

Example 4.4. *We use the dataset illustrated in example 4.3. We consider task t_4 and $\Omega^{t_4} = \{\omega_1, \omega_2, \omega_3, \omega_5\}$. Largest group of bidder having offered bids on t_4 is $C = \{b_1, b_2, b_3, b_4, b_5, b_6, b_7\}$. Bidder offered sensors are $\Omega^{b_1} = \{\omega_3\}$, $\Omega^{b_2} = \{\omega_1\}$, $\Omega^{b_3} = \{\omega_3\}$, $\Omega^{b_4} = \{\omega_1, \omega_2, \omega_3, \omega_5\}$, $\Omega^{b_5} = \{\omega_3, \omega_5\}$, $\Omega^{b_6} = \{\omega_2, \omega_3\}$, and $\Omega^{b_7} = \{\omega_1, \omega_3\}$.*
Classified bidders according to sensor names are class$(\omega_1) = \{b_2, b_4, b_7\}$, class$(\omega_2) = \{b_4, b_6\}$, class$(\omega_3) = \{b_1, b_3, b_4, b_5, b_6, b_7\}$, class$(\omega_4) = \{\perp\}$, and class$(\omega_5) = \{b_4, b_5\}$. Cardinalities of these classes are 3, 2, 6, 1, and 2, respectively. So, we have $3 \times 2 \times 6 \times 1 \times 2 = 72$ minimal groups verifying equation (4.2). The optimal one must minimize/maximize the considered objective function. The symbol $\perp$ represents a fictional bidder, and its role is to denote that corresponding sensor is not requested by the considered task. Some instances of minimal groups are $\{b_2, b_4, b_4, \perp, b_5\}$, $\{b_7, b_6, b_1, \perp, b_5\}$, $\{b_2, b_6, b_6, \perp, b_4\}$, $\cdots$

4.6.2 Heuristic method

Given a task t and its largest group of bidders C– $C \in 2^B$. Next we explain the executed steps of the GA to determine the best minimal group satisfying equation (4.2).
Encoding scheme, an individual is composed of one chromosome. The chromosome encodes a minimal group of bidders, and its length is $|\Omega|$ – we use Φ to denote it. Thus, each gene encodes a sensor. Gene values are strings. If the value of a gene is $\perp$, it means that the corresponding sensor is not required by the considered task, otherwise, we should find a bidder identifier – it means that this bidder offers the corresponding sensor to the considered task. For example, we consider $\Omega = \{\omega_1, \omega_2, \omega_3, \omega_4, \omega_5\}$, $t_1 = \{\omega_2, \omega_3, \omega_5\}$, $b_1 = \{\omega_1, \omega_2\}$, $b_2 = \{\omega_2, \omega_3, \omega_4\}$, $b_3 = \{\omega_3, \omega_4, \omega_5\}$. One chromosome can be encoded as $\Phi = [\perp, b_1, b_2, \perp, b_3]$.
Fitness function, the fitness function $F(\Phi)$ assigns to each chromosome a numerical value measuring its quality – that can be used to sort and compare chromosomes. We use equation (4.26) to calculate the fitness value of a chromosome.

$$F(\Phi_i) = \frac{f(\Phi_i)}{\Sigma_{q=1}^{N} f(\Phi_q)} \tag{4.26}$$

where Φ_q is the q^{th} individual, N is the population size, and $f(\Phi)$ is the objective function – optimization criterion (defined in Table 5). Before that, objective values, i.e.$f(\Phi)$, are scaled to avoid premature convergence of the population and maintain fairly constant selective pressure on the population. We use equation (4.27) – "sigma truncation scaling" (Oscar, Francisco and Frank, 2001) – to scale objective value of an individual Φ_i.

$$f(\Phi_i) = \frac{f(\Phi_i) - \frac{\Sigma_{q=1}^{N} f(\Phi_q)}{N} - \sigma}{\sigma} \tag{4.27}$$

where $\frac{\Sigma_{q=1}^{N} f(\Phi_q)}{N}$ and σ are average and standard deviation, respectively.

Selection, we use tournament selection. k individuals – $k \leq \frac{N}{4}$ – are randomly selected from the current population. The one having best fitness value is taken, and inserted into the mating pool. The process is repeated until mating pool reaches a given size N' – $N' \leq \frac{N}{2}$. With this method, we ensure that bad individuals are not selected, and best ones will not dominate. The value of k is directly related to the selective pressure, i.e. a reasonable value would ensure optimal solution (Siddique and Adeli, 2013).

Crossover, once the mating pool is created, we apply the crossover operator. Its aim is to create the individuals of next population. Initially, next population contains the individuals of mating pool, and rest of individuals will be created using the crossover operator. To apply this operator, two individuals are randomly selected from the mating pool, and a uniform crossover (Siddique and Adeli, 2013) is applied to them to produce a new one. This process is repeated until the size of the next generation becomes the same as the current population. The uniform crossover principle is shown in Figure 3.

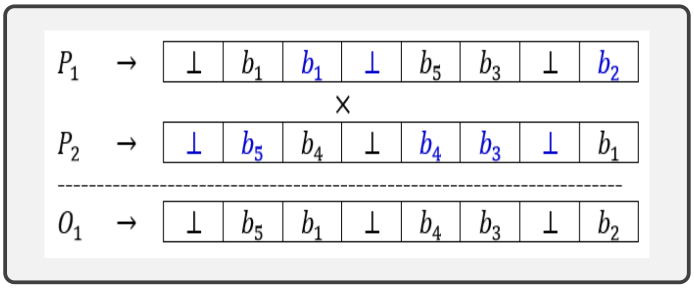


Figure 3: The principle of uniform crossover operator.

Mutation, random initialization of first population can sometimes limit exploited search space – we find solutions that are not close enough to the optimal one. This problem can be bypassed using mutation operator. When a new individual is created, a uniform mutation (Siddique and Adeli, 2013) is applied to its genes before its insertion into the next population. The uniform mutation replaces, with a certain probability P, the value of a given gene by a new value belonging to a certain set. The uniform mutation principle is shown in Figure 4.

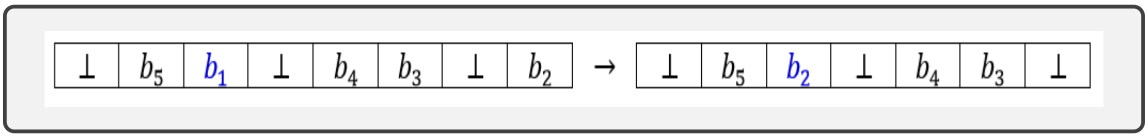


Figure 4: The principle of uniform mutation operator.

4.7 Algorithms' discussion and analysis

We discuss and analyze the algorithms presented in the previous sections (4.6.1 and 4.6.2). It is worth pointing out that the algorithms use the Contract NET Protocol (Smith, 1980b) to coordinate robots actions. It has been formally defined as interaction protocol by Foundation of Intelligent Physical Agents (FIPA) (Wooldridge, 2009). It is frequently used in practical applications. The Contract NET protocol has many advantages over other protocols, such as: find

robots that are the most appropriate for given tasks and it is the only protocol that is accepted as a standard by FIPA.
We start with the convergence and stopping criteria of algorithms. The convergence criterion will guarantee that the algorithm eventually finds the optimal solution in a finite time, e.g. if the mutation rate is 0, then a genetic algorithm may never find the optimal solution. The stopping criterion is a user-specified thing: when do we stop looking for better solutions? e.g. a relative change in objective values between generations or number of generations. We know for sure that the algorithms converge in the case of the exact method, and will find an optimal allocation in a finite time (the time depends on problem scale). However, in the case of heuristic method (GA) there is no guarantee that it will find an optimal solution in finite time (GA do have the advantage of being able to escape local optima through randomness). The algorithms stop when the search space is exhausted in the case of the exact method and after a given number of generations in the case of the heuristic method.
The only three parameters of the algorithms are used in the case of the heuristic method. They are the population size, crossover probability and mutation probability. Their values are chosen according to the most used values in the literature (Siddique and Adeli, 2013).
We have two random parameters in the algorithms (case of the heuristic method). The first one stipulates if two parents are used to produce a new offspring or not. The second one says how often parts of an offspring will be mutated. Their values are chosen randomly, and they have no effects on performance and found results.

5 Simulation and result discussion

Several simulations have been performed to evaluate proposed algorithms and objective functions. Due to a lack of standard testing data on the web, we have randomly generated our own data. We explain the structure of the used testing data in Section 5.1. The adopted evaluation metrics are described in Section 5.2. Found results are discussed in Section 5.3. Finally, a comparative study is given in section 5.4.

5.1 Used datasets

We have developed a software to generate our testing data. Figure 5 shows its graphical interface. The number of sensors, tasks, and bidders are easily regulated. Also, height, width, and altitude of simulation environment can be adjusted, as wished. Finally, simulation duration – hours and minutes – can be personalized. If we clique on the button “generate”, two files are generated. They contain information about tasks and bidders. Tables 7 and 8 show the structure of these files. The code of programs and datasets are available at https://github.com/farouqzitouni/task-allocation-with-datasets.

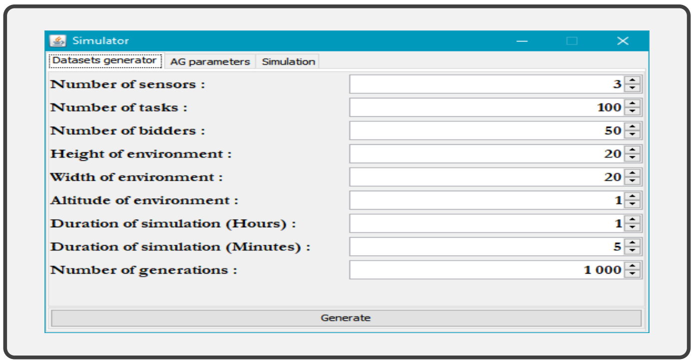


Figure 5: The graphical interface of our datasets' generator.

Table 7: The structure of bidders' file.

Bidder ID	Offered sensors	Costs	Working currents	Location	Physical information
⋮	⋮	⋮	⋮	⋮	⋮

Table 8: The structure of tasks' file.

Task ID	Required senors	Rewards	Working rates	Location	Temporal information
⋮	⋮	⋮	⋮	⋮	⋮

Columns of Table 7 are explained as follows. The first column contains bidders' identifiers. The second column contains their sensors. The third and fourth columns represent sensor costs and their working currents. The fifth column gives their spatial positions. The last column recapitulates their physical properties, such as velocity (v_b), the percentage of energy gauge, mass (m_b), altitude (a_b), battery voltage (U^b), battery capacity (A^b), Peukert's Exponent (η), and C-rate (R^b).
Columns of Table 8 are explained as follows. The first column contains tasks' identifiers. The second column contains sensors they need. The third and fourth columns represent sensor rewards and their working rates. The fifth column gives their spatial positions. The last one recapitulates their temporal properties, such as priority, duration (DUR_t), earliest start date (ES_t), latest start date (LS_t), and latest finish date (LF_t).
In order to validate the proposed solution, ten datasets have been randomly generated using our software. This choice was taken to target the main complexity source of MRTA problems, which is the system size – the number of bidders and tasks. So, it was important to test qualitatively and quantitatively proposed algorithms and objective functions, to observe their capability of handling heavily constrained MRTA problems. Table 9 summarizes used configurations.

Table 9: The information of used datasets.

		Attributes				
		$\|\Omega\|$	$\|T\|$	$\|B\|$	**Environment size**	**Simulation duration**
Datasets	**A**	3	10	5	$20 \times 20 \times 1$	[00:00-00:04]
	B	3	20	10	$20 \times 20 \times 1$	[00:00-00:04]
	C	3	30	15	$20 \times 20 \times 1$	[00:00-00:04]
	D	3	40	20	$20 \times 20 \times 1$	[00:00-00:04]
	E	3	50	25	$20 \times 20 \times 1$	[00:00-00:04]
	F	3	60	30	$20 \times 20 \times 1$	[00:00-00:09]
	G	3	70	35	$20 \times 20 \times 1$	[00:00-00:34]
	H	3	80	40	$20 \times 20 \times 1$	[00:00-00:59]
	I	3	90	45	$20 \times 20 \times 1$	[00:00-00:59]
	J	3	100	50	$20 \times 20 \times 1$	[00:00-00:59]

According to objective functions and datasets, we have 60 different scenarios – 6 objective functions $\times$ 10 datasets. Each one has been run 10 times. In each scenario, both exact and heuristic methods are applied and found results are compared, in terms of the adopted evaluation metrics. All scenarios have two inputs which are the tasks' and bidders' files. The expected output are best allocations that maximize or minimize the considered objective function.

5.2 Evaluation metrics

The evaluation of mobile robot performance and assessing their behaviors in real-life applications are always an open research question (Calisi and Nardi, 2009). Although substantial progress has been brought in terms of standards for the evaluation process. Unfortunately, these frameworks are often related to the considered problem. The following evaluation metrics have been adopted to assess our solution.

1. **Rate of allocated tasks:** ratio between the number of allocated tasks and all tasks.
2. **Total time of allocations:** for all assigned tasks, this indicator represents the total time that the auctioneer takes to compute an allocation to each one.
3. **Fitness values:** for all assigned tasks, this indicator represents the total of objective function values of each allocation.

5.3 Results and analysis

To test the proposed framework and see how effective it is – for solving MRTA problems, we have implemented our own simulator using Java programming language – JADE platform was used to simulate auctioneer and bidders' behaviors. For hardware configuration, all simulations were run on a DELL laptop Intel(R) Core(TM) i3 CPU M 380 @ 2.53GHz 2.53GHz, RAM 3.00 Go, having Windows 10-64 bits operating system. Figures 6, 7, and 8 summarize obtained values of previously exposed evaluation metrics.

Figure 6 shows a comparison between rates of allocated tasks of each objective function, according to allocation methods and used datasets. As a first observation, we can clearly see that exact method has the best allocation rates for datasets $[A, \cdots, G]$ – rate = 100%, but these rates are significantly degraded for the others – i.e. datasets $[H, \cdots, J]$. The main reason for such degradation is the size of datasets, i.e. number of tasks and robots – as the size of datasets increases, rates are increasingly degraded. However, for the heuristic method, it is obvious that rates of allocated tasks are close to each other. Theses verdicts certainly let us decide, according to the confronted situation, what is the best allocation method. In other words, if the size of datasets is small, we better opt for exact method, alternatively, we choose the heuristic one.

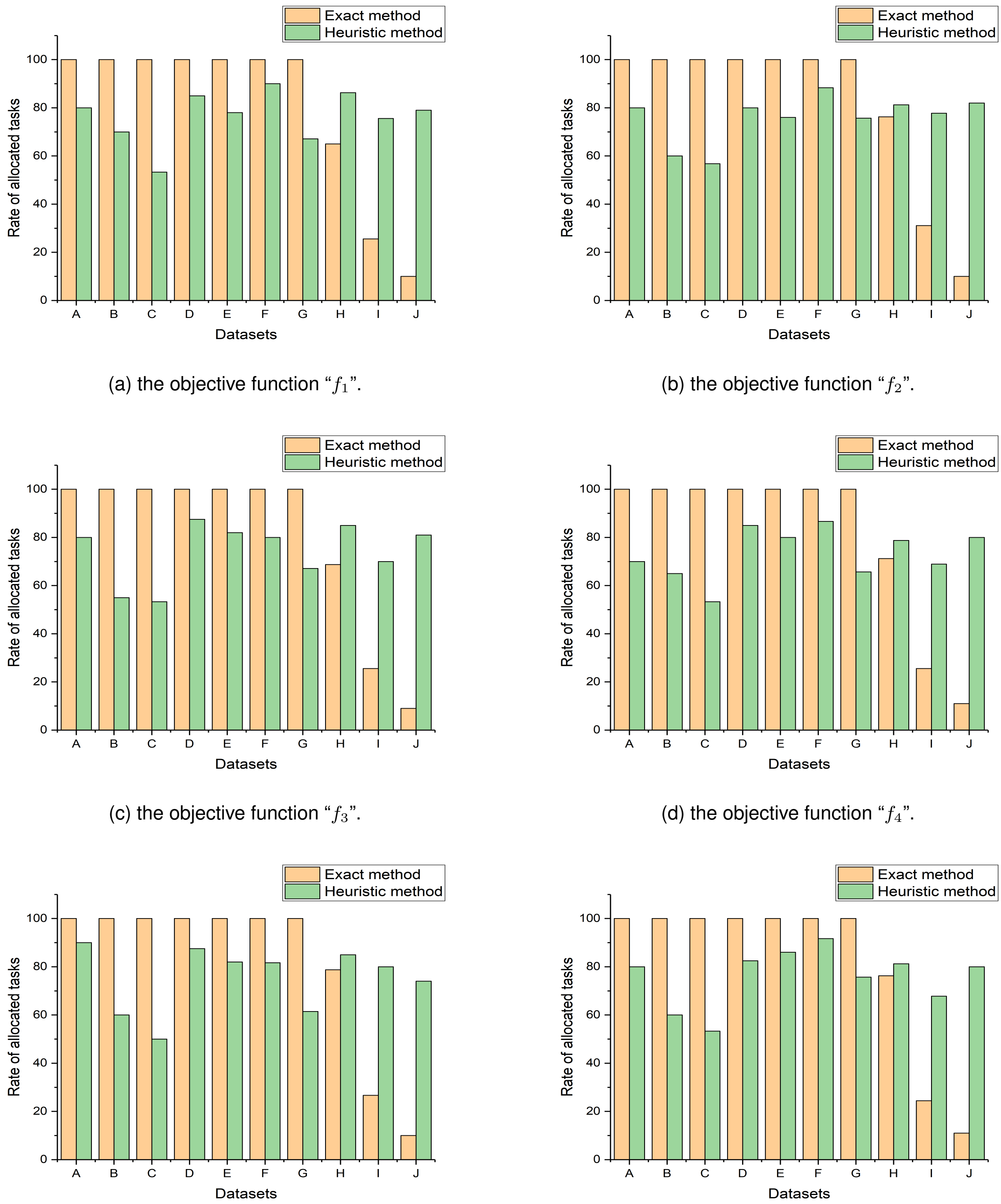
Figure 7 shows a comparison between total time of allocations of each objective function, according to assignment methods and used datasets. As a first observation, we can clearly say that the exact method is relatively effective for datasets $[A, \cdots, E]$ – less than one minute, but total time is significantly increased for the others – i.e. datasets $[F, \cdots, J]$. The main reason for such rapid increase is the size of datasets, i.e. number of tasks and robots – we have a combinatorial explosion. However, for the heuristic method, it is obvious that total time of allocations is nearly constant, even for large datasets – we have a linear trend. Theses verdicts certainly let us choose, according to the confronted situation, what is the best allocation method. In other words, if the size of datasets is small, we better opt for exact method, alternatively, we choose the heuristic one.

Figure 8 shows a comparison between considered fitness values of objective functions, according to allocation methods and used datasets. This graphic can be seen as a compliment to previous ones. It allows us to compare fitness values of considered objective functions, and decide which one is better according to the faced case. As a summary, it is import to note that the size of datasets and the choice of allocation method have a direct impact on system performance. However, in most real-life situations, it is often unavoidable to find a compromise between evaluation metrics presented above. For instance, in Urban Search And Rescue scenarios we can take into consideration two metrics, such as rescue a maximum number of victims with a minimum energy consumptions. In light of previous works, our framework could be used as a decision support system, since it lets us decide – according to the faced situation – what are the quantities to be minimized and the ones to be maximized.

5.4 Comparative study

We evaluate the performance of our methodology, named SOLUTION, by comparing it to the two solutions proposed in (Wei, Hindriks and Jonker, 2016) – named “AUCTION” and “PREDICTION”, respectively. The solutions presented in (Wei et al., 2016) discuss the MRTA problem in the foraging field, where groups of robots search for targets in an environment, i.e. tasks, and then retrieve them back to a home base. The following parameters are taken into account: size of the environment, size of robots’ groups, and initial positions of robots. Table 10 shows the different experimental configurations used to perform the comparison.

We use two environments: small or large. All three solutions were tested with one, five, and ten robots. For deployment of robots, we use two alternatives: all robots are initially in the



(a) the objective function “f_1”.

(b) the objective function “f_2”.

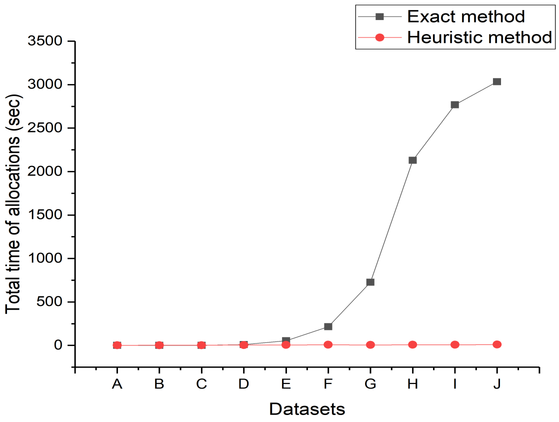
(c) the objective function “f_3”.

(d) the objective function “f_4”.

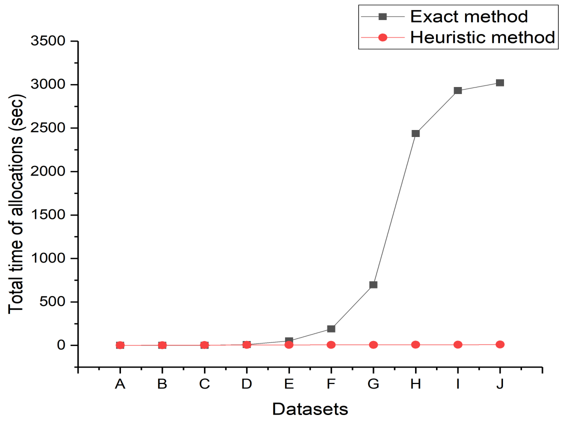
(e) the objective function “f_5”.

(f) the objective function “f_6”.

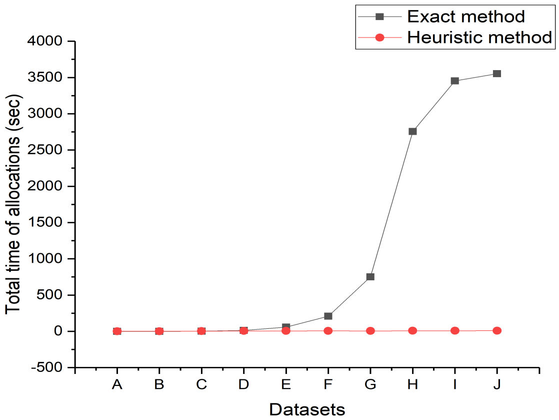
Figure 6: Rates of allocated tasks of each objective function, according to allocation methods and datasets.



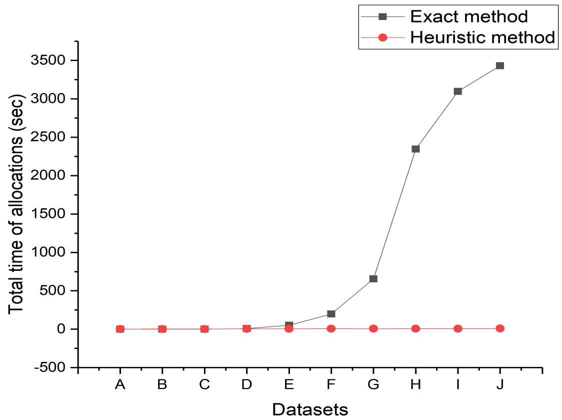
(a) the objective function “f_1”.



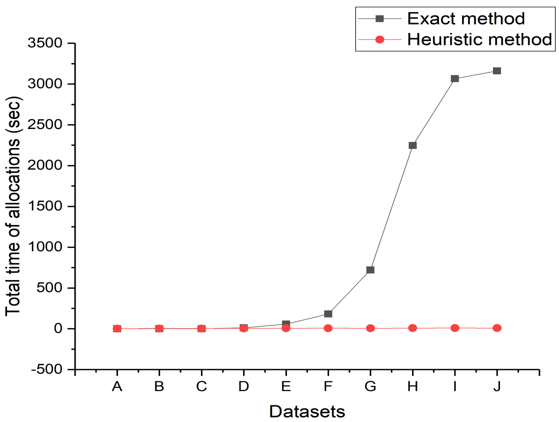
(b) the objective function “f_2”.



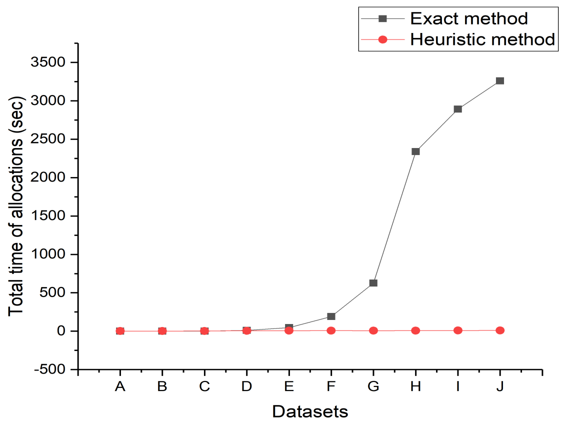
(c) the objective function “f_3”.



(d) the objective function “f_4”.

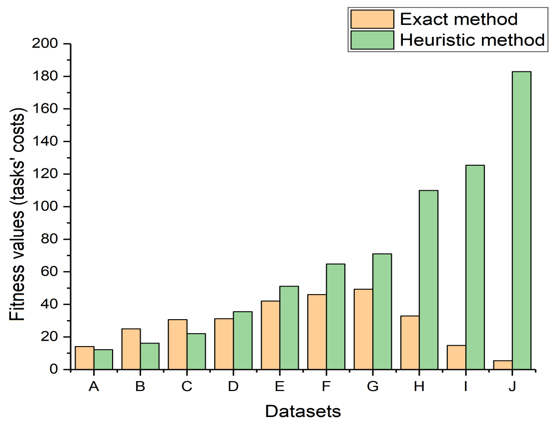


(e) the objective function “f_5”.

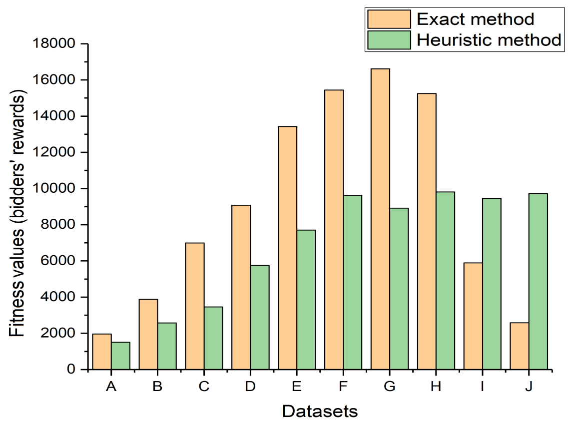


(f) the objective function “f_6”.

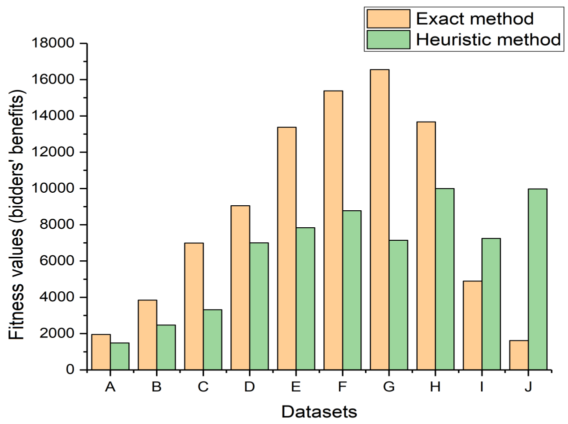
Figure 7: Total time of allocations of each objective function, according to allocation methods and datasets.



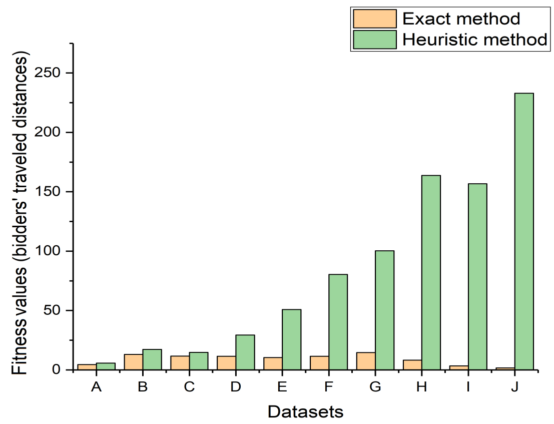
(a) the objective function "f_1".



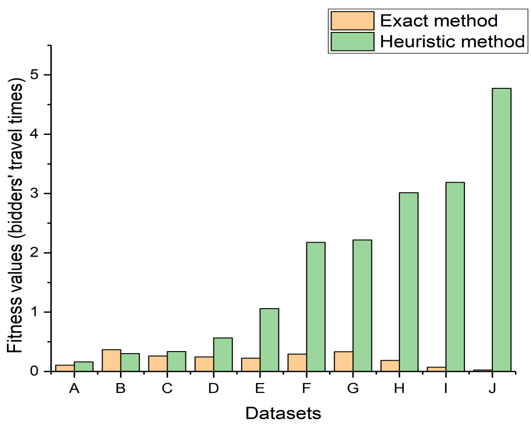
(b) the objective function "f_2".



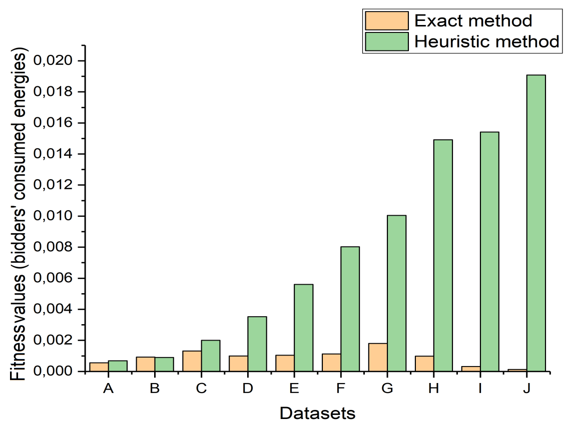
(c) the objective function "f_3".



(d) the objective function "f_4".



(e) the objective function "f_5".



(f) the objective function "f_6".

Figure 8: Fitness values of each objective function, according to allocation methods and datasets.

Table 10: Experimental configurations used for the comparison.

Solutions	Map	Deployment	Robots
[AUCTION, PREDICTION, SOLUTION]	[SMALL, LARGE]	[CLOSE, DISPERSAL]	[1, 5, 10]

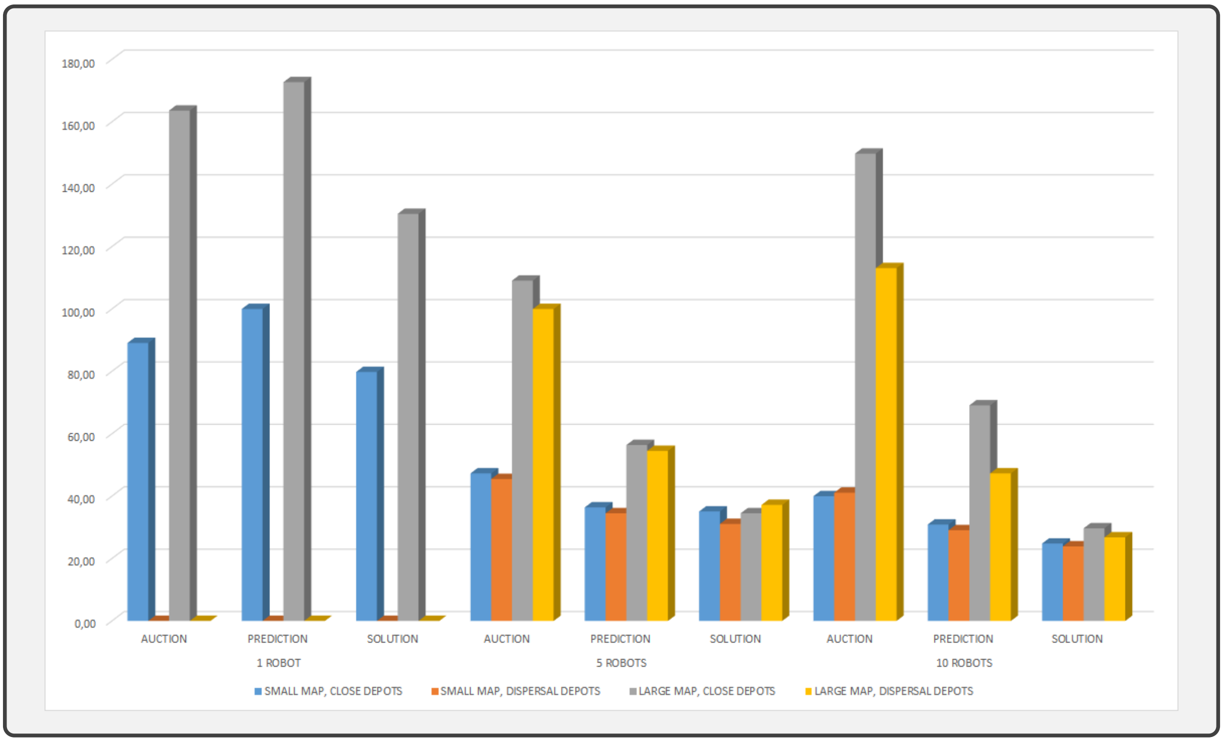


Figure 9: Completion time for the comparative study.

same place, or they are dispersed in the environment. The goal of simulations is to accomplish ten tasks that are in the environment, their locations are set randomly. To compare the three solutions, we use the completion time as an evaluation criterion. Each condition was run 50 times – to reduce the variance and filter the noise effects in our experiments.

Figure 9 shows the results of the comparative study, we compared the two solutions proposed in (Wei et al., 2016), named “AUCTION” and “PREDICTION” respectively, with our solution, named “SOLUTION”.

In the first case, i.e. use of a single robot, we observe that the completion time of the tasks, for the three solutions, is almost the same because no workload is shared – it should be noted that the execution time of agents’ programs may also influence this criterion. However, it is clear that our solution gives the best performance for both possible cases.

In remaining cases, i.e. use of five and ten robots, where several robots are engaged in tasks’ accomplishment, we observe that our solution gives the best completion time, followed by “PREDICTION”, and finally “AUCTION”. This trend becomes nettly visible when the size of the environment and the number of robots increase – in Figure 9, we find that the completion times of “PREDICTION” and “AUCTION” are double and triple of our solution, respectively.

This trend is explained as follows, in the “AUCTION” solution the auctioneer must consider all the robots’ bids in a round before it determines the winners – obviously, the completion

time increases with the number of robots. In contrast, in our solution, the completion time progressively decreases because the number of bids diminishes – robots performing tasks cannot always offer bids on new tasks. Finally, in the "PREDICTION" solution, a robot is not allowed to submit a new bid if it is currently doing a task – this is not really optimal, especially in the case when tasks are critical. In conclusion, our solution is between them – a robot submits a bid if it can, even if it is now carrying out another task. For example, a robot is performing a task that it will finish soon, so it can submit a bid for a new task.
Regarding the initial deployment of robots, we find that it is directly related to the size of the environment and the number of robots, and therefore to the completion time of tasks. As conclusion, the initial layout of robots in the environment is very important in MRTA problems. Finally, one might wonder: what is the connection between the applications and previous theoretical concepts? we give the answer. As we have mentioned above, the MRTA problem is an optimization problem. So, it is important to resolve it in a formal way to ensure finding the best solution. We gave and modified the MRTA formulation in Table 3 to consider energetic, spatial and temporal constraints. Also, we proposed six objective functions that deals with these constraints.

6 Conclusion and perspectives

In this paper, we dealt with the MRTA problem. First, based-auction algorithms are used with two robot types, such as auctioneer and bidders. Also, some objective functions are proposed and extensively tested on some datasets. Both exact and heuristic allocations are implemented and compared considering temporal, spatial, and energetic constraints. Discussion section showed the structure of used datasets, explanation of adopted evaluation metrics, and their critics. Finally, we compared our solution to two well-known solutions (Wei et al., 2016) and demonstrated its superiority in terms of completion time.
In the future, first we plan to improve our framework to be a Decision Support System (DSS) in MRTA field – of course, other evaluation metrics may be added to cover the maximum of fields concerned with MRTA problems. Second, we plan to use real robots to assess the performance of our solution.

References

Allen, J. F. 1983. Maintaining knowledge about temporal intervals, *Communications of the ACM* **26**(11): 832–843.

Aragues, R., Cortes, J. and Sagues, C. 2011. Distributed consensus algorithms for merging feature-based maps with limited communication, *Robotics and Autonomous Systems* **59**(3-4): 163–180.

Atay, N. and Bayazit, B. 2006. Mixed-integer linear programming solution to multi-robot task allocation problem.

Badreldin, M., Hussein, A. and Khamis, A. 2013. A comparative study between optimization and market-based approaches to multi-robot task allocation, *Advances in Artificial Intelligence* **2013**: 11.

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

Borja, F.-G., Jose Manuel, L. G., Ekaitz, Z., Zelmar, E. and Manuel, G. n. 2011. Basic results and experiments on robotic multi-agent system for hose deployment and transportation, *International Journal of Artificial Intelligence* **6**(S 11): 183–202.

Botelho, S. S. D. C. and Alami, R. 1999. M+: A scheme for multi-robot cooperation through negotiated task allocation and achievement, *1999 IEEE International Conference on Robotics and Automation, Marriott Hotel, Renaissance Center, Detroit, Michigan, May 10-15, 1999, Proceedings*, IEEE Robotics and Automation Society, pp. 1234–1239.

Calisi, D. and Nardi, D. 2009. Performance evaluation of pure-motion tasks for mobile robots with respect to world models, *Autonomous Robots* **27**(4): 465.

Campbell, T., Johnson, L. and How, J. P. 2013. Multiagent allocation of markov decision process tasks, *American Control Conference*, pp. 2356–2361.

Chaimowicz, L., Campos, M. and Kumar, V. 2002. Dynamic role assignment for cooperative robots, *Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No.02CH37292)*, Vol. 1, pp. 293–298.

Chapman, A. C., Micillo, R. A., Kota, R. and Jennings, N. R. 2009. Decentralised dynamic task allocation: A practical game: Theoretic approach, *Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems - Volume 2*, AAMAS '09, International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, pp. 915–922.

Chen, W.-H. and Lin, C.-S. 2000. A hybrid heuristic to solve a task allocation problem, *Computers & Operations Research* **27**(3): 287–303.

Coltin, B. and Veloso, M. 18-22 Oct. 2010. Mobile robot task allocation in hybrid wireless sensor networks, *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2932–2937.

Davis, R. I. and Burns, A. 2011. A survey of hard real-time scheduling for multiprocessor systems, *ACM Computing Surveys (CSUR)* **43**(4): 35:1–35:44.

Dechter, R., Meiri, I. and Pearl, J. 1991. Temporal constraint networks, *Artificial Intelligence* **49**(1-3): 61–95.

Dias, B. and Stentz, A. 2002. Opportunistic optimization for market-based multirobot control, *In: International Conference on Intelligent Robots and Systems*, pp. 2714–2720.

Dias, M. B. 2004. *TraderBots: A New Paradigm for Robust and Efficient Multirobot Coordination in Dynamic Environments*, PhD thesis, Carnegie Mellon University, Pittsburgh, PA.

Dias, M. B. and Stentz, A. 2000. A free market architecture for distributed control of a multi-robot system, *In 6th International Conference on Intelligent Autonomous Systems (IAS-6)*, pp. 115–122.

Ding, Y., He, Y. and Jiang, J. 2003. Multi-robot cooperation method based on the ant algorithm, *Swarm Intelligence Symposium, 2003. SIS '03. Proceedings of the 2003 IEEE*, pp. 14–18.

Ding, Y., Zhu, M., He, Y. and Jiang, J. 2006. An autonomous task allocation method of the multi-robot system, *9th International Conference on Control, Automation, Robotics and Vision*, pp. 1–6.

Diwekar, U. M. 2003. *Optimization Under Uncertainty*, Springer US, Boston, MA, pp. 145–208.

Dong, Y. and Zheng-ou, W. 2004. Improved ant algorithm for assignment problem, *Journal of Tianjing University* pp. 373–376.

Du, J., Zhou, L., Qu, P., Shi, Z. and Yang, L. 2010. Task allocation in multi-agent systems with swarm intelligence of social insects, *ICNC*, IEEE, pp. 4322–4326.

Echegoyen, Z., Villaverde, I., Moreno, R., na, M. G. and d'Anjou, A. 2010. Linked multi-component mobile robots: Modeling, simulation and control, *Robotics and Autonomous Systems* **58**(12): 1292–1305. Intelligent Robotics and Neuroscience.

Espina, M. V., Grech, R., De Jager, D., Remagnino, P., Iocchi, L., Marchetti, L., Nardi, D., Monekosso, D., Nicolescu, M. and King, C. 2011. Multi-robot teams for environmental monitoring, *in* P. Remagnino, D. N. Monekosso and L. C. Jain (eds), *Innovations in Defence Support Systems – 3: Intelligent Paradigms in Security*, Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 183–209.

Fang, T. and Parker, L. E. 2005. Asymtre: Automated synthesis of multi-robot task solutions through software reconfiguration, *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, pp. 1501–1508.

Fatima, S. S. and Wooldridge, M. 2001. Adaptive task resources allocation in multi-agent systems, *Proceedings of the Fifth International Conference on Autonomous Agents*, AGENTS '01, ACM, New York, NY, USA, pp. 537–544.

Gelenbe, E. 2009. Analysis of single and networked auctions, *ACM Transactions on Internet Technology (TOIT)* **9**(2): 8:1–8:24.

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

Gerkey, B. P. and Mataric, M. J. 2002. Sold!: auction methods for multirobot coordination, *IEEE Transactions on Robotics and Automation* **18**(5): 758–768.

Ghosn, S. B., Drouby, F. and Harmanani, H. M. 2016. A parallel genetic algorithm for the open-shop scheduling problem using deterministic and random moves, *International Journal of Artificial Intelligence* **14**(1): 130–144.

Gini, M. L. 2017. Multi-robot allocation of tasks with temporal and ordering constraints, *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA.*, pp. 4863–4869.

Haidegger, T., Kovacs, L., Precup, R., Benyo, B., Benyo, Z. and Preitl, S. 2012. Simulation and control for telerobots in space medicine, *Acta Astronautica* **81**: 390–402.

Horst, R. 2002. *Introduction to Global Optimization (Nonconvex Optimization and Its Applications)*, Springer-Verlag New York, Inc., Secaucus, NJ, USA.

Hussein, A., Adel, M., Bakr, M., Shehata, O. M. and Khamis, A. 2014. Multi-robot task allocation for search and rescue missions, *Journal of Physics: Conference Series* **570**(5): 052006.

Jones, E. G., Dias, M. B. and Stentz, A. 2011. Time-extended multi-robot coordination for domains with intra-path constraints, *Autonomous Robots* **30**(1): 41–56.

Juedes, D., Drews, F., Welch, L. and Fleeman, D. 2004. Heuristic resource allocation algorithms for maximizing allowable workload in dynamic, distributed real-time systems, *18th International Parallel and Distributed Processing Symposium, 2004. Proceedings.*, p. 117.

Kalra, N. and Martinoli, A. 2006. Comparative study of market-based and threshold-based task allocation, *in* M. Gini and R. Voyles (eds), *Distributed Autonomous Robotic Systems 7*, Springer Japan, Tokyo, pp. 91–101.

Khamis, A., Hussein, A. and Elmogy, A. 2015. *Multi-robot Task Allocation: A Review of the State-of-the-Art*, Springer International Publishing, Cham, pp. 31–51.

Kmiecik, W., Wojcikowski, M., Koszalka, L. and Kasprzak, A. 2010. Task allocation in mesh connected processors with local search meta-heuristic algorithms, *Intelligent Information and Database Systems*, Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 215–224.

Kong, Y., Zhang, M. and Ye, D. 2014. A group task allocation strategy in open and dynamic grid environments, *ANAC@AAMAS*, Vol. 638 of *Studies in Computational Intelligence*, Springer, pp. 121–139.

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

Lenagh, W., Dasgupta, P. and Munoz-Melendez, A. 2015. A spatial queuing-based algorithm for multi-robot task allocation, *Robotics* **4**(3): 316–340.

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

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

Liu, D. K. and Kulatunga, A. K. 2007. *Simultaneous Planning and Scheduling for Multi-Autonomous Vehicles*, Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 437–464.

Liu, S.-H., Zhang, Y., Wu, H.-Y. and Liu, J. 2010. Multi-robot task allocation based on swarm intelligence, **40**: 123–129.

Liu, S., Sun, T. and cheng Hung, C. 2011. Multi-robot task allocation based on swarm intelligence.

Liu, W., Winfield, A. F. T., Sa, J., Chen, J. and Dou, L. 2007. Towards energy optimization: Emergent task allocation in a swarm of foraging robots, *Adaptive Behavior* **15**(3): 289–305.

Manathara, J. G., Sujit, P. B. and Beard, R. W. 2011. Multiple uav coalitions for a search and prosecute mission, *Journal of Intelligent & Robotic Systems* **62**(1): 125–158.

Marino, A., Parker, L. E., Antonelli, G. and Caccavale, F. 2013. 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* **71**(3): 423–444.

Michalak, T., Sroka, J., Rahwan, T., Wooldridge, M., McBurney, P. and Jennings, N. R. 2010. A distributed algorithm for anytime coalition structure generation, *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems: Volume 1 - Volume 1*, AAMAS '10, International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, pp. 1007–1014.

Mosteo, A. 2010. Multi-robot task allocation for service robotics: from unlimited to limited connection range.

Mosteo, A. and Montano, L. 2006. Simulated annealing for multi-robot hierarchical task allocation with flexible constraints and objective functions.

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

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

Nedjah, N., de Mendona, R. M. and de Macedo Mourelle, L. 2015. Pso-based distributed algorithm for dynamic task allocation in a robotic swarm, *Procedia Computer Science* **51**: 326–335. International Conference On Computational Science, ICCS 2015.

Nunes, E. and Gini, M. 2015. Multi-robot auctions for allocation of tasks with temporal constraints, *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, AAAI'15, pp. 2110–2216.

Nunes, E., Manner, M., Mitiche, H. and Gini, M. 2017. A taxonomy for task allocation problems with temporal and ordering constraints, *Robotics and Autonomous Systems* **90**(C): 55–70.

Oscar, C., Francisco, H. and Frank, H. 2001. *Genetic Fuzzy Systems: Evolutionary Tuning And Learning Of Fuzzy Knowledge Bases*, Advances In Fuzzy Systems-applications And Theory, World Scientific Publishing Company.

Parker, J., Nunes, E., Godoy, J. and Gini, M. 2016. Exploiting spatial locality and heterogeneity of agents for search and rescue teamwork, *Journal of Field Robotics* **33**(7): 877–900.

Parker, L. E. 1998. Alliance: an architecture for fault tolerant multirobot cooperation, *IEEE Transactions on Robotics and Automation* **14**(2): 220–240.

Parker, L. E. and Tang, F. 2006. Building multirobot coalitions through automated task solution synthesis, *Proceedings of the IEEE* **94**: 1289–1305.

Petcu, A. and Faltings, B. 2005. Superstabilizing, fault-containing distributed combinatorial optimization, *AAAI*, AAAI Press / The MIT Press, pp. 449–454.

Ramchurn, S. D., Polukarov, M., Farinelli, A., Truong, C. and Jennings, N. R. 2010. Coalition formation with spatial and temporal constraints, *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems: Volume 3 - Volume 3*, AAMAS '10, International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, pp. 1181–1188.

Sandholm, T., Larson, K., Andersson, M., Shehory, O. and Tohmé, F. 1999. Coalition structure generation with worst case guarantees, *Artificial Intelligence* **111**(1-2): 209–238.

Sariel, S. and Balch, T. 2006. *A Distributed Multi-robot Cooperation Framework for Real Time Task Achievement*, Springer, pp. 187–196.

Scerri, P., Farinelli, A., Okamoto, S. and Tambe, M. 2005. Allocating tasks in extreme teams, *Proceedings of the Fourth International Joint Conference on Autonomous Agents and Multiagent Systems*, AAMAS '05, ACM, New York, NY, USA, pp. 727–734.

Shea, P. J., Alexander, K. and Peterson, J. 2003. Group tracking using genetic algorithms, *Sixth International Conference of Information Fusion, 2003. Proceedings of the*, Vol. 1, pp. 680–687.

Shehory, O. and Kraus, S. 1998. Methods for task allocation via agent coalition formation, *Artificial Intelligence* **101**(1-2): 165–200.

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

Shkurti, F., Xu, A., Meghjani, M., Higuera, J. C. G., y. Girdhar, Giguere, P., Dey, B. B., Li, J., Kalmbach, A., Prahacs, C., Turgeon, K., Rekleitis, I. and Dudek, G. 2012. Multi-domain monitoring of marine environments using a heterogeneous robot team, *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Portugal, pp. 1447–1753.

Siddique, N. and Adeli, H. 2013. *Computational Intelligence: Synergies of Fuzzy Logic, Neural Networks and Evolutionary Computing*, 1st edn, Wiley Publishing.

Smith, R. G. 1980a. The contract net protocol: High-level communication and control in a distributed problem solver.

Smith, R. G. 1980b. The contract net protocol: High-level communication and control in a distributed problem solver, *IEEE Transactions on Computers* **C-29**(12): 1104–1113.

Spall, J. C. 2003. *Introduction to Stochastic Search and Optimization*, 1 edn, John Wiley & Sons, Inc., New York, NY, USA.

Tang, F. and Parker, L. E. 2007. A complete methodology for generating multi-robot task solutions using asymtre-d and market-based task allocation, *Proceedings 2007 IEEE International Conference on Robotics and Automation* pp. 3351–3358.

Wang, J.-P., Gu, Y. and Li, X.-M. 2012. Multi-robot task allocation based on ant colony algorithm, *Journal of Computers* **7**(9): 2160–2167.

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

Werger, B. B. and Mataric, M. J. 2000. Broadcast of local eligibility: Behavior-based control for strongly cooperative robot teams, *Proceedings of the Fourth International Conference on Autonomous Agents*, AGENTS '00, ACM, New York, NY, USA, pp. 21–22.

Wooldridge, M. 2009. *An Introduction to MultiAgent Systems*, 2nd edn, Wiley Publishing.

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

Zhang, D., Xie, G., Yu, J. and Wang, L. 2007. Adaptive task assignment for multiple mobile robots via swarm intelligence approach, *Robotics and Autonomous Systems* **55**(7): 572–588.

Zhang, Y., Liu, S., Fu, S. and Wu, D. 2009. A quantum-inspired ant colony optimization for robot coalition formation, *Chinese Control and Decision Conference*, pp. 626–631.

Zhang, Y. and Parker, L. 2013a. Iq-asymtre: Forming executable coalitions for tightly coupled multirobot tasks, *IEEE Transactions on Robotics* **29**(2): 400–416.

Zhang, Y. and Parker, L. E. 2013b. Multi-robot task scheduling, *IEEE International Conference on Robotics and Automation*, pp. 2992–2998.

Zhang, Y., Parker, L. E. and Kambhampati, S. 2014. Coalition coordination for tightly coupled multirobot tasks with sensor constraints, *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1090–1097.

Zheng, X. and Koenig, S. 2008. Reaction functions for task allocation to cooperative agents, *AAMAS (2)*, IFAAMAS, pp. 559–566.

Zitouni, F. and Maamri, R. 2016. *An adaptive protocol for dynamic allocation of tasks in a multi-robot system*, PhD thesis.

Zitouni, F. and Maamri, R. 2017. Cooperative learning-agents for task allocation problem, *Interactive Mobile Communication Technologies and Learning - Proceedings of the 11th IMCL Conference, 30 November - 1 December 2017, Mediterranean Palace Hotel, Thessaloniki, Greece*, pp. 952–968.

Zitouni, F. and Maamri, R. 2018. Fa-setpower-mrta:a solution for solving the multi-robot task allocation problem, *Computational Intelligence and Its Applications*, Springer-Verlag LNCS IFIP Advances in Information and Communication Technology (IFIP AICT).

Zlot, R. M. 2006. *An Auction-based Approach to Complex Task Allocation for Multirobot Teams*, PhD thesis, Pittsburgh, PA, USA. AAI3250901.

Zlot, R. M., Stentz, A. T., Dias, M. B. and Thayer, S. 2002. Multi-robot exploration controlled by a market economy, *in* IEEE (ed.), *IEEE International Conference on Robotics and Automation*, Vol. 3, pp. 3016–3023.

Zlot, R. and Stentz, A. 2006. Market-based multirobot coordination for complex tasks, *The International Journal of Robotics Research* **25**(1): 73–101.