

FA-SETPOWER-MRTA: A Solution for Solving the Multi-Robot Task Allocation Problem

Farouq Zitouni, Ramdane Maamri

▶ **To cite this version:**

Farouq Zitouni, Ramdane Maamri. FA-SETPOWER-MRTA: A Solution for Solving the Multi-Robot Task Allocation Problem. 6th IFIP International Conference on Computational Intelligence and Its Applications (CIIA), May 2018, Oran, Algeria. pp.317-328, 10.1007/978-3-319-89743-1_28 . hal-01913866

HAL Id: hal-01913866
https://inria.hal.science/hal-01913866

Submitted on 6 Nov 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

FA-SETPOWER-MRTA
A solution for Solving the Multi-Robot Task Allocation Problem

ZITOUNI Farouq and MAAMRI Ramdane

Kasdi Merbah University, Ouargla — Algeria
Constantine 2 — Abdelhamid Mehri University, Constantine — Algeria
{farouq.zitouni,ramdane.maamri}@univ-constantine2.dz

Abstract. The Multi-Robot Task Allocation problem (MRTA) is the situation where we have a set of tasks and robots; then we must decide the assignments between robots and tasks in order to optimize a certain measure (e.g. allocate the maximum number of tasks). We present an effective solution to resolve this problem by implementing a two-stage methodology: first a global allocation that uses the Firefly Algorithm (FA), next a local allocation that uses the set theory properties (Power Set algorithm). Finally, results of the different simulations show that our solution is efficient in terms of the rate of allocated tasks and the calculated allocations are locally optimal.

Keywords: multi-robot systems, task allocation, firefly algorithm, set theory properties, power set algorithm.

1 Introduction

Nowadays, Multi-Robot Systems (MRS) are receiving a great attention and become omnipresent in our daily life. Consequently, intensive researches have been conducted on them to validate their applicability and adequacy with different real-life issues. These systems have several advantages that allow them solving easily various problems, such as: industrial applications, surveillance, target tracking and rescue missions [1].

It should be noted that the design of MRS must respect the different interaction and coordination aspects between their components (e.g. robots), otherwise it may produce non-deterministic systems with reduced performances [2]. Thus, one of the coordination issues that researchers must consider is the Task Allocation (TA) problem, which is intuitively the process of assigning a set of tasks to some robots [3]. In MRS, the TA problem can be expressed by the following expression: given a set of robots and tasks, how to select the appropriate robots to perform the desired tasks? so as to achieve in a cooperative manner the overall objective of the considered system. Therefore, the goal here is to coordinate the robot behaviors and find an optimal way for the allocation of different tasks [4].

Mainly, we can find two taxonomies for the categorization of MRTA problems. In fact, the main objective of these taxonomies is to propose a classification

of different TA configurations in categories, so that the different problems encountered in the real-life can be inserted into one of them (i.a. a mathematical formulation is generally given).

Authors of [5] have categorized the TA problem by proposing a classification which is articulated around three axes. The first axis distinguishes robots according to their abilities to perform a single task or probably several tasks at a time {single-task robots (ST), multi-task robots (MT)}. The second axis distinguishes tasks according to their needs to be executed by a single robot or several robots at the same time {single-robot tasks (SR), multi-robot tasks (MR)}. The third axis distinguishes between tasks that require an instantaneous assignment without considering future allocations and tasks that require assignments that consider both current and future allocations {instantaneous assignment (IA), time-extended assignment (TA)}. Finally, despite its good coverage of the majority of encountered MRTA problems; however, the authors of this taxonomy state that it does not capture problems with interrelated utilities and constraints on tasks [5], hence the need to propose a taxonomy that addresses these limitations.

To remedy the limitations of the previous taxonomy, the authors of [6] have taken and modified it by adding a higher level. In reality, this level expresses interdependence degrees of utilities between robots and tasks. Thus, this taxonomy is a hierarchy that has two levels, where in the second one we find the different classes proposed in the previous taxonomy. On the other hand, in the first level we find four interdependence degrees, i.e. No Dependencies (ND), In-Schedule Dependencies (ID), Cross-Schedule Dependencies (XD), Complex Dependencies (CD).

The rest of the paper is organized as follows. First, we give, in the section 2, an overview of the related works already done to address the MRTA problem. Then, we present, in the section 3, the solution that we have proposed to address this problem and explain its different algorithms. After, we simulate, in the section 4, our solution and discuss the obtained results. Finally, we give, in the section 5, a conclusion and some perspectives.

2 Related works

In this section, we will give some solutions that have been proposed in the literature to address the MRTA problem. More specifically, these solutions will be categorized according to the first level of the taxonomy 2, that is to say these approaches will be divided into four categories. Moreover, the list of approaches presented bellow is not exhaustive, but shows the most frequently encountered in the literature [4].

In the first family, we can clearly see that several approaches have been identified in the literature [5] to address more specifically ND[ST-SR-IA] problems. For example, we can cite the paper [7] that uses potential fields and the papers [8, 9] that use auction-based methods.

In the second family, we can find several works devoted to the coordination in MRS. Generally, these approaches focus on ID[ST-SR-TA] problems. For example, we can cite the work exposed in [10] that uses the Traveling Salesman Problem (TSP) and multiple TSP to model the TA problem. Also, the paper [11] provides a good solution for the routing problem in MRS using mixed linear programming. Finally, auction-based approaches [12–15] have been widely used to resolve this problem, because of their distributed nature which is well suited to MRS.

In the third family, we can easily identify lot of developed works. For example, we find the system M+ [16] which addresses the problem of instantaneous allocation of tasks using a market system with supporting precedence constraints. Also, we cite the work presented in [17] that proposes a solution manipulating constraints between tasks and adopting a market-economy approach. In addition, the work exposed in [18] addresses the problem of routing, where some robot teams must accomplish a set of scientific missions. Besides, we can cite the work presented in the paper [19], which exposes a solution for the MRTA problem by using quantum genetic algorithms and reinforcement learning. Also, in the paper [20], the authors manipulate the case of a simple task scheduling using the heuristic: the task x must be executed n second(s) before the beginning of the task y. Another way to consider the TA problem is to bring it to a coalition formation problem [21, 22]. Accordingly, the authors of the paper [23] have taken the works presented in the last two papers and improved the proposed algorithms by minimizing communications and imposing constraints on agents capabilities. Likewise, the authors of the papers [24, 25] use auctions and coalitions to address the TA problem. On the other hand, a Framework has been proposed in [26] that imposes the constraint of shared resources, e.g. communication mediums and processors. Finally, the papers presented in [27, 28] address the TA problem, where robots must consider current tasks and future allocations.

In the fourth family, we can unfortunately make a short list of done works. In fact, this is due to the high difficulty of these problems and the lack of mathematical formulations [6]. However, we can cite the work exposed in [29] which addresses the TA problem, with time-extended assignment, by managing an environment hit by a natural disaster. In addition, we can also mention the work presented in [30] which exposes a task allocation approach using coalitions (this approach has two versions: centralized version or ASyMTRe and distributed version or ASyMTRe-D). Finally, we mention the work given in [31], which is an improved version of [12], that manipulates the time-extended assignments.

3 Proposed solution

In this section, an auction-based solution (named FA-POWERSET-MRTA for Firefly Algorithm-Power Set algorithm-Multi Robot Task Allocation) is presented to address the MRTA problem. In fact, auction-based approaches have been widely used because of the multiple advantages they offer [32]. In this kind

of approaches, we consider an auctioneer, a set of bidders and a set of goods. In a MRTA problems, a particular robot is the auctioneer, the rest of robots are the bidders and tasks are the goods. In our solution, we use the Contract Net Protocol (CNP) [33] for exchanging messages between the auctioneer and bidders.

Algorithm 1: behavior of the auctioneer a

Input : the set of bidders B and the set of unallocated tasks T.
Output: allocations between bidders and tasks.

```
1  while (a is active) do
2      PeriodicBehaviour {n_1 seconds}
3          if (T ≠ ∅) then
4              foreach b_i ∈ B do
5                  sendMessageTo(b_i,CHOOSE-TASK,T);
6              end
7          end
8      end
9      CyclicBehaviour
10         message ← receiveMessageFrom(b_i);
11         if (message.subject = CHOSEN-TASK) then
12             foreach t ∈ T do
13                 if (t is feasible) then
14                     compute the allocation;
15                     T ← T/{t};
16                     inform concerned bidders;
17                 end
18             end
19         end
20     end
21 end
```

3.1 Problem configuration and assumptions

In our methodology (i.e. solution), we use a particular robot called auctioneer and several robots called bidders. The auctioneer must first communicate the tasks (i.e. goods) to be allocated to the different bidders and receive their offers; then it must decide the best assignments between robots and tasks; finally, it notifies bidders of its results. Although the use of an auctioneer can be seen as a bottleneck for the system (e.g. if it breaks down then the system will be too); in fact, this limitation can be omitted, because it decreases considerably the communication rates and maintains an overall view of different allocations. Also, the auction-based methods have the advantage of sharing computations on the different bidders of the system (scalability), which would maintain the assignment process even if a bidder breaks down (robustness) [32].

Furthermore, the bidders calculate their utilities for the considered tasks according to the different information they have. So, the overall cost function

(i.e. fitness) is divided into sub-functions, which are estimated in a decentralized and independent way by bidders. In our solution, the allocation of a given task means the presence of the resources it requires. In other words, a given task is allocated if and only if the resources it requires are all offered. On the other hand, these resources are offered by bidders, i.a generally these resources are sensors and actuators. In our solution, we use confusedly the term skill to represent these sensors and actuators.

Algorithm 2: behavior of a bidder b_i

```
   Input  : S_i,C_i,v_i,e_i,a_i and p_i.
   Output: the chosen task and its costs.
 1 while (b_i is active) do
 2 |  PeriodicBehaviour {n_2 seconds}
 3 |  |  if (T ≠ ∅) then
 4 |  |  |  if (state = AVAILABLE) then
 5 |  |  |  |  for k ← 1 to 1000 do
 6 |  |  |  |  |  foreach t ∈ T do
 7 |  |  |  |  |  |  move the bidder b_i towards the task t;
 8 |  |  |  |  |  end
 9 |  |  |  |  end
10 |  |  |  |  t ← choose the closest task to the bidder b_i;
11 |  |  |  |  U ← estimate the costs of the bidder b_i for the task t;
12 |  |  |  |  sendMessageTo(a,CHOSEN-TASK,t,U);
13 |  |  |  else
14 |  |  |  |  sendMessageTo(a,CHOSEN-TASK,⊥,⊥);
15 |  |  |  end
16 |  |  |  T ← ∅;
17 |  |  end
18 |  end
19 |  CyclicBehaviour
20 |  |  message ← receiveMessageFrom(a);
21 |  |  if (message.subject = CHOOSE-TASK) then
22 |  |  |  T ← message.content;
23 |  |  end
24 |  end
25 end
```

Now we consider the following assumptions, we assume the class of ST-MR-IA problems, that is to say that each robot can execute only one task at the same time, the tasks might be executed by several robots at the same time and the allocation of tasks to robots is instantaneous. Moreover, we assume two non-empty sets of bidders B and tasks T. Also, we consider the both cases where all the tasks are previously known or gradually inserted into the system. Accordingly, each bidder $b_i \in B$ is defined by six variables $\langle S_i, C_i, v_i, e_i, a_i, p_i \rangle$, where S_i is a vector representing the skills that has the bidder b_i, C_i is a vector representing the skill costs of the bidder b_i and the rest of variables represent respectively

the speed, energy level, aging factor and spatial position of the bidder b_i. In a similar way, each task $t_j \in T$ is defined by three variables $\langle S_j, d_j, p_j \rangle$, where S_j is a vector representing the skills required by the task t_j and the rest of variables represent respectively the execution duration and spatial position of the task t_j. Now, we present and explain the proposed algorithms to address the MRTA problem.

3.2 Proposed algorithms

The algorithms 1 and 2 show respectively the behaviors adopted by the auctioneer and bidders to address the TA problem. In fact, we can clearly see that these algorithms have two sub-behavior types that we have named "PeriodicBehavior" and "CyclicBehavior", where in the first one we have put the instructions that are executed in a periodic manner (i.e. every n seconds) and in the second one we have put the instructions that are executed immediately after every message receiving.

Now, we will explain the behaviors (i.e. instructions) of previous algorithms. To do this, we can resume their working principle in the following three sections.

A. Task announcement

Firstly, the task allocation process begins when the auctioneer announces that there are tasks that must be allocated (i.e. "PeriodicBehavior" of the algorithm 1, on the page 4) . To do this, the auctioneer broadcasts a message, i.e. to all bidders, which contains the list of tasks to be allocated, their required skills and spatial locations.

B. Global allocation

Secondly, when the bidders receive a message from the auctioneer (i.e. "CyclicBehavior" of the algorithm 2, on the page 5), each one must choose a task to perform, calculate its costs and answer the auctioneer (i.e. "PeriodicBehavior" of the algorithm 2, on the page 5).

To choose a task, one bidder executes the firefly algorithm [34, 35] on all the received tasks and converges gradually towards one of them. For the convergence of a bidder to a given task, we have used the equation adopted by the firefly algorithm, which is:

$$p_r = p_r + \beta_0 e^{-\gamma (d_{rt})^2} (p_t - p_r) + \alpha (rand - 0.5) \tag{1}$$

Where p_r and p_t are the respective locations of the bidder r and a task t, β_0 is a regularization parameter and its value is always 1, the term $\beta_0 e^{-\gamma (d_{rt})^2} (p_t - p_r)$ is called the attraction degree of the robot r by the task t, the parameter γ characterizes the attraction variation and its value is very important for the convergence speed of the algorithm and the behavior of the robot r ($\gamma \in [0.01, 100]$), the parameter α represents physically the environment noise and its value affects

the visibility of a task by a robot ($\alpha \in [0,1]$), $rand \in [0,1]$ is a random variable and the d_{rt} represents the weighted distance between the robot r and the task t and it is given by the following equation:

$$d_{rt} = \delta \|p_t - p_r\| + (1-\delta) H(S_t, S_r) \tag{2}$$

Where the term $\|p_t - p_r\|$ represents the Euclidean distance between the locations of the robot r and the task t, the term $H(S_t, S_r)$ represents the Hamming distance between the skill vectors of the robot r and the task t and the parameter δ makes the balance between the two distances.

Once a task has been selected, one bidder must now estimate its costs for each offered skill to the chosen task. To do this, we have proposed the following equation:

$$u^{rt}_{i \in \Delta_{rt}} = [\delta c^r_i + (1-\delta)(\frac{v_r}{\|p_t - p_r\|})^{\frac{e_r}{a_r}}] \times |\Delta_{rt}| \tag{3}$$

Where the term Δ_{rt} represents the common skills between the robot r and the task t (i.e. the skills that the robot r offers to the task t), again the parameter δ make the balance between the two costs and the other variables are explained above.

Finally, the bidder sends its result to the auctioneer. It should be noted that the symbol $\perp$ means that the bidder has sent an empty result to the auctioneer (i.e. it is not able to perform any task: busy or do not have the required skills).

C. Local allocation

Thirdly, when all the responses of bidders are received, now the auctioneer should estimate an allocation for each selected task ("CyclicBehavior" of the algorithm 1, on the page 4). To do this, first the auctioneer browses the list of the chosen tasks and considers for each one the set of all skills offered by the bidders, then it calculates all the useful subparts (the power set algorithm) of this set of skills (i.e. the concept of a useful subpart will be explained in the following example), finally the fitness of each useful subpart is estimated and the one with the smallest fitness value will be considered as an allocation for the considered task. The fitness value of a given subpart is the sum of cost values of its bidder skills. At the end of this step, the bidders concerned by the calculated allocations are notified.

4 Simulation and result discussion

In this section, we simulate the proposed algorithms on some numerical data and we evaluate the found results. Moreover, we assume that all the tasks are known in advance; however, our algorithms can also handle the case where tasks are inserted gradually in the system.

4.1 Generation of simulation data

In fact, the numerical data that we have used for our simulations are generated randomly, on a 2D grid of 100×100 cells, and the number of used skills is 10. To simplify things, we suppose that the speed, energy level and aging factor values are constant for all bidders.

Besides, these simulation data are divided into three different data sets named "dataset 1", "dataset 2" and "dataset 3". First, in the "dataset 1" there is from 10 to 50 bidders (bidders are incremented by 10) and 100 tasks (this dataset is created to see the effect of the number of used bidders on allocation performances). Second, in the "dataset 2" there is from 10 to 100 tasks (tasks are incremented by 10) and 10 bidders (this dataset is created to see the effect of the number of used tasks on allocation performances). Thirdly, in the "dataset 3" there is also from 10 to 100 tasks (the tasks are incremented by 10) and 50 bidders (similarly this dataset is created to see the effect of the number of used tasks on allocation performances).

4.2 Results and analysis

In the figure 1 (on the page 9), we present in the left sub-figure the allocation times needed to allocate all tasks and in the right sub-figure the fitness values found for the "dataset 1". For this experimentation, we remember that we have used 100 tasks to be allocated and the number of bidders varies from 10 to 50.

As a first observation, we can clearly notice that the number of used bidders does not greatly improve the allocation time of all tasks (approximately we have an allocation time of 20,199 seconds, whatever the number of used bidders); therefore, the increase in the number of bidders for the allocation of all tasks is not imperative in our proposed solution, hence its power to allocate the tasks with a small number of bidders.

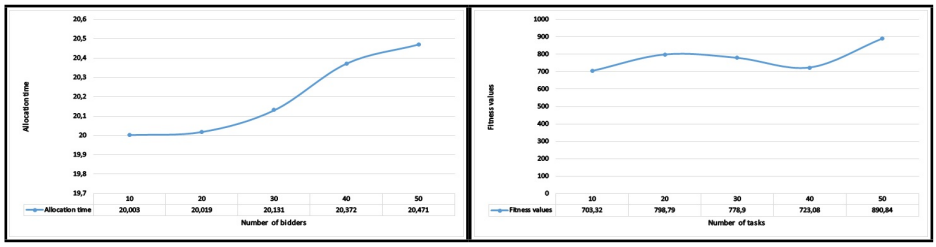
In the figure 2 (on the page 10), we present in the left sub-figure the allocation times needed to allocate all tasks and in the right sub-figure the fitness values found for the "dataset 2" and "dataset 3". For this experimentation, we remember that we have used respectively 10 bidders ("dataset 2") and 50 bidders ("dataset 3") and the number of tasks to be allocated varies from 10 to 100 in both datasets.

In fact, we can clearly notice again that the number of bidders does not greatly improve the allocation time, whatever the number of tasks to be allocated (with 10 or 50 bidders we have approximately the same times). However, the allocation times increase almost linearly with the increase in the number of tasks, which is quite natural.

It should be noted that the allocation times shown in the figures 1 (on the page 9) and 2 (on the page 10) encompass all the possible actions for the allocation of tasks, i.e. sending and receiving messages between the auctioneer and bidders, selection of tasks by bidders (global allocation) and allocation of tasks by the auctioneer (local allocation) and the periodicity of the different behaviors.

Now, we graphically show the impact of the number of used bidders on the quantity of allocated tasks for each cycle. To do this, in the figure 3 (on the page 10) we present the approximate number of allocated tasks by iteration, according to the number of used bidders. For this experimentation, we specify that we have used 100 tasks to be allocated and the number of bidders varies from 10 to 50 (from the left to the right).

Finally, we can obviously notice that all considered tasks are allocated to bidders, whatever the used configuration; therefore, the proposed algorithms are efficient in terms of the rate of allocated tasks. On the other hand, we can also say and affirm that the increase in the number of used bidders only enhances the number of allocated tasks by iteration but does not minimize the allocation times.

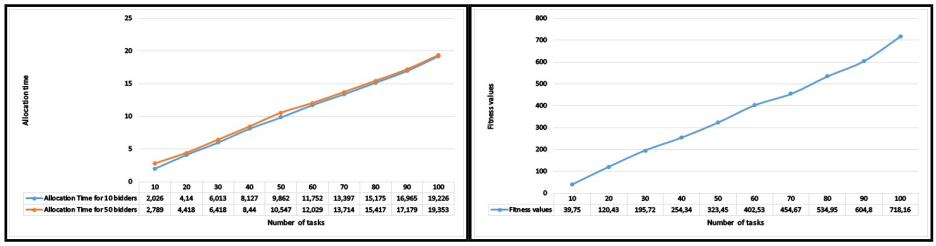


(a) allocation times. (b) fitness values.

Fig. 1. measures for the "dataset 1".

5 Conclusion and perspectives

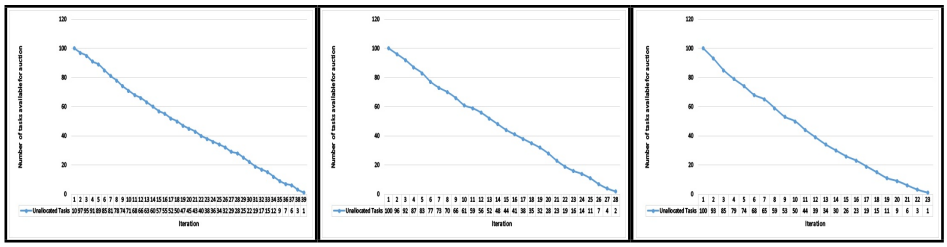
In this paper, we have exposed a solution to address the MRTA problem. In fact, this solution uses auctions with two robot types, which are: auctioneer (with a single instance) and bidder (with several instances). Also, this solution exploits two allocation types, which are: global done by bidders and local done the auctioneer. Finally, the discussion section shows that the proposed solution is effective in terms of the rate of allocated tasks (the rate of allocated tasks is 100%, whatever the used configuration) and it supports both cases where the tasks are all known in advance or inserted dynamically in the system. As perspectives, we expect to improve our solution by imposing temporal and spatial constraints on tasks to make it more general.



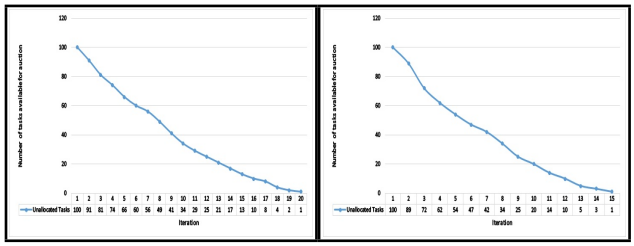
(a) allocation times for "dataset 2" and "dataset 3".

(b) fitness values the "dataset 2".

Fig. 2. measures for the "dataset 2" and "dataset 3".



(a) number of bidders = 10. (b) number of bidders = 20. (c) number of bidders = 30.



(d) number of bidders = 40. (e) number of bidders = 50.

Fig. 3. number of allocated tasks for each iteration according to the number of used bidders

References

1. H. Ahmed, A. Mohamed, B. Mohamed, S. Omar, and K. Alaa, "Multi-robot task allocation for search and rescue missions," *Journal of Physics: Conference Series*, vol. 570, no. 5, p. 052006, 2014.

2. H. Ahmed and K. Alaa, "Market-based approach to multi-robot task allocation," *International Conference on Individual and Collective Behaviors in Robotics (ICBR)*, pp. 69–74, 2013.
3. H. Yanyan, L. Deshi, C. Jian, Y. Xiangguo, H. Yuxi, and Z. Guangmin, "A multi-robots task allocation algorithm based on relevance and ability with group collaboration," *International Journal of Intelligent Engineering and Systems*, vol. 3, no. 2, pp. 33–41, 2010.
4. K. Alaa, H. Ahmed, and E. Ahmed, "Multi-robot task allocation: a review of the state-of-the-art," *Springer International Publishing : Cooperative Robots and Sensor Networks*, pp. 31–51, 2015.
5. B. P. Gerkey and M. J. Mataric, "A formal analysis and taxonomy of task allocation in multi-robot systems," *The International Journal of Robotics Research*, vol. 23, no. 9, pp. 939–954, 2004.
6. A. G. Korsah, A. Stentz, and B. M. Dias, "A comprehensive yaxonomy for multi-robot task allocation," *The International Journal of Robotics Research*, vol. 32, no. 12, pp. 1495–1512, 2013.
7. D. Vail and M. Veloso, "Multi-robot dynamic role assignment and coordination through shared potential fields," *Multi-Robot Systems, Kluwer*, 2003.
8. B. P. Gerkey and M. J. Mataric, "Sold!: auction methods for multirobot coordination," *IEEE Transactions on Robotics and Automation*, vol. 18, no. 5, pp. 758–768, 2002.
9. R. G. Simmons, D. Apfelbaum, W. Burgard, D. Fox, M. Moors, S. Thrun, and H. L. S. Younes, "Coordination for multi-robot exploration and mapping," *Proceedings of the Seventeenth National Conference on Artificial Intelligence and Twelfth Conference on Innovative Applications of Artificial Intelligence*, pp. 852–858, 2000.
10. C. Barnhart, E. L. Johnson, G. L. Nemhauser, M. W. P. Savelsbergh, and P. H. Vance, "Branch-and-price: column generation for solving huge integer programs," *Operations Research*, vol. 46, no. 3, pp. 316–329, 1998.
11. J. Melvin, P. Keskinocak, S. Koenig, C. A. Tovey, and B. Y. Ozkaya, "Multi-robot routing with rewards and disjoint time windows," *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2332–2337, 2007.
12. B. M. Dias, "Traderbots: a new paradigm for robust and efficient multirobot coordination in dynamic environments," *Carnegie Mellon University*, 2004.
13. M. Berhault, H. Huang, P. Keskinocak, S. Koenig, W. Elmaghraby, P. Griffin, and A. Kleywegt, "Robot exploration with combinatorial auctions," *IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 2, pp. 1957–1962, 2003.
14. S. Koenig, C. A. Tovey, X. Zheng, and I. Sungur, "Sequential bundle-bid single-sale auction algorithms for decentralized control," *Proceedings of the 20th International Joint Conference on Artificial Intelligence*, pp. 1359–1365, 2007.
15. M. G. Lagoudakis, E. Markakis, D. Kempe, P. Keskinocak, A. Kleywegt, S. Koenig, C. Tovey, A. Meyerson, and S. Jain, "Auction-based multi-robot routing," *Proceedings of Robotics: Science and Systems*, 2005.
16. S. Silva, C. Botelho, and R. Alami, "M+: a scheme for multi-robot cooperation through negotiated task allocation and achievement," *IEEE International Conference on Robotics and Automation*, pp. 1234–1239, 1999.
17. D. C. Mackenzie, "Collaborative tasking of tightly constrained multi-robot missions," *In Multi-Robot Systems: From Swarms to Intelligent Automata: Proceedings of the 2003 International Workshop on Multi-Robot Systems*, vol. 2, pp. 39–50, 2003.

18. S. Chien, A. Barrett, T. Estlin, and G. Rabideau, "A comparison of coordinated planning methods for cooperating rovers," *Proceedings of the Fourth International Conference on Autonomous Agents*, pp. 100–101, 2000.
19. F. Zitouni and M. Ramdane., "Cooperative learning-agents for task allocation problem," *Proceedings of the International Conference on Interactive Mobile Communication, Technologies and Learning*, 2017.
20. T. Lemaire, R. Alami, and S. Lacroix, "A distributed tasks allocation scheme in multi-uav context," *IEEE International Conference on Robotics and Automation*, vol. 4, pp. 3622–3627, 2004.
21. O. Shehory and S. Kraus, "Task allocation via coalition formation among autonomous agents," *Proceedings of the 14th International Joint Conference on Artificial Intelligence*, vol. 1, pp. 655–661, 1995.
22. O. Shehory and S. Kraus, "Methods for task allocation via agent coalition formation," *Artificial Intelligence*, vol. 101, no. 1, pp. 165–200, 1998.
23. L. Vig and J. A. Adams, "Multi-robot coalition formation," *IEEE Transactions on Robotics*, vol. 22, no. 4, pp. 637–649, 2006.
24. G. Jose and O. Gabriel, "Multi-robot task allocation strategies using auction-like mechanisms," *Artificial Research and Development in Frontiers in Artificial Intelligence and Application*, vol. 100, pp. 111–122, 2003.
25. L. Lin and Z. Zheng, "Combinatorial bids based multi-robot task allocation method," *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, pp. 1145–1150, 2005.
26. P. M. Shiroma and M. F. M. Campos, "Comutar: a framework for multi-robot coordination and task allocation," *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 4817–4824, 2009.
27. M. Koes, I. Nourbakhsh, and K. Sycara, "Constraint optimization coordination architecture for search and rescue robotics," *Proceedings 2006 IEEE International Conference on Robotics and Automation*, pp. 3977–3982, 2006.
28. A. G. Korsah, "Exploring bounded optimal coordination for heterogeneous teams with cross-schedule dependencies," *Carnegie Mellon University*, 2011.
29. G. E. Jones, B. M. Dias, and A. Stentz, "Time-extended multi-robot coordination for domains with intra-path constraints," *Autonomous Robots*, vol. 30, no. 1, pp. 41–56, 2011.
30. L. E. Parker and F. Tang, "Building multirobot coalitions through automated task solution synthesis," vol. 94, no. 7, pp. 1289–1305, 2006.
31. R. M. Zlot, "An auction-based approach to complex task allocation for multirobot teams," *Carnegie Mellon University*, 2006.
32. F. Tang and L. E. Parker, "A complete methodology for generating multi-robot task solutions using asymtre-d and market-based task allocation," *IEEE International Conference on Robotics and Automation*, 2007.
33. R. Smith, "Communication and control in problem solver," *IEEE Transactions on Computers*, 1980.
34. X.-S. Yang, "Firefly algorithm: nature-inspired metaheuristic algorithms," *Luniver Press*, 2008.
35. X.-S. Yang, "Firefly algorithms for multimodal optimization," *Proceedings of the 5th International Conference on Stochastic Algorithms: Foundations and Applications*, vol. 5792, pp. 169–178, 2009.