Task Allocation of Multi-robot Coalition Formation

Chuang Huang, Hao Zhang(✉), and Zhuping Wang

College of Electronic and Information Engineering, Tongji University,
Shanghai 200092, People's Republic of China
zhang_hao@tongji.edu.cn

Abstract. Due to the lack of capability of a single robot, multiple robots need to form coalition to implement tasks. This paper proposes a coalition formation task allocation approach for a case where multiple robots are set off to perform multiple tasks with resource constraints. Since the robots taken into consideration are heterogeneous, the resources carried by each robot are distinct. For a task, there are robot-level constraint and coalition-level constraint that needs to be satisfied by the corresponding task coalition. Task-priority which indicates the urgency of completing the tasks is also considered before allocating the tasks. In the calculation of utility, not only the contribution of resources, but also the path cost of robots executing tasks is considered in this paper. In the above approach, the coalition will be selected in a greedy way to perform the task. The simulation results verify the validity of the proposed approach.

Keywords: Task allocation · Coalition formation · Resource constraints · Task-priority

1 Introduction

In recent years, multi-robot task allocation problem, namely MRTA, has attracted extensive attention. MRTA problem can be regarded as an optimal allocation problem. Its goal is to optimally allocate a group of robots to a group of tasks, so as to optimize the performance of the whole system under a set of constraints. Common MRTA methods include market-based methods and optimization based methods. The market-based approach is based on contract net protocol and uses the way of auction to allocate tasks. [1] proposes two decentralized auction algorithms CBAA and CBBA to coordinate a fleet of autonomous vehicles. A stochastic clustering auction is applied to address the problem of optimal task allocation for heterogeneous teams in [2]. The method based on optimization focuses on finding the optimal solution from a set of feasible solutions. For example, [3] proposes PSO-based distributed algorithm for dynamic task allocation. [14] presents an ACO-based heuristic algorithm for solving the constraint task allocation problem.

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2023
Z. Ren et al. (Eds.): *Proceedings of 2021 5th Chinese Conference on Swarm Intelligence and Cooperative Control*, LNEE 934, pp. 221–230, 2023.
https://doi.org/10.1007/978-981-19-3998-3_22

According to the taxonomy for MRTA proposed by [4], the coalition formation problem belong to ST-MR-IA type. Since it needs to assign multiple single-task robots to execute multi-robot task, it is mathematically equivalent to the set partition problem [5] and it is a NP-hard problem. [6] points out the limitations of multi-agent coalition formation algorithm in practical robot system, and then proposes a multi-robot coalition formation algorithm applied in practical robot system. [7] concludes and analyses market-based multi robot coordination approaches. [8] proposes a dynamical formed heterogeneous robot teams approach to execute coordinated tasks based on market-based allocation scheme.

The above coalition formation approach does not consider the resource constraints of actual robots and tasks. [9] proposes a leader-follower based coalition task allocation methodology for a group of heterogeneous mobile robots with resource constraints. In [10] and [15], they considers resource constraints of UAVs and tasks and a two-stage coalition formation algorithm is proposed for multiple UAVs to perform multiple tasks. Quantum Multi-Objective Particle Swarm Optimization (QMOPSO) is proposed in [11] to address task allocation in large-scale natural disasters. A hedonic coalition formation game model is used to adapt to wireless communication networks in [12].

Based on the sequential coalition algorithm proposed in [13], this paper considers the priority and path cost of executing tasks while considering resource constraints.

2 Modeling of Resource Constrained Multi-robot System

In the multi-robot system, we assume that there are N robots and M tasks. Each robot is denoted by $R(i)$, where $i = 1, 2, ..., N$ is the index of robots and N is the total number of robots. Each task is denoted by $T(j)$, where $j = 1, 2, ..., M$ is the index of tasks and M is the total number of tasks.

2.1 Resource Modeling

In multi robot system, resource capability is denoted by c_k where $k = 1, 2, ..., K$ is the index of resources and K is the total number of resources. c_k is consisted of several elements which can contribute to different capability to accomplish an action. A set S_k that contains all elements of c_k is denoted by $S_k = \{(s_l)_k | (s_l)_k > 0, l = 1, 2, ..., L_k\}$, where l is the index of different elements, L_k is the total number of elements in c_k. $(sl)_k$ is the maximum value of the lth element.

2.2 Robot Modeling

Each robot carries n different types of resource with limited quantity which can be denoted by $R_i =< r_{i1}, r_{i2}, ..., r_{in} >$, where R_i is the resource capability vector of the i^{th} robot. $r_{ik} (k = 1, 2, ..., n)$ is the k^{th} resource which can be denoted by $r_{ik} =< r_{ik}^1, r_{ik}^2, ..., r_{ik}^{L_k} >$, where $r_{ik}^m (m = 1, 2, ..., L_k)$ is the m^{th} element of resource c_k.

2.3 Task Modeling

There are two types of task constraints: robot-level constraints and coalition-level constraints. Robot-level constraints are the constraints that each robot individual needs to meet. Coalition-level constraints are the constraints that the alliance as a whole needs to meet.

Robot-level constraints can be denoted by $T_j^A = <t_{j1}^a, t_{j2}^a, ..., t_{jn}^a>$,where T_j^A is resource capability vector by the robot-level constraints of the j^{th} task. $t_{jk}^a(k = 1, 2, ..., n)$ is the k^{th} resource which can be denoted by $t_{jk}^a = <t_{jk}^{a1}, t_{jk}^{a2}, ..., t_{jk}^{aL_k}>$, where $t_{jk}^{am}(m = 1, 2, ..., L_k)$ is the m^{th} element of resource c_k. $t_{jk}^{am}(m = 1, 2, ..., L_k)$ is normalized between 0 and 1.

Coalition-level constraints on number of resource can be denoted by $N_j^C = <n_{j1}^c, n_{j2}^c, ..., n_{jn}^c>$, where N_j^C is resource number vector by the coalition-level constraints of the j^{th} task. $n_{jk}^c \geq 0(k = 1, 2, ..., n)$ is the number of the k^{th} resource.

Coalition-level constraints on elements of resource can be denoted by $T_j^C = <t_{j1}^c, t_{j2}^c, ..., t_{jn}^c>$, where T_j^C is resource capability vector by the coalition-level constraints of the j^{th} task. $t_{jk}^c(k = 1, 2, ..., n)$ is the k^{th} resource which can be denoted by $t_{jk}^c = <t_{jk}^{c1}, t_{jk}^{c2}, ..., t_{jk}^{cL_k}>$, where $t_{jk}^{cm}(m = 1, 2, ..., L_k)$ is the m^{th} element of resource c_k. $t_{jk}^{cm}(m = 1, 2, ..., L_k)$ is normalized between 0 and 1.

2.4 Priority Modeling

When the robots execute multiple tasks, the order of tasks execution needs to be considered. Different order of task execution will lead to different utility and task completion. In this paper, priority is used to define the order of task execution. The priority of tasks can be denoted by the following formula:

$$Pr(j) = \omega_1 \times Re(j) + \omega_2 \times \sum_{p=1}^{n} \sum_{q=1}^{L_k} t_{jp}^{aq}, \tag{1}$$

where $Pr(j)$ is the priority of the j^{th} task. ω_1 and ω_2 are two weight parameters and their sum is 1. $Re(j)$ is the j^{th} task's reward value, which indicates the benefits of performing tasks $T(j)$. $\sum_{p=1}^{n} \sum_{q=1}^{L_k} t_{jp}^{aq}$ is the sum of resources required for each task. The greater the $Pr(j)$, the more prior the task is to execute.

2.5 Utility Modeling

We define the number of useful resources required by N_j^C in the calculation of utility as n_{ik}, which can be denoted by the following formula:

$$n_{ik} = \begin{cases} \mathrm{min(1, n_{jk}^c)} & \mathrm{if\ r_{ik} \geq t_{jk}^c} \\ 0 & \mathrm{otherwise} \end{cases}$$

$n_{ik} = 1$ if and only if $r_{ik} \geq t_{jk}^c$ and $n_{jk}^c \geq 1$. Otherwise, $n_{ik} = 0$.

In this paper, we define the utility of the mth functional element of the resource c_k on robot $R(i)$ as $u(r_{ik}^{m})$, which can be calculated as follows:

$$u(r_{ik}^{m}) = (1 - e^{-r_{ik}^{m}}) \times Re(j) \tag{2}$$

Then the profit of robot $R(i)$ performing task $T(j)$ is defined as p_j. p_j is calculated as follows:

$$p_j = \sum_{k=1}^{n} \{\lambda_{jk} \frac{\min(\mathrm{n_{jk}^{c}}, \mathrm{n_{ik}})}{L_k} \sum_{m=1}^{L_k} u(r_{ik}^{m})\} \times Re(j), \tag{3}$$

where $\lambda_{jk} \in [0, 1]$ denotes a task-specified weight parameter for measuring the relative importance of different resources in performing $T(j)$. λ_{jk} does not necessarily sum to 1.

We also consider the cost of completing a task in this paper. The cost of a task can be calculated as follows:

$$c_j = S_{ij} \times R(i)_{fuel}, \tag{4}$$

where S_{ij} is the Euclidean distance between robot $R(i)$ and task $T(j)$. $R(i)_{fuel}$ is the fuel consumption of robot $R(i)$.

According to the formula (3) and (4), the utility of robot $R(i)$ performing task $T(j)$ can be defined as follows:

$$u_i(t_j) = p_j - c_j. \tag{5}$$

Let $C(j)$ denotes the robot coalition of task $T(j)$. Hence, the objective function is proposed as follows:

$$\mathrm{Maximize} : \sum_{\mathrm{j=1}}^{\mathrm{M}} \mathrm{u^{c}(t_j)} \tag{6}$$

Subject to:

$$\sum_{i=1}^{C(j)} N_i(j) \geq N_j^{C}, \tag{7}$$

where $u^c(t_j)$ is the utility of robot coalition $C(j)$ for performing task $T(j)$.

3 Coalition Formation Algorithm

To solve the multi-robots task allocation with resource constraints, we consider the task priority before allocating tasks and propose the coalition formation algorithm. The purpose of the coalition formation algorithm is to form a robot coalition to perform tasks and maximize the total utility. The steps of coalition formation algorithm is as follows.

3.1 Determining Task Priority (Line 2)

*A.1)*The centralized sever collects all task's robot-level constraints T_j^A and calculates the priority $P_r(j)$ of the j^{th} task according to formula (1).

*A.2)*Then the centralized sever issues tasks to the robots in descending order of priority.

Algorithm 1: Coalition Formation Algorithm

```
   Input: Robots' resource capability vector $R_i$;
          Tasks'robot-level constraints $T_j^A$;
          Tasks'coalition-level constraints $N_j^C$, $T_j^C$;
          Robots' and Tasks' location information;
          Robots'speed $R(i)_{fuel}$;
          weight parament $\lambda_{jk}$;
1  Initialize: $r_{max} = 0; u_{max} = 0; N_{max} = 0$;
2  Calculate the task priority $P_r(j)$;
3  for $i = 1$ to $N$ do
4      if (robotfree(i) && $R_i \geq T_j^A$) is satisfied then
5          qualified robot ← $R(i)$;
6      else
7          disqualified robot ← $R(i)$;
8          $i = i + 1$;
9  Repeat:
10 for $i = 1$ to $N$ do
11     if ($R(i) \in$ qualified robot) is satisfied then
12         for $k = 1$ to $K$ do
13             if $r_{ik} \geq t_{jk}^c$ then
14                 Update $n_{ik}$, $u_i(t_j)$;
15                 $k = k + 1$;
16         if $\hat{u}_i(t_j) > u_{max}$ then
17             Update $r_{max}$, $u_{max}$, $N_{max}$;
18     $i = i + 1$;
19 if $u_{max} == 0$ then
20     return failure;
21 else
22     Update $C(j)$, $u^c(t_j)$, $N_{p+1}^c$;
23     disqualified robot ← $r_{max}$;
24     if $N_{p+1}^c == 0$ then
25         Return $C(j)$, $u^c(t_j)$;
26     else
27         $P = P + 1$;
   Output: Robots coalition for each task.
```

3.2 Selecting Qualified Robots (Lines 3–8)

B.1) When tasks are distributed, the first step is to find qualified robots. The **qualified robot** is defined when it can meet $robotfree(i)$ && $R_i \geq T_j^A$, where $robotfree(i)$ is a set including all robots that do not participate in any task. In the initial stage, all robots belong to this set. $R_i \geq T_j^A$ means that the robot can meet robot-level constraints.

B.2) Otherwise, what does not satisfy the above formula belongs to disqualified robot. The disqualified robot can not participate in the execution of any task.

3.3 Finding the Robots with the Highest Utility (Lines 10–18)

C.1) The key of the coalition formation algorithm is to find the robots with the highest utility in a greedy manner. When the number of cycles P is less than the number of robots N, enter the cycle and initialize the parameters r_{max}, u_{max}, $N_{max}.r_{max} = 0, u_{max} = 0, N_{max} = 0$, where r_{max} is the robot with the highest utility. u_{max} is the highest utility of robot in coalition. N_{max} is the number of necessary resource of r_{max}. The three parameters will be updated in *C.4)*.

C.2) If it satisfies the formula $r_{ik} \geq t_{jk}^c$, where it proves that the k^{th} resource of qualified robot meets the coalition-level constraints and the number of necessary resources and the utility of the robot will be updated in *C.3)*.

C.3) Next step is to update the number of necessary resources and the utility of the robot i. $\hat{n}_{ik} = \mathrm{min}(\mathrm{n_{ik}(t_j)}, (\mathrm{n_{jk}^c})_\mathrm{P})$, $\hat{u}_i(t_j) = \sum_{k=1}^{n} \hat{n}_{ik} u_{ik}(t_j)$, where $\hat{n}_{ik}$ is the number of the k^{th} updated resource. $(n_{jk}^c)_P$ is the number of the k^{th} resource on coalition-level constraints for the P^{th} cycle. $\hat{u}_i(t_j)$ is the updated utility of $R(i)$ for task $T(j)$.

C.4) When $\hat{u}_i(t_j) > u_{max}$, update the parameters r_{max}, u_{max}, N_{max}. $r_{max} = r_i$, $u_{max} = \hat{u}_i(t_j)$, $N_{max}(k) = \hat{n}_{ik}$.

3.4 Forming the Coalition of Robots (Lines 22–23)

D.1) After finding the robots with the highest utility, it needs to add the r_{max} to the coalition $C(j)$. The utility u_{max} of the r_{max} will be added to $u^c(t_j)$. The number of necessary resources of r_{max} will be deleted from the coalition-level constraints N_p^c. $C(j) = C(j) + r_{max}$, $u^c(t_j) = u^c(t_j) + u_{max}$, $N_{p+1}^c = N_p^c - N_{max}$.

D.2) Once the r_{max} joins the coalition $C(j)$, it means that the r_{max} will no longer participate in the execution of other tasks. Therefore, the robot will be removed from the set of **qualified robot** to the set of **disqualified robot**.

3.5 Controlling Cycle Conditions (Lines 24–30)

The cycle will continue until $N_{p+1}^c = 0$, then judge the coalition whether meets resource distribution constraint. If it satisfies, then return the coalition $C(j)$ and its utility $u^c(t_j)$.

4 Simulations

In this section, we will assign 3 different tasks to 10 heterogeneous robots to verify the effectiveness of the priority based coalition formation algorithm. Table 1 shows the tasks' resources constraints in this simulation.

Table 1. Tasks' resources constraints

		c_1	c_2	c_3	c_4	c_5	c_6	c_7
t_1	T_1^A	0	0	0	$0, 0$	$0.8, 0.8$	$0.6, 0.6, 0.5$	$0, 0, 0$
	N_1^C	1	1	1	2	0	0	0
	T_1^C	0.5	0.6	0.5	$0.5, 0.3$	$0, 0$	$0, 0, 0$	$0, 0, 0$
t_2	T_2^A	0	0	0	$0, 0$	$0.8, 1$	$0, 0, 0$	$0, 0, 0$
	N_2^C	0	0	0	0	0	2	0
	T_2^C	0	0	0	$0, 0$	$0, 0$	$0.8, 0.8, 0.8$	$0, 0, 0$
t_3	T_3^A	0	0	0	$0, 0$	$0, 0$	$0, 0, 0$	$0.8, 1, 1$
	N_3^C	1	1	2	0	0	2	0
	T_3^C	0.5	0.6	0.5	$0, 0$	$0, 0$	$0.8, 0.8, 0.5$	$0, 0, 0$

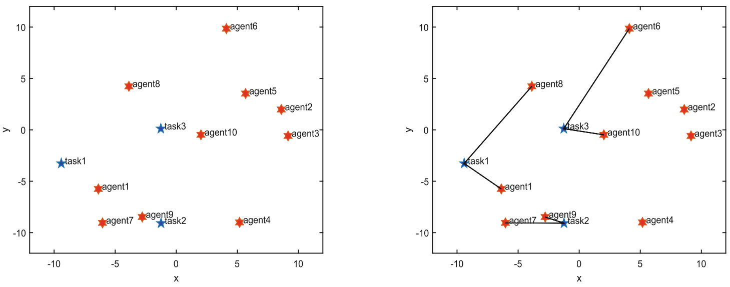
Table 2 shows the 10 robots' resources in the simulation.

Table 2. Robot resources

		c_1	c_2	c_3	c_4	c_5	c_6	c_7
r_1	R_1	0.5	0	0.5	$0.5, 0.3$	$0.8, 1$	$0.8, 0.8, 0.5$	$1, 1, 1$
r_2	R_2	1	0.6	0	$0, 0$	$0.8, 1$	$0.8, 0.8, 0.8$	$0, 0, 0$
r_3	R_3	0	0.6	0.6	$0.5, 0.5$	$0.8, 1$	$1, 1, 0.8$	$1, 1, 1$
r_4	R_4	0	1	0.5	$0.5, 0.8$	$0.8, 1$	$0.8, 1, 0.8$	$0.8, 1, 1$
r_5	R_5	0.5	0	0	$1, 1$	$0.8, 0.8$	$0.8, 0.8, 0.5$	$1, 1, 1$
r_6	R_6	0.5	0.6	0.5	$0, 0$	$1, 1$	$0.8, 0.8, 0.5$	$1, 1, 1$
r_7	R_7	0	0.8	1	$0.3, 0.5$	$0.8, 1$	$1, 1, 0.8$	$1, 1, 1$
r_8	R_8	1	1	0.5	$0.5, 0.3$	$1, 1$	$0.8, 0.8, 0.8$	$0.8, 1, 1$
r_9	R_9	0	0	0.5	$1, 0.3$	$0.8, 1$	$1, 1, 1$	$0.8, 1, 1$
r_{10}	R_{10}	0.5	0	1	$0.5, 0.3$	$0.8, 0.8$	$0.8, 0.8, 1$	$0.8, 1, 1$

In the simulation, task's reward value $Re(j)$ is listed as follows: $Re(1) = 100$, $Re(2) = 150$, $Re(3) = 200$. To decide the priority of the three tasks in the simulation, we set the weight parameters $\omega_1 = 0.3$, $\omega_2 = 0.7$. According to the formula (1), the value of the three tasks' priority $Pr(j)$ is calculated as follows: $Pr(1) = 32.31$, $Pr(2) = 46.26$, $Pr(3) = 61.96$. The weight parameter for measuring the relative importance of different resources in performing $T(j)$ is listed as follows: $\lambda_{1k} = [0.9; 0.9; 0.8; 1; 0; 0; 0]$, $\lambda_{2k} = [0; 0; 0; 0; 1; 0; 0]$, $\lambda_{3k} = [0.9; 0.9; 0.8; 0; 1; 0; 0]$.

In the simulation, the positions of robots and tasks are randomly distributed, the fuel consumption of each robot is 3. The three tasks will be assigned to the robots according to the value of the task's priority.



(a)

(b)

Fig. 1. Simulation result. (a) Initial distribution of robots and tasks. (b) The result of task allocation.

Figure 1 shows the result of the above simulation. Figure 1(a) presents the initial distribution of robots and tasks in a plane area. The blue five pointed star represents the task, and the red six pointed star represents the robot or agent. Figure 1(b) shows the result of coalition formation task allocation. The straight line with an arrow between the task and the robot represents that the corresponding robot performs the task along the path.

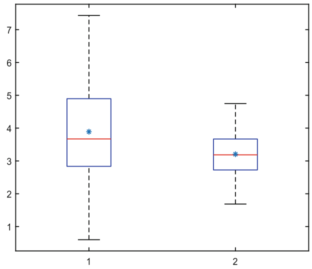


Fig. 2. Comparison of task completion time

Figure 2 shows the average task completion time of 100 repeated tests under the two methods. On the left is the average task completion time of the coalition formation algorithm considering only resource constraints, and on the right is the average task completion time of the coalition formation algorithm based on priority and considering path cost proposed in this paper. It is obvious that the task completion time of the proposed method is less than that of the previous algorithm, which can improve the efficiency of task completion.

5 Conclusions

This paper proposes a coalition formation algorithm with task priority and considers path cost in the calculation of utility to address MRTA problem. Since the robots forming the coalition are selected in a sequential greedy way, this method can optimally solve the multi robot task allocation problem and ensure the highest overall utility. Simulation result shows that compared with the method considering only resource constraints, the execution time of the task is effectively shortened after considering the path cost, so the efficiency of task completion can be improved.

Acknowledgement. This work was supported by the National Natural Science Foundation of China (61922063, 61773289), Shanghai Shuguang Project (18SG18), Shanghai Natural Science Foundation (19ZR1461400), Shanghai Hong Kong Macao Taiwan Science and Technology Cooperation Project 21550760900, Shanghai Municipal Science and Technology Major Project (2021SHZDZX0100) and the Fundamental Research Funds for the Central Universities.

References

1. Choi, H.-L., Brunet, L., How, J.P.: Consensus-based decentralized auctions for robust task allocation. IEEE Trans. Robot. **25**(4), 912–926 (2009). https://doi.org/10.1109/TRO.2009.2022423
2. Zhang, K., Collins, E.G., Shi, D.: Centralized and distributed task allocation in multi-robot teams via a stochastic clustering auction. ACM Trans. Auton. Adapt. Syst. **7**(2), 1–22 (2012). https://doi.org/10.1145/2240166.2240171
3. Choi, H.-L., Brunet, L., How, J.P.: PSO-based distributed algorithm for dynamic task allocation in a robotic swarm. Procedia Comput. Sci. **51**, 326–335 (2015). https://doi.org/10.1016/j.procs.2015.05.250
4. Gerkey, B.P., Mataric, M.J.: A formal analysis and taxonomy of task allocation in multi-robot systems. Int. J. Robot. Res. **23**, 939–954 (2004). https://doi.org/10.1177/0278364904045564
5. Shehory, O., Kraus, S.: Methods for task allocation via agent coalition formation. Artif. Intell. **101**(1), 165–200 (1998). https://doi.org/10.1177/0278364904045564
6. Vig, L., Adams, J.A.: Multi-robot coalition formation. IEEE Trans. Robot. **22**(4), 637–649 (2006). https://doi.org/10.1109/TRO.2006.878948
7. Dias, M.B., Zlot, R., Kalra, N., Stentz, A.: Market-based multirobot coordination: a survey and analysis. Proce. IEEE **94**(7), 1257–1270 (2006). https://doi.org/10.1109/JPROC.2006.876939
8. Jones, E.G., Browning, B., Dias, M.B., Argall, B., Veloso, M., Stentz, A.: Dynamically formed heterogeneous robot teams performing tightly-coordinated tasks. In: Proceedings 2006 IEEE International Conference on Robotics and Automation, pp. 570–575. IEEE Press, Orlando (2006). https://doi.org/10.1109/ROBOT.2006.1641771

9. Chen, J., Sun, D.: Resource constrained multirobot task allocation based on leader follower coalition methodology. Int. J. Robot. Res. **30**(12), 1423–1434 (2011). https://doi.org/10.1177/0278364910396552
10. Tang, B., Zhu, Z., Shin, H., Tsourdos, A.: Task-priority based task allocation of multiple UAVs with resource constraint. In: 2015 23rd Mediterranean Conference on Control and Automation (MED), pp. 8–13. IEEE Press, Torremolinos (2015). https://doi.org/10.1109/MED.2015.7158722
11. Mouradian, C., Sahoo, J., Glitho, R.H., Morrow, M.J., Polakos, P.A.: A coalition formation algorithm for Multi-Robot Task Allocation in large-scale natural disasters. In: 2017 13th International Wireless Communications and Mobile Computing Conference (IWCMC), pp. 1909–1914. IEEE Press, Valencia (2017). https://doi.org/10.1109/IWCMC.2017.7986575
12. Saad, W., Han, Z., Basar, T., Debbah, M., Hjorungnes, A.: Hedonic coalition formation for distributed task allocation among wireless agents. IEEE Trans. Mob. Comput. **10**(9), 1327–1344 (2011). https://doi.org/10.1109/TMC.2010.242
13. Chen, J., Sun, D.: Coalition-based approach to task allocation of multiple robots with resource constraints. IEEE Trans. Autom. Sci. Eng. **9**(3), 516–528 (2012). https://doi.org/10.1109/TASE.2012.2201470
14. Pendharkar, P.C.: An ant colony optimization heuristic for constrained task allocation problem. J. Comput. Sci. **7**, 37–47 (2015). https://doi.org/10.1016/j.jocs.2015.01.001
15. Dutta, A., Ufimtsev, V., Said, T., Jang, I., Eggen, R.: Distributed hedonic coalition formation for multi-robot task allocation. In: 2021 IEEE 17th International Conference on Automation Science and Engineering (CASE), pp. 639–644. IEEE Press, Lyon (2021). https://doi.org/10.1109/CASE49439.2021.9551582