Multimedia Tools and Applications (2020) 79:14529–14538
https://doi.org/10.1007/s11042-018-7080-4

An auction-based rescue task allocation approach for heterogeneous multi-robot system

Jieke Shi[1] · Zhou Yang[1] · Junwu Zhu[1]

Received: 13 June 2018 / Revised: 7 December 2018 / Accepted: 12 December 2018 /
Published online: 20 December 2018

Abstract

Nowadays, robots are faced with real-time, dynamic, complex and confrontational working environment. It is significant to analyze task allocation in multi-robot systems. In this paper, a dynamic auction approach for differentiated tasks under cost rigidities (DAACR) is proposed, which can obtain optimal results in the task allocation of rescue robots. To verify the feasibility of the proposed approach, we investigate the optimality of the DAACR and compare it with other task allocation approaches based on the Hungarian algorithm. The results show that robots using this algorithm can adapt to a variety of complicated work environments, accomplish more tasks in limited time, reduce the delay of task allocation, and improve the overall utility of multi-robot systems.

Keywords Multi-robot system · Task allocation · Auction · Optimization

1 Introduction

With the rapid development of technology, robots are used in more and more fields. However, the environment that robots face today is always dynamic, real-time, complex and adversarial. What's more, the current single-robot system has limitations in obtaining information and solving problems. When faced with such complicated tasks and stochastic working environment, a single-robot system is complicated to perform operations.

Compared to single-robot systems, multi-robot systems have more advantages when executing tasks. Robots in the system can complete tasks faster and better.

Therefore, the critical issue we need to study is how to assign tasks based on the work environment and situation accurately which is called multi-robot task assignment, MRTA. It aims to increase the efficiency of the entire robotic system and take full advantage of the collaborative functions in multi-robot systems. When the number and state of robots

✉ Junwu Zhu
jwzhu@yzu.edu.cn

Jieke Shi
jiekeshi25@outlook.com

[1] College of Information Engineering, Yangzhou University, Yangzhou Jiangsu, China

are precise, if the number and state of tasks are also precise, the MRTA problem can be considered as an assignment problem. We assume that there are N different robots and M different tasks in the multi-robot system. Then how to map N robots to M tasks correctly is an assignment problem.

Under normal circumstances, accomplishing tasks can provide some rewards for the system. The first object of task allocation is to maximize the utility the whole system can achieve. Moreover, it costs some to accomplish a task. So the second object is to minimize the cost of the whole system. The allocation of tasks directly affects the efficiency of the entire system. We aim to find the optimized task scheduling strategies.

However, in robot rescue tasks, completing tasks will not provide any positive return for the system, and tasks will continue to damage the system. For example, there is a firing building in a city. Extinguishing the fire cannot offer any positive income. However, if the fire brigade leaves the fire as it was, then the fire will cause continuous damage to the city until it stops.

This paper mainly studies task allocation in the rescue environment. Our goal is to minimize system damage caused by sudden disasters. Many task allocation methods are static at present. The task allocation method can be divided into two types: centralized and distributed. Based on the analysis of related work, a merging approach based on auction is proposed.

We made three contributions in this paper. First, a dynamic auction approach for differentiated tasks under cost rigidities (DAACR) is proposed. Secondly, we prove that DAACR is ϵ-optimal. Finally, a detailed experiment is designed, we demonstrate the results to analyze our approach.

2 Related work

In recent years, scholars have made great progress in the research of multi-robot system and its cooperation mechanism. Some scholars proposed to use the dual programming representation process to solve decision problems [1], or use reinforcement learning and neural network in recent years to to control the robots [10, 11, 13]. The traditional dual programming algorithm is generally based on known and determined static environment, but in fact, the rescue environment is a dynamic environment. In order to improve the efficiency of search and rescue (SAR) in disaster relief, some researchers used auction-based task allocation scheme to develop a cooperative rescue plan [2, 5, 16].

Auction algorithm is a feasible task assignment method. It has proved to be effective in producing suboptimal solutions. Generally speaking, agents bid for tasks, and the highest bid wins the assignment [4, 7]. The traditional way of calculating the winner is to have a central agent as an auctioneer to receive and evaluate every bid in the bidder list [14]. Considering multiple resources of the robots and limited robot communication range, Lee [7] proposes a resource-oriented, decentralized auction algorithm (RODAA) for multi-robot task allocation. Lu [9] applied the idea of second-price auction to determine the final price in the double auction.

However, the results that auction generated cannot necessarily achieve the goals in the ideal time [5], at the same time, the response of robots when tasks are operated also has certain hysteresis [15].

The disadvantage of these methods is that the bidding of each agent must be transmitted to the auctioneers in some way [8]. A standard way to avoid communication constraints is to sacrifice auction performance by running auctions through adjacent auctioneers. To achieve an ideal task allocation plan, we must give full consideration to the tasks we have accomplished to obtain the corresponding costs and benefits [3, 6, 12].

3 Model description

We define E_t to represent the robots' work environment at time t. Mathematically speaking, E_t can be formalized as below:

$$E_t =< T_1, T_2, R > \tag{1}$$

In the formula (1), T_1 is the set of tasks that have been allocated while T_2 is the set of the tasks waiting to be assigned. Apparently the set T consisting of all the tasks in the system is $T = T_1 \cup T_2$. A task can be formalized as below:

$$task_j =< taskID_j, location_j, state_j, C_j(A_j^t), t > \tag{2}$$

$taskID_j$ is uesd to identify a task uniquely and $taskID_j = j$. $location_j$ represents the robots current location. $state_i$ indicates the status of task j, and it can be described as below:

$$state_j = \begin{cases} 0, & \text{if task } j \text{ is not assigned;} \\ taskID_j, & \text{if robot } i \text{ takes care of it.} \end{cases} \tag{3}$$

We need to point out that A_j^t is the accomplishment rate of $task_j$ at time t. Clearly, $A_j^t \in [0, 1]$. We also define a new parameter called damaging rate function, labeled with $C_j(AR_j^t)$. It represents how much damages $task_j$ will cause to the system in per unit of time when the accomplishment rate equals to A_j^t.

R is the set of all the robots in this system, it can be formalized as below:

$$R =< r_1, r_2, ..., r_n > \tag{4}$$

r_i is the ith robot in the system. $n = |R|$ represents the total number of robots. One of robots r_i can be described as below:

$$r_i =< robotID_i, location_i, state_i, t > \tag{5}$$

In this formula, $robotID_i$ uniquely identifies a robot. $location_i$ represents the robot's location. $state_i$ indicates the operational status of robot r_i, whose value rules are demonstrated below:

$$state_i = \begin{cases} 0, & \text{if robot } r_i \text{ is free;} \\ taskID_j, & \text{if } task_j \text{ is assigned to } r_i. \end{cases} \tag{6}$$

We use auction as a basic approach to assign tasks, an auction can be formalised as below:

$$A =< T_2, B > \tag{7}$$

T_2 is the set of tasks waiting to be assigned in this auction. B is the set of robots waiting to be assigned to tasks, in other words, bidders.

So far, if t_j is assigned to bi at time t_0, then under such allocation result, the damage that $task_j$ will cause to the system can be formalized as below:

$$cost_i^j = \int_{t_0}^{t_0+t_1^{ij}} C_j(0)dt + \int_{t_0+t_{ij}^1}^{t_0+t_{ij}^1+t_{ij}^2} C_j(A_j^t)dt \tag{8}$$

For a certain task t_j, bidder b_i will calculate two time parameters: t_{ij}^1 and t_{ij}^2 . t_{ij}^1 represents the time it takes for b_i to go to task t_j . t_{ij}^2 represents the time it takes from starting working on t_j to accomplishing it.

4 Auction algorithm description

4.1 *N* tasks for *N* bidders

In this section, we consider one to one auction model of task allocation with N robots matching N rescue tasks. We use $taskID$ and $robotID$ to represent a task and a robot, respectively. The allocation problem to be discussed here is to minimise the damage to the whole system by matching the N robots with the N tasks, and the constraints can be expressed as the following mathematical formula:

$$\begin{aligned} \min\ & \sum_{i\in B}\sum_{j\in T_2} q_{ij}cost_i^j \\ \text{s.t.}\ & \sum_{j|(i,j)\in D} q_{ij} = 1, \forall i = 1,2,...,n \\ & \sum_{i|(i,j)\in D} q_{ij} = 1, \forall j = 1,2,...,n \\ & q_{ij} = 0,1, \forall (i,j) \in D \end{aligned} \tag{9}$$

In this format group above, we use set D to represent the tuples of all possible distributions. It can be formalized as below:

$$D = \{(i,j)|j \in T_2, \forall i = 1,2,...,N\} \tag{10}$$

We use $D(i)$ to represent the set of all tasks that can be assigned to the robot i. To reduce the complexity of the algorithm, we set the time limit using the $\bar{t}$. If the total time of robot i solving task j is more than t, task j cannot be assigned to robot i.

$$D(i) = \{j|t_{ij} \le \bar{t}\} \tag{11}$$

The set A is used to represent the two-tuples (i, j) consisting of robots and tasks. Each robot can have one two-tuples $(i, j) \in A$ at most. Each robot can have one two-tuples $(i, j) \in A$ at most either. For the set A, if there is a two-tuples$(i, j) \in A$, it means that task j is assigned to robot i.

In the algorithm, we set a positive value called ϵ and a price set $p = \{p_1, ..., p_n\}$. For robot i, if the difference between its absolute value of the relative gains obtained from task j and the optimal relative gains obtained from all the allocation schemes is not greater than ϵ, we call robot i and task j satisfy complementary slackness condition. This two-tuples (i, j) is the optimal result, which can be described as below:

$$|p_j - cost_i^j - \max_{k\in D(i)}\{p_k - cost_i^k\}| \le \epsilon \tag{12}$$

The specific process of auction algorithm is described as follows:

Step 1: prepare phase Select a $\epsilon > 0$, set $p_k = 0, \forall k = 1,2,...,n$, and the set of robots which are tender to the task j in the bidding phase is denoted as $B(j)$;

Step 2: decision phase For each robot i, if $state_i = 0$, get the maximum relative gains u_i and the assigned task j_i when the maximum relative gains is obtained:

$$u_i = \max_{k \in D(i)} \{p_k - cost_i^k\} \tag{13}$$

And its second relative gains v_i:

$$v_i = \max_{k \in D(i), k \neq j} \{p_k - cost_i^k\} \tag{14}$$

If $D(i)$ has only one task j, we define v_i as $-\infty$.

Step 3: bidding phase All robots bid for j_i which is the most gainful task, the bidding price of the robot is determined as:

$$a_{ij_i} = p_{j_i} - u_i + v_i - \epsilon = cost_i^{j_i} + v_i - \epsilon \tag{15}$$

Step 4: allocation phase For each task j, if the $B(j)$ is not empty, we update the price of the task j to the highest bid price:

$$p_j = \max_{i \in B(j)} a_{ij} \tag{16}$$

The task is assigned to the highest bidding robot i_j, and at the same time, the two-tuples related to the robot i_j and the task j are removed from the A which is an infeasible allocation, and the new two-tuples (i_j, j) are added to the set A.

Optimization proof In this iterative process of the algorithm, we use p_j and p'_j respectively to represent the price corresponding to the task before and after the iteration. In the iterative process, if robot i bids to the task j_i and is successfully assigned to the task j_i, its price will be updated, which can be described as follow:

$$p'_{j_i} = cost_i^{j_i} + v_i - \epsilon \tag{17}$$

Therefore, we can get the format below:

$$p'_{j_i} - cost_i^{j_i} = v_i - \epsilon = \max_{k \in D(i), k \neq j} \{p_k - cost_i^k\} - \epsilon \tag{18}$$

For each task j, there are $p_j \geq p'_{j_i}$, so:

$$|p'_{j_i} - cost_i^{j_i} - \max_{k \in D(i), k \neq j} \{p_k - cost_i^k\}| \leq \epsilon \tag{19}$$

It can be seen that every two tuple (i, j) always satisfies complementary slackness condition in each iteration.

4.2 *M* tasks for *N* bidders

In the actual disaster environment, the number of rescue tasks and the number of robots cannot be guaranteed to be equal. In order to ensure the feasibility of the algorithm in different environments, it is necessary to discuss the reasonable allocation of N rescue robots and M rescue tasks. According to the number of rescue tasks, there are two kinds of situations, namely, $N > M$ and $N < M$.

4.2.1 $N > M$

The number of rescue robots is larger than the number of rescue missions. Set the $N - M$ virtual tasks and join the auction, we set the price of the virtual task as $p_j = +\infty$. After all of the tasks have been assigned to the robots, robots which are assigned to operate virtual tasks can join the auction again to get real tasks until all robots have been assigned real tasks so far.

4.2.2 $M > N$

The number of rescue tasks is greater than the number of rescue robots, unable to deal with all tasks in a timely manner, in order to ensure the minimum loss of the system, we should select rescue tasks which can have the greater loss for priority allocation. We set $M - N$ virtual robots to add to the allocation, and the tasks allocated to the virtual robot or the less loss task should be discarded in the allocation.

4.3 Dynamic adjustment of task assignment

Because the rescue environment of the robot is changing at all time, the results of the auction are not necessarily feasible. Also, there may be other emergencies in the process of responding to tasks. In this case, we need to adjust the task executors. In this way, we quantify the emergency results using E_j. The definition of urgency is:

$$E_j = \frac{w}{\beta v} \tag{20}$$

In the above formula, w is the quantitative result of the severity of the disaster in the task, and the v is the speed of the task execution. β is the ratio coefficient. In the process of task execution, if E_j is greater than the set threshold h, it is considered that the execution robot cannot finish the task under the existing circumstances. Station will reassess the task and add the task to the auction list, then launch a new round of auction.

5 Experiment analysis and future work

In this paper, we proposed a new auction model and use the auction algorithm to study the multi-robot task allocation problem. Experiments show that the algorithm can effectively allocate tasks.

We chose the classical Hungarian algorithm for comparative experiments. For all cost matrices, the same optimal allocation result can be obtained by iteration. Through experiments, it is found that the results of the two task assignment algorithms are the same. For all matrices, the same optimal assignment results or the near optimal results can be obtained by iteration.

For example, when the number of tasks and robots is 5, we give the cost matrix shown in Table 1.

The final allocation results of the two algorithms are the same:

- Task allocation results:
  Robot r_1 gets $task_3$
  Robot r_2 gets $task_1$
  Robot r_3 gets $task_2$

Table 1 Cost matrix

	$cost_{r_1}$	$cost_{r_2}$	$cost_{r_3}$	$cost_{r_4}$	$cost_{r_5}$
$task_1$	10	3	5	5	7
$task_2$	7	9	1	8	6
$task_3$	1	7	7	4	4
$task_4$	6	7	5	4	8
$task_5$	8	1	10	2	4

Robot r_4 gets $task_4$
Robot r_5 gets $task_5$

- The total cost of tasks is 13. This is a global optimal result.

For all cost matrices in the experiment, the distribution results may be different, but the total cost will be the same and the least. The two algorithms can show the same precision.

However, the proposed auction algorithm has excellent computational complexity obviously. We can improve the time complexity to a fairly low level by changing the value of ϵ. When the size of task matrix that needs to be assigned is 250×250, the running time of Hungarian algorithm is 15 times that of Auction algorithm. When processing the assigned task of 500×500, this multiple reduced to 5. The number of tasks has a greater impact on the running time of the Hungarian algorithm than the algorithm we proposed. The time spent on auction algorithm and Hungarian algorithm is as follows (see Fig. 1).

Due to the limited ability of the author, there are many deficiencies in this paper. On the one hand, there may be new tasks in the execution of tasks. For more urgent tasks, it should

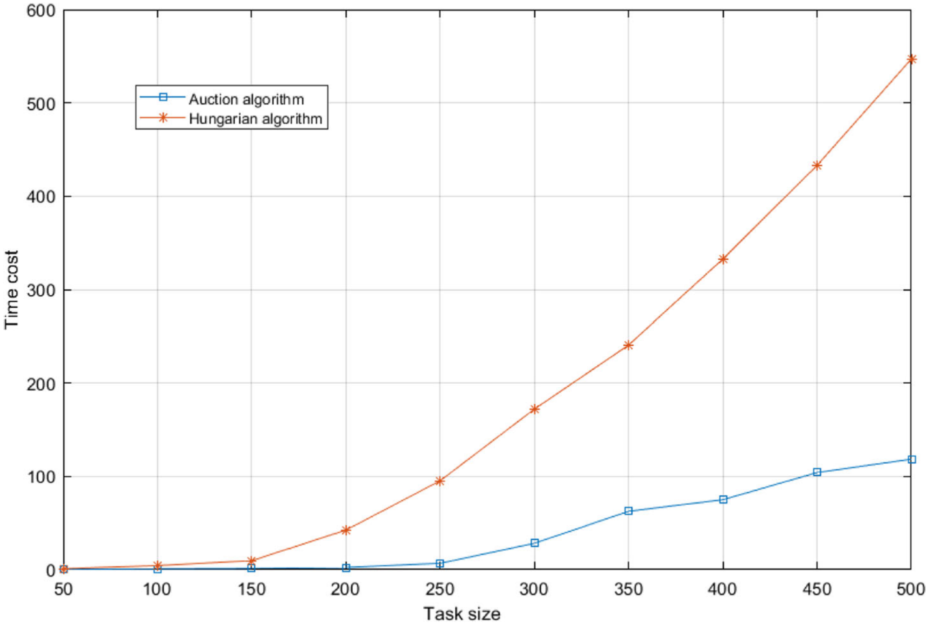

Fig. 1 Time cost comparison between Auction algorithm and Hungarian algorithm

have priority and can be done by more agents, and the robot may need to stop the current work and perform new tasks.

On the other hand, the auction algorithm can only be terminated in the feasible distribution, so a mechanism must be added to the auction algorithm to discover the infeasibility of the problem when the problem is unsolvable. Also, the experiment we carried out in this paper are only theoretical simulation experiments. A real world experiment should be carried out later to verify the method.

Acknowledgments This work was supported by the National Nature Science Foundation of China under Grant 61872313, and Grant 61472344, Grant 61170201, Grant 61070133, the key research projects in education informatization in Jiangsu Province (20180012), in part by the Innovation Foundation for graduates of Jiangsu Province Province under Grant CXLX12 0916, in part by the Natural Science Foundation of the Jiangsu Higher Education Institutions under Grant 14KJB520041, in part by the Advanced Joint Research Project of Technology Department under Grant BY201506106 and Grant BY201506108, and in part by the Yangzhou Science and Technology under Grant YZ2017288 and YZ2018076, and in part by Yangzhou University Jiangdu High-end Equipment Engineering Research Institute Open Project under Grant YDJD201707, in part by the Postgraduate Research and Practice Innovation Program of Jiangsu Province under Grant KYCX18/2366, and in part by Jiangsu Students' Innovation and Entrepreneurship Training Program under Grant 201811117029Z.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

References

1. Booth KE, Nejat G, Beck JC (2016) A constraint programming approach to multi-robot task allocation and scheduling in retirement homes. In: International conference on principles and practice of constraint programming. Springer, pp 539–555
2. Das GP, McGinnity TM, Coleman SA, Behera L (2015) A distributed task allocation algorithm for a multi-robot system in healthcare facilities. J Intell Robot Syst 80(1):33–58
3. Elango M, Nachiappan S, Tiwari MK (2011) Balancing task allocation in multi-robot systems using k-means clustering and auction based mechanisms. Expert Syst Appl 38(6):6486–6491
4. Garg R, Kapoor S (2006) Auction algorithms for market equilibrium. Math Oper Res 31(4):714–729
5. Hooshangi N, Alesheikh AA (2017) Agent-based task allocation under uncertainties in disaster environments: an approach to interval uncertainty. Int J Disaster Risk Reduc 24:160–171
6. Jiang L, Zhang R (2011) An autonomous task allocation for multi-robot system. J Comput Inf Syst 7(11):3747–3753
7. Lee DH, Zaheer SA, Kim JH (2015) A resource-oriented, decentralized auction algorithm for multirobot task allocation. IEEE Trans Autom Sci Eng 12(4):1469–1481
8. Liu Y, Yang J, Zheng Y, Wu Z, Yao M (2013) Multi-robot coordination in complex environment with task and communication constraints. Int J Adv Robot Syst 10(5):229
9. Lu L, Yu J, Zhu Y, Li M (2017) A double auction mechanism to bridge users' task requirements and providers' resources in two-sided cloud markets. IEEE Transactions on Parallel and Distributed Systems
10. Lu H, Li B, Zhu J, Li Y, Li Y, Xu X, He L, Li X, Li J, Serikawa S (2017) Wound intensity correction and segmentation with convolutional neural networks. Concurr Comput Practice Exp 29(6):e3927
11. Lu H, Li Y, Mu S, Wang D, Kim H, Serikawa S (2017) Motor anomaly detection for unmanned aerial vehicles using reinforcement learning. IEEE Internet of Things Journal
12. Lu H, Li Y, Chen M, Kim H, Serikawa S (2018) Brain intelligence: go beyond artificial intelligence. Mob Netw Appl 23(2):368–375
13. Lu H, Li Y, Uemura T, Kim H, Serikawa S (2018) Low illumination underwater light field images reconstruction using deep convolutional neural networks. Future Generation Computer Systems
14. Ponda SS, Johnson LB, How JP (2012) Distributed chance-constrained task allocation for autonomous multi-agent teams. In: American control conference (ACC). IEEE, pp 4528–4533

15. Serikawa S, Lu H (2014) Underwater image dehazing using joint trilateral filter. Comput Electr Eng 40(1):41–50
16. Tang J, Zhu K, Guo H, Gong C, Liao C, Zhang S (2018) Using auction-based task allocation scheme for simulation optimization of search and rescue in disaster relief. Simul Model Pract Theory 82:132–146

Jieke Shi is an undergraduate studying at School of Information Engineering of Yangzhou University. His major is Electronic and Information Engineering and his research interests include algorithmic game theory, deep learning and multi-agent system.

Zhou Yang is an undergraduate studying at School of Information Engineering of Yangzhou University. His major is Software Engineering and his research interests include algorithmic game theory, security game and multi agent system.

Junwu Zhu received PHD degrees in computer science from the Nanjing University of Aeronautics and Astronautics in 2008. Currently, he is a professor in School of Information Engineering, Yangzhou University, and also a visiting professor in School of Computer Science, University of Guelph, Canada. His research interests include knowledge engineering, ontology, mechanism design, cloud computing.