Research on Task Assignment Optimization Algorithm Based on Multi-Agent

Jie Zhang[1],Gang Wang[2],Xiaoqiang Yao[2], Yafei Song[1], Fangzheng Zhao[1]
1.Graduate School, Air Force Engineering University
2.Air Defense and Anti-missile Academy of Air Force Engineering University
Xi'an, China
824519928@qq.com

***Abstract*—This paper studies the task assignment problem in multi-agent field. The maximum number of tasks for a single agent and the number of agents required for a single task are taken as objective functions. Considering the characteristics of the air defense unmanned combat agent, the Multi-agent task allocation model based on the improved auction algorithm is established. The first step is to establish a mathematical model for the task assignment strategy and confirms the objective function. Then, the resource utilization efficiency is improved by optimizing the auction algorithm. Finally, the simulation experiments prove the effectiveness and convergence of the algorithm.**

Keywords—multi-agent; task assignment; auction algorithm; resource utilization; convergence

I. INTRODUCTION

With the development of network and fighter, the future air defense combat mode not only deploys close-range combat of single-machine to single-machine, but also gradually develops into a distributed air combat coordinated with each fighter under command. There are many researches on air combat task assignment at home and abroad. The literature [1, 2] mainly introduces the basic theory of multi-task assignment in air operations and the decision-making problem of multi-objectives. The literature [3, 4] discusses the task assignment problem of UVA operations. The literature [5~8] studies the problem of multi-machine cooperative guidance. For task assignment problem, the auction algorithm is applied frequently. It has improved the auction algorithm by strengthening the auction algorithm. This is a solution for the multi-agent environment, which provides a solution for the related problems.

II. TASK ASSIGNMENT STRATEGE MODELING (AUCTION)

A. Conceptual Description

In the view of the characteristics of the unmanned combat agent (such as position, distance, speed, performance, mission revenue, execution capability, battlefield situation and difficulty of completion, etc.). This model distributes and describes our multi-agent distributed combat tasks for the enemy air strike mission.

B. Variable Definition

(1) Attribute B of The Agent:

$$B = \{B_j\} \quad (j = 1, 2, \dots N)$$

N is the number of agents.

The value of the agent C:

$$C = \{C_j\} (j = 1, 2, \dots N)$$

C_j is the value of the agent B_j, and its size is related to the ability of the agent.

Sensor distance D of the agent:

$$D = \{D_j\} \quad (j = 1, 2, \dots N)$$

D_j is the sensor distance of the agent B_j.

The average moving rate of the agent V:

$$V = \{V_j\} \quad (j = 1, 2, \dots N)$$

V_j is the average moving rate of the agent.

(2) Subtask T: The task is derived from combat mission.

$$T = \{T_k\} \quad (k = 1, 2, \dots M)$$

M is the number of subtasks. The coordinate position of T_k is (x_k, y_k), and T_d is the difficulty of subtasks.

(3) Task execution quality Q: How good or bad the task is. The task execution quality is only related to the destructed probability P of target.

Then,

$$Q_{jk} = f\,(P_{jk})$$

The characteristics of the relationship between task execution quality and target destruction probability are as follows:

a: The trend of the function relationship $Q_{jk} = f\left(P_{jk}\right)$ is controlled by the threshold δ.

b: This function is a monotonically increasing function. And when P_{jk} approaches 0, Q_{jk} approaches 0. When P_{jk} approaches 1, Q_{jk} also approaches 1.

c: When P_{jk} is located near the threshold δ, the rate of change of Q_{jk} is large, and when P_{jk} is far from the threshold δ, the rate of change of Q_{jk} is small.

C. Problem Modeling

Let the entire agent system perform tasks by N agents, M subtasks need to be assigned and assign combat tasks to each agent. Each agent can receive multiple subtasks, or multiple agents can receive one subtask, the subtask assignment matrix A^(N×M) is:

$$A^{N\times M} = \begin{cases} 1, & B_j \textit{ need to execute subtask} \\ 0, & B_j \textit{ needn't to execute subtask} \end{cases}$$

in which $j = 1, 2, \cdots N; k = 1, 2, \cdots M$

The aim to construct the subtask matrix A^{M*N} like that is to maximize the operational effectiveness of the entire agent system.

1) Benefits of Task Assignment

The basic strategy of task assignment: each agent should determine the most reasonable subtask according to the subtask and priority order that it is best at. The benefit function E(A) of the execution task determined by the strategy of task assignment is:

$$E(A) = \sum_{k=1}^{M} \theta_k G_k$$

Where θ_k is the benefit value of the execution subtask; G_k is the priority of the subtask. θ_k can be obtained by the following formula:

$$\theta_k = 1 - \prod_{j=1}^{N} (1 - A_{jk} \cdot Q_{jk})$$

The priority of task execution is determined by many factors, such as threat size, target distance, superior command and operational value. We adapt the calculation method of priority calculation weight, and the priority value is the priority value of the subtask. Then, we obtain the following equations:

$$G_k = (h_k)^{-\frac{1}{2}}$$

Where h_k is the priority order of the subtasks, and the value of h_k can be 1, 2, ..., M.

2) The Cost of Task Assignment

A cost function I(A):

$$I(A) = \sum_{j=1}^{N} [\sum_{k=1}^{M} \xi_j(T_k) A_{jk}] \mu_j$$

Where $\xi_j(T_k)$ is the loss probability of the agent B_j when performing the subtask T_k:

$$\xi_j(T_k) = f(I_{jk}^{le}, I_{jk}^{th}, I_{jk})$$

I_{jk} is the cost of B_j being destroyed when performing subtask T_k, which is the consequence of the destruction of the agent.

I_{jk}^{le} is the cost of road length B_j when performing subtask T_k, which means the fuel loss of the agent B_j on the flight segment. It is related to the segment length I_{km} and the moving speed V_j of the agent, so the cost of road length agent B_j in the segment is:

$$I_{jkm}^{le} = \alpha \cdot f(v_j, l_{km})$$

in which f is the fuel loss function of B_j and α is the normalization factor.

I_{jk}^{th} is the average path threat cost of B_j when performing subtask T_k, which is the degree of local threat that the agent receives on the segment. Therefore, the average path threat cost of the agent B_j on the segment is:

$$I_{jkm}^{th} = (1-\alpha) \cdot r_j \cdot le_{km} / (d_{km})^4$$

Where r_j is the effective scattering cross-sectional area of B_j. le_{km} is the threat level and d_{km} is the shortest distance from the segment to threat.

Considering if the path cost, threat cost and the agent B_j is destroyed when they carry out the subtask T_k. The loss function $\xi_j(T_k)$ can be established as follows:

$$\xi_j(T_k) = I_{jk}[\sum_{m=1}^{X} (I_{jkm}^{le} + I_{jkm}^{th})]$$

3) Objective Function

The benefits and costs of task assignment have been described in detail above. Therefore, the objective function for the multi-agent distributed task assignment problem model can be constructed as

$$\max J(A) = \beta \sum_{k=1}^{M} [1 - \prod_{j=1}^{N} (1 - A_{jk} \cdot Q_{jk})] G_k - \gamma \sum_{j=1}^{N} \left[\sum_{k=1}^{M} \xi_j(T_k) A_{jk} \right] \mu_i$$

Then,

$$\sum_{j=1}^{N} A_{jk} \geq X_{\min}$$

$$\sum_{k=1}^{M} A_{jk} \leq Y_{max}$$

$$A_{jk} \in \{0,1\}$$

Where, β and γ are the weighting factors of benefit and cost. $X_{\min}$ represents the minimum number of agents required to complete a subtask, and $Y_{\max}$ represents the maximum of total tasks of a single agent.

III. MANNED OR UNMANNED COMBAT MULTI-AGENT DISTRIBUTED TASK ASSIGNMENT METHOD BASED ON AUCTION ALGORITHM

A. Auction Mechanism

Auctions, it is known that sellers provide goods and show its low prices and buyers bid according to the rules. The highest bidder will win the right to buy. Auction will stop immediately if it beyond the limited time. Then the highest bidder will win the goods. In combat, the auction means that the combat agent establishes its own plans according to the tasks that it performed. In the process of bidding, according to the greed algorithm, the agent selects the best task plan among all the task plans [9]. However, when the number of agents and the performed tasks increase, the number of plans will increase accordingly. As a result, the time to select the best task plan will be needed more. In order to make the algorithm practicable, the auction mechanism should be modified and the improvements are as follows:

Before the auction, each agent needs to prepare a plan according to its tasks. The coordination center is responsible for coordinating the bidding sequence. When it is in the turn of a certain group of auctions, the agents in this group will select the best plan randomly from the prepared plans [10].

Let the number of agents be O_A in a group and the auction time of this group needed to bid be O_A. If each group is independent from each other, the number of schemes constructed by all the agents is $(M+1)O_A$ in a group. Compared with the original plans that all agents allocate task together, this mechanism reduces the number of schemes and redundant calculations, and the time that spent meet requirements.

B. Auction Principles

1) Program preprocessing

According to the constraint of the minimum number of agents required to complete a subtask and the maximum of total tasks of a single agent in the objective function, the solution is selected and all the solution sets of a single agent are written as R:

$$R = \{\emptyset, T_1, T_2, \ldots, T_M, \{T_1, T_2\}^{\pm}, \ldots, \{T_{M-1}, T_M\}^{\pm}\}$$

Let program r_i be a subset of R and (i = 1, 2, ... U), U represents the number of elements in set R, then $U = M^2 + 1$. The number of schemes that agents can construct increases exponentially with the task increase, so we need to conduct a further selection. At that time, the proximity factor φ, which indicates the degree of proximity of the subtask, is introduced. When the proximity of the subtask of the agent B_j is too large, the program will be deleted.

$$\varphi = \frac{1}{M}\sum_{k=1}^{M}\sqrt{(x_k - x_{med})^2 + (y_k - y_{med})^2}$$

$$\emptyset = \sqrt{(x_k - x_i)^2 + (y_k - y_i)^2}$$

$$\tau = \varepsilon\varphi$$

Where (x_k, y_k) is the position coordinate of the subtask T_k. (x_{med}, y_{med}) is the average position coordinate of all subtasks. (x_i, y_i) is the position coordinate of the subtask T_i. ε is the coefficient factor, and its value range is （0， 1].

2) Scheme Construction

a. The benefit of the implementation r_i of the Agent B_j is:

$$E_j(r_i) = \sum_{m=1}^{d(z_l)} \theta_{j\cdot km} G_{km}$$

And is $d(Z_l)$ the number of subtasks.

b. The cost of $d(Z_l)$ the implementation r_i of agent B_j is:

$$I_j(r_i) = \xi_j(r_i)\mu_j$$

$$\xi_j(r_i) = I_{jk}[\psi \cdot Th_j(r_i) + (1-\psi) \cdot Le_j(r_i)]$$

Where ψ is a normalization factor, the value range of it is [0,1]. $Th_j(r_i)$ is the threat cost of the agent B_j in the case of using the scheme r_i. $Le_j(r_i)$ is the path cost of agent B_j in the case of using the scheme r_i.

c. The selection function of implementation program r_i agent B_j is

$$V_{ji} = \beta \cdot E_j(r_i) - \gamma \cdot I_j(r_i)$$

Then, we can obtain the maximum allocation scheme by the following formula:

$$r(j) = V_i^{-1}(\max(V_{ji}))$$

3) The update criteria of priority ordering for tasks

In the process of the task auction, when an agent is bidden, it will lower its priority sequence in the next task auction, and then continue to perform the task auction and update its own bidding task plan [11]. This is conducive to the overall optimization, so as to get the best bidding solution. When the agent B_j bids to the task plan r_i , it will update the priority sequence of the task with the following formula:

$$G_k(new) = (1 - Q_{jk}) \cdot G_k$$

4) Auction Algorithm Flow

As shown below

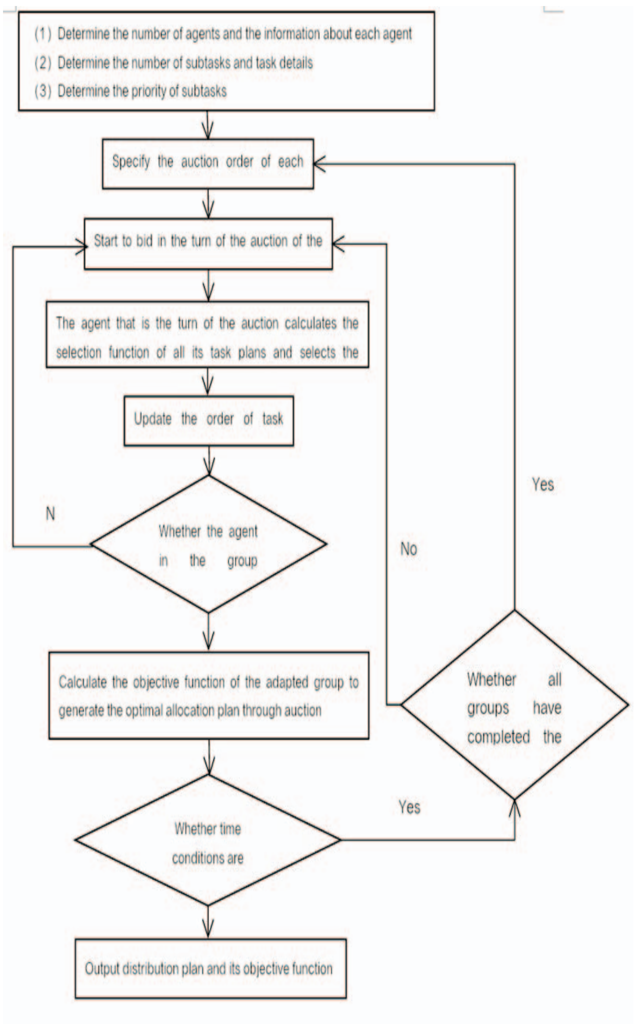

Fig.1. Multi-agent distributed task allocation algorithm flow chart

Before executing the auction algorithm, the number of agents and the information of each agent should be input first to make group. The following step is to input the number of subtasks and the task details but also determine the rules and constraints of each subtask. Agents determine the priority order of the subtasks according to the detailed task. Each agent lists all possible task scenarios according to the subroutine and filters out feasible solutions according to the constraints [12].

When the auction algorithm [13] is executed, the auction order of the group is determined first. The auction is performed according to the auction order. The selection function is calculated during the auction, determining the optimal task plan and updating the task priority order. Then the group bidding is completed. Determining if the time exceeds the limit, and if so, output the current plan; if not, continue the process. When all the group auctions have been completed, a new round of auctions is started. The best solution is output until the time reaches the limit.

IV. SIMULATION EXPERIMENTS AND RESULTS ANALYSIS

In the same problem of task assignment, comparing the optimization effect of the algorithm designed in this paper with the unmodified auction algorithm under the same conditions, as the following table.

TableI. ALGORITHMIC ITERATION EFFICIENCY COMPARISON

Algorithm	*Number of agents*	*Number of auction iterations*	*Average objective function value*
Algorithm in this paper	10	20	8.458
	20	50	7.816
	50	100	6.054
Unimproved auction algorithm	10	20	13.224
	20	50	10.682
	50	100	9.545

The improved algorithm takes the minimum number of agents required for a subtask and the maximum value of the total task amount of a single agent as the objective function values. After a certain times of auction iterations, we can obtain the average objective function. By the experimental data, it is known that the objective function proves that the improved algorithm has superior convergence and high efficiency [14].

Comparing the algorithm of this paper with the literature algorithm in the same number of initial auction function [15], the algorithm of this paper obtains a lower objective function value and makes the task distribution more reasonable after giving an initial function value. It proves that the algorithm has better superiority [16].

TableII. ALGORITHM PERFORMANCE COMPARISON

Algorithm	*Number of auction iterations*	*Initial function value*	*Objective function*
reference 【9】	24	2. 536	1. 435
reference 【13】	24	2. 536	1. 145

Algorithm in this paper	2	2.536	1.007
reference【9】	69	1.5	1.234
reference【13】	69	1.5	0.823
Algorithm in this paper	69	1.5	0.427
reference【9】	15	0.9959	0.9903
reference【13】	15	0.9959	0.9950
Algorithm in this paper	15	0.9959	0.9877

In performance algorithm comparison., setting the same initial function value and iteration number are set for the literature algorithm and the algorithm in this paper, the algorithm has better convergence under the same number of iterations [17].

The algorithm is used for allocation, and the distribution result is shown in Table 3. The attack situation map and the dominant histogram are shown in the figure. According to this allocation scheme, the overall advantage value is 1.826, and the simulation time is 0.6481s. However, the enemy's key targets No.1 and No.2 are assigned to our No.1 fighter with the greatest advantage, which achieves a blow to the enemy's key targets.

Our plane number	enemy's plane number							
	1	2	3	4	5	6	7	8
1	1	1	0	0	0	0	0	0
2	0	0	0	0	1	1	0	0
3	0	0	0	0	0	0	1	1
4	0	0	1	1	0	0	0	0

The results show that the algorithm in this paper has a good adaptability for key task assignments and has certain advantages over other algorithms for task assignment problems of key objectives [18].

V. CONCLUSIONS

This paper studies the problem of multi-intelligent task assignment in distributed environment, introducing an improved auction algorithm and designing a multi-agent alliance model based on this algorithm. By analyzing the relationship between iteration number and objective function value and the final algorithm, it proves the superiority of the algorithm in task assignment when the improved auction algorithm is compared with literature algorithm [19]. The algorithm of this paper has better convergence and higher efficiency than other algorithms and can reach the target function value faster. It is specifically applied to the project engineering, which solves the optimization problem of the real battlefield task assignment.

ACKNOWLEDGMENT

This work was supported by the Young Scientists Fund of the National Natural Science Foundation of China under Grant No. 61703412.

REFERENCES

[1] Harcholbalter M, Crovella M E, Murta C D. On Choosing a Task Assignment Policy for a Distributed Server System. Journal of Parallel & Distributed Computing, 1998, 59(2):204-228.

[2] Harcholbalter M, Crovella M E, Murta C D. On Choosing a Task Assignment Policy for a Distributed Server System. Journal of Parallel & Distributed Computing, 1998, 59(2):204-228.

[3] Lo V M. Heuristic Algorithms for Task Assignment in Distributed Systems. IEEE Trans on Comput, 1988, 37(11):1384-1397.

[4] Kafil M, Ahmad I. Optimal task assignment in heterogeneous distributed computing systems. Concurrency IEEE, 1997, 6(3):42-50.

[5] Bertsekas D P. Auction algorithms for network flow problems: A tutorial introduction. Computational Optimization & Applications, 1992, 1(1):7-66.

[6] Mitola J I. Cognitive Radio : An Integrated Agent Architecture for Software Defined Radio. 2000.

[7] Nicholas R. Jennings. On agent-based software engineering. Artificial Intelligence, 2000, 117(2):277-296.

[8] Jennings N R, Sycara K, Wooldridge M. A Roadmap of Agent Research and Development. Autonomous Agents and Multi-Agent Systems, 1998, 1(1):7-38.

[9] Olfati-Saber R, Fax J A, Murray R M. Consensus and Cooperation in Networked Multi-Agent Systems. Proceedings of the IEEE, 2007, 95(1):215-233.

[10] Bellifemine F, Poggi A, Rimassa G. Developing Multi-agent Systems with JADE. Intelligent Agents VII Agent Theories Architectures & Languages, 2000, 12(3):89-103.

[11] Bertsekas D P. The auction algorithm: A distributed relaxation method for the assignment problem. Annals of Operations Research, 1988, 14(1):105-123.

[12] Bertsekas D P, CastaÑ D A, on. A FORWARD/REVERSE AUCTION ALGORITHM FOR ASYMMETRIC ASSIGNMENT PROBLEMS1. Computational Optimization & Applications, 1992, 1(3):277-297.

[13] Zavlanos M M, Spesivtsev L, Pappas G J. A distributed auction algorithm for the assignment problem. Decision and Control, 2008. Cdc 2008. IEEE Conference on. IEEE, 2008:1212-1217.

[14] Sun J, Modiano E, Zheng L. Wireless channel allocation using an auction algorithm. IEEE Journal on Selected Areas in Communications, 2006, 24(5):1085-1096.

[15] Garg R, Kapoor S. Auction algorithms for market equilibrium. Mathematics of Operations Research, 2006, 31(4):511-518.

[16] Huang J, Berry R A, Honig M L. Auction-Based Spectrum Sharing. Mobile Networks & Applications, 2006, 11(3):405-408.

[17] Gerkey B P, Mataric M J. Sold!: auction methods for multirobot coordination. IEEE Transactions on Robotics & Automation, 2002, 18(5):758-768.

[18] Skreta V, Figueroa N. Optimal Auction Design For Multiple Objects with Externalities. Ucla Economics Online Papers, 2004, 6(1):58-73.

[19] Olfati-Saber R. Flocking for multi-agent dynamic systems: algorithms and theory. IEEE Transactions on Automatic Control, 2006, 51(3):401-420.