Postprint: A Solution Method for Decision Entity Configuration Problem Based on Ant Colony Optimization with Global Update Rules

Authors: Chen Guanyu, Sun Peng, Mengchen Liao, Zhang Jieyong, Wu Junsheng

Date: 2018-06-19T00:00:00+00:00

Abstract

To solve the clustering problem of platform resources in battlefield C2 organizations, this paper first defines the various elements in the decision entity configuration process. Subsequently, the workload of decision entities is defined from both external and internal perspectives in a mission-oriented manner. A decision entity configuration model is established with the root mean square (RMS) of workload as the objective function. Taking the platform scheduling scheme as input information, an ant colony algorithm based on global update rules is adopted to solve the problem model, thereby generating decision entity configuration schemes that satisfy operational requirements. Finally, the proposed method is validated and analyzed through simulation examples. Experimental results demonstrate that the method can obtain superior platform clustering schemes and exhibits advantages over traditional methods in terms of both clustering results and convergence speed.

Full Text

Preamble

Solving Method for Decision-Makers Configuration Problem Based on Global Update Rule of Ant Colony Algorithm

Chen Guanyu[1], Sun Peng[1,2], Liao Mengchen[3], Zhang Jieyong[1], Wu Junsheng[2]

(1. College of Information & Navigation, Air Force Engineering University, Xi' an 710077, China;
2. a. School of Computer Science, b. School of Software & Micro-electronics, Northwestern Polytechnical University, Xi' an 710072, China;
3. Unit 95445 of People' s Liberation Army, Dali, Yunnan 672100, China)

Abstract: To solve the clustering problem of platform resources in battlefield C2 organizations, this paper first defines the various elements in the decision-making entity configuration process. It then defines the workload of decision-making entities from both external and internal perspectives in a combat-task-oriented manner, establishing a decision-making entity configuration model with the root mean square (RMS) of workload as the objective function. Using the platform scheduling scheme as input information, the paper employs an ant colony algorithm based on global update rules to solve the problem model and generate decision-making entity configuration schemes that meet operational requirements. Finally, simulation examples verify and analyze the proposed method. Experimental results demonstrate that this method can obtain better platform clustering schemes and shows superiority over traditional methods in both clustering results and convergence speed.

Keywords: decision-makers configuration; combat-task-oriented; root mean square; ant colony algorithm

0 Introduction

With the deepening influence of information theory on military warfare, establishing relationships among platform resources, combat tasks, and decision-making entities in battlefield environments has become a key issue in command and control (C2) organization design. The platform resource clustering problem, also known as the decision-making entity configuration problem in C2 organization design, involves designing command and control relationships between operational platforms and decision-making entities to enable different platforms to better and faster complete their assigned combat tasks under the control of corresponding decision-making entities, thereby improving task execution effectiveness and quality.

How to scientifically and reasonably describe and measure workload and establish corresponding mathematical models to solve the platform clustering problem constitutes the main content of the C2 organization decision-making entity configuration process. Previous research has established mathematical models with different objectives: some aimed to minimize the maximum workload of decision-making entities, measuring workload from the perspective of platform and task quantities; others measured workload based on combat task execution time, establishing models that minimize the RMS value of workload; some defined the concept of command and control complexity, establishing workload minimization models from the perspective of collaboration among different platforms on combat tasks; some proposed alternative solution strategies to address the local optima issue in traditional hierarchical clustering methods; and others generated a series of frontier solutions for multi-objective problem models and obtained platform clustering results through specific screening methods.

Building upon these studies, this paper further discusses a method that combines external and internal workload descriptions for combat tasks to measure decision-making entity workload, establishes a decision-making entity configuration problem model with the RMS value of decision-making entity workload as the objective function, and utilizes an ant colony algorithm based on global update rules to solve the problem model. Finally, simulation examples discuss the optimal parameter settings for the algorithm and verify its applicability and superiority. Experimental results show that selecting appropriate algorithm parameters can yield scientifically reasonable decision-making entity configuration schemes, and compared with traditional ant colony algorithms and adaptive ant colony algorithms, this algorithm demonstrates faster convergence speed and better clustering results.

1 Problem Model Establishment

1.1 Variable Definition

Entities in the battlefield environment are mainly divided into three categories: decision-making entities (DM), platforms (P), and combat tasks (T). The platform clustering problem involves clustering platforms and assigning them to different decision-making entities for command and control, given the allocation relationship between tasks and platforms. For the problem addressed in this paper, relevant variables are defined as follows:

Let $\{DM_1, DM_2, ..., DM_{ND}\}$ represent the set of all decision-making entities, where ND is the number of decision-making entities. Let $\{P_1, P_2, ..., P_{NP}\}$ represent the set of all platforms, where NP represents the number of platforms. Let $\{T_1, T_2, ..., T_{NT}\}$ represent the set of all combat tasks, where NT represents the number of tasks.

The task-platform allocation matrix $\omega = (\omega_{ij})_{NT\times NP}$ indicates whether task T_i requires platform P_j for execution. If $\omega_{ij} = 1$, task T_i requires platform P_j; if $\omega_{ij} = 0$, task T_i does not require platform P_j. This matrix serves as known input information for the model.

The decision-making entity-platform command relationship matrix $m = (m_{nj})_{ND\times NP}$ indicates whether decision-making entity DM_n commands and controls platform P_j. If $m_{nj} = 1$, decision-making entity DM_n commands and controls platform P_j; if $m_{nj} = 0$, decision-making entity DM_n does not command and control platform P_j. This matrix represents the final output information of the model.

The decision-making entity-task execution matrix $x = (x_{ni})_{ND\times NT}$ indicates whether decision-making entity DM_n executes task T_i. If $x_{ni} = 1$, decision-making entity DM_n executes task T_i; if $x_{ni} = 0$, decision-making entity DM_n does not execute task T_i.

The decision-making entity collaboration relationship vector $y = (y_{mn})_{ND\times ND}$ indicates collaboration between different decision-making entities. If $y_{mn} = 1$, decision-making entities DM_m and DM_n have collaboration on tasks; if $y_{mn} = 0$, they have no collaboration. The relationship can be derived from x and m as follows: for platform P_j, if $\sum_{i=1}^{NT} \omega_{ij} = 1$, then $\sum_{n=1}^{ND} m_{nj} = 1$.

Therefore, from the above description, we know: - $y_{mn} = x_{mi} \cdot x_{ni}$ for $(m, n = 1, 2, ..., ND; m \neq n; i = 1, 2, ..., NT)$ - $y_{mn} \leq x_{mi} \cdot m_{nj}$ for $(m, n = 1, 2, ..., ND; m \neq n; i = 1, 2, ..., NT; j = 1, 2, ..., NP)$ - $y_{mn} \leq \sum_{j=1}^{NP} m_{nj}$ for $(m, n = 1, 2, ..., ND; m \neq n)$

1.2 Constraint Condition Analysis

The force organization structure can be divided into three layers: decision-making layer, task layer, and platform layer. Based on this structure, the following constraints exist in the decision-making entity configuration problem:

a) The relationship between decision-making entities and platforms is one-to-many. One decision-making entity can command and control multiple platforms, while one platform can only belong to one decision-making entity. That is, $\sum_{n=1}^{ND} m_{nj} = 1$ for $j = 1, 2, ..., NP$.

b) From the variable definitions, when and only when $\sum_{j=1}^{NP} \omega_{ij} \cdot m_{nj} \geq 1$, then $x_{ni} = 1$. Therefore, the relationship $x_{ni} = \sum_{j=1}^{NP} \omega_{ij} \cdot m_{nj}$ holds.

c) Each decision-making entity must participate in commanding and controlling platforms to execute their assigned combat tasks, meaning each decision-making entity must command and control at least one platform: $\sum_{j=1}^{NP} m_{nj} \geq 1$ for $n = 1, 2, ..., ND$.

d) The collaboration relationship between decision-making entities is represented by vector y_{mn}: $y_{mn} = x_{mi} \cdot x_{ni}$ for $(m, n = 1, 2, ..., ND; m \neq n; i = 1, 2, ..., NT)$.

1.3 Problem Model Establishment

1.3.1 Definition of Workload Workload indicates the situation of platforms commanded and controlled by each decision-making entity executing tasks. Considering that combat tasks have different completion times and bring different workload amounts to different decision-making entities, this paper selects task execution time as the workload measure and describes it from both internal load and external load aspects.

1) **Internal Load Definition and Measurement Method**

The internal load of decision-making entity DM_n, denoted as $In(DM_n)$, represents the workload generated by the decision-making entity commanding and controlling its own platforms to complete combat tasks. Numerically, it equals the sum of execution times of all platforms controlled by the decision-making entity:

$$In(DM_n) = \sum_{j=1}^{NP} m_{nj} \cdot S_j$$

where S_j represents the workload of platform P_j executing tasks, defined as the sum of execution times of combat tasks performed by platform P_j:

$$S_j = \sum_{i=1}^{NT} \omega_{ij} \cdot t_i$$

where t_i represents the execution time of task T_i.

Therefore, from equations (10) and (11), we know:

$$In(DM_n) = \sum_{j=1}^{NP} m_{nj} \cdot \sum_{i=1}^{NT} \omega_{ij} \cdot t_i = \sum_{i=1}^{NT} \sum_{j=1}^{NP} \omega_{ij} \cdot m_{nj} \cdot t_i$$

2) **External Load Definition and Measurement Method**

The external load of decision-making entity DM_n, denoted as $En(DM_n)$, represents the workload generated by collaboration between two different decision-making entities to complete combat tasks. Numerically, it equals the sum of workloads of platforms controlled by all decision-making entities collaborating to execute tasks:

$$En(DM_n) = \sum_{m=1, m \neq n}^{ND} y_{mn} \cdot \sum_{i=1}^{NT} \sum_{j=1}^{NP} \omega_{ij} \cdot m_{nj} \cdot t_i$$

3) **Total Workload of Decision-Making Entity** DM_n

Combining equations (12) and (13), the total workload of decision-making entity DM_n is:

$$CW_n = a \cdot In(DM_n) + b \cdot En(DM_n)$$

where weight coefficients a and b represent the influence of internal load and external load on total workload respectively, with $a + b = 1$.

1.3.2 Objective Function Construction Since workload is described from both internal load and external load aspects, different clustering schemes produce different workloads for decision-making entities commanding and controlling platforms to complete tasks, resulting in different configuration outcomes. For a given platform scheduling scheme, the sum of internal loads of all decision-making entities remains constant regardless of clustering results, but the sum of external loads varies across different clustering schemes. Therefore, an appropriate clustering scheme can be selected to minimize the sum of external loads of all decision-making entities.

This paper selects the minimization of average workload per decision-making entity and minimization of workload differences among decision-making entities as the objective functions of the clustering model. The average workload of decision-making entities is expressed as:

$$\mu = \frac{1}{ND} \sum_{m=1}^{ND} CW_m$$

The differences among decision-making entities are represented by variance:

$$\sigma^2 = \frac{1}{ND} \sum_{m=1}^{ND} [CW_m - \mu]^2$$

Therefore, the RMS value is used to comprehensively evaluate the workload of decision-making entities:

$$RMS = \sqrt{\mu^2 + \sigma^2}$$

From equation (16), we can derive:

$$RMS = \sqrt{\frac{1}{ND} \sum_{m=1}^{ND} CW_m^2}$$

1.3.3 Mathematical Model Establishment Based on the above constraint analysis and objective function construction, the mathematical model of the decision-making entity configuration problem established in this paper is expressed as:

$$\min \quad RMS = \sqrt{\frac{1}{ND} \sum_{m=1}^{ND} CW_m^2}$$

Subject to:

$$\sum_{n=1}^{ND} m_{nj} = 1, \quad j = 1, 2, ..., NP$$

$$\sum_{j=1}^{NP} m_{nj} \geq 1, \quad n = 1, 2, ..., ND$$

$$x_{ni} = \sum_{j=1}^{NP} \omega_{ij} \cdot m_{nj}, \quad n = 1, 2, ..., ND; i = 1, 2, ..., NT$$

$$y_{mn} \leq \sum_{j=1}^{NP} \omega_{ij} \cdot m_{nj}, \quad m, n = 1, 2, ..., ND; m \neq n; i = 1, 2, ..., NT$$

$$y_{mn} \leq \sum_{j=1}^{NP} m_{nj}, \quad m, n = 1, 2, ..., ND; m \neq n$$

2 Decision-Making Entity Configuration Problem Based on Global Update Rule Ant Colony Algorithm

Decision-making entity configuration is essentially a platform clustering process that assigns different platforms to a certain number of decision-making entities, which then control each platform to execute predetermined combat tasks. During platform clustering, the optimal solution that satisfies problem constraints and objective functions must be obtained, making the platform clustering scheme reflect the rationality of the designed configuration relationship between platforms and decision-making entities.

The decision-making entity configuration problem is fundamentally a combinatorial optimization problem that can be solved using intelligent algorithms. Ant Colony Optimization (ACO), as an intelligent algorithm with strong search capabilities and fast convergence, has been widely applied to combinatorial optimization problems, demonstrating good performance in path planning, flow shop scheduling, optimal solution searching, building location selection, and UAV combat target allocation. Therefore, ant colony algorithm can serve as a method for seeking optimal clustering schemes in the platform clustering process.

The algorithm follows heuristic ideas, using positive feedback and pheromone evaporation mechanisms. Through the induction of pheromones released by ants on traversed paths, it gradually converges to the global optimal solution through multiple iterations. The core idea can be briefly described as: all ants leave pheromones on traversed paths, with different paths corresponding to different pheromone trails. Since ants on the shortest path reach their destination earlier, increasing their round-trip frequency, the pheromone density on this path becomes greater. Other ants select paths with stronger pheromones during pathfinding, causing continuous pheromone accumulation on this path. Through this positive feedback mechanism, ants can eventually find the optimal path from the nest to the food source.

For the decision-making entity configuration problem, this paper describes the various components of the ant colony algorithm in the platform clustering process based on existing literature.

a) **Construction Graph**: Ants randomly walk on the construction matrix to establish the solution construction process. The graph fully connects all elements in set C, which contains all platforms and decision-making entities. A feasible solution represents a matrix established by ND decision-making entities and NP platforms in the pairing form (P_j, DM_n). If $m_{nj} = 1$, it indicates that platform P_j is assigned to decision-making entity DM_n for command and control. During the allocation process, one platform can only be assigned to one decision-making entity, while one decision-making entity can command and control multiple platforms.

b) **Pheromone**: Pheromone τ_{mj} indicates the desirability of assigning platform P_j to decision-making entity DM_n. It encodes information about the ant' s path search process and is updated by the ants.

c) **Pheromone Global Update**: To avoid excessive pheromone accumulation caused by redundant pheromones, each ant dilutes old information and strengthens new information after completing clustering of NP platforms. Therefore, at time $t + n$, the pairing between platforms and decision-making entities is updated according to the global update rule:

$$\tau_{mj}(t+n) = (1-\rho) \cdot \tau_{mj}(t) + \Delta\tau_{mj}(t+n)$$

where ρ is the evaporation coefficient, representing the decay degree of pheromones over time, and $\Delta\tau_{mj}(t+n)$ represents the pheromone increment.

$$\Delta\tau_{mj}(t+n) = \sum_{k=1}^{M} \Delta\tau_{mj}^{k}(t+n)$$

where $\Delta\tau_{mj}^{k}(t+n)$ represents the pheromone amount left by the k-th ant on path (P_j, DM_n) at time $t + n$, with specific values determined by the ant' s performance:

$$\Delta\tau_{mj}^{k}(t+n) = \begin{cases} \frac{Q}{RMS_k}, & \text{if the } k\text{-th ant traverses path } (P_j, DM_n) \\ 0, & \text{otherwise} \end{cases}$$

where Q is a control constant and RMS_k represents the workload of the allocation scheme represented by the k-th ant.

d) **Termination Rule**: The algorithm ends when all platforms have been allocated or when the number of algorithm iterations exceeds a certain limit.

The transfer probability is calculated based on pheromones as follows:

$$P_{mj}^{k}(t) = \frac{[\tau_{mj}(t)]^{\alpha} \cdot [\eta_{mj}]^{\beta}}{\sum_{s \in \text{allowed}_k} [\tau_{ms}(t)]^{\alpha} \cdot [\eta_{ms}]^{\beta}}$$

where η_{mj} represents the heuristic information.

The decision-making entity configuration process based on the global update rule ant colony algorithm is as follows:

1. Initialize all parameters. Set the number of ants as M, maximum cycle number as $NC_{\max}$, initialize each pairing (P_j, DM_n), and set initial time $\tau_{mj}(0) = 0$, $\Delta\tau_{mj}(0) = 0$. Randomly generate M configuration schemes between decision-making entities and platforms, and eliminate infeasible configurations.
2. Calculate the workload of decision-making entities in different configuration schemes.
3. Calculate pheromone increments $\Delta\tau_{mj}^{k}(t+n)$ and solve for pheromone $\tau_{mj}(t+n)$.
4. Calculate transfer probabilities according to the formula above, select clustering schemes based on probabilities, and record the current optimal allocation scheme.
5. Determine whether the cycle number is less than the maximum cycle number. If yes, save the optimal individual and return to step 2; if greater, output the final result.

The platform clustering process based on the global update rule ant colony algorithm is shown in [Figure 1: see original paper].

3 Example Analysis

This paper uses a joint operation scenario as a simulation example to verify and analyze the proposed method. The case involves 18 tasks and 20 platforms, with 8 types of platform resource capabilities and task resource requirements. Task attribute data and platform attribute data can be found in existing literature. The allocation relationship between tasks and platforms is represented by a platform scheduling scheme Gantt chart, which serves as known input information for the decision-making entity configuration problem, as shown in [Figure 2: see original paper].

Simulation Experiment 1

Since the ant colony algorithm based on global update rules involves many parameters, and different parameter settings significantly affect the final platform clustering results, this experiment first compares and analyzes the impact of different algorithm parameters on clustering results to determine the optimal parameter settings for solving this problem.

1) Impact of Decision-Making Entity Quantity ND on Clustering Results

The algorithm initialization parameters are set as follows: number of ants $M = 25$, number of platforms $NP = 20$, number of tasks $NT = 18$, maximum cycle number $NC_{\max} = 50$, pheromone residual coefficient $Rho = 0.8$, pheromone total amount $Q = 100$, internal load weight $a = 0.5$, external load weight $b = 0.5$, information heuristic factor $\alpha = 0.8$, and expectation heuristic factor $\beta = 3$. Let the number of decision-making entities be $ND = 2, 3, ..., 10$. The variation of RMS value with ND is shown in [Figure 3: see original paper].

From [Figure 3: see original paper], we can see that the number of decision-making entities affects the variation trend of the objective function value. Generally, the fewer the decision-making entities, the larger the objective function RMS value, indicating that each decision-making entity' s load is greater when the number is small. However, more decision-making entities do not necessarily mean better clustering schemes. Constrained by operational reality, the number of decision-making entities is limited. Too many decision-making entities in a single campaign task will increase costs. From the curve trend, when the number of decision-making entities is between 4 and 6, the curve' s slope changes fastest, so the number of decision-making entities should be selected between 4 and 6. Considering all factors, we select $ND = 5$ as the parameter value for subsequent clustering processes.

2) Impact of Pheromone Residual Coefficient ρ on Clustering Results

Pheromone evaporation degree refers to the volatilization degree of pheromones left by ants on traversed paths. In the algorithm, ρ is the pheromone evaporation coefficient, and $1-\rho$ is the pheromone residual coefficient. The pheromone evaporation coefficient directly affects the convergence speed and global search capability of the ant colony algorithm. Due to the existence of the information volatilization coefficient, pheromones on paths not traversed by ants will gradually volatilize until reduced to 0. When the pheromone evaporation coefficient ρ is too large, pheromones volatilize too quickly, and some paths not traversed by ants will not be selected, reducing the algorithm' s global search capability. When ρ is too small, although the algorithm' s global search capability can be improved, its convergence speed will decrease.

The relationship between pheromone residual coefficient and algorithm performance is shown in . The relationship between pheromone residual coefficient and algorithm output RMS is shown in [Figure 4: see original paper]. The relationship between pheromone residual coefficient ρ and algorithm convergence iteration number is shown in [Figure 5: see original paper].

From the curve trend, we can see that when ρ is between 0.7 and 0.9, the output RMS is minimized. Therefore, considering convergence speed and global search capability, we select the pheromone evaporation coefficient $\rho = 0.3$, i.e., pheromone residual coefficient $1-\rho = 0.7$, as the parameter value for subsequent clustering processes.

3) Impact of Ant Quantity M on Clustering Results

In ant colony algorithms, the number of ants directly affects the group' s capability. For a single ant, completing one allocation constitutes a valid solution; all ants completing one allocation forms a solution set. Obviously, more ants mean a larger solution set and stronger global search capability, but slower convergence speed. Conversely, fewer ants mean a smaller solution set and weaker global search capability, but faster convergence speed. When the ant quantity is too small, although convergence speed improves, the algorithm easily falls into local optima, reducing global search effectiveness and causing premature stagnation.

The relationship between ant quantity M and RMS value and algorithm iteration number is shown in . From , we can see that when the ant quantity is too large, convergence speed decreases and affects the clustering results. The figure shows that when selecting $M = 25$ ants, the output result is smaller and iteration number is fewer, so we set $M = 25$ as the parameter value for subsequent clustering processes.

4) Heuristic Factors α and β

The information heuristic factor α reflects the relative importance of accumulated information in guiding ant colony search during movement, while the expectation heuristic factor β reflects the relative importance of heuristic information. A larger β value increases the possibility of ants selecting locally shortest paths at certain points, accelerating algorithm convergence but reducing randomness in optimal path search and making it easier to fall into local optima. The size of α reflects the intensity of random factors in path search; larger values increase the possibility of ants selecting previously traversed paths, reducing search randomness, and excessively large values can also cause premature convergence to local optima.

The impact of heuristic factors α and β on output results and algorithm convergence is shown in . From , we can see that when the information heuristic factor $\alpha = 0.8$ and the expectation heuristic factor $\beta = 3$, the algorithm output is superior and convergence speed is faster. Therefore, we set parameters $\alpha = 0.8$ and $\beta = 3$ as parameter values for subsequent clustering processes.

Simulation Experiment 2

Using the results from Experiment 1, the algorithm parameters are initialized as follows: ant quantity $M = 25$, decision-making entity quantity $ND = 5$, platform quantity $NP = 20$, task quantity $NT = 18$, maximum cycle number $NC_{\max} = 50$, pheromone residual coefficient $Rho = 0.8$, pheromone total amount $Q = 100$, internal load weight $a = 0.5$, external load weight $b = 0.5$, information heuristic factor $\alpha = 0.8$, and expectation heuristic factor $\beta = 3$. Using the ant colony algorithm based on global update rules, a platform clustering scheme is generated with the platform-task allocation relationship shown in [Figure 2: see original paper] as input information. The clustering results are shown in .

The algorithm convergence curve is shown in [Figure 6: see original paper]. From [Figure 6: see original paper] and , we can see that the ant colony algorithm based on global update rules is feasible for solving platform clustering problems. From the algorithm output results, the average workload of decision-making entities is 246, the workload RMS value is 256.2421, and the workload variance is 80.1873. From the algorithm convergence curve, the algorithm converges at the 33rd generation. Therefore, the ant colony algorithm demonstrates good performance in solving platform clustering problems, verifying the feasibility and applicability of the method.

Simulation Experiment 3

To further verify the superiority of the ant colony algorithm based on global update rules, we first compare it with hierarchical clustering algorithm and adaptive ant colony algorithm. Key parameters are set as: ant quantity $M = 25$, decision-making entity quantity $ND = 5$, platform quantity $NP = 20$, task quantity $NT = 18$, maximum cycle number $NC_{\max} = 50$, pheromone residual coefficient $Rho = 0.8$, pheromone total amount $Q = 100$, internal load weight $a = 0.5$, information heuristic factor $\alpha = 0.8$, and expectation heuristic factor $\beta = 3$. The output platform clustering results using hierarchical clustering method and adaptive ant colony algorithm are shown in and respectively.

Comparing the results in through , the ant colony algorithm based on global update rules obtains a final clustering objective function value of $RMS = 256.2421$, hierarchical clustering method obtains $RMS = 258.1859$, and adaptive ant colony algorithm obtains $RMS = 257.8759$. In terms of clustering results, the ant colony algorithm based on global update rules achieves the smallest RMS value, followed by adaptive ant colony algorithm, with traditional hierarchical clustering method being the largest. Therefore, ant colony algorithm is superior to traditional hierarchical clustering method for solving platform clustering problems.

To further verify the superiority of the global update rule, we compare the algorithm convergence using global update rule ant colony algorithm, adaptive ant colony algorithm, ordinary ant colony algorithm, and Quantum Genetic Algorithm (QGA). The convergence curves of each algorithm are shown in [Figure 7: see original paper].

From the convergence curve trends in [Figure 7: see original paper], we can intuitively see that the convergence effect of ant colony algorithm using global update strategy is similar to that of adaptive ant colony algorithm, but both the convergence effect and iteration number to reach convergence of the global update strategy are superior to QGA. This further verifies the superiority of ant colony algorithm using global update strategy in solving platform clustering models.

4 Conclusion

This paper primarily addresses the decision-making entity configuration problem in C2 organizations, also known as the platform clustering problem. It first defines the workload of decision-making entities from external and internal aspects in a combat-task-oriented manner, establishes a decision-making entity configuration problem model with the RMS value of workload as the objective function, and uses an ant colony algorithm based on global update rules to solve the problem. Finally, simulation examples verify the algorithm' s applicability and superiority. Experimental results show that reasonable algorithm parameter settings can achieve better clustering results. Compared with traditional hierarchical clustering method, this method can obtain smaller objective function values, and in terms of convergence speed, it outperforms traditional ant colony algorithm and adaptive ant colony algorithm, effectively solving the decision-making entity configuration problem.

Future work will focus on research regarding optimization and adjustment of decision-making entity configuration schemes under uncertain environments.

References

[1] Bui H, Han X, Mandal S, et al. Optimization-based decision support algorithm for a team-in-the-loop planning experiment [C]// Proc of IEEE International Conference on Systems, Man, and Cybernetics, 2009.

[2] Yang Dongsheng, Zhang Weiming, Liu Zhong, et al. Command and Control Organization Design Method [M]. Beijing: National Defense Industry Press, 2010: 161-189.

[3] Han X, Mandal S, Pattipati K R, et al. An optimization-based distributed planning algorithm: A blackboard-based collaborative Framework [J]. IEEE Trans on Systems, Man, and Cybernetics-Part A: Systems and Humans, 2014, 44 (6): 673-684.

[4] Sun Yu, Yao Peiyang, Zhang Jieyong. Measurement and comprehensive evaluation of C2 organizational information structure efficiency [J]. Systems Engineering and Electronics, 2015, 37 (6): 1313-1318.

[5] David S, Kevin S. On optimizing command and control structures [C]// Proc. of the 16th International Command and Control Research Symposium. 2011: 1-10.

[6] Levchuk G M, Levchuk Y N, Luo J, et al. Normative design of organizations –Part : mission planning [J]. IEEE Trans on Systems, Man, and Cybernetics-Part A: Systems and Humans, 2002, 32 (3): 311-322.

[7] Levchuk G M, Levchuk Y N, Luo J, et al. Normative design of organizations–Part : organizational structure [J]. IEEE Trans. on Systems, Man, and Cybernetics-Part A: Systems and Humans, 2002, 32 (3): 322-339.

[8] Zhang Jieyong, Yao Peiyang. Model and solving method for collocating problem of decision-makers in C2 organization [J]. Systems Engineering and Electronics, 2012, 34 (4): 737-742.

[9] Wu Ruijie, Sun Peng, Sun Yu, et al. Self-adaption quantum genetic algorithm used in design of command and control structure [J]. Application Research of Computers, 2016, 33 (7): 113-119.

[10] Zhou Xiangxiang, Yao Peiyang, Yin Zhonghai. A method for design of command decision-making entity' s hiberarchy architecture [J]. Electronics Optics & Control, 2012, 19 (2): 68-72.

[11] Sun Yu, Yao Peiyang, Wu Jixiang, et al. Design method of flattening command and control structure of army organization [J]. Systems Engineering and Electronics, 2016, 38 (8): 1833-1839.

[12] Naz-afarin M, Vahidreza G. Integration of efficient multi-objective ant-colony and a heuristic method to solve a novel multi-objective mixed load school bus routing model [J]. Applied Soft Computing, 2018, 68: 566-579.

[13] Wang Shengyao, Wang Ling, Xu Ye, et al. An estimation of distribution algorithm for solving hybrid flow-shop scheduling problem [J]. Acta Automatica Sinica, 2012, 38 (3): 437-443.

[14] Soumya B, Abhishek P. Test Suite Minimization in Regression Testing Using Hybrid Approach of ACO and GA [J]. International Journal of Applied Metaheuristic Computing, 2018, 9 (3): 88-104.

[15] Xin Ning, Jingyan Qi, Chunlin Wu, Wenjuan Wang. A tri-objective ant colony optimization based model for planning safe construction site layout [J]. Automation in Construction, 2018, 89: 1-12.

[16] Ziyang Zhen, Dongjing Xing, Chen Gao. Cooperative search-attack mission planning for multi-UAV based on intelligent self-organized algorithm [J]. Aerospace Science and Technology, 2018.

[17] Zhang Bangchu, Peng Chen, Xu Rongxin, et al. Cooperative target assignment algorithm of multiple missiles based on ant colony optimization [J]. Journal of Projectiles, Rockets, Missiles and Guidance, 2012, 32 (4): 69-73.

Note: Figure translations are in progress. See original paper for figures.