

A Self-adaption Quantum Genetic Algorithm Used in the Design of Command and Control Structure

SUN Peng^{1,3,a}, WU Jun-sheng², ZHANG Jie-yong^{3,*}, LIAO Meng-chen³

1. School of computer science, Northwestern Polytechnical University, Xi'an, China
2. School of software and microelectronics, Northwestern Polytechnical University, Xi'an, China
3. College of Information and Navigation, Air Force Engineering University, Xi'an, China

^a281126096@qq.com

Keywords: Work load, Root mean square, Quantum genetic algorithm, Self-adaption.

Abstract. The paper mainly solved the problem of the design of command and control structure. First, the key elements in command and control structure are defined and we introduced the concept of work load of decision makers to describe the problem mathematically. Next, a mathematic model aimed at minimizing the root mean square of decision makers' work load is developed. Finally, we combine the quantum genetic algorithm with self-adaption strategy and get the self-adaption quantum genetic algorithm. Major characteristic of this algorithm is adjusting the quantum rotation gate, generating the crossover probability and the mutation probability in a self-adaption way. Experimental results show that the self-adaption quantum genetic algorithm has a feature of evolving fast and searching precise, and it can cluster the platforms well to accomplish the design of command and control structure.

1 Introduction

The design of command and control structure is the key issues in army organization. A reasonable command and control structure can improve the command efficiency of the whole army command [1][2][3], and balance the workload of each decision makers [4][5]. In the design of the command and control structure, we divide all platforms into several non-overlapping groups. Each group is distributed a decision maker [6]. So the key problem of command and control structure design is clustering platforms.

Platform clustering is the second part of content of the three-phase design of adaptive organization [7][8][9]. Levchuk sets up the mathematical model with the target of minimize the maximum workload of decision makers [8]. Zhang set the task processing time as the measurement of decision-makers workload and use hierarchical clustering method to solve the problem in [10]. Hierarchical clustering method use greedy strategy in each step which can only get a local optimal solution. The above models are single objective model and Sun put forward multi-objective

optimization model in [11] which can get several forefront solutions.

In this paper, we come up with the concept of task complexity to measure the workload of decision-makers and set up a mathematical model minimizing the RMS (Root Mean Square) of decision-makers workload as the objective function. Finally, we put forward a SAQGA (Self-Adaption Quantum Genetic Algorithm, SAQGA) to solve the model.

2 Definition

2.1 Elements in the design of command and control

(1) A task is a goal that needs to be accomplished during combat. Typically, completing a task requires the use of one or more platforms. Task set is $T = \{T_i\} (i = 1, \dots, I)$.

(2) Platform is the basic unit for executing tasks. Platform set is $P = \{P_j\} (j = 1, \dots, J)$.

(3) Decision-maker(DM) is the essential element of command and control in military organization, which has the ability of information processing and commanding. DM set is $DM = \{DM_m\} (m = 1, \dots, D)$.

2.2 Variable

(1) Assignment variable between task and platform $\omega_{ij} (i = 1, \dots, I; j = 1, \dots, J)$, $\omega_{ij} = 1$ means assign platform j to task i ; $\omega_{ij} = 0$ means there is no relationship between platform and task.

(2) Control variable between DM and platform $x_{mj} (m = 1, \dots, D; j = 1, \dots, J)$, $x_{mj} = 1$ means DM_m control platform P_j ; $x_{mj} = 0$ means there is no relationship between platform and DM.

(3) Execution variables for DM $u_{mi} (m = 1, \dots, D; i = 1, \dots, I)$, $u_{mi} = 1$ means DM_m execute task T_i ; $u_{mi} = 0$ means there is no relationship between task and DM.

3 Problem model

Figure 1 simply describes the relationship among DMs, tasks, and platforms, and there is collaborative relationships among DMs.

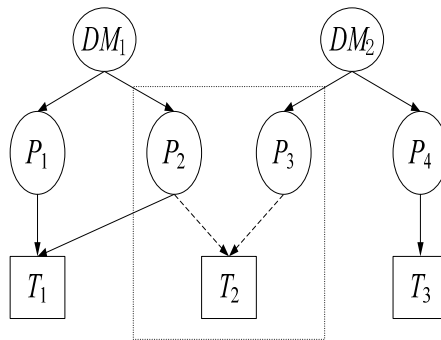


Fig.1 Relationship among DMs, tasks and platforms

3.1 Definition of working load of decision-makers

In the operational command, the decision-makers command platform to perform a number of tasks. Different tasks have different complexity.

(1) In the case of the same number of platforms, the complexity of the charge of the collaborative task is greater than the complexity of the task that does not require collaboration.

(2) In the case of the same collaboration relationship, the complexity of the number of tasks required for the platform is larger than the number of platforms, and the complexity of the task is less than the number.

κ_{mi} is the complexity of decision-maker m executing task i :

$$\kappa_{mi} = \sqrt{\sum_{j=1}^J x_{mj} \cdot \omega_{ij}} + \sum_{\substack{n=1, \\ n \neq m}}^D \sum_{j=1}^J x_{nj} \cdot \omega_{ij} \quad (1)$$

The workload of decision-maker m executing task i is:

$$load(m, i) = \kappa_{mi} \cdot t_i \quad (2)$$

The workload of decision-maker m is sum of all tasks workload:

$$CW(m) = \sum_{i=1}^{i=l} \left(\sqrt{\sum_{j=1}^J x_{mj} \cdot \omega_{ij}} + \sum_{\substack{n=1, \\ n \neq m}}^D \sum_{j=1}^J x_{nj} \cdot \omega_{ij} \right) \cdot t_i \quad (3)$$

3.2 Optimization objective

The essence of command and control structure design is to generate platform clustering scheme. All platforms are divided into a number of platforms, each of which is controlled by a decision entity. According to the 2.1 decision of the decision-makers workload, we measure the objective function from two aspects.

(1) Average workload of decision-makers:

$$\mu = \frac{1}{D} \sum_{m=1}^D CW(m) \quad (4)$$

(2) Variance workload of decision-makers:

$$\sigma^2 = \frac{1}{D} \sum_{m=1}^D [CW(m) - \mu]^2 \quad (5)$$

This paper uses the mean square root entity of decision load measure clustering scheme is good or bad. Literature [9] shows in a team, minimizing the mean and variance of the decision-makers workload is equal to minimize the RMS. RMS is equal to the expectation and standard deviation of the square value.

$$RMS = \sqrt{\frac{1}{D} \sum_{m=1}^D CW^2(m)} = \sqrt{\sigma^2 + \mu^2} \quad (6)$$

To sum up, the optimization objectives of the whole model are as follows:

$$\begin{aligned} & \min_x RMS \\ s.t. & \sum_{m=1}^D x_{mj} = 1 \quad \sum_{j=1}^J x_{mj} \geq 1 \quad x_{mj} \in \{0, 1\} \end{aligned} \quad (7)$$

4 Problem solving

4.1 Quantum genetic algorithm

Quantum genetic algorithm [13] and the traditional genetic algorithm is mainly reflected in two points. One is using quantum bit encoding to represent chromosomes, genes expressed in the form of probability amplitude, the other is using quantum rotating gate to update population instead of

selecting, crossover and mutation operation.

M bit quantum bit code can be expressed as:

$$\begin{bmatrix} \alpha_1 & \alpha_2 & \cdots & \alpha_m \\ \beta_1 & \beta_2 & \cdots & \beta_m \end{bmatrix} \quad (8)$$

(α_i, β_i) is the i^{st} qubit of the chromosome, and it is a pair of plural. It meets $|\alpha|^2 + |\beta|^2 = 1$, and $|0\rangle, |1\rangle$ represent two different bit states. Without observation, the quantum is in the basic state, after observing, the quantum will collapse to 0 or 1 with probability 0.5 or 0.8. The qubit is updated as follows

$$\begin{bmatrix} \alpha'_i \\ \beta'_i \end{bmatrix} = \begin{bmatrix} \cos \Delta\theta_i & -\sin \Delta\theta_i \\ \sin \Delta\theta_i & \cos \Delta\theta_i \end{bmatrix} \begin{bmatrix} \alpha_i \\ \beta_i \end{bmatrix} \quad (9)$$

(α'_i, β'_i) represents the qubits updated, and $\Delta\theta_i$ is the quantum rotation angle.

4.2 Self-adaptive strategy

The calculation formulas of the quantum rotation angle, crossover probability and mutation probability in the adaptive strategy are as follows:

$$\Delta\theta = \theta_{\min} + \frac{f_{\max} - f_x}{f_{\max} - f_{\min}} (\theta_{\max} - \theta_{\min}) \quad (10)$$

$$P_c = \begin{cases} \frac{k_1(f_{\max} - f)}{f_{\max} - f_{avg}}, & f \geq f_{avg} \\ k_2, & f < f_{avg} \end{cases} \quad (11)$$

$$P_m = \begin{cases} \frac{k_3(f_{\max} - f')}{f_{\max} - f_{avg}}, & f' \geq f_{avg} \\ k_4, & f' < f_{avg} \end{cases} \quad (12)$$

f_{\max} is the maximum fitness in the population. f_{avg} is the average fitness of population. f is the greater fitness of the two individuals to be crossed. f' is the adaptive degree of individual variation. f_x is the current individual fitness. θ_{\max} is the maximum value for interval $\Delta\theta$. θ_{\min} is the minimum value for interval $\Delta\theta$. k_1, k_2, k_3, k_4 is constant.

4.3 SAQGA

Algorithm steps are as follows:

Step 1: Initial quantum bit code, $\alpha_i, \beta_i (i=1, \dots, m) = 1/\sqrt{2}$, ensure the quantum bit collapse to the same probability of 0 or 1 at the beginning;

Step 2: Observed quantum bit code and got the binary code, calculate the fitness of each individual.

Step 3: Select optimal individuals as the next generation of evolutionary goals. Adjust the quantum rotation angle adaptively to update population according to the formula (10). Do quantum crossover operation.

Step 4: Generating crossover and mutation probability adaptively according to the formula (11)(12). If the iteration stops, output the result; otherwise return to step 2.

5 Experiment simulation

5.1 Experimental case

The experimental data used in this paper is from the MOC-1 experiment carried out by the US naval Graduate School (NPS) supported by the U.S. Department of defense [5]. The matching

relationship between task and platform is shown in Table 1, Simulation runs on Intel (R) Core CPU 2.5GHz i5-2450M computer.

Tab. 1 Assignment relationship between tasks and platforms

Task	Platform	Execution time (h)	Task	Platform	Execution time (h)
T ₁	P ₁	30	T ₇	P ₆ P ₁₂	24
T ₂	P ₂ P ₄ P ₁₇	30	T ₈	P ₁₅	24
T ₃	P ₉ P ₁₃ P ₁₉	30	T ₉	P ₃ P ₁₁	30
T ₄	P ₈ P ₁₀	30	T ₁₀	P ₁₆ P ₁₈	18
T ₅	P ₅ P ₁₄	10	T ₁₁	P ₇ P ₂₀	30
T ₆	P ₅ P ₇	4			

5.2 Analysis

Experiment 1: In this experiment, the number of decision-makers is 4. The maximum workload is 94.7, and the minimum workload is 81.9. The average working load is 86.85 and the RMS value is 86.98. Hierarchical clustering algorithm is a common method to solve the clustering problem. The clustering scheme used in this case is shown in Table 3. The maximum load is 115.90, the minimum working load is 84.85, the average working load is, and the RMS is about 92.32. Compared with the hierarchical clustering algorithm, the average workload using SAQGA algorithm is lower than that of the hierarchical clustering algorithm 9.78%, and the RMS increased by 10.54%. It is also noted that the workload of DM2 has fallen from 94.7 to 86.2, which shows that the SAQGA algorithm can effectively balance the load of each decision entity, avoid the load variance is too large.

Tab. 2 SAQGA algorithm

DM	SAQGA		Average RMS
	Platform	Workload	
DM ₁	P ₃ P ₈ P ₁₀ P ₁₁	84.8528	86.98
DM ₂	P ₅ P ₇ P ₁₄ P ₁₅ P ₁₆ P ₁₈ P ₂₀	94.7107	
DM ₃	P ₁ P ₂ P ₄ P ₁₇	81.9615	
DM ₄	P ₆ P ₉ P ₁₂ P ₁₃ P ₁₉	85.9026	

Tab. 3 Hierarchical clustering algorithm based on minimum RMS

DM	Hierarchical clustering algorithm based on minimum RMS		Average RMS
	Platform	Workload	
DM ₁	P ₃ P ₈ P ₁₀ P ₁₁	84.8528	96.15
DM ₂	P ₅ P ₇ P ₁₄ P ₁₅ P ₂₀	86.2254	
DM ₃	P ₂ P ₄ P ₁₆ P ₁₇ P ₁₈	94.3879	
DM ₄	P ₁ P ₆ P ₉ P ₁₂ P ₁₃ P ₁₉	115.9026	

Experiment 2: In order to further verify the superiority of SAQGA algorithm, we compare the quantum genetic algorithm (Quantum Genetic Algorithm, QGA) with SAQGA algorithm. Two search algorithms were iterated for 200 times, population size of 200, the evolution curve shown in Figure 2. QGA algorithm converges in the 130 generation, RMS value is 96, and SAQGA algorithm

converges in the 40 generation, RMS value of 87. In contrast, the SAQGA algorithm has the characteristics of fast convergence and good search results.

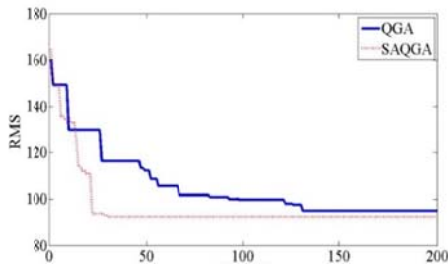


Fig. 2 Iterative process

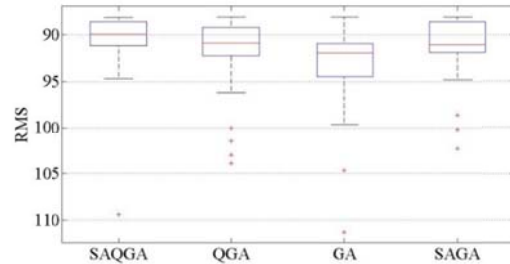


Fig. 3 Monte Carlo simulation

Experiment 3: In order to analyze the statistical characteristics of SAQGA algorithm, the SAQGA algorithm is compared with QGA algorithm, genetic algorithm (GA) and adaptive genetic algorithm (SAGA). We have achieved across 50 Monte-Carlo simulation experiments for four kinds of algorithms, the experimental results are shown in Figure 3. The horizontal line represents the median statistics in boxplots, highly representative of distribution box, outside the box of discrete points represent outliers. The experimental results show that the SAQGA algorithm has the best convergence, the least number of outliers, and the median RMS value is the lowest.

6 Summary

In this paper, a mathematical model is established to minimize the RMS, and the SAQGA algorithm is used to solve the model. The experimental results show that the SAQGA algorithm can effectively solve the problem of clustering platform.

References

- [1] Mandal S, Han X, Pattipati K R: *Agent-Based Distributed Framework for Collaborative Planning*. Proceedings of the 2010 IEEE International Conference on Aerospace conference. Big Sky, MT(2010)p.1-11.
- [2] Mandal S, Bui H and Han X, in: *Optimization-based decision support software for a team-in-the-loop experiment: Asset package selection and planning*. IEEE Transactions on Systems, Man, and Cybernetics (2013).
- [3] Yu F, Tu F and Pattipati K R, in: *Integration of a holonic organizational control architecture and multi-objective evolutionary algorithm for flexible distributed scheduling*. IEEE Transactions on System, Man, and Cybernetics-Part A: System and Humans (2008).
- [4] Hutchins S G, Kemple W G and Kleinman D L: *Maritime operations centers with integrated and isolated planning teams*, in proc. 15th ICCRTS (2010) p. 1-26.
- [5] Hutchins S G, Kemple W G and Kleinman D L: *Maritime headquarters with maritime operations center: A research agenda for experimentation*, in proc. 14th Int. Command Control Res. Technol. Symp (2009), p. 1-32.
- [6] Han X, Mandal S, Pattipati K R, and Kleinman D L, in: *Optimization-based Distributed Planning Algorithm: A Blackboard-Based Collaborative Framework*. IEEE Transactions on Systems, Man, and Cybernetics (2014).
- [7] Levchuk G M, Levchuk Y N and Luo J, in: *Normative Design of Organizations—Part I: Mission Planning*. IEEE Transactions on Systems, Man, and Cybernetics (2002).
- [8] Levchuk G M, Levchuk Y N, Luo J and Pattipati K R. *Normative design of organizations-PartII: Organizational structure*. IEEE Transactions on Systems, Man, and Cybernetics (2002).
- [9] Levchuk G M, Levchuk Y N, Luo J, Pattipati K R, and Kleinman D L. *Normative design of project-based organizations-PartIII: Modeling congruent, robust, and adaptive organizations*. IEEE Transactions on Systems, Man, and Cybernetics (2004).
- [10] Zhang J Y, Yao P Y: *Model and solving method for collocating problem of decision-makers in C2 organization*. Systems Engineering and Electronics (2012).

- [11] Sun Y, Yao P Y and Wu J X: *Design method of flattening command and control structure of army organization*. Systems Engineering and Electronics (2016).
- [12] Sun Y, Yao P Y and Li M H. *Adaption adjusting method of command and control structure of army organization*. Systems Engineering and Electronics (2016).
- [13] Han K H, Kim J H: *Quantum-inspired evolutionary algorithm for a class of combinatorial optimization*. Evolutionary Computation, IEEE Transactions on (2002) p.580-593.
- [14] Shor PW: *Algorithms for quantum computation: discrete logarithms and factoring*. Foundations of Computer Science, 1994 Proceedings, 35th Annual Symposium on. IEEE (1994) p. 123-134.
- [15] Li S Y, Li P C: *Quantum computation and quantum optimization algorithm*. Harbin industrial university press (2009).