

RESEARCH ARTICLE

Multi-AUV autonomous task planning based on the scroll time domain quantum bee colony optimization algorithm in uncertain environment

Jianjun Li^{1,2*}, Rubo Zhang^{1,3}, Yu Yang²

1 College of Computer Science and Technology, Harbin Engineering University, Harbin, China, **2** School of Computer and Information Engineering, Harbin University of Commerce, Harbin, China, **3** College of Electromechanical & Information Engineering, Dalian Nationalities University, Liaoning Dalian, China

* hrbec2008@126.com



Abstract

Research on distributed task planning model for multi-autonomous underwater vehicle (MAUV). A scroll time domain quantum artificial bee colony (STDQABC) optimization algorithm is proposed to solve the multi-AUV optimal task planning scheme. In the uncertain marine environment, the rolling time domain control technique is used to realize a numerical optimization in a narrowed time range. Rolling time domain control is one of the better task planning techniques, which can greatly reduce the computational workload and realize the tradeoff between AUV dynamics, environment and cost. Finally, a simulation experiment was performed to evaluate the distributed task planning performance of the scroll time domain quantum bee colony optimization algorithm. The simulation results demonstrate that the STDQABC algorithm converges faster than the QABC and ABC algorithms in terms of both iterations and running time. The STDQABC algorithm can effectively improve MAUV distributed tasking planning performance, complete the task goal and get the approximate optimal solution.

OPEN ACCESS

Citation: Li J, Zhang R, Yang Y (2017) Multi-AUV autonomous task planning based on the scroll time domain quantum bee colony optimization algorithm in uncertain environment. PLoS ONE 12 (11): e0188291. <https://doi.org/10.1371/journal.pone.0188291>

Editor: Yong Deng, Southwest University, CHINA

Received: March 6, 2017

Accepted: September 12, 2017

Published: November 29, 2017

Copyright: © 2017 Li et al. This is an open access article distributed under the terms of the [Creative Commons Attribution License](https://creativecommons.org/licenses/by/4.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Data Availability Statement: All relevant data are within the paper and its Supporting Information files.

Funding: The authors received no specific funding for this work.

Competing interests: The authors have declared that no competing interests exist.

Introduction

The Autonomous Underwater Vehicle Task Planning (AUVTP) is based on the AUV task, AUV task load and quantity, pre-setting and coordinating the specific combat mission. Task planning includes task allocation, route planning, sensor planning and link planning. From time division, AUVTP can be divided into real-time planning and pre-planning. Real-time planning is a viable route for AUV during the voyage, based on actual navigation conditions and subsea environmental changes, including options for pre-planned revisions and contingency plans[1,2]. Pre-planning is AUV pre-sailing to develop, comprehensive mission requirements, the marine environment and communication conditions and other information, the development of long-term mission planning. Real-time mission planning, due to the process

of navigation, emphasizes real-time planning capabilities in dynamic, uncertain battlefield environments, and faces more complex and greater challenges than pre-planning.

In the high combat environment, the battlefield information changes, the need for AUV to reduce the manual participation and reduce the dependence on other systems, can independently adjust the program and re-planning, to achieve truly intelligent self-planning. Intelligent autonomous planning is the further development of artificial intelligence technology, is the ultimate goal of mission planning. At present, some local independent planning and control technology has made some development, such as autonomous search, autonomous target recognition, autonomous fire distribution and so on [3–5].

Beyond achieving various underwater applications, the motion control of AUV is one of the essential problems in attaining underwater operational objectives [6,7]. However, the modelling parameters of the underwater vehicle is difficult to be accurately acquired and the vehicle is also vulnerable by environmental disturbances including ocean currents and waves [8]. These internal and external uncertainties along with the nonlinearity of the vehicle dynamics render the AUV control problem difficult.

With the rapid development of swarm intelligence algorithms, many researchers have simulated insect foraging behavior and have introduced a response threshold model to assign tasks; other experts have introduced the ant colony algorithm to solve the large-scale task allocation problem based on the time series [9–11]. Some experts have also designed a task model and proposed an improved discrete particle swarm optimization algorithm to solve the problem [12,13]. These methods provide a new way to solve the problem of allocating tasks among multiple robots. The present paper explores multi-AUV autonomous task planning, especially the rolling time domain AUV mission planning in uncertain environment, and use the bionic task planning method of quantum bee optimization algorithm to explore the AUV autonomous mission planning theory.

Quantum bee colony optimization algorithm

Quantum artificial bee colony algorithm

The bee colony optimization algorithm is a type of meta-heuristic optimization method to imitate the behavior of natural bees. Ferrante et al. [14] proposed a self-organization model, which was applied to task partitioning. Grozinger [15] proposed a self-organizing model, which showed the communication in the bee colony through many methods, including "swing dance" and odor. This self-organization model can complete different tasks in different social classes. Karaboga et al. [16] successfully applied the colony algorithm to the problem of function extremum optimization and systematically introduced the artificial bee colony (ABC) model. Civicioglu and Besdok [17] analyzed a conceptual comparison of the Cuckoo search, particle swarm optimization, differential evolution and artificial bee colony algorithms. Loubière et al [18] proposed a sensitivity analysis method for driving the artificial bee colony algorithm's search process, a new approach to random selection in neighborhood search. Quantum calculation is a new computational method based on quantum mechanics theory. Quantum Artificial Bee Colony Algorithm (QABC) uses the quantum revolving door to realize the optimal position search of the bee, and uses the qubit to encode the current position of the bee. The quantum bezel is used to realize the variation of the bee position and avoid premature convergence Phenomenon occurs.

In the quantum space, the particle state $ih\frac{\partial}{\partial t}\varphi(X, t) = H\varphi(X, t)$ is represented by the wave function $\varphi(X, t)$, where H is the Hamiltonian operator and h is Planck's constant. If the particle undergoes a one-dimensional potential well movement at the center point of Q , the position

determined by the stochastic equation is $X = Q \pm \frac{h^2}{2m\gamma} \ln(1/u)$, where m is the particle mass and u is a random number in the interval of $(0, 1)$ [19].

Thus, we can obtain a formula of the quantum bee colony optimization algorithm:

$$X_{i,j}(t + 1) = Q_{i,j}(t) \pm \lambda |X_{i,j}(t) - X_{i \neq j,j}(t)| \ln(1/u_{i,j}(t)) \tag{1}$$

In the formula, i is the bee number, j is the dimension, $X_{i,j}$ is the bee optimization position, and λ is a constant. In addition,

$$Q_{i,j}(t) = \alpha_j(t) \times Q_{i,j}(t) + (1 - \varphi_j) \times G_j(t) \tag{2}$$

In the formula, α_j is a random number in the interval of $(0, 1)$, $Q_{i,j}(t)$ is the best estimate of the current position of an individual bee, and $G_j(t)$ is the best estimate of the current position of all bees.

The best estimate of the position of the i -th bee is

$$Q_i(t) = \begin{cases} X_i(t) & f[X_i(t)] < f[Q_i(t - 1)] \\ Q_i(t - 1) & f[X_i(t)] \geq f[Q_i(t - 1)] \end{cases} \tag{3}$$

The best estimate of the global position is determined by $g = \arg \min_{1 \leq i \leq m} \{f[Q_i(t)]\}$ and $G(t) = Q_g(t)$.

Task allocation model based on quantum bee colony optimization algorithm

The managers are denoted by AUV_α in the distributed contract net. They are responsible for managing the task, and the other AUV_i are responsible for bidding the task. The task allocation process includes four steps: task bidding, bid, bid winning and task execution based on the contract net. The contract net task allocation model based on the differential evolution quantum bee colony algorithm is as follows:

Assume that there are N_V AUVs, $Task = \{Task_1, Task_2, \dots, Task_{N_M}\}$, the number of task targets is N_M , $V = \{V_1, V_2, \dots, V_{N_V}\}$, the number of AUVs is N_V , $Menace = \{Menace_1, Menace_2, \dots, Menace_{N_Q}\}$, and the number of threat sources is N_Q . The AUVs, task targets, and threat sources can include many types. If the same type of task is performed by different AUVs, the implementation effect is different. Assuming that the task set assigned to AUV_i is $T_i = \{Task_i^1, Task_i^2, \dots, Task_i^{n_i}\}$, the multi-AUV distributed task allocation problem can be translated as follows: Assign the existing tasks to multiple AUVs in the shortest possible time, i.e., $\bigcup_{i=1}^{N_V} T_i = Task$; each AUV has only one task, i.e., $\forall i, j \in \{1, \dots, N_V\}, i \neq j$, and $T_i \cap T_j = \emptyset$. If the maximum number of tasks executed by the multi-AUV system is less than the number of tasks that should be allocated, the assignment can be optimized to improve the overall efficiency of the multi-AUV task allocation system according to the following objectives.

Objective one: To maximize the overall effectiveness $\sum_{i=1}^{N_V} \theta_i(T_i)$ of the AUV after finishing the task, $\theta_i(T_i)$ is the performance after the task set T_i is completed by V_i .

Objective two: To minimize the required time $\max_{i \in V} Time_i(T_i)$ of the task to be completed by the AUV, $Time_i(T_i)$ is the time at which the task set T_i is finished by V_i .

Objective three: To balance the task load of each AUV, $\sum_{i=1}^{N_V} |Tload_i(T_i) - \overline{Tload}|$ is minimized, where $Tload_i(T_i)$ is the task load of V_i and \overline{Tload} is the average task load for each AUV.

Multi-AUV autonomous task planning in uncertain environment

Scroll time domain task allocation model

Scroll time domain control is a numerical optimization problem in a reduced time range. For multi-AUV applications, rolling time domain control is one of the better task planning techniques. In an uncertain environment, since the sensor information can be embedded in the online solution, the task execution can be handled very well; at the same time, because only the local information is embedded, the calculated workload can be greatly reduced. The cost function is weighed between the AUV dynamics state, the environment and the cost, and the final cost function is submitted to the online task planning to ensure that the task objectives are completed and the approximate optimal solution is obtained.

The scroll time domain task assignment is a given task set T , The distance between tasks $d(i, j)$, And lists the specified length L_c the sequence of tasks in memory[20, 21]. The estimate of each task sequence is

$$R_{VP} = \sum \lambda T_{ip} R_{wd} \tag{4}$$

Where T_{ip} is the time at which task p is completed in task sequence i , R_{wd} is the assigned task weight, and λ is the time scale factor.

When the values of all task sequences R_{VP} are given, the scrolling time domain task allocation algorithm can select the optimal task sequence for AUV.

$$\begin{aligned} \max L &= \sum_{p=1}^{N_{VP}} \sum_{V=1}^{N_V} R_{VP} X_{VP} \\ \sum_{V=1}^{N_V} \sum_{P=1}^{N_{VP}} A_{VP} X_{VP} &\leq 1 (X_{VP} \in \{0, 1\}) \\ \sum_{p=1}^{N_{VP}} x_{vp} &\leq 1 (\forall v \in 1, \dots, N_V) \end{aligned} \tag{5}$$

Where x_{vp} is a binary variable, its value is 1 when the P^{th} task sequence is selected, otherwise its value is 0; when task P in the task sequence i is searched, the value of A_{VP} is 1, otherwise it is 0.

Time adjustment parameter setting, parameter λ_{et} indicates K^{th} update task arrival time estimation value, completion time of task time depends on test state time T_c , h is the time value given by the bottom clock. If $\lambda_{et} > \delta_{min}$, it will be updated, where δ_{min} is a relatively fixed user fixed value.

$$\lambda_{et} = |T^k - T_c - h| \tag{6}$$

Space adjustment parameter setting, X_t for the reference configuration; X for the measured configuration value.

$$\lambda_s = |X_t - X| \tag{7}$$

If $\lambda_s > \alpha_{min}$, it will be updated, and parameters δ_{min} and α_{min} affect the number of updates.

Multi-AUV distributed task allocation with balanced coefficient

To enable multiple AUVs to quickly complete the task and achieve global optimization, first, the task is distributed to the entire AUV team with the smallest cost using the contract net to ensure the global optimization of task implementation. Then, the balance coefficient is used to make the entire AUV team distribute and achieve the tasks in the shortest time.

The balanced coefficient B_{eq}^R is introduced in the contract net distributed robot task allocation. Each robot uses its cost function to count the workload: the workload is the cost of robot R in the entire process of the work. Each robot broadcasts its workload to the entire team and calculate its B_{eq} . The formula of the balance coefficient for robot R is as follows:

$$B_{eq}^R = \frac{wa(R) - \overline{wa}}{\overline{wa}} \tag{8}$$

where \overline{wa} is the average workload of all robots in the team.

$B_{eq}^R < 0$: robot R has a lighter workload than the other robots;

$B_{eq}^R > 0$: robot R has a heavier workload than the other robots;

$B_{eq}^R > B_{eq}^{R1}$ 0: robot R has a heavier workload than robot R1;

In the contract net, the robot can take the task at the minimum cost, and the workload to be obtained should not be excessive. Thus, the task can be estimated from the balance coefficient B_{eq} . The formula of the task is estimated by robot R as follows:

$$rt'^R(T_1) = rt^R(T_1) - B_{eq}^R \times |rt^R(T_1)| \tag{9}$$

The task can be estimated using the balance coefficient B_{eq} of robot R. The following effects can be obtained:

1. A robot with a larger workload cannot easily obtain new tasks, and its tasks are more likely to be reassigned because its task utility is low.
2. A robot with a smaller workload easily obtains new tasks and does not easily give up its task because its task utility is high.

AUV autonomous learning mechanism

AUV has an autonomous learning mechanism that needs to include an implementation component, Multi-Level Executive (MLE) and Deliberative Layer (DL). Including two major execution instruction mechanisms.

First, the dynamic task is inserted. Insert the new task into the relative position that has been ranked in the task sequence according to the insertion model.

Second, the dynamic task is terminated. If the task has not yet started running, the termination of the task is canceled. If the task has already started running, the termination of the task is interrupted.

Multi-AUV system flexibility and openness, multi-AUV system can be combined with the group of swarm intelligence algorithm. AUV as an agent can not only interact with the optimal AUV in the current population, but also can self-study in multiple iterations to complete the accumulation of knowledge, so as to improve the ability to solve the problem, to achieve group intelligence optimization.

Let the position of AUV S_{ij} in the solution space be $S_{ij} = (s_1, s_2, \dots, s_n)$, the size of the solution space is $mS_{size} \times mS_{size}$, and the position formula of each AUV $mS_{i'j'} (i', j' = 1, 2, \dots, mS_{size})$ is:

$$mS_{i'j'} = \begin{cases} S_{i,j} & i' = 1, j' = 1 \\ SS_{i',j'} & \text{Other} \end{cases} \tag{10}$$

Formula $SS_{i',j'} = (SS_{i',j',1}SS_{i',j',2} \dots SS_{i',j',n})$, where $SS_{i',j',n}$ is:

$$SS_{i',j',n} = \begin{cases} L_{nmin}, & S_n rand(1 - mR, 1 + mR) < L_{nmin} \\ L_{nmax}, & S_n rand(1 - mR, 1 + mR) > L_{nmax} \\ S_n rand(1 - mR, 1 + mR), & \text{Other} \end{cases} \tag{11}$$

In the formula, mR is the local search radius, and $mR \in [0,1]$, $rand(1 - mR, 1 + mR)$ is a random number of $(1 - mR, 1 + mR)$.

Because AUV has the ability of autonomous learning mechanism, so as long as the optimal state of each iteration process of self-learning, you can achieve their own task control ability to improve. This not only improves the efficiency of the algorithm, but also re-searches the current optimal state to improve the accuracy of the search algorithm.

Experimental results and analysis

Experimental parameter assignment

To evaluate the performance of the Multi-AUV autonomous task planning based on the scroll time domain quantum bee colony optimization algorithm in uncertain environment. The simulation experiment were performed on a laptop computer which has a dual core 3.2 GHz CPU and 8 GB RAM using MATLAB. The MATLAB parallel computing toolbox is used to execute the algorithm for all clusters in parallel. The conditions of the simulation experiment are as follows:

A set of thirty task items to be assigned is selected. The thirty tasks can be divided into three categories: T_1 , T_2 , and T_3 . AUV_1 , AUV_2 , and AUV_3 are involved in the bidding of the AUVs and all tasks of the bid. The bid value of the completed task, trust and initial ability are shown in Table 1. The influence factors of the AUV load, ability and trust degree are 0.5, 0.3 and 0.2, respectively, in the bidding strategies of the contract net task allocation based on the scroll time domain quantum bee colony algorithm.

The simulation experiment has 2 objectives. When the bidding and tendering stage are identical, the first objective is to test and compare the contract net model based on the scroll time domain quantum bee colony algorithm and the traditional contract net model. The second objective is to compare the performance in four aspects: efficiency of task allocation, average AUV load, number of bid AUV allocated tasks, and proportion relation of the corresponding type of task ability.

Table 1. Initial value of the completed task, trust and ability.

	T_1			T_2			T_3		
	Bidding value	Trust degree	Ability	Bidding value	Trust degree	Ability	Bidding value	Trust degree	Ability
AUV_1	3	0.7	0.8	2	0.9	0.9	2	0.8	0.7
AUV_2	4	0.8	0.6	3	0.8	0.8	3	0.7	0.9
AUV_3	5	0.6	0.7	3	0.7	0.7	4	0.6	0.8

<https://doi.org/10.1371/journal.pone.0188291.t001>

Experimental verification

After the experiment, the simulation results are as follows. Fig 1 shows the average load of AUV_1 , AUV_2 and AUV_3 in the traditional contract net model AUV_1 AUV_2 AUV_3 . Fig 2 shows the reduced proportion (%) when AUV_1 , AUV_2 and AUV_3 AUV_2 AUV_3 execute tasks in the traditional contract net model.

Fig 3 shows the average load of the AUVs in the contract net model based on the differential evolution quantum bee colony algorithm. Fig 4 shows the reduced proportion (%) of execution time in the contract net model with the introduced balance coefficient based on the differential evolution quantum bee colony algorithm.

Comparing the front and back images, we observe that the traditional contract net does not consider the load balance of the bidding AUV, which causes a large load difference for the bidding AUV. The improved contract net model satisfies the requirement of load balance because the proportions of load and task execution time of three bidding AUVs are basically equivalent.

Fig 5 shows that the comparison of the executive entirety effectiveness of multiple AUVs in the distributed task planning experiment in the traditional contract net model and the contract net model with the introduced scroll time domain balance coefficient based on the quantum bee colony algorithm.

Fig 6 shows the comparison of the convergence performance of the ABC, QABC, and STDQABC algorithms in the process of multi-AUV distributed task planning.

Figs 7 and 8 show the comparison of the number of iterations and running time when the ABC, QABC, and STDQABC algorithms are used to solve 10 task allocation cases to obtain the optimal solution.

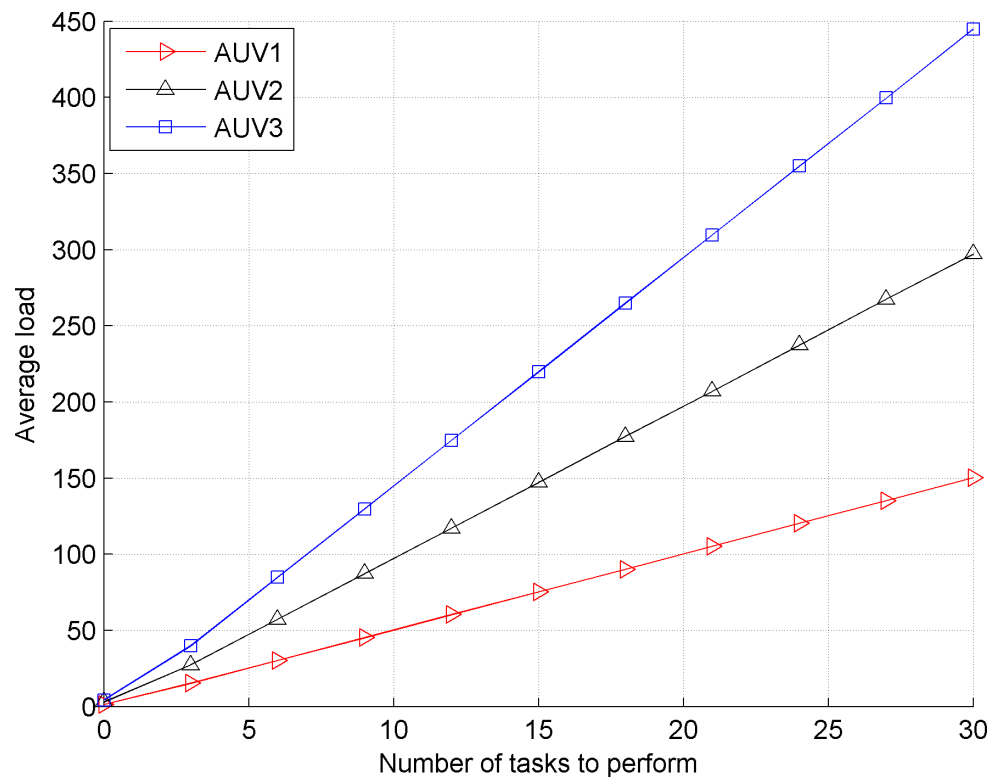


Fig 1. Average load of AUV_1 , AUV_2 and AUV_3 in the traditional contract net model AUV_2 AUV_3 .

<https://doi.org/10.1371/journal.pone.0188291.g001>

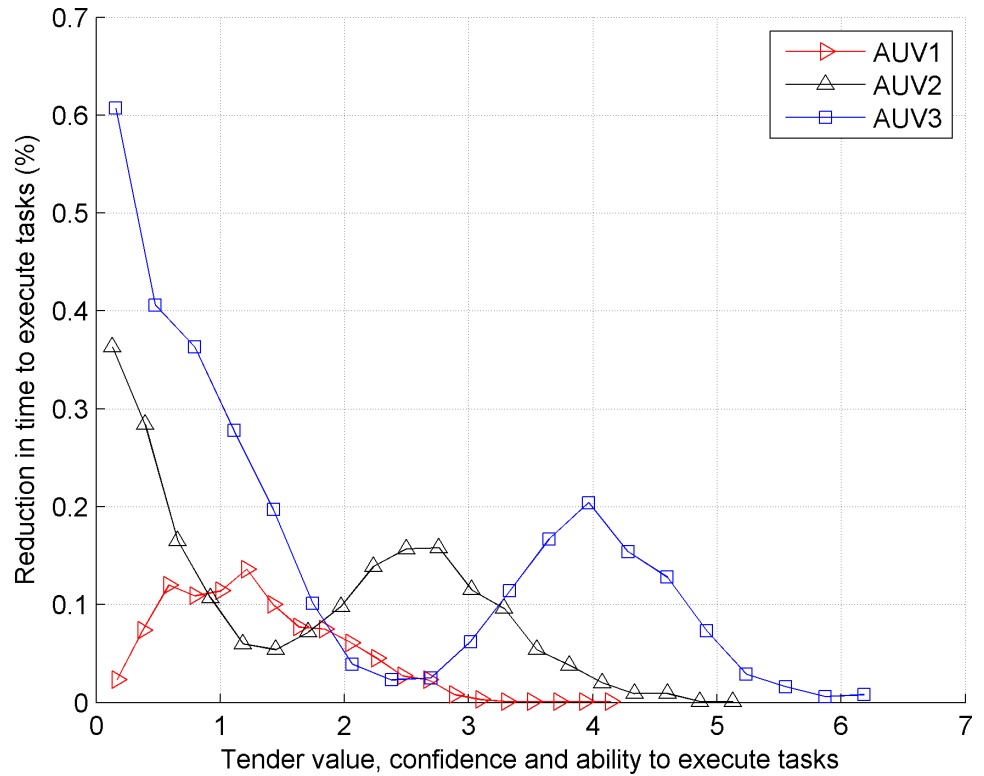


Fig 2. Reduced proportion (%) of AUV₁, AUV₂ and AUV₃ AUV₂ AUV₃ when they executed tasks in the traditional contract net model.

<https://doi.org/10.1371/journal.pone.0188291.g002>

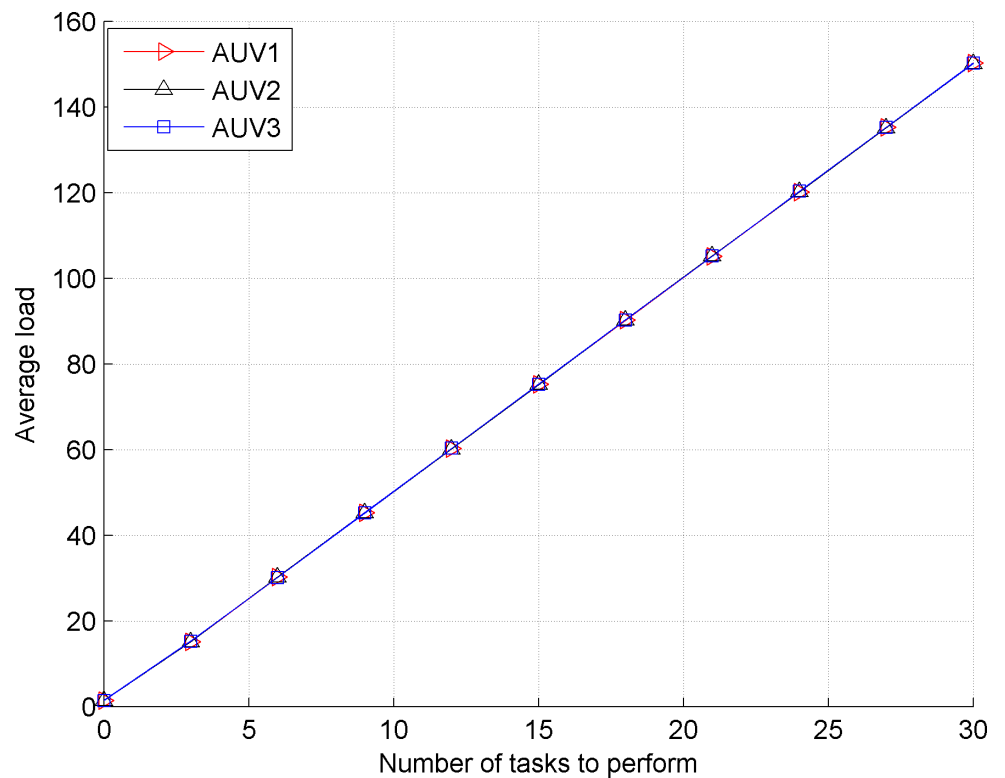


Fig 3. Average load of AUV₁, AUV₂ and AUV₃ in the improved contract net model AUV₃.

<https://doi.org/10.1371/journal.pone.0188291.g003>

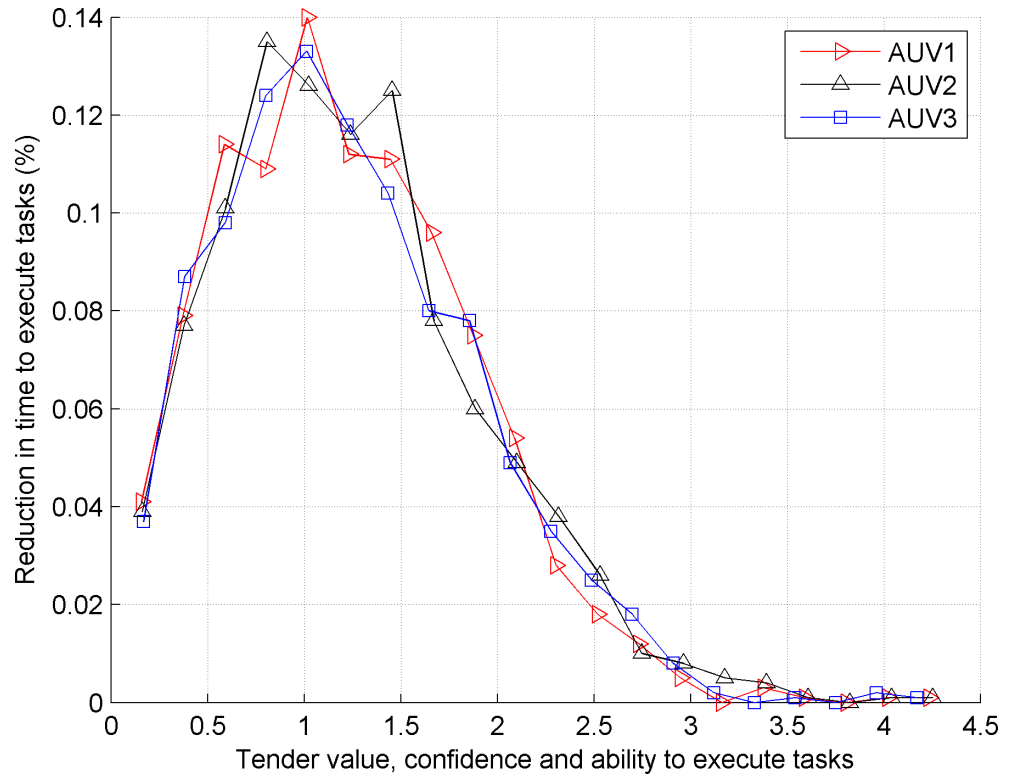


Fig 4. Reduced proportion (%) of AUV₁, AUV₂, and AUV₃ tasks in the improved contract net model.

<https://doi.org/10.1371/journal.pone.0188291.g004>

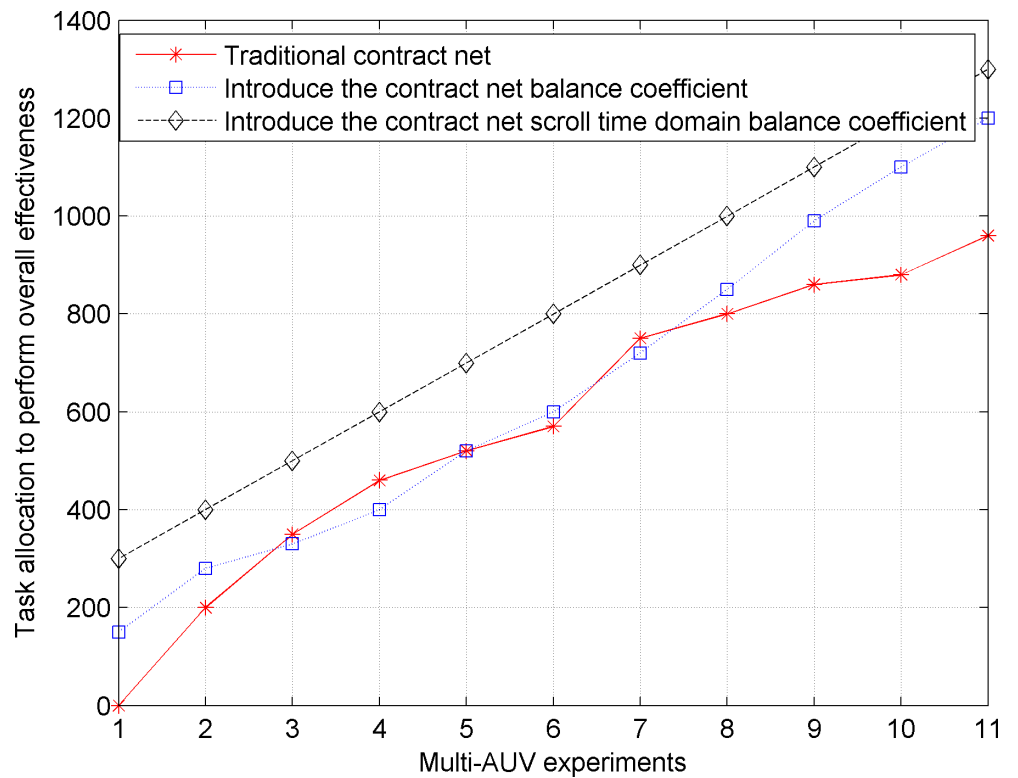


Fig 5. Comparison of the executive entirety effectiveness of the multi-AUV distributed task planning.

<https://doi.org/10.1371/journal.pone.0188291.g005>

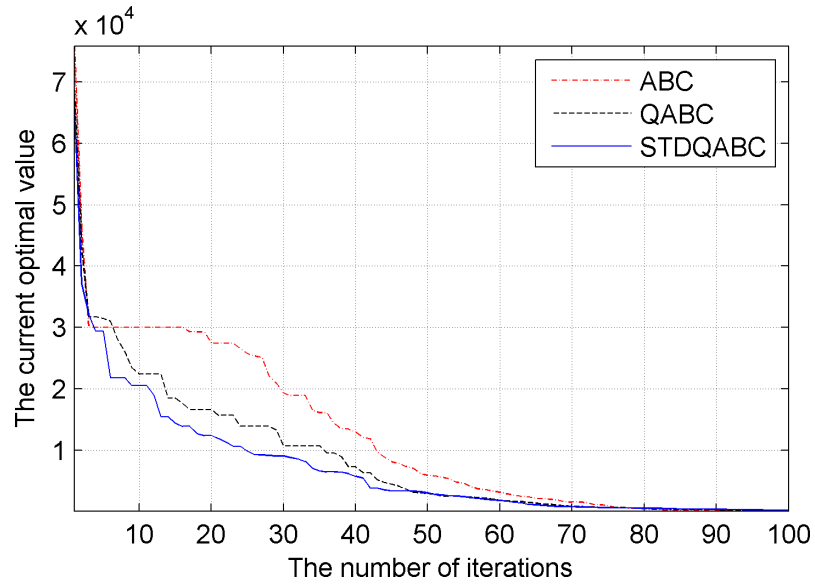


Fig 6. Comparison of the convergence performance of the ABC, QABC, and STDQABC algorithms.

<https://doi.org/10.1371/journal.pone.0188291.g006>

For the test functions with both unimodal and multi modal regions $F_1(x) = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$, DIM = 30, Range = [-100,100], fmin = 0 and $F_2(x) =$

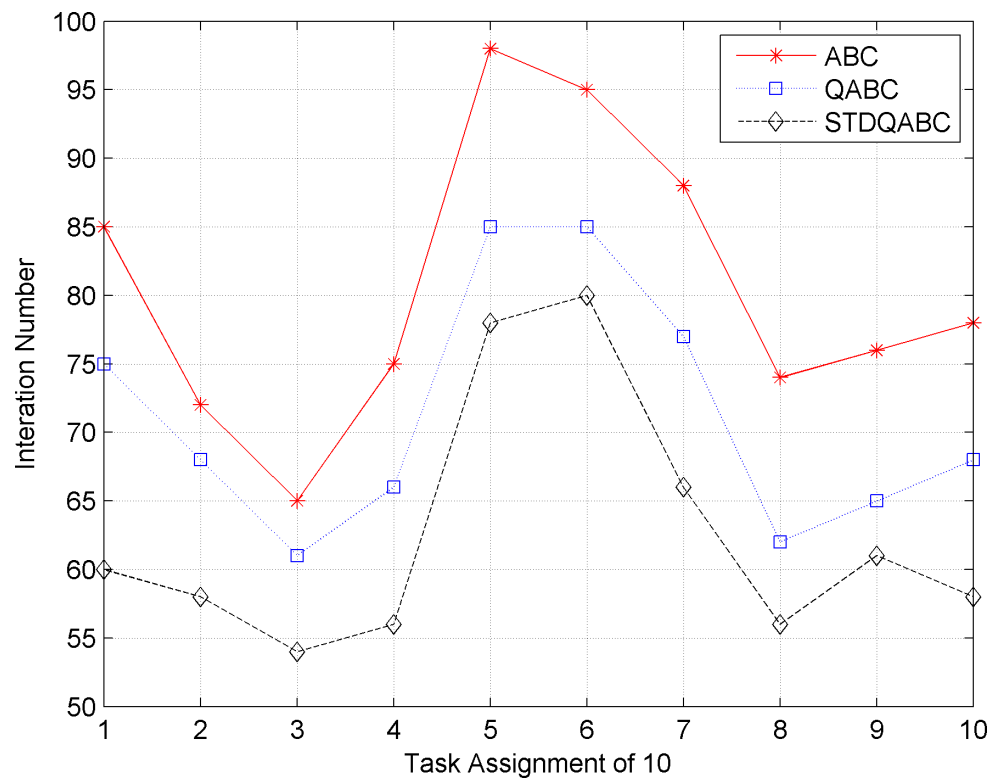


Fig 7. Comparison of the number of iterations for the ABC, QABC, and STDQABC algorithms.

<https://doi.org/10.1371/journal.pone.0188291.g007>

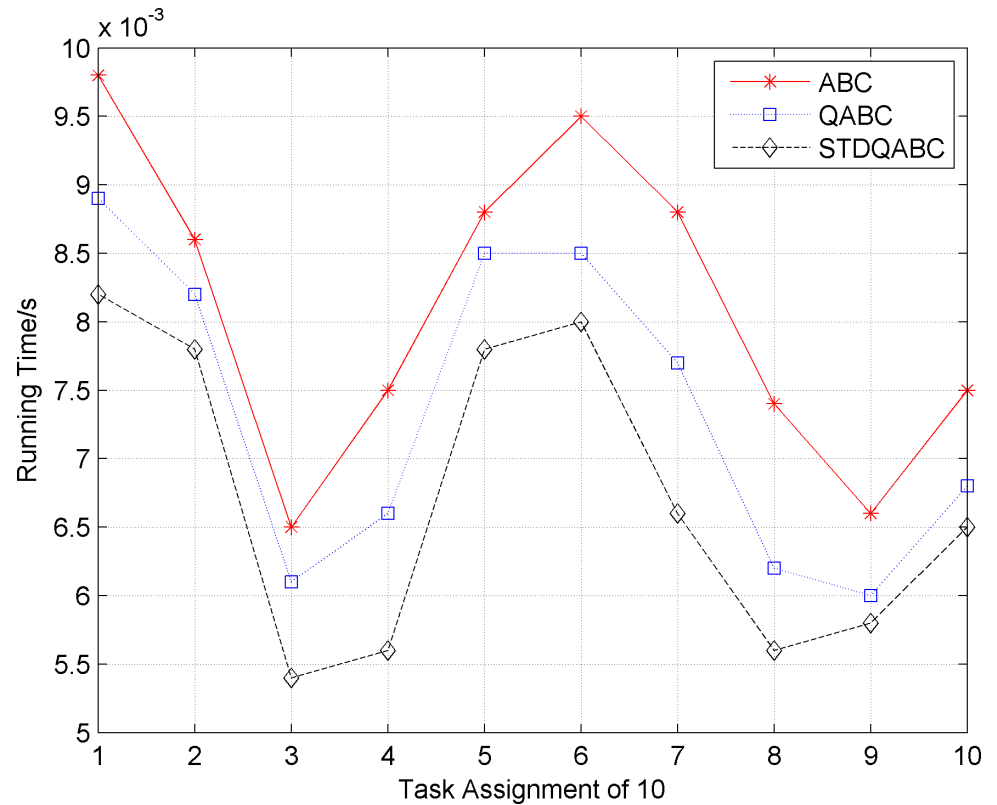


Fig 8. Comparison of the running time for the ABC, QABC, and STDQABC algorithms.

<https://doi.org/10.1371/journal.pone.0188291.g008>

$-20\exp(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^n x_i^2} - \exp(\frac{1}{n}\sum_{i=1}^n \cos(2\pi x_i))) + 20 + e$, DIM = 30, Range = [-30,30], fmin = 0, Fig 9 and Fig 10 These results again demonstrate that DEQABC efficiently balances exploration and exploitation to approximate the global optimum.

The experimental results demonstrate that the STDQABC algorithm is faster than the QABC algorithm in terms of both number of iterations and running time and Get global optimal. Thus, the STDQABC algorithm can effectively improve the performance of the multi-AUV task planning. The STDQABC algorithm has faster and better stability than the ABC algorithm and STDQABC algorithm in solving the task planning problem for multiple AUVs and improving the system performance.

Conclusion

In this paper, we propose a distributed task planning model based on the scroll time domain quantum bee colony algorithm to allow more rapid task planning for a greater number of AUVs and achieve global optimization in the multi-AUV distributed task planning. The balance coefficient is introduced to distribute the AUV task planning of the traditional contract net. The unbalanced load and other defects are improved in the multi-AUV distributed task planning of the traditional contract net. The scroll time domain quantum bee colony algorithm is applied to the process of multi-AUV dynamic distributed task planning in Uncertain Environment. The simulation experiment verifies that the quantum bee colony based on scroll time domain can avoid falling into local optima; shorten the convergence time; reduce the number of iterations; enhance the global, dynamic and adaptive capability of the bee colony

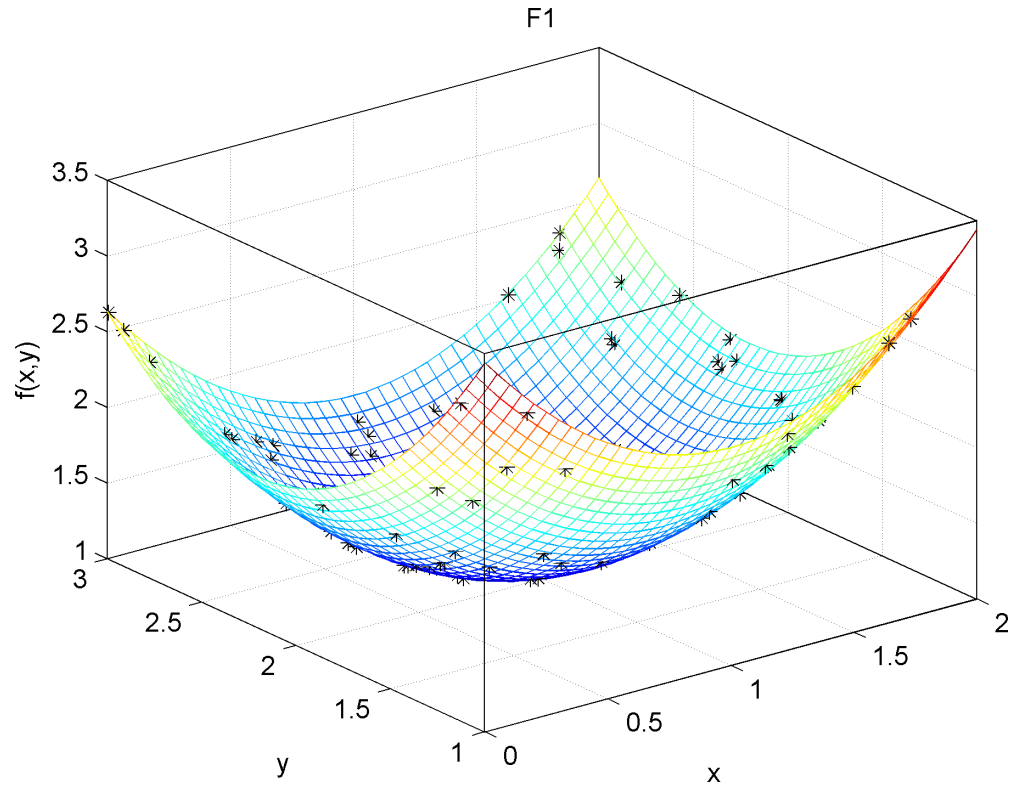


Fig 9. STDQABC algorithm single peak area test function F1 effect.

<https://doi.org/10.1371/journal.pone.0188291.g009>

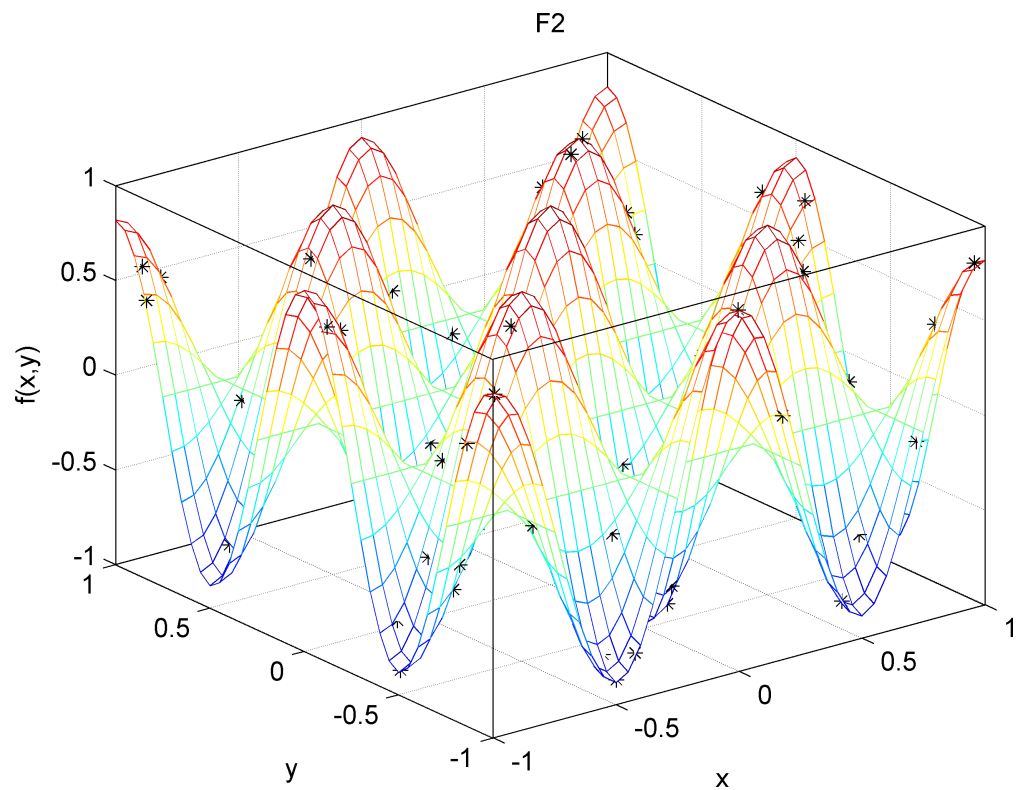


Fig 10. STDQABC algorithm multi-peak area test function F2 effect.

<https://doi.org/10.1371/journal.pone.0188291.g010>

algorithm; and effectively improve the overall performance of distributed task planning for multiple AUVs.

Supporting information

S1 Appendix. An overview of the process of the combinatorial model in mission scenario. (CSV)

S2 Appendix. Productivity of the model in 10 mission scenarios. (CSV)

Author Contributions

Conceptualization: Jianjun Li, Rubo Zhang, Yu Yang.

Data curation: Jianjun Li, Rubo Zhang, Yu Yang.

Formal analysis: Jianjun Li, Rubo Zhang, Yu Yang.

Software: Yu Yang.

Supervision: Rubo Zhang.

Validation: Rubo Zhang, Yu Yang.

Writing – original draft: Jianjun Li.

Writing – review & editing: Jianjun Li.

References

1. SE H, SM R. Trajectory planning for motion-constrained AUVs in uncertain environments. *Oceans*. 2014; 1: 1–5.
2. Liam P, Mae S, Howard L. Area coverage planning that accounts for pose uncertainty with an AUV seabed surveying application. *IEEE International Conference on Robotics & Automation*. 2014:6592–6599.
3. Yan Z, Wu D, Zhou J, Hao L. Recursive subspace identification of AUV dynamic model under general noise assumption. *Mathematical Problems in Engineering*. 2014; 178:368–381.
4. Zhang SJ, He B, Ying LL, Li MH, Yuan G. Autonomous navigation with constrained consistency for c-ranger. *International Journal of Advanced Robotic Systems*. 2014; 11:1–13
5. He B, Ying L, Zhang S, Feng X, Yan T, Nian R, et al. Autonomous navigation based on unscented-FastSLAM using particle swarm optimization for autonomous underwater vehicles. *Measurement*. 2015; 71: 89–101.
6. Bibuli M, Pascoal A, Ridao P, Zereik E. Introduction to the special section on navigation, control, and sensing in the marine environment. *Annual Reviews in Control*. 2015; 40: 127–128.
7. Malisoff M, Zhang F. Robustness of adaptive control under time delays for three-dimensional curve tracking. *Siam Journal on Control & Optimization*. 2015; 53:2203–2236.
8. Peng Z, Wang D, Shi Y, Wang H, Wang W. Containment control of networked autonomous underwater vehicles with model uncertainty and ocean disturbances guided by multiple leaders. *Information Sciences An International Journal*. 2015; 316: 163–179.
9. Özkale C, Fiğlalı A. Evaluation of the multiobjective ant colony algorithm performances on biobjective quadratic assignment problems. *Appl Math Model*. 2013; 37: 7822–7838.
10. Pendharkar PC. An ant colony optimization heuristic for constrained task allocation problem. *J Comput Sci*. 2015; 7: 37–47.
11. Yuan DF, Cong-Ying L. Application of improved ant colony algorithm for quadratic assignment problems. *Comput Mod*. 2013; 3: 9–11.
12. Awad AI, El-Hefnawy NA, Abdel_Kader HM. Enhanced particle swarm optimization for task scheduling in cloud computing environments. *Proced Comput Sci*. 2015; 65: 920–929.
13. Beheshti Z, Shamsuddin SM. Non-parametric particle swarm optimization for global optimization. *Appl Soft Comput*. 2015; 28: 345–359.

14. Ferrante E, Turgut AE, Duéñez-Guzmán E, Dorigo M, Wenseleers T. Evolution of self-organized task specialization in robot swarms. *PLoS Comput Biol*. 2015; 11:e1004273. <https://doi.org/10.1371/journal.pcbi.1004273> PMID: 26247819
15. Grozinger CM, Richards J, Mattila HR. From molecules to societies: mechanisms regulating swarming behavior in honey bees (*Apis* spp.). *Apidologie*. 2014; 45: 327–346.
16. Karaboga D, Gorkemli B, Ozturk C, Karaboga N. A comprehensive survey: artificial bee colony (ABC) algorithm and applications. *Artif Intell Rev*. 2014; 42: 21–57.
17. Civicioglu P, Besdok E. A conceptual comparison of the Cuckoo-search, particle swarm optimization, differential evolution and artificial bee colony algorithms. *Artif Intell Rev*. 2013; 39: 315–346.
18. Loubière P, Jourdan A, Siarry P, Chelouah R. A sensitivity analysis method for driving the artificial bee colony algorithm's search process. *Appl Soft Comput*. 2016; 41: 515–531.
19. Sun J, Fang W, Wu X, Palade V, Xu W. Quantum-behaved particle swarm optimization: analysis of individual particle behavior and parameter selection. *Evol Comput*. 2012; 20: 349–393. https://doi.org/10.1162/EVCO_a_00049 PMID: 21905841
20. Kang Q, He H. A novel discrete particle swarm optimization algorithm for meta-task assignment in heterogeneous computing systems. *Microprocessors & Microsystems*. 2011; 35:10–17.
21. Pamosoaji AK, Hong GY, Widyotriatmo A, Hong KS. Minimum-Time Task Assignment Algorithm for Multiple-Vehicle Systems. *Asian Control Conference*. 2011; 1:85–89.