A Quantum-Inspired Ant Colony Optimization for Robot Coalition Formation

Zhang Yu, Liu Shuhua, Fu Shuai，Wu Di

School of Computer Science, Northeast Normal University, Changchun 130117
Email: zhangy234@nenu.edu.cn,liush129@nenu.edu.cn

Abstract: A Quantum-Inspired Ant Colony Optimization (QACO), based on the concept and principles of quantum computing is proposed in this paper to improve the ability to search and optimization of Ant Colony Optimization (ACO). Each ant is a quantum individual and instead of Q-bit code, we use the probability of choosing robots, and QACO is successfully applied to solve robot coalition formation. The simulated results show that QACO has the better diversity of population and ability to search and optimization, and performs well, even with a small population, without premature convergence as compared to ACO.

Key Words: QACO, task allocation, Large-scale robots, Task Allocation, Robot Coalition Formation

1 Introduction

Similarly to agent coalition formation problem, robot coalition formation problem finds the robot coalition with the greatest value that can complete the task t. A coalition may be formed by several arbitrary robots in the system. In order to have a satisfactory result, you must consider all or most of the combinations. Therefore it's a complex combinatorial optimization problem. In MAS, many achievements in the theoretical study of agent coalition formation have been achieved [1,2].But because of the difference between MAS and MRS, the existing agent coalition formation algorithm can't directly used in MRS. So the research of robot coalition formation has just started[3,4].

Ant Colony Optimization (ACO) is a new simulation evolvement algorithm which is proposed by Italian scholars Dorigo M. It solves some difficult problems in the optimization of discrete system using the ability of optimization in the process of ant colony searching food. TSP[5], assignment problem[6],agent coalition formation problem[7] and so on has successfully solved by it , and have achieved better effect. However, in practical application, as the other evolvement algorithms, ACO also has some disadvantages which are mainly premature convergence, low searching ability and slow convergence speed.

Quantum computing is a new type of computation theory, which is the combination of quantum physics and computer science. It use of quantum superposition, entanglement and interference in quantum theory and is possible to solve many puzzles in the classical calculating. It becomes research hotspot with its unique computing performance[8,9]. Several researchers have introduced quantum computing theory into traditional algorithms and proposed quantum genetic algorithm and quantum-behaved particle swarm optimization [11]. Their searching ability and efficiency are superior to the traditional genetic algorithm and particle swarm optimization.

We propose a novel Quantum-Inspired Ant Colony Optimization (QACO) based on the concept and principles of quantum computing and it is successfully applied in solving robot coalition formation. The simulated results show that QACO has the better diversity of population and ability to search and optimization, and performs well, even with a small population, without premature convergence as compared to ACO.

2 KEY ISSUES OF ROBOT COALITION FORMATION

2.1 Validity of Robot Coalition

Agents are allowed to exchange resources, so the formed coalition freely redistributes resources amongst the members. However, this is not possible in a multiple-robot domain. Robot capabilities correspond to sensors (camera, laser, sonar, or bumper) and actuators (wheels or gripper) which cannot be autonomously exchanged. This implies that a robot coalition that simply possesses the adequate resources is not necessarily performing a task, and other location constraints have to be represented and met.

Correct resource distribution is an important issue in robot coalition formation. The box-pushing task[12]is used to illustrate this point. Three robots, two pushers (with one bumper and one camera) and one watcher (with one laser range finder and one camera) cooperate to complete the task. The total resource requirements are: two bumpers, three cameras and one laser range finder. However, this information is incomplete, as it does not represent the constraints related to sensor locations. Correct task execution requires that the laser range finder and camera reside on a single robot while the bumper and laser range finder reside on different

978-1-4244-2723-9/09/$25.00 © 2009 IEEE

robots. Therefore each candidate coalition must be verified feasible.

Checking the feasibility of robot coalition is a Constraint Satisfaction Problem (CSP).It is defined by a set of variables, a set of the domain values for each variable and a set of constraint relationships between variables, which is denoted as (V,D,C). Where V is the set of variables $\{v_1,\ldots,v_n\}$ which are resources and capabilities requirements, in box-pushing task, $v_1,\ldots,v_n$ are the bumper, camera and laser range finder. D is the set of the domain values which is the sum of the available robots possessing the required resources and capabilities, $D=\{D_1,\ldots,D_n\}$, where D_i is the limited domain of v_i 's all possible values. C is the set of constraint relationships between variables, $C=\{C_1,\ldots,C_m\}$, each constraint includes a subset of V, that is $\{v_i,\ldots,v_j\}$ and a constraint relationship $R\subseteq D_i\times\ldots\times D_j$. For box-pushing task ,two types of constraints exist, the sensors and actuators must reside on the same robot or on different robots. As shown in Fig.1, locational constraints represented as arcs labeled *s*(same robot) or *d*(different robot).

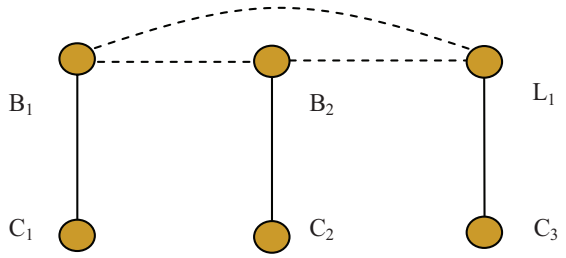

Figure 1. Box-pushing task constraint graph

2.2 The Evaluation Criteria of Robot Coalition Formation

A coalition where one or more agents have a predominant share of the capabilities may have the same utility as a coalition with evenly distributed capabilities. However robots are typically unable to redistribute their resources. Therefore coalitions with one or more dominating members tend to be heavily dependent on those members for task execution. These dominating members then become indispensable. Such coalitions should be avoided in order to improve fault tolerance as over reliance on dominating members can cause task execution to fail or considerably degrade, Coalition imbalance is defined as the degree of unevenness of resource contributions made by individual members to the coalition. The perfectly balanced coalition is where each member contributes equally (*taskvalue/n*) to the task. The Balance Coefficient (*BC*) quantifies the coalition imbalance level. *BC* can be calculated as follows:

$$\mathrm{BC}=\frac{\gamma_1\times\gamma_2\times\cdots\times\gamma_n}{\left[\frac{\mathrm{taskvalue}}{n}\right]^n} \tag{1}$$

Where $(\gamma_1,\gamma_2,\cdots,\gamma_n)$ is a resource distribution with a coalition C. For the coalitions of the same size, the higher *BC*, the more balanced the coalition.

Although more balanced coalitions are more desirable, generally, larger coalitions imply that the average individual contribution and the capability requirements from each member is lower, larger coalitions have much more costs. Thus the comparison across coalitions of different sizes requires that metric subsume elements of both balance and size. The Fault Tolerance Coefficient (*FTC*) is such a metric and has the following from:

$$\mathrm{FTC}=\delta\mathrm{BC}+\mu\mathrm{f}(n) \tag{2}$$

Where $\delta+\mu=1$, $\mathrm{f}(n)=1-e^{-\lambda n}$ is the function of coalition size. After a particular point, increasing n does not result in a significant increase to the function value. This is desirable from a coalition formation perspective since after a certain point, increasing coalition size does not yield improved performance.

2.3 The Description of Robot Coalition Formation

(1) The Ability Description of Robots

All robots in the system form a robot set $R=\{R_1,R_2,\cdots,R_n\}$.The ability vector of R_i is $B_{R_i}=(b_{i1},b_{i2},\cdots,b_{im})^T$, and the ability cost vector is $cost_{R_i}=(cost_{i1},cost_{i2},\cdots,cost_{im})^T$ where $cost_{ij}$ is the cost of the ability b_{ij}.When $b_{ij}=0$,it denotes R_i without the ability b_{ij}.The cost of R_i is $\sum_{j=1}^{m}cost_{ij}b_{ij}$, who has m kinds of abilities.

(2) The Ability Description of Robot Coalition

Robot coalition is a set of robots in which robots can cooperate to complete a task. A coalition C is the nonempty subset of R . Based on the different ability attributes of the robots, there are different ability vectors of the coalition. For the additive capacity (such

as handling, etc.), the ability of the coalition C is $B_C = \sum_{R_i \in C} B_{R_i}$ .For the merger capacity (such as video distance, etc.),the ability of the coalition C is $B_C = \bigcup_{R_i \in C} B_{R_i}$.

The cost of the coalition ability is defined as follows:

The additive capacity: $D(C) = \sum_{R_i \in C} \sum_{j=1}^{m} cost_{ij} b_{ij}$

The merger capacity: $D(C) = \bigcup_{R_i \in C} \sum_{j=1}^{m} cost_{ij} b_{ij}$

(3) The Requirement Description of Task Capacity

There are k tasks, denoted $T = \{t_1, t_2, \cdots, t_k\}$ .The task t has the ability demanded vector: $B_t = (b_1, b_2, \cdots, b_m)^T$.

The essential condition for the coalition C to finish the task t is as follows： $B_C \geq B_t$.

(4) The Definition of Coalition 's Income

We define a *reward* function which is a mapping from the set of tasks to the set of real numbers, denoted $reward : T \to R^+$ .A *cost* function is defined as $cost : C \to R^+$, which is a mapping from the set of coalitions to the set of real numbers. We consider two types of cost:

- A *coalition-inherent* cost measures the inherent cost(e.g., in terms of energy consumption or computational requirements)of using particular capabilities of the coalition. Here the main consideration is the consumption of the robot's ability to accomplish the tasks, including the communication between the robots in the coalition and the cost of the coalition ability. We denote it by C_cost .
- A *task-specific* cost measures cost according to task-related metrics, such as time, distance, etc. Here we consider the distance mainly. We denote the cost of the coalition performing the task by T_cost .

Thereby, the *cost* function of the coalition *C* performing task *t* is denoted as: $Cost(C,t) = \omega_1 C\_cost + \varpi_2 T\_cost$, where ϖ_1 and ϖ_2 are weighted parameters of both the *coalition-inherent* cost and *task-specific* cost, $\varpi_1 > 0$, $\varpi_2 > 0$. According to the differences between agent coalitions and robot coalitions, the income of the robot coalition should be defined as:

$$Inc(C) = FTC \times [rew(t) - Cost(C,t)] \quad (3)$$

3 Solving Robot Coalition using Ant Colony Algorithm

Put m ants on n robots at random, the probability that ant k chooses Robot j is as follows:

$$p_{ij}^{\ k} = \frac{[\tau_{ij}(t)]^{\alpha} [1/d_{ij}]^{\beta}}{\sum_{u \in J_k} [\tau_{ij}(t)]^{\alpha} [1/d_{iu}]^{\beta}} \quad (4)$$

J_k ,the robot set that ant k hasn't chosen; $\tau_{ij}(t)$, the quantity of pheromone remained on the line between robot *i* and robot *j*; $d_{ij}(i, j = 1,2,\cdots,n)$,the distance between robot *i* and robot *j*, called communication spending; α and β control the relative importance of pheromone and communication spending. The ant will stop seeking route when it arrives at a certain robot and finds that the current robot coalition can accomplish the task. When all ants have formed their task-oriented coalitions, one circle finishes. Then each candidate coalition is checked to verify if it is feasible. Update the maximal income, and according to equation (5) updates the intensity of pheromone.

$$\tau_{ij}(t+1) \leftarrow \rho \tau_{ij}(t) + \sum_{k=1}^{m} \Delta \tau_{ij}^{\ k} \quad (5)$$

Here $\Delta \tau_{ij}^{\ k}$,the increment of the familiar degree between robot *i* and robot *j* given by ant k in this circle, is defined as :

$$\Delta \tau_{ij}^{\ k} = \begin{cases} \dfrac{Inc(C_k)}{\sum_{k=1}^{m} C_k}, & \text{If the coalition formed by ant k including robot ;} \\ 0, others \end{cases} \quad (6)$$

$Inc(C_k)$ is the income given to the coalition formed by ant k. The optimal combination of parameter α and β in this algorithm can be confirmed by the experimental method. The stop of the program may be controlled by a fixed evolving generation, or when the evolving trend is inconspicuous, the calculating process will end. The algorithm's complexity degree is $O(NC \cdot m \cdot n^2)$, NC is the number of circle.

4. Solving Robot Coalition using Quantum-Inspired Ant Colony Optimization

4.1 Coding with Q-bits

In binary quantum computer, information unit is called Q-bit. A state of a Q-bit can be defined as: $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ or $\begin{bmatrix}\alpha\\ \beta\end{bmatrix}$, α, β is the probability amplitude. $|\alpha|^2$ is the probability of $|0\rangle$, $|\beta|^2$ is the probability of $|1\rangle$, and $|\alpha|^2 + |\beta|^2 = 1$. Besides the state of 0 or 1, it also can be a linear superposition state of both. 0 and 1 exists in the probability, through measure and observation collapsing to 0 or 1[13]. Obviously, now widely used in the classical information (0 or 1 as the information unit) is an exception of quantum information, namely α and β are zero.

$$\begin{bmatrix}\alpha_1 & \alpha_2 & \cdots & \alpha_m\\ \beta_1 & \beta_2 & \cdots & \beta_m\end{bmatrix} \tag{7}$$

Where $|\alpha_i|^2 + |\beta_i|^2 = 1 \quad i = 1,2,\cdots,m$ .It can denote 2^m states, a binary string of m bits can only denote one state. Therefore, Q-bit coding has better colony diversity.

4.2 Quantum-Inspired Ant Colony Optimization

The basic idea of quantum-inspired ant colony optimization is to make ants have quantum characteristics, every ant is a quantum individual and using the probability of choosing robots instead of Q-bit coding. Due to introducing quantum computing, compared with ACO algorithm, QACO are added to the corresponding observation process and repairing process[14].

The probability coding is defined as: $\begin{bmatrix}P_0\\ P_1\end{bmatrix}$, where $P_0 + P_1 = 1$. The individual is denoted as:

$$q_k = \begin{bmatrix}P_{k_{10}} & P_{k_{20}} & P_{k_{j0}} & P_{k_{m0}}\\ P_{k_{11}} & P_{k_{21}} & P_{k_{j1}} & P_{k_{m1}}\end{bmatrix} \tag{8}$$

Where $k = 1,2,\cdots,n$, $j = 1,2,\cdots,m$, $P_{k_{j1}} = P_{ij}^{\ k}$, $P_{k_{j0}} = 1 - P_{k_{j1}}$. The t -th generation population of QACO is denoted as: $Q(t) = \{q_1^{\ t}, q_2^{\ t}, \cdots, q_n^{\ t}\}$. $P(t) = \{X_1^{\ t}, X_2^{\ t}, \cdots, X_n^{\ t}\}$, where $X_k^{\ t}$ is the state of observing k -th individual, $X_k^{\ t} = \{x_{k1}^{\ t} x_{k2}^{\ t} \cdots x_{km}^{\ t}\}$, $x_{kj}^{\ t}$ is 0 or 1, when it is 0, denoting robot j not chosen. And 1 denoting robot j is chosen.

The pseudocode of QACO is as follows:

Step1: initializing $t = 0 \quad NC = 0 \quad NC_{\max} = N$
$numAnt = n \qquad numRobot = m$
$\Delta\tau_{ij} = 0 \quad \tau_{ij}(0) = \tau_0$

Step2: Put n ants on m robots randomly
for $k = 1$ to n
for $j = 1$ to m
{ if ant k starts from robot j , then $P_{k_{j1}} = 1$. According to formulation (4), calculate the probability of choosing robots, $P_{k_{j1}} = P_{ij}^{\ k}$, $P_{k_{j0}} = 1 - P_{ij}^{\ k}$ }

Step3: observing the individuals of $Q(t)$, and getting the states $P(t)$.

Step4: checking whether every state in $P(t)$ is solution, if not then repair it, and go to Step9.

Step5: According to the formulation (3), calculate the income $Inc(X_j^{\ t})$ of $X_j^{\ t}$.

Step6: Saving the optimization coalition b and it's income $Inc(b)$.

Step7: Updating the pheromone by the formulation (5) and (6).

Step8: Set $t = t + 1 \quad NC = NC + 1 \quad \Delta\tau_{ij} = 0$

Step9: if $(\mathrm{NC} < \mathrm{NC}_{\max})$ and not no evolving for a long time then go to Step2 else output the optimization coalition and it's income.

Step10: Repairing the state which is not solution through repairing process. If states in $P(t)$ are all solutions then go to Step4.

5 SIMULATION

In order to verify the effectiveness of the algorithm, we implemented QACO algorithm at TeamBots simulation platform. And to check the performance of our algorithm, we made a contrastive experiment with basic ACO algorithm. The parameters are as follows: the number of robot $m = 20$, The ability and cost vector of robots is shown in Table1, the ability of task is (3,2,3), $\alpha = 1.5$, $\beta = 2$, $\rho = 0.7$, $\tau_0 = 50$,

the maximum iteration $NC_{max} = 1000$, $rew(t) = 10000$, and we made 20 experiments for different population sizes.

Table 1 The ability and cost vector of robots

	Ability	Cost		Ability	Cost
R_0	1, 0, 1	1, 2, 1	R_{10}	2, 0, 1	1, 4, 3
R_1	1, 1, 1	1, 1, 2	R_{11}	1, 3, 3	2, 2, 1
R_2	2, 1, 2	2, 3, 1	R_{12}	2, 1, 3	1, 3, 1
R_3	1, 2, 1	3, 2, 1	R_{13}	0, 2, 1	3, 1, 2
R_4	0, 1, 1	1, 1, 1	R_{14}	1, 2, 3	3, 1, 2
R_5	1, 1, 2	2, 1, 1	R_{15}	2, 1 , 1	4, 2, 1
R_6	0, 0, 1	3, 2, 3	R_{16}	3, 2, 2	2, 4, 1
R_7	2, 2, 1	3, 2, 1	R_{17}	3, 1, 2	1, 3, 3
R_8	3, 2, 1	2, 1, 3	R_{18}	1, 3, 1	2, 1, 2
R_9	3, 1, 1	2, 3, 2	R_{19}	0, 1, 2	1, 2, 1

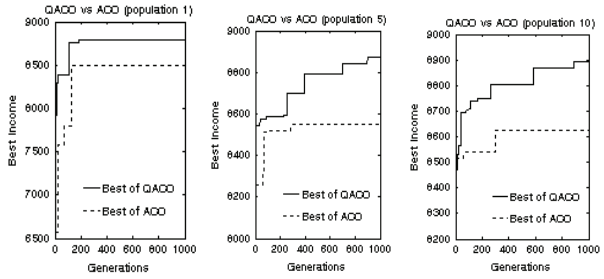

Figure 2. The comparison of single best test result between QACO and ACO

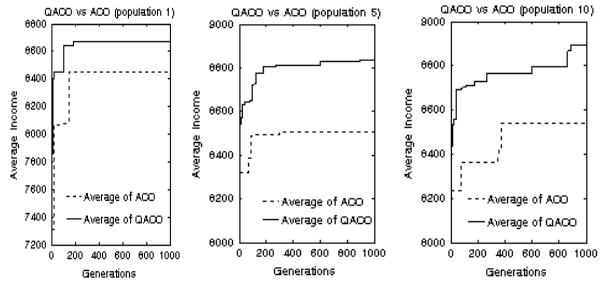

Figure 3. The comparison of the mean in 20 test between QACO and ACO

Figure 2 are the population size of 1, 5, 10 for the best evolutionary curve, Figure 3 is an average of evolutionary curve. According to Figures 2 and 3,we can find that the convergence rate of basic ACO algorithm is slightly better than QACO algorithm, but it's optimization weaker and it has premature convergence. When the population size is 1, basic ACO and QACO both appear premature. With the increasing population size, the optimal solution of QACO algorithm is superior to the basic ACO algorithm, and the evolution of QACO is more and more obvious, and the optimal capacity is also more and more stronger, until close to 1000 it approaches to the convergence. However, the evolution of basic ACO has no significant changes, and it is easy to fall into local optimum, and difficult to jump out of local optimal solution.

We can see from Table 2, the coalition with FTC has better fault-tolerance and robustness. If a member of the coalition fails, the coalition can still complete the task. When the size of the population is small, basic ACO algorithm is difficult to get a satisfactory solution. However, when the size of the population is 1,QACO algorithm gets the optimal solution 7 times, and when the size of the population is 5 ,it gets almost all of the optimal solution.

6 CONCULUTION

This paper proposed a Quantum-Inspired Ant Colony Optimization based on the concept and principles of quantum computing. Each ant is a quantum individual and instead of Q-bit code, we use the probability of choosing robots. We also discussed the key issues of robot coalition problem and QACO is successfully applied to solve robot coalition formation. In contrast with basic ACO algorithm, QACO has the better diversity of population and ability to search and optimization, and performs well, even with a small population, without premature convergence as compared to ACO.

REFERENCES

[1] M Klush, A Gerber. Dynamic coalition formation among rational Agents. IEEE Intelligent Systems, 2002,17(3):42−47.

[2] ZHANG Xin-Liang, SHI Chun-Yi. A Dynamic Formation Algorithm of Multi-Agent Coalition Structure. Journal of Software. 2007, 18(3): 574-581.

[3] Lovekesh Vig and Julie A. Adams. Multi-Robot Coalition Formation. IEEE International Conference on Robotics and Automation, Florida, 2006.

[4] Fang Tand and Lynne E.Parker. A Complete Methodology for Generating Multi-Robot Task Solutions using ASyMTRe-D and Market-based Task Alloction . Proc.of IEEE International Conference on Robotics and Automation,Rome,Italy,2007.

[5] WU Bin,SHI Zhong-Zhi. An Ant Colony Algorithm Based Partition Algorithm for TSP. Chinese Journal of Computers.2001, 24(12): 1-5.

[6] LIANG Yao, QIN Zheng, YANG Li-ying, HUANG Ru. Mutated Ant Colony Algorithm for Assignment Problem. Microelectronics & Computer. 2005, 42(5):734-739.

[7] Xia Na , Jiang Jianguo , Wei Xing , Zhang Ling. Searching for Agent Coalition for Single Task Using Improved Ant Colony Algorithm. Journal of Computer Research and Development. 2005, 42(5):734-939.

[8] L K Grover. Algorithms for quantum computation discrete logarithms and factoring .In Proceedings of the 35th Annual Symposium on Foundations of Computer Science,Piscataway,NJ,1994.

[9] P W Shor.Quantum computing. documenta mathematica, extra volume. Proceedings of the International Congress of Mathematiclans, Berlin, Germany 1998.

[10] WANG Yu-Ping , LI Ying-Hua. A Novel Quantum Genetic Algorithm for TSP. Chinese Journal of Computers. 2007, 30(5):748-755.

[11] KANG Yan, SUN Jun , XU Wen-bo. Parameter selection of quantum-behaved particle swarm optimization. Computer Engineering and Applications. 2007, 43(23):40-42.

[12] B.P Gerkey, M.J.Mataric. Pusher-watcher: An approach to fault-tolerant tightly-coupled robot coordination. IEEE International Conference on Robotics and Automation, Washington, 2002.

[13] XIA Pei-Su. Quantum Computing. Journal of Computer Research and Development. 2001, 38(10):1153-1171.

[14] Kuk-Hyun Han. Quantum-Inspired Evolutionary Algorithm for a Class of Combinatorial Optimization. IEEE Transactions on Evolutionary Computation, 2002,6(6):580-593.