Multi-robot Task Allocation Based on Ant Colony Algorithm

Jian-Ping WANG
School of Information Engineer, Henan Institute of Science and Technology, Henan, Xinxiang, 453003, China
E-Mail:xunji2002@163.com

Yuesheng Gu and Xiao-Min LI
School of Information Engineer, Henan Institute of Science and Technology, Henan, Xinxiang, 453003, China and
School of Technique and Electricity, Henan Institute of Science and Technology, Henan, Xinxiang, 453003, China
E-Mail:hz34567@126.com and lxm0707@163.com

***Abstract*—With the development of information technology, the capability and application fields of robots become wider. In order to complete a complex task, the cooperation and coordination of robots are needed to be adopted. As the main problem of the multi-robot systems, multi-robot task allocation (MRTA) reflects the organization form and operation mechanism of the robots system. The cooperation and allocation for large-scale multi-robot system in loosely environment is the hot issue. As a popular bionic intelligence method, ant colony algorithm is powerful for solving MRTA. By analyzing the existing algorithms, this paper proposed a new solution for MRTA based on ant colony algorithm, built up the model of the algorithm and described the robots coalition, high-level task allocation process in details. Finally, we realized the simulation of ant colony algorithm based on MATLAB, and then compared the robustness and the best incomes of the four algorithms. The simulation results show that, ant colony algorithm is a high degree of ability and stability for solving MRTA.**

***Index Terms*— Ant Colony Algorithm, Multi-Robot Task Allocation, Robot Coalition Formation, multi-robots systems, MATLAB**

I. INTRODUCTION

A. Background

Along with the development of robotics, multi-robot coordination has received more attention. Multi-robot systems can provide several advantages over single-robot systems: robustness, flexibility and efficiency among others. To benefit from these potential aspects the robots must cooperate to carry out a common mission.

In particular, multi-robot task allocation (MRTA) has recently risen to prominence and become a key research topic in its own right. The general idea of multi-robot systems is that, teams of robots, deployed to achieve a common project, are not only able to perform tasks that a single robot is unable to, but also can outperform systems of the individual robot, in terms of efficiency and quality. MRTA is the basic problem of the multi-robots systems. It means to optimize the task allocation scheme, in order to improve the operation efficiency of multi-robot systems. As the increasing task and robots, task allocation becomes more difficulty. MRTA is a typical combinatorial optimization problem [1]. The formulation of MRTA with multiple of robots of different types to take up large number tasks consists of several parameters that make it as NP-hard.

At present, some bionic algorithms have been proposed for solving the problems of NP-hard. In this paper, we proposed a new methodology for MRTA based on ant colony algorithm, built up the model of the algorithm and described the robots coalition, high-level task allocation process in details. Finally, we realized the simulation of ant colony algorithm based on MATLAB, and then compared the robustness and the best incomes of the four algorithms. The simulation results show that, ant colony algorithm is a high degree of ability and stability for solving MRTA.

B. Overview of MRTA

The clear definition of MRTA was proposed by Gerkey [2]. In his article, MRTA was defined ad follows:

Given are m robots, each capable of executing one task, and n possibly weighted tasks, each requiring one robot. Furthermore given for each robot is a nonnegative efficiency rating estimating its performance for each task (if a robot is incapable of executing a task, then the robot is assigned a rating of zero for that task). The goal is to assign robots to tasks so as to maximize overall expected performance, taking into account the priorities of the tasks and the efficiency ratings of the robots.

At present, some methods have been proposed for solving MRTA list as follows:

(1) Market-Based Approaches

The methods based on the market mechanism are the most popular way for solving MRTA, like first price auctions, Dynamic role assignment, Trade robots, Murdoch, Demircf [3], M+ and so on.

In these approaches, each distributed agent computes a cost for completing a task, and broadcasts the bid for that task. The auctioneer robot decides the best available bid, and the winning bidder attempts to perform the task won. They effectively meet the practical demands of robot teams, while producing efficient solutions by capturing

doi:10.4304/jcp.7.9.2160-2167

the respective strengths of both distributed and centralized approaches.

First, they can distribute much of the planning and execution over the team and thereby retain the benefits of distributed approaches, including robustness, flexibility, and speed.

They also have elements of centralized systems to produce better solutions: auctions concisely gather information about the team and distribute resources in a team-aware context.

Because of its well extensibility, these methods are particularly suitable for distributed robotics area. Theoretically, they can ensure the optimal task allocation. However, if the communication costs are too high in the task allocation process, once there are failures in robot’s communication, the performance will degrade noticeably [4], so these methods fit for small and medium-scale task allocation.

M+ was proposed by Botelho and Alami in Ref. [5], the algorithm used a task allocation protocol based on the Contract Net protocol with formalized capabilities and task costs. The need to pre-define the capabilities and costs limits the applicability of the M+ algorithm to domains where these are known.

(2) Behaviors-based approaches

In this way, the approaches rely on the behaviors of robots, like ALLCANCE, BLE and ASyMTRe, etc.

ALLIANCE [6] is an architecture that has been proposed for fault tolerant instantaneous allocation, and integrates impatience and acquiescence into each robot. ALLIANCE uses motivational behaviors such as robot impatience and robot acquiescence to perform tasks that can’t be done by other robots, and gives up the tasks they can’t perform efficiently.

BLE [7] means the broadcast of local eligibility technique. It is another behavior based architecture, which uses cross inhibition of behaviors between robots. It is based on a calculated task eligibility measure which robots compute individually and broadcast to the team.

ASyMTRe [8] is a behavior-based architecture. It is based on mapping environmental, perceptual, and motor control schemas to the required flow of information through multi-robot systems, automatically reconfiguring the connections of schemas within and across robots to synthesize valid and efficient multi-robot behaviors for accomplishing the team objectives.

These approaches are stronger in real-time capability, fault-tolerant and robustness, but still only the part optimal for solving MRTA.

(3) Approaches based on linear programming

Gerkey and Mataric regard MRTA as the linear programming problem of 0-1 type [9], in this approach, they find n^2 nonnegative integers, and in order to maximize α_{ij} . The formula is defined as (1):

$$\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_{ij}\mu_{ij}\omega_{j} \tag{1}$$

In (1), the conditions satisfy (2).

$$\begin{aligned} &\sum_{i=1}^{n}\alpha_{ij} = 1, 1 \le j \le n \\ &\sum_{j=1}^{n}\alpha_{ij} = 1, 1 \le i \le n \end{aligned} \tag{2}$$

The methods based on linear programming can handle only MRTA of single-robot tasks and single-task robots, but can’t handle a task for which need multi-robots to cooperate to accomplish. Early, the main methods to solve liner programming were simple type method and the hungry method. These two methods are essentially matrix calculations, as the increasing of tasks and robots. The computational complexity will grow of exponential. Some mixed integer linear programming methods of MRTA can find the optimal solution successfully, but usually need to collect the information of all the tasks and robots. The expansibility and efficiency of these methods are weak.

(4) Approaches based on swarm intelligence

There approaches simulate the behaviors of insects to assign the task of robots, swarm intelligence methods [10] include the threshold value method and ant colony algorithm, are mainly using for robot system in unknown environment. Because the group cooperation among individuals is distributed, a few of the individual's fault can’t affect the entire task to solving, Swarm intelligence methods have high robustness, scalability, are very suitable for distributed multi-robot systems. In this paper, we proposed a solution for MRTA based on ant colony algorithm.

II. Ant Colony Algorithm

In the natural world, ants run randomly around their colony to search for food. Ants deposit a chemical substance called pheromone along the traveled paths. Other ants upon finding these paths would follow the trail. Then, if they discover a food source they will return to the nest and deposit another path along with the previous one, and will be strengthened the preceding path.

Over time, however, the pheromone trail starts to evaporate, thus reducing its attractive strength. The more time it takes for an ant to travel down the path and back again, the more time the pheromones have to evaporate. Pheromone evaporation also has the advantage of avoiding the convergence to a locally optimal solution. If there were no evaporation at all, the paths chosen by the first ants would tend to be excessively attractive to the following ones. In that situation, the exploration of the solution space would be constrained.

Thus, when one ant finds a path from the colony to a food source, other ants is more likely to follow that path, and positive feedback eventually leads all the ants following a single path.

The idea of the ant colony algorithm is to mimic this behavior with “simulated ants” walking around the graph representing the problem to solve. Ant colony algorithm is the result of research on computational intelligence approaches to combinatorial optimization originally conducted by Dr. Marco Dorigo, in collaboration with

Alberto Colorni and Vittorio Maniezzo [11]. The first algorithm was aiming to search for an optimal path in a graph, based on the behavior of ants seeking a path between their colony and a food source.

In operation research, ant colony algorithm is a probabilistic technique for solving computational problems, which is reduced to finding good paths through graphs. This algorithm is a member of ant colony algorithms family, in swarm intelligence methods, and it constitutes some metaheuristic optimizations. It is a Bionic algorithm. It simulates ant's foraging behavior in nature: the intelligent ants through the exchange of information and collaboration between individuals to find the optimal path from the nest to the food source.

At present, ant colony algorithm is mainly applied in TSP, quadratic assignment problem, job shop scheduling problem and network routing problem. It is also applied in pattern recognition fields, and many achievements have been made up to now.

A. The Basic Principles of Ant Colony Algorithm

The basic principles are understood as follows:

Ants communicate and cooperate with each other by releasing pheromones [12], ants through the concentration of pheromones to choose their path during movement, and release their own pheromones. The denser the pheromone concentration of the path the more likely ants will choose the path. Therefore, the pheromone concentration will be greater on the path which ants usually move. As time goes by, pheromone concentration will get smaller, and the pheromone concentration will get smaller and smaller on the path ants seldom choose.

When the quantity of ants is large, there will appear the positive feedback of pheromone, until ants find the shortest path from the nest to the food source.

The execution flow of ant colony algorithm is described as Fig. 1. When the algorithm starts, it will initialize the ant colony, and build up the search path by the ants individually.

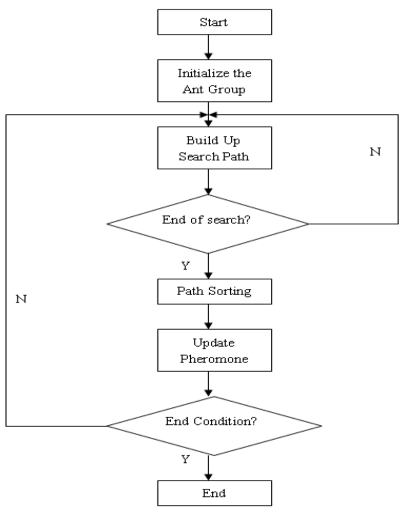


Figure 1. Execution Flow of Ant Colony Algorithm

When all the ants accomplish to build up the search path, the system will sort the entire path, if not, the ants will continue to build up the search path. After then, the Pheromone which has been left on the path will update. If the entire ants go to the end condition, the algorithm will be end. If not, the algorithm will continue.

B. The Mathematical Model of the Ant Colony Algorithm

Ant colony algorithm was proposed first in solving TSP, TSP means traveling salesman problem, in order to seek an optimal way for salesman [13]. In ant colony algorithm, the salesman is simulated as individual ants. In the walking process, all the ants calculate the state transition probability according to the amount of information on various paths.

The ant system simply iterates a main loop where m ants construct in parallel their solutions, thereafter updating the trail levels.

Given $Ant_k (k = 1, 2, \ldots, m)$, set $p_{ij}^k(t)$ as the state transition probability from $City_i$ to $City_j$ at the moment of t . $p_{ij}^k(t)$ is described as (3).

$$p_{ij}^k(t) = \begin{cases} \dfrac{[\tau_{ij}(t)]^\alpha \bullet [\eta_{ik}(t)]^\beta}{\sum\limits_{s \subset allowed_k} [\tau_{is}(t)]^\alpha \bullet [\eta_{is}(t)]^\beta}, & j \in allowed_k \\ 0, & otherwise \end{cases} \quad (3)$$

In (3), α means the information stimulating factor, this parameter reflects the relative importance of track, β means the expected stimulating factor, said the relative importance of visibility. $\eta_{ij}(t)$ shows the inspire function, it is described as (4).

$$\eta_{ij}(t) = 1 / d_{ij} \quad (4)$$

In (4), d_{ij} means the distance from $City_i$ to $City_j$. $\eta_{ij}(t)$ shows the expectations of ants move from $City_i$ to $City_j$.

After a moment, all the ants complete a cycle. The pheromone left on each path will be adjusted as (5).

$$\begin{cases} \tau_{ij}(t+n) = (1-\rho)\tau_{ij}(t) + \Delta\tau_{ij}(t) \\ \Delta\tau_{ij}(t) = \sum\limits_{k=1}^{m} \Delta\tau_{ij}^k(t) \end{cases} \quad (5)$$

In (5) ρ means the information volatile factor. $1-\rho$ means the information of track attenuation coefficients, $\Delta\tau_{ij}(t)$ means the pheromone incremental left on the path from $City_i$ to $City_j$, $\Delta\tau_{ij}^k(t)$ means the amount of information left on the path from $City_i$ to $City_j$.

According to the updating strategy of pheromone, M.Dorigo brought up three different ant colony algorithm models: Ant-Cycle, Ant-Quantity and Ant-Density.

The difference between the three algorithms is $\Delta\tau_{ij}^k(t)$, among the three algorithms, Ant-Quantity and Ant-Density use the local information, Ant-Cycle uses the global information. In Ant-Quantity and Ant-Density models, each ant lays its trail at each step, without waiting for the end of the tour.

In Ant-Density model a quantity Q of trail is left on edge (i, j), every time an ant goes from i to j, Therefore, in the ant-density model, $\Delta\tau_{ij}^{k}(t)$ is described as (6).

$$\Delta\tau_{ij}^{k}(t)=\begin{cases}Q, & Ant_k\in Collation\\ 0, & Others\end{cases} \tag{6}$$

In Ant-Quantity model, an ant going from i to j leaves a quantity $\frac{Q}{d_{ij}}$ of trail on edge (i, j), every time it goes from i to j. So, in Ant-Quantity model, $\Delta\tau_{ij}^{k}(t)$ is described as (7).

$$\Delta\tau_{ij}{}^{k}=\begin{cases}\frac{Q}{d_{ij}}, Ant_k\in Collation\\ 0, Others\end{cases} \tag{7}$$

III. Solution Of MRTA Based On Ant Colony Algorithm

A Model of MRTA

So far, some models of multi-robot coordination in the literature list as follows:

Ref. [14] proposed a formalism of information invariants. In these formalism models, the information requirements of a coordination algorithm and provides a mechanism to perform reductions between algorithms. Ref. [15] developed a prescriptive control-theoretic model of multi-robot coordination and showed that it is used to produce a precise multi-robot box-pushing.

MRTA based on ant colony algorithm is abstracted as the hierarchy model shown in Fig. 2. Low-level refers to the task group composed by disparate ant individuals. Each task is undertaken by different ants. High-level refers to task assignment group formed by task groups.

At the low level, as to multi-robot task coalition 1, the robots coalition is formed based on ant colony algorithm [16]. At high level, task is assigned also based on ant colony algorithm. High level ant colony algorithm is aimed at achieving optimal task allocation, and let ants represent tasks and choose the holders for each task. The low level is to form the robot coalition to generate tight coupling task solutions.

The system's overall workflow is as follows:

At first, the ants are standing at the high level, starting from the first task, to choose the proper holder. When the ants encounter a tight coupling task, they move to the low level and call the corresponding coalition to form the algorithm, and then accomplish the task cooperative.

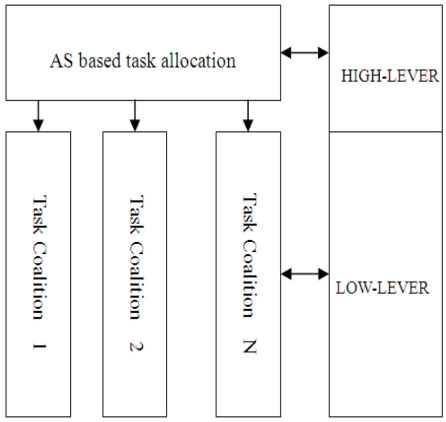


Figure 2. Hierarchy Architecture of System

B Coalition of Robot

In this paper, ant colony algorithm is informally defined as a multi-robot system inspired by the observation of some real ant colony behavior exploiting simmer. All the tasks is seen as foods, the ant is seen as individual ants, coordination among ants is achieved by exploiting the stigmatic communication mechanism.

Given m tasks on n robots randomly, for Ant_k, the probability of choosing $Robot_j$ is (8).

$$p_{ij}^{k}=\frac{[\tau_{ij}(t)]^{\alpha}[1/d_{ij}]^{\beta}}{\sum_{u\in J_k}[\tau_{iu}(t)]^{\alpha}[1/d_{iu}]^{\beta}}, j\in J_k \tag{8}$$

In (8), J_k is the robot collection for which Ant_k doesn't select. $\tau_{ij}(t)$ means the residual pheromone amount on connection i, j at the moment of t. $d_{ij}(i, j=1,2,\ldots n)$ means the distance from $Robot_i$ to $Robot_j$, i.e. communication cost. The two parameters α and β show the intensity of accumulated pheromones on the path and the weight of communication cost.

After selecting one robot, if the ant finds the present coalition can complete the task, it will stop path searching. When all the ants have completed a solution, one cycle is completed. Take the coalition which gets the maximal income in this cycle as the present optimal solution, and updates the intensity of pheromone as (9) and (10).

$$\tau_{ij}(t+1)\leftarrow\rho\tau_{ij}(t)+\sum_{k=1}^{m}\Delta\tau_{ij}^{k} \tag{9}$$

$$\Delta\tau_{ij}^{k}=\begin{cases}\frac{Inc(C_k)}{\sum_{k=1}^{m}C_k}, Robot_K\in Coalition\\ 0, Others\end{cases} \tag{10}$$

In (8), $Inc(C_k)$ means the income of coalitions for which Ant_k formed. In the algorithm, for parameters α,

β and ρ, experiment methods are used to determine the optimal combination; fixed evolution generation is used as the stop condition, or when the evolutionary trend is not obvious, the calculation will be stopped.

In order to compute the complexity of the algorithm for MRTA, we usually take the qualitative quantitative way. In a project composed of m tasks and n robots, we use ant colony algorithm for distributing, T is the iteration time, time complexity [17] of the algorithm is described as (11):

$$T(o) = O(f(n)) = m \times (m-1)^2 \times n \times T / 2 \qquad (11)$$

Given $m < n$, $T = k \times n$, when $n \to \infty$, $T(o)$ is described as (12).

$$T(o) = O(n^4) \qquad (12)$$

We know that, with the increased amount of transaction and the robots number, the time complexity of the algorithm becomes higher.

C Task Allocation

The process of task allocation is described as follows:

At first, the entire ants stand at the high level. For the first task, the algorithm chooses the suitable ants as the undertakers.

Then, if the current task is accomplished by a single robot, the system will find the optimal solution directly.

If the task can't be accomplished by a single robot, that means the task is tightly coupled, all the ants will move to the low level, some suitable ants will be assigned to undertake the task. In MRTA, it seems robots as ants, uses ant colony algorithm to solve the problem.

Set the number of ants is m, each task is node 0, candidate robots or robot coalition are nodes 1～n, then given Ant_k start walking from node 0, the probability of choosing node j is (13).

$$p_{ij}^k = \frac{[\tau_{ij}(t)]^\alpha [1/Cost_{ij}]^\beta}{\sum_{u \in J} [\tau_{iu}(t)]^\alpha [1/Cost_{iu}]^\beta}, j \in J_k \qquad (13)$$

In (11), J_i is the collection of candidate robots or robot coalition of $Task_i$. $Cost_{ij}$ is the cost of robot or robot coalition completing $Task_i$. For each robot, the cost of completing a task is the distance between the robot, task and the consumption of its ability. For robot coalition, the cost of completing task is $Cost(C,t)$.

For Ant_k, the first task node of to be assigned task list is the starting point for optimal path. After Ant_k finished choosing the holder for the task, it will move to the next task, and choose another holder for it. After Ant_k finished choosing the holders for all the tasks, one task allocation is completed. When all the ants have completed the solutions, one cycle is completed. Take the coalition which gets the maximal income in this cycle as the present optimal solution, and updates the intensity of pheromone as (14).

$$\Delta \tau_{ij}^k = \frac{Q}{\sum_{k=1}^{m} Cost_{kj}}, Q \in C \qquad (14)$$

Algorithm flow is as follows:

Given $t = 0$, $NC = 0$, $\tau_{ij}(0) = \tau_0$, $\Delta \tau_{ij}^k = 0$, $NumTask = s$, $NumAnt = m$, $NumRobot = n$, the capacity needs of the task, the ability each robot has and the corresponding ability cost.

STEP 1

for i = 1 to s

for k = 1 to m do

Ant_k Starts from the first task, and determines whether the current $Task_i$ is a tight coupling task, if it is, then turn to step6. If not, Ant_k will choose a task holder from J_i according to the probability P_{ij}^k of (6) and calculate the income of completing the task. Then, Ant_k will move to the next task and repeat the above actions until all tasks are assigned the holders.

STEP 2

Calculate the total income of task allocation corresponding to each ant. Update the maximal income and the corresponding allocation plan.

STEP 3

Update the intensity of pheromone [18] $\tau_{ij}(t+1)$ according to (9) and (10).

STEP 4

Set (13) as follows.

$$\begin{cases} t = t+1 \\ NC = NC+1 \\ \Delta \tau_{ij} = 0 \end{cases} \qquad (15)$$

STEP 5

If ($NC < NC_{\max}$ [19]) then go to Step1, output the task allocation plan for which can get maximal incomes.

STEP 6

Organize the robot coalition of tasks using ant colony algorithm, and then return to step 1.

After the 6 steps above, the high-level ant colony algorithm completes the task allocation of the whole system by using the low-level ant colony algorithm.

IV. SIMULATION PROCESS

A. Parameters of Experiment

In order to validate the algorithm, we take information transmission as the background, there are a number of tasks, some of them are loose type tasks, i.e. task is completed by single robot independently [20]; some are tight coupling tasks, i.e. the task is completed by some robots' cooperation. The number of robots and tasks is set as it is required, and each task and each robot has a corresponding capacity vector. The work space is 5×5 of four zones, shows as Fig. 3.The algorithm is achieved on MATLAB. It is a numerical computing environment and fourth-generation programming language developed by MathWorks. MATLAB can support a dynamic environment control system of multi-agent simulation, and provide visualization capabilities.

-5, 5	-4, 5	-3, 5	-2, 5	-1, 5	0, 5	1, 5	2, 5	3, 5	4, 5	5, 5
-5, 4	-4, 4	-3, 4	-2, 4	-1, 4	0, 4	1, 4	2, 4	3, 4	4, 4	5, 4
-5, 3	-4, 3	-3, 3	-2, 3	-1, 3	0, 3	1, 3	2, 3	3, 3	4, 3	5, 3
-5, 2	-4, 2	-3, 2	-2, 2	-1, 2	0, 2	1, 2	2, 2	3, 2	4, 2	5, 2
-5, 1	-4, 1	-3, 1	-2, 1	-1, 1	0, 1	1, 1	2, 1	3, 1	4, 1	5, 1
-5, 0	-4, 0	-3, 0	-2, 0	-1, 0	0, 0	1, 0	2, 0	3, 0	4, 0	5, 0
-5, -1	-4, -1	-3, -1	-2, -1	-1, -1	0, -1	1, -1	2, -1	3, -1	4, -1	5, -1
-5, -2	-4, -2	-3, -2	-2, -2	-1, -2	0, -2	1, -2	2, -2	3, -2	4, -2	5, -2
-5, -3	-4, -3	-3, -3	-2, -3	-1, -3	0, -3	1, -3	2, -3	3, -3	4, -3	5, -3
-5, -4	-4, -4	-3, -4	-2, -4	-1, -4	0, -4	1, -4	2, -4	3, -4	4, -4	5, -4
-5, -5	-4, -5	-3, -5	-2, -5	-1, -5	0, -5	1, -5	2, -5	3, -5	4, -5	5, -5

Figure 3. Work Space of Robots

The entire robot scattered among the space of a certain position, each robot can move in the space, goods are stored in a certain position of the space.

Given the numbers of ants as $m = 8$, the maximum number of iterations as $NC_{\max} = 300$, $rew(T_i) = 700$, $\varpi_1 = \varpi_2 = 1$, $\delta = \mu = \lambda = 0.5$, $Q = 1$, $\alpha = 1.5$, $\beta = 2$, $\rho = 0.7$. The capacity and cost vector of robots are provided in TABLE I.

TABLE I
CAPACITY AND COST VECTOR OF ROBOTS

Robot	Capacity	Cost vector
R_0	80	103
R_1	102	160
R_2	210	258
R_3	120	178
R_4	062	104
R_5	087	115
R_6	185	230
R_7	164	210

As there exist the different situations: some tasks are completed by single robot independently. Some tasks are completed by some robots' cooperation. In the following part, some tasks belong to single robots, and someone belong to multi-robots [21]. Table II provides the information about the tasks, where the ant stands.

Using the parameters above, we do the experiment for the information transmission task, after the experiment, we can estimate the performance of ALLIANCE, BLE, M+ and ant colony algorithm for MRTA. The Performance indicators of MRTA contain robustness, speed, extensibility, heterogeneity, flexibility and so on.

In this paper, we make the main contrast of robustness and speed for the four algorithms.

TABLE II
INFORMATION OF TASKS

Tasks	Required capacity	Location
T_0	210	(4,3)
T_1	300	(0,0)
T_2	130	(-5,-3)
T_3	120	(-2,4)
T_4	229	(-4,-1)
T_5	112	(3,0)
T_6	332	(0,-4)
T_7	106	(2,3)

B. Robustness

Robustness means the multi-robots systems can continue to accomplish the task, even if some robots' Failure. In the test, we take the failure times of the task allocations for measuring the Robustness of the four algorithms. In the process of cooperation, the conflict may appear. This will lead to error in distribution and handling, so contrast the collaboration efficiency is crucial to the distribution process.

By comparing the ALLIANCE, BLE, M+ and ant colony algorithm, we know that, conflict times of ant colony algorithm is fewer. The results are shown as Fig. 4.

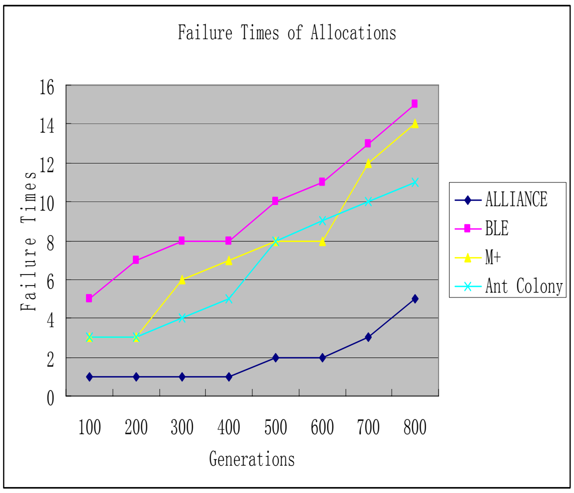


Figure 4. Failure Times of allocations

By anglicizing of the experiment results, we know that the Robustness of ant colony algorithm is strong, as the increasing generation, the failure times will present linear growth trend.

C. The Best Incomes

Speed is the most important index for the multi-robots systems. The best incomes describe the task execution efficiency. Using the data for which provided by table I and table II, we make the test of ALLIANCE, BLE, M+ and ant colony algorithm for MRTA, in the test, we set the population 3, the best incomes of the algorithms is shown as Fig. 5.

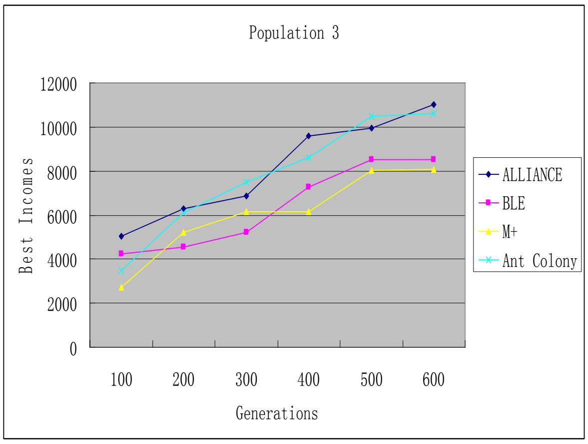


Figure 5. The Best Incomes of Population 3

D. Results of the Experiments

At present, ant colony algorithm has been used for solving the cooperation of the robots, but the algorithm is only limited to the loosely distributed tasks, solving the large-scale distributed tasks by ant colony algorithm is uncommon. This paper introduces a new methodology for MRTA based on Ant Colony Algorithm.

The main contributions of this paper are the following:

(1) We discuss the coral-problem to the formation of the robot alliance, and use ant colony algorithm for resolving the alliance formation of multi-robots.

(2) We use ant colony algorithm for solving the large-scale task allocation.

(3) We realize the task allocation process by building up the hierarchy model, using the hierarchy structure we divide MRTA problem into low-level and high-level.

(4) We realize the algorithm simulation program by MATLAB, using the program we do the experiment for the information transmission task, and estimate the performance of ALLIANCE, BLE, M+ and ant colony algorithm for MRTA.

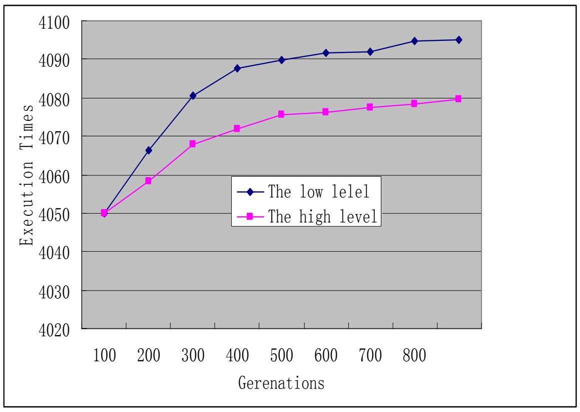


Figure 6. Average Solution Evolving Curve

40 times of calculation of ant colony algorithm are made respectively by using the above parameters and the calculation results of robot coalition and high-level task allocation is obtained. The achieved results are analyzed and compared as shown in Fig. 6.

V. Conclusion

We do the simulation of MRTA based on ant colony algorithm, in order to comparing with ALLIANCE, BLE, M+. By analyzing the experiment data of the four algorithms, we get the conclusion as follows:

(1) Robustness

By test the four algorithms, we know that the robustness of ALLANCE is the stranger one. In the beginning of system, the robustness between M+ and ant colony is the same, as the growing of the generations, the failure times of M+ will grow rapidly, but ant colony is more stable. As the generation grows, the robustness of ant colony will be stranger.

(2)The Best Incomes

By analyzing the experiment data, we know that , in the beginning of the system, the best incomes of ant colony is the lower, but as the generations grows, the best incomes of ant colony will be more higher.

(3)The efficiency of solution evolving between the low level and the low level:

By testing the executing times of the low level and the low level, we know that, the coalition of robots will take longer times. And when the coalitions have been shaped, all the tasks allocation will becomes more easily.

So, we can draw the conclusion that, ant colony algorithm is suitable for solving robot coalition problems because of its high degree of ability and stability. While the corresponding high-level task allocation has relative low efficiency as the numbers of ants and tasks are not limited. Considering the pheromones distribution problems, this may cause the imbalance in assigning to a task, the problem will lead to the heavy workload of some robots, and the others have few workloads. Therefore, in the actual use, the algorithm may need guidance.

As the development of the robotics, how to improve the stability and the efficiency of the algorithm becomes more important. Future work includes the implementation of ant colony algorithms with real robots, and the improvement of the stability and the efficiency of ant colony algorithm for MRTA. To realize fair distribution based on ant colony algorithm of MRTA need to be studied further.

Acknowledgment

This work was supported by a grant from the Social Science Association of Henan Province (No. SKL-2010-1146 and No. SKL-2011-2381).

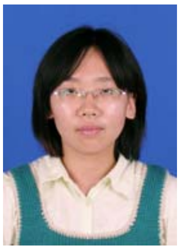
References

[1] ZHANG Yu, LIU Shu-Hua, “Survey of Multi-Robot Task Allocation,” CAAI Transactions on Intelligent Systems, Vol.3 No.2, pp: 115-120, April 2008.

[2] Gerkey B P, Mataric M J. “A framework for studying multi-robot task allocation,” In Proceedings of the NRL Workshop on Multi-robot systems, pp: 17-19, 2003.
[3] Kalra N, Martinoli, “A Comparative Study Of Market-Based And Threshold-Based Task Allocation,” Distributed Autonomous Robotic Systems, Tokyo: Springer, 2006.
[4] Fang Tang, “Automated Synthesis of Multi-Robot Task Solution through Software Reconfiguration,” Proc of IEEE Int Conf on Robotics and Automation, Barcelona, Spain, pp:1501-1508, 2005.
[5] Botelho.S, Alami.R, “M+: a scheme for multi-robot cooperation through negotiated task allocation and achievemen,” Proc of the 1999 IEEE International Conf on Robotics and Automation, Detroit, Michigan, pp: 1234-1239, 1999.
[6] L. E. Parker, “ALLIANCE: An Architecture for Fault Tolerant Multi-robot Cooperation,” IEEE Transactions on Robotics and Automation, Vol. 14, No. 2, pp. 220–240, April 1998.
[7] B. B. Werger, M. J. Matari, “Broadcast of Local Eligibility for Multi-Target Observation,” in Distributed Autonomous Robotic Systems, Springer, pp: 347–356, 2001.
[8] Fang Tang, Lynne E. Parker. “ASyMTRe: Automated Synthesis of Multi-Robot Task Solutions through Software Reconfiguration,” Proc of IEEE Inte Conf on Robotics and Automation, Barcelona, Spain, May 2005.
[9] M. Dorigo, M. Birattari, and T. Stitzle, “Ant Colony Optimization: Arificial Ants as a Computational Intelligence Technique,” IEEE computational intelligence magazine, November, 2006.
[10] ZLOT R, STENTZ A, “Complex task allocation for multiple robots,” Proc of IEEE Int Conf Robot Atom, pp: 1515-1522, 2005.
[11] VIG L, ADAMS J A, “Market-based multi-robot coalition formation,” Proc s of the 8th Inte Symposium on Distributed Autonomous Robotic Systems, Minneapolis, USA, pp: 227-236, 2006.
[12] Liu, S. H, Zhang Y, “Multi-robot task allocation based on particle swarm and ant colony optimal,” Journal of Northeast Normal University, Vol.41, No.4, pp: 68-72, 2009.
[13] Huang Bo, Yan Li-Na, “Superiority evaluation algorithm-based task allocation strategy of robot soccer systems,” Journal of Huazhong University of Science and Technology, pp: 38-44, 2010.
[14] Donald. B, Jennings. J and Rus. D, “Information invariants for distributed manipulation,” The Intl. J. of Robotics Research, pp: 673–702, 1997.
[15] Spletzer. J.R, Taylor. C. J, “A Framework for Sensor Planning and Control with Applications to Vision Guided Multi-robot Systems,” Proc of Computer Vision and Pattern Recognition Conf, Kauai, Hawaii, pp: 378–383, 2001.
[16] Liu, S. H, Zhang Y, “Multi-robot task allocation based on swarm intelligence,” Journal of Jilin University, Vol.40, No.1, pp: 123-129, 2010.
[17] XU Ju-xiang, LIU Guo-dong, “Ability Evaluation of Multi-Robot Based on Set-Pairs Proximity,” Journal of Jiangnan University, pp: 56-72, 2009.
[18] XIE Ping, LIU Zhi-jie, “Error Compensation Method of Parallel Robot Based on Ant Colony Algorithm,” Computer Engineering, pp: 56-70, 2011.
[19] XU Jiang-le, XIAO Zhi-tao, “An Improved Intelligent Ant Colony Optimization Based on Genetic Algorithm,” Microelectronics Computer, pp: 67-74, 2011.
[20] Zhang Dan-dan, “Adaptive Task Assignment for Multiple Mobile Robots Via Swarm Intelligence Approach,” Robotics and Autonomous Systems, pp: 52-58. 2007.
[21] GERKEY B P, MATARIC M J, “A formal analysis and taxonomy of task allocation in multi-robot systems,” International Journal of Robotics Research, pp: 939-954, 2004.

Jian-Ping WANG (1981-), was born in BaoJi, Shaanxi Province of China. He received his B.S degree in 2004 from Shaanxi Normal University, his M.S degree in 2010 from Nanjing university of science and technology. His current research area is computer network technology and robot system. He is a lecture in the School of Information Engineer, Henan Institute of Science and Technology, Xinxiang, Henan Province of China.

Yuesheng Gu: birth 1973, male, vice professor of Henan institute of science and technology. His current research area is computer network technology and artificial intelligence.



Xiao-Min Li received bachelor's degree in Electrical Engineering at Southwest Jiaotong University, then received a master's degree in electronics and communication engineering at Nanjing University of Science and technology.
She is teaching at Henan Institute of Science and Technology now, more than ten papers have been published.