Information Technology Journal
Year: 2009 | Volume: 8 | Issue: 2 | Page No.: 156-164
DOI: 10.3923/itj.2009.156.164

Research Article

A Hybrid Heuristic Ant Colony System for Coordinated Multi-Target Assignment

Bo Liu, Zheng Qin, Rui Wang, You-Bing Gao and Li-Ping Shao

Abstract

The aim of this study is to solve the target assignment of coordinated distributed multi-agent systems. Earlier methods (e.g., **neural network**, genetic algorithm, ant colony algorithm, particle swarm optimization and auction algorithm) used to address this problem have proved to be either too slow or not stable as far as converging to the global optimum is concerned. To address this problem, a new algorithm is proposed which combines heuristic ant colony system and decentralized cooperative auction. Based on ant colony system, the decentralized cooperative auction is used to construct ants` original solutions which can reduce the numbers of blind search and then the original solutions are improved by heuristic approach to increase the system stability. The performance of the new algorithm is studied on air combat scenarios. Simulation experiment results show present method can converge to the global optimum more stably and faster by comparing the original methods.

INTRODUCTION

Target assignment of coordinated distributed multi-agent systems is an important yet difficult task. Research performed about it as task allocation, weapon-target assignment, has already made remarkable progresses and almost all researches have focused on the coordinated system which is a case of explicit cooperation (Khashayar *et al*., 2003) where agents in a team must work synchronously with respect to time or space in order to achieve a goal. Examples of target (task) assignment of multi-agent systems are coordinated multiple target attack in air combat, environmental monitoring, scientific data collection (e.g., Moon car on the surface of Moon). Typically, the target assignment can be formulated as a nonlinear integer programming problem and is known to be NP-complete. The goal of this problem is to find a proper assignment of agents-to-targets with the objective of maximizing the global utility function subject to a set of constrains.

Consider an air combat scenario where a command center has to reallocate their weapons (referred to as the agent hereafter) carried by one or more of the platforms (fighters) to a set of targets. Agents access to the situation information from corresponding platforms which exchange the information with the Airborne Warning and Control System (AWACS) or Ground-based Radar. For example, in an air combat situation, as shown in Fig. 1, six agents carried by three platforms has to be assigned to four targets.

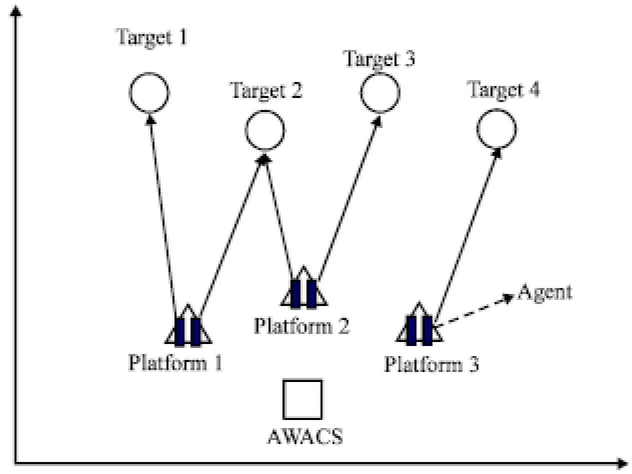

Fig. 1: Situation of a sample task assignment

Many kinds of different methods to solve it have been proposed, such as methods based on B-P **neural network** (Li and Tong, 1999), genetic or improved genetic algorithm (Luo *et al.*, 2005), ant colony algorithm and ant colony system (Lee *et al.*, 2002; Luo *et al.*, 2006a), particle swarm optimization algorithm (Luo *et al.*, 2006b), market-based approaches (Dias *et al.*, 2006) or market-based cooperation planning system (Anawat and Rysdyk, 2004) which combines the flexibility of evolution-computation with the distributed nature of market strategy, auction or decentralized cooperation auction (Palmer, 2003), collective intelligence (Agogino, 2003; David and Kagan, 1999). Bionic algorithm, represented by genetic algorithm and ant colony system, is applied widely in the general coordinated multi-target assignment, because it is an effective algorithm that can converge to global optimum stably. Auction algorithm, represented by decentralized cooperative auction and other auctions (Bogdanowicz and Coleman, 2007; Sujit *et al.*, 2006), is usually applied to solve target assignment problems subject to time or communication restraints. Collective intelligence, proposed by NASA/Ames Research Center, is major used to systems which agents do not need centralized control. But these methods have no ability to converge to global optimum both stably and fast. Generally, bionic algorithms converge slowly, auction algorithm cannot converge stably and collective intelligence is not suitable for heterogeneous multi-agent system.

To address this problem, in this study, a new algorithm is proposed that combines heuristic ant colony system and decentralized cooperative auction. Based on ant colony system, the decentralized cooperative auction is used to construct ants original solutions which can reduce the numbers of blind search and then the original solutions are improved by heuristic approach to increase the system stability. Simulation experiment results based on air combat scenarios show this method can converge to the global optimum more stably and faster to compare the original methods; the scalability of the algorithm has also been improved.

RELATED WORK ABOUT COORDINATED MULTI-TARGET ASSIGNMENT

Ant Colony System (ACS) for target assignment: For a collection of n homogeneous or heterogeneous agents that have to perform m interchangeable tasks at different costs because different initial conditions, this is a combinatorial optimization problem and the complexity of finding the optimal solution is np-hard if n and m are allowed to vary independently and agents are allowed to vary multiple or no tasks (Palmer *et al.*, 2003). In recent years, Ant Colony System (ACS) and varied improved ACS were proposed to address this problem. To avoid premature convergence or to locally optimize the solutions found by ants, a lot of work has been done to define and use heuristics to improve ACS. Zne-Jung *et al.* (2002) improved the ACS using immune system which could quickly find good solutions within a small region of the search space. Luo *et al.* (2006a) introduce an improvement heuristics which using special information determined by the value-driven approach (Lazarus, 1997) to locally optimize original solutions found by ants. Most existing researches on target (or task) assignment using ACS or other similar method, such as Particle Swarm Optimization (PSO), have focus on local optimization using improvement heuristics (Macro and Gambardella, 1997). These methods effectively improved the accuracy of convergence. But algorithm cannot converge to the global optimum fast because there are too many blind search progresses in the progress of constructing original solutions by ants. The effect of coordinated work among the agents is rarely considered.

Decentralized cooperative auction for target assignment: It is a challenging and difficult work that a collection of agents have to effectively map agents to multiple tasks in order to perform the tasks quickly with limited resources such as time-limited, communication bandwidth-constrained. To address this problem, the Cooperative Assignment of Simultaneous Tasks (CAST) auction (Palmer *et al*., 2003) is proposed. The algorithm starts by assuming each agent knowing information about itself and all possible tasks in the auction and builds task cost table for every agent. Agent bids on a task when it is its turn to do so. In first round of the auction, each agent computes an ordered choice list based on task cost. In second round and beyond, each agent`s biding order is generated by a pre-determined pseudo-random number generator, then the computed first bidder selects first choice from its cost table and broadcasts it and cost to all agents. Subsequent agents select their best task from their remaining choices until all tasks have been selected. The process repeats as time and resource constraints allow, with the lowest cost mapping retained (Palmer *et al*., 2003). The algorithm also introduces two technologies for allowing agents to make a bid for more tasks or nothing for the overall good. Bogdanowicz and Coleman (2007) introduce another auction algorithm for solving dependent sensor/weapon-target pairing problems. A negotiation scheme is proposed by Sujit *et al*. (2006) to allocate tasks for multiple UAVs. These auction algorithms has several desirable features, the most prominent feature is that it generates usable solutions every iteration. However, because the algorithm based on random approach, it cannot ensure stability of convergence.

The algorithms mentioned above cannot converge to the global optimum both stably and fast. So, we need to find other method to solve the problem.

COORDINATED MULTI-TARGET ASSIGNMENT PROBLEM

Scenario of the problem: Suppose in an air combat scenario, there are M(M ε ℕ) agents carried by Z platforms have to perform N(N ε ℕ) targets. Denote the agent set A = {i, i = 0, 1,...,M-1}, the target set T = {j, j = 0,1,...,N-1}, the platform set F = {k, k = 0,1,...,Z-1}. The probability of achieving target j by agent i can be viewed as the threat of i to j. For simplicity, suppose the threat of agent to target is equal to the threat of platform which carried the agent to target. The situation between platform k and target j is as shown in Fig. 2.

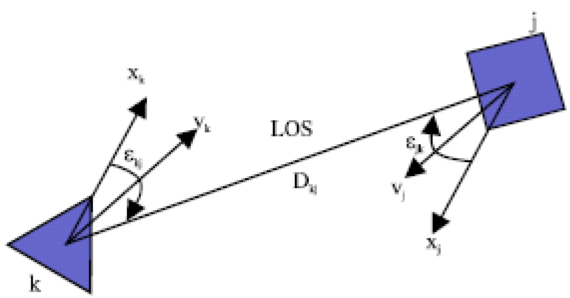

Fig. 2: The air combat situation between k and j

Where, LOS is the line of sight between the platform k and target j, D_{kj} is the distance between k and j. x_k and v_k are the body axis and velocity of k. ε_{kj} is the bore of sight angle of k to j. x_j, v_j and ε_{jk} are defined in the similar way. The threat of platform k to target k, th_{kj}, can be described as a composite function of its threat factors:

$$th_{kj} = \omega_1 P_{kj}^{D} \bullet P_{kj}^{\varepsilon} + \omega_2 P_{kj}^{v} + \omega_3 P_{kj}^{E} \quad (1)$$

where, ω_1, ω_2, ω_3 are non-negative weight coefficients and satisfy:

$$\omega_1 + \omega_2 + \omega_3 = 1 \quad (2)$$

where, P_{kj}^{D} is the distance threat factor, $P_{kj}^{D} \in [0,1]$, P_{kj}^{ε} is the angle threat factor, $P_{kj}^{\varepsilon} \in [0.1,1]$, P_{kj}^{v} is the velocity threat factor, $P_{kj}^{v} \in [0.1,1]$ and P_{kj}^{E} is the effectiveness threat factor, $P_{kj}^{E} \in [0,1]$ (Luo *et al*., 2006a). Thus, there is th_{kj} ε [0,1].

Based on the fore-mentioned assumption, the threat of i to j, $th_{ij} = th_{kj}$, th_{ij} ε [0,1]. The threat of agent i to target j can be viewed as a probability of j is destroyed by i which sharing the information and effectiveness of the corresponding platform. The threat of target j to platform k or agent i, th_{jk}, th_{ji}, is defined in the similar way.

The model for the problem: Based on the fore-mentioned scenario, the problem is to find a proper agent-target assignment solution π* to optimize the global utility function G(π) which is mathematically defined and can rate all possible combination of agent-target.

$$\pi^* = G^{-1}\left(\underset{\pi\in\Omega}{\text{optimal}}\, G(\pi)\right) \quad (3)$$

where, π represent the agent-target assignment solution, Ω represent solution space.

Suppose every platform carried L agents, so the total agent number is:

$$M = Z \cdot L \quad (4)$$

and suppose

$$N \le M \le 2N \quad (5)$$

Given the i-th agent of M is the h-th agent of the z-th platform, the i-th agent can be defined as:

$$i = (z-1) \cdot L + h,\ h = 0,1,\ldots,L-1 \quad (6)$$

Equation 6 indicates that i-th agent of m carried by z-th platform.

Based on actual situation of air combat, two assumptions are made. First, each target must be assigned at least one and no more than two agents. Second, all agents must be assigned to targets (Luo *et al*., 2006a). The threat th_{ij} can be viewed as the probability of agent i to destroy the target j. If the i-th agent is assigned to the target j, j survives with a probability of l-th_{ij}. The expect remaining threat of the target j is:

$$(1 - th_{ij}) \bullet \sum_{k=0}^{Z-1} th_{jk}$$

The global utility can be expressed as G(π), defined in Eq. 7, as far as the total remaining threat is concerned, or G`(π), defined in Eq. 8, as far as the total elimination of threat is concerned.

$$G(\pi) = \sum_{j=0}^{N-1}\sum_{k=0}^{Z-1}\left\{th_{jk} \bullet [\prod_{i=0}^{M-1}(1 - th_{ij})^{X_{ij}}]\right\}, \pi \in \Omega \quad (7)$$

$$G'(\pi) = \sum_{j=0}^{N-1}\sum_{k=0}^{Z-1}\left\{th_{jk} \bullet [1 - \prod_{i=0}^{M-1}(1 - th_{ij})^{X_{ij}}]\right\}, \pi \in \Omega \quad (8)$$

where, X_{ij} indicates the assignment situation of agent i, X_{ij} = I, indicates i is assigned to target j, X_{ij} indicates i is not assigned to j. The constraints of Eq. 7 and 8 are two assumptions mentioned above, that is:

$$\text{s.t.}\begin{cases}\sum_{i=0}^{M-1} X_{ij} \le 2, \forall j \in \{0,1,\ldots,N-1\} \\ \sum_{j=0}^{N-1} X_{ij} = 1, \forall i \in \{0,1,\ldots,M-1\}\end{cases} \tag{9}$$

It can be easily deduced:

$$G'(\pi) = \sum_{j=0}^{N-1}\sum_{k=0}^{Z-1} th_{jk} - G(\pi), \pi \in \Omega \tag{10}$$

The situation information is the same in a time slice, so the

$$\sum_{j=0}^{N-1}\sum_{k=0}^{Z-1} th_{jk}$$

is a fixed value, Eq. 8 is equivalent to Eq. 7. The difference is that it must be maximize for the G(π) and minimize for the G\`(π).

HYBIRD HEURISTIC ANT COLONY ALGORITHM

Decentralized cooperative auction: For the decentralized cooperative auction (Palmer *et al.*, 2003), the solution of a proper assignment of agents-to-targets based on the model defined in the Eq. 7 is produced as follows:

- **Step 1:** Each agent builds its cost table using the information about itself and all possible tasks in the auction
- **Step 2:** In first round, each agent computes its choice order list and biding order based on the cost table
- **Step 3:** In second round and beyond, algorithm randomly generates the biding order for each agent by pre-determined pseudo-random number generator
- **Step 4:** The first bidder generated in step 2 or 3 selects a task from the cost table using a greedy algorithm and broadcasts it to all agents
- **Step 5:** Subsequent agents select their best tasks in turn from their remain choices until all tasks have been selected
- **Step 6:** If the constraints allowed, repeat the step 3 to 5, each time retains the best solution

Heuristic ACS: Decentralized cooperative auction algorithm can find a good solution with less computation and communication. But it cannot converge to the global optimum stably. So, the solution generated by decentralized cooperative auction has to be optimized in order to enhance its stability. ACS and heuristic ACS have been proved that it outperform other nature-inspired algorithms such as simulated annealing and evolutionary computation in the middle-scaled combination optimization problems (Macro Dorigo and Gambardella, 1997). So, in this research, ACS is used to solve this problem.

Generally, the ACS (Macro and Gambardella, 1997) works as follows: m ants are initially positioned on agent, then assigning the target chosen according to some initialization rule (e.g., randomly) to it. Each ant builds a solution by repeatedly applying the state transition rule until all agents have been assigned. While constructing its solution, an ant also modified the amount of pheromone on the assigned edges by applying the local updating rule. Once all ants have conducted their solution well, the amount of pheromone on edges is modified by applying the global updating rule.

The state transition rule of ant s, s ε {0,1,...,S-1} (S is the number of ants in the system) in the time (or step) t is as follows:

$$j = \begin{cases} \arg\max_{u \in allowed_s(t)} \{[\tau_{ru}(t)]^{\alpha} \cdot [\eta_{ru}]^{\beta}\}, & \text{if } q \le q_0 \\ J, & \text{otherwise} \end{cases} \tag{11}$$

where, τ_{ru} (t) is the pheromone, η_{ru} is the visibility of ant. α and β are parameters that control the relative importance of trail versus visibility. q is a random number uniformly distributed in [0,1], q_0 is a parameter ($0 \le q_0 \le 1$). J is chosen by Eq. 12:

$$p_{rj}^{k}(t) = \begin{cases} \dfrac{[\tau_{rj}(t)]^{\alpha} \cdot [\eta_{rj}]^{\beta}}{\sum_{u \in allowed_s(t)} [\tau_{ru}(t)]^{\alpha} \cdot [\eta_{ru}]^{\beta}}, & \text{if } j \in allowed_s(t) \\ 0, & \text{otherwise} \end{cases} \tag{12}$$

The local updating rule is as follows:

$$\tau_{rj}(t+1) = (1-\xi) \bullet \tau_{rj}(t) + \xi \bullet \tau_0 \tag{13}$$

where, ξ(0<ξ<1) is the pheromone decay parameter, $\tau_0 = (m \cdot E_{mn})$, m is the number of agents, n is the number of targets, E_{mn} is the value of solution produced by the nearest neighbor heuristic.

The global updating rule is as follows:

$$\begin{aligned} &\tau_{rj}(t+1) = (1-\rho) \bullet \tau_{rj}(t) + \rho \bullet \Delta\tau_{rj}(t) \\ &\Delta\tau_{rj}(t) = \begin{cases} [E(\pi_{elitist})]^{-1} & \text{if } (r,j) \in \pi_{elitist} \\ 0 & \text{otherwise} \end{cases} \end{aligned} \tag{14}$$

where, ρ(0<ρ<1) is the pheromone decay parameter and $E(\pi_{elititst})$ is the value of the globally best solution from the beginning of the trial.

To speed up the convergence, Luo *et al*. (2006a) introduces the Eq. 15 and 16 as the heuristic information to locally optimize the solutions found by ants.

$$\eta_{ij} = ASM(i,j) = th_{ij} \bullet \sum_{k=0}^{Z-1} th_{jk} \tag{15}$$

$$DIF(r,i,j) = |ASM(r,j) - ASM(i,j)| \tag{16}$$

ASM(i, j) named assignment value and DIF (r, i, j) named difference assignment value are based on prior attack principle (Luo *et al*., 2006a; Lazarus, 1997). Based on ASM (i, j), the proposed algorithm generates solutions and then DIF(r, i, j) is applied to locally optimize them.

Hybrid Heuristic ACS(HHACS): Although the search efficiency of the fore-mentioned heuristic ACS (Luo *et al*., 2006a) is superior to that of the basic Ant Colony Algorithm (ACA), it did not consider the effect of multi-agent coordinated work (attack) which would cause the algorithm to converge to the global optimum not stably and fast.

To address this problem, we defined the new ASM` (i, j) and DIF` (r, i, j) in which the effect of coordinated work (attack) among agents is considered. And the η_{ij} is also defined to adapt to the coordinated work of multi-agent.

$$\eta_{ij} = th_{ij} \cdot [\prod_{r=0, r\neq i}^{M-1} (1 - th_{rj})^{X_{rj}}] \cdot \sum_{k=0}^{Z-1} th_{jk} \quad (17)$$

$$ASM'(i, j \mid r) = th_{ij} \cdot (1 - th_{rj})^{X_{rj}} \cdot \sum_{k=1}^{Z-1} th_{jk} \quad (18)$$

$$DIF'(r, i, j) = \left| ASM'(r, j \mid i) - ASM'(i, j \mid r) \right| \quad (19)$$

The ACS can be effectively improved by using Eq. 7 as constructive heuristics to generate the original solutions and then apply the defined ASM` (i, j|r) and DIF` (r, i, j) as the improvement heuristics to locally optimize the original solutions. The procedure of using heuristic improvement to locally optimize the solutions is described as follows:

- **Step 1:** Find out targets that only allocated 1 agent in π_k, put them into the set C, assuming the number of target in C is n
- **Step 2:** According to Eq. 19, find out the agent pair (r, i, j) that has the largest difference assignment value. Compare ASM` (r, j|i) to ASM` (i, j|r), choosing the agent that have the smaller assignment value. For example, i; (if ASM` (r, j|i) = ASM` (i, j|r), select one randomly)
- **Step 3:** Select a target u for i from set C randomly, if the i` s assignment value to u is bigger than the to the j, i is reassigned to u, otherwise keeping the current assignment
- **Step 4:** Remove u
- **Step 5:** Repeat step 3 and 4 for n_l times, n_l ($0_l \leq |C = n|$) is a random integer

In order to converge to the global optimum faster and stably, a new algorithm (HHACS) is proposed that combines heuristic ant colony system and decentralized cooperative auction. The procedure of the algorithm can be described in pseudo code as:

```
{
 Initialize and set the parameters S, q0, α, β, ξ, ρ, Emn;
 //In the process of computing the Emn, the
     decentralized cooperative auction is used, the first
     ant construct the solution according to the first
     round in auction, and other ants construct the
     solutions according to the second round and
     beyond in auction. The process is as mentioned
     above.
```

```
t ← 0;

for (every possible agent-to-target (i, j)){ τ_ij (t) ← τ_0; }
//initialize the pheromone trail, where i∈{0,1,...,M-1},
j∈{0,1,...,N-1}
while(the stop criterion is not satisfied){
 for (every ant s, s∈{0,1...,S-1}){
   allowed_s(t) ←{j_0, j_1,..., j_N-1}; //initialize the allowed
 targets set of every ant
   agent_s(t) ←{i_0,i_1,...,i_M-1}; //initialize the usable agent set
   of every ant
   }
   While(agent_s(t)≠Ø){
     Select an agent randomly, for example i, then update
     agent_s(t) ← agent_s(t)- Eq. 11;
     Construct π_s←j according to the Eq. 17;
     //in Eq. 11, η_ij, is computed by Eq. 17;
     Update allowed_s(t) subject to the constraint defined
     in Eq. 9;
          Update the local pheromone τ_ij(t) according to
          the Eq. 13;
     }
    }
     For(every ant s){
   Locally optimize the solution π_s found by s according
     to the procedure of using heuristic improvement to
     locally optimize the solutions;
       }
   Find out the best solution π_elitist experienced so far;
   Update the global pheromone according to the Eq.  14;
    t ← t+1;
   }
   Output the best agents-targets assignment solution
   π_elitist;
   }
```

SIMULATIONS RESULTS

In the experimental scenario (Luo *et al*., 2006a), we consider N = 14, Z = 4 and L = 4, so there is M = 16.

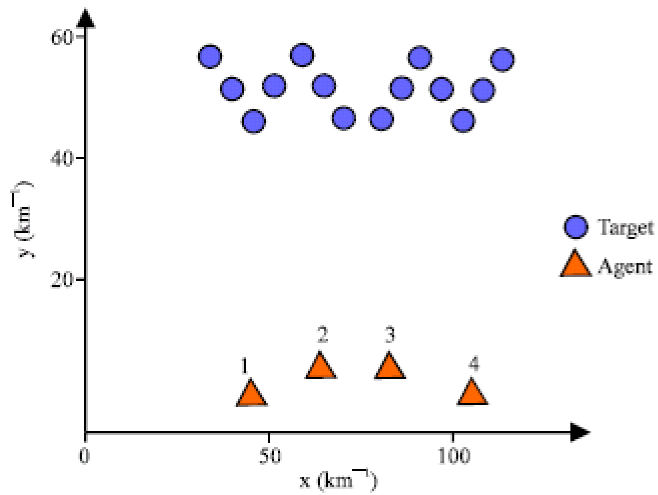

Fig. 3: The situation between agents and targets

The other conditions that determine the components of threats of agents-to-targets and targets-to-agents are the same as (Luo *et al*., 2006a) defined. The difference is that every weapon (missile) is defined as an agent in this paper other than every platform is defined as an agent.

The situation of agents and targets is shown with Fig. 3, in which the agents and the targets are supposed at the same altitude, for simplicity and agents has to be assigned to targets in the same time slice.

To verify our new algorithm`s efficiency, the algorithm is compared with the HGA presented in (Luo *et al*., 2005), HPSO presented by Luo *et al*. (2006b) and HACA presented by Luo *et al*. (2006a). The parameters are set according the settings in the corresponding paper. For the HGA, choosing the population size P_{size} = 50, the crossover rate P_c = 0.8 and the mutation rate P_m = 0.1. For the HPSO, choosing the number of particles p = 50, inertia weight w = 0.2, the acceleration coefficient $c_1 = c_2 = 2$. For the HACA, choosing the number of ants m = 30, q_0 = 0.9, α = 1, β = 2, ξ = 0.1, ρ = 0.8.

Based on the situation of Fig. 3, 100 trials are performed respectively under the same experiment condition with each running a fixed time and then the average result is taken as the result in order to eliminate the accidental circumstance caused by running the random number generator in the system. The results of four algorithms are shown in Table 1.

Where, N represents number of convergence to the global optimum, N_m represents minimal iteration number of convergence, N_a represents average iteration number of convergence, V_b represents optimal value that the algorithm achieved, V_a represents average value that the algorithm achieved, T_a/ms represents the average time of convergence. The global optimum is 1.914609 in this situation.

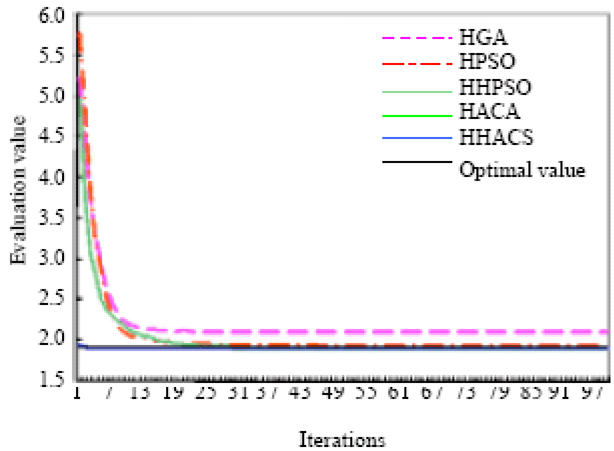

Fig. 4: Iterative processes of the algorithms based on situation of Fig. 3

Table 1: Comparison of simulation results based on scenario of Fig. 2

Algorithm	N	N_m	N_a	V_b	V_a	T_a msec^{-1}
HGA	23	8	18.61	1.914609	2.110114	157.52
HPSO	80	4	21.03	1.914609	1.945314	25.04
HACA	99	7	28.16	1.914609	1.915352	98.78
HHACS	100	1	1.27	1.914609	1.914609	0.96

As shown in Table 1, our new algorithm can find out the global optimum in each trial. To compare with the other three algorithms, our algorithm can converge to the global optimum faster and stably. Figure 4 shows the average iterative processes of four algorithms based on the situation of Fig. 3.

To verify the algorithm`s generality, the situation of simulation is randomly generated. The number of agents and targets in unchanged. The distance between the platforms (or agents) and the targets is 40 km, targets are randomly generated in the range of 5-145 km (x-axis) and 50-70 km (y-axis), platforms are randomly generated in the range of 15-135 km (x-axis) and 0-10 km (y-axis). One possible situation is as shown in Fig. 5.

Based on the situation of Fig. 5, the results of four algorithms are shown in Table 2 and the average iterative processes of four algorithms are shown in Fig. 6.

As shown in Table 2 and Fig. 6, the experiment results show that although our algorithm cannot converge to the global optimum by 100%, it has the best stability and can converge to the global optimum faster by comparing with the other three algorithms.

Furthermore, to verify the scalability of our algorithm, we investigate the performance of the algorithms in a situation with more agents and targets. The situation is shown in Fig. 7. There are 8 platforms and 26 targets, every platform carries 4 agents. The distance between the platforms and targets is 30 km, both sides are placed regularly. The average iterative processes of the algorithms are shown in Fig. 8.

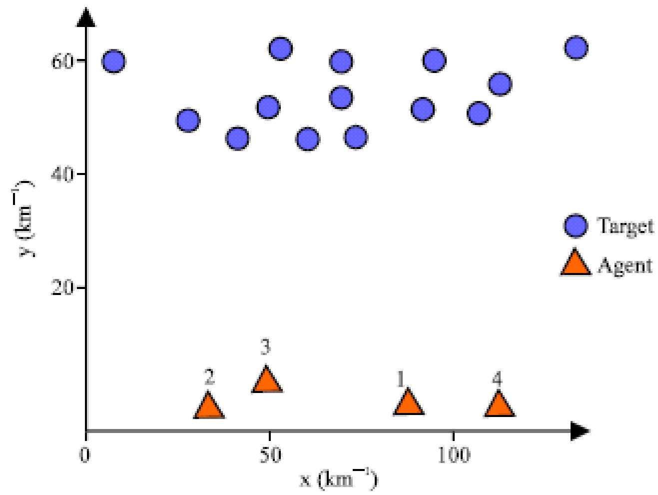

Fig. 5: The situation between randomly generated agents and targets

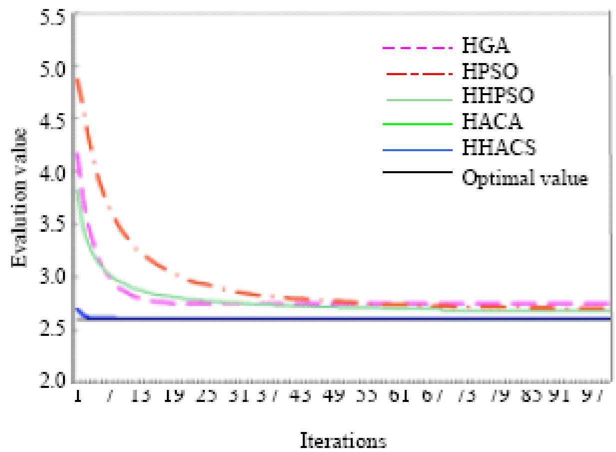

Fig. 6: Iterative processes of the algorithms based on situation of Fig. 5

Table 2: Comparison of simulation results based on scenario of Fig. 5

Algorithm	N	N_m	N_a	V_b	V_a	T_a $msec^{-1}$
HGA	13	13	17.15	2.6040123	2.7669973	145.16
HPSO	17	32	63.18	2.6040123	2.6983840	75.22
HACA	22	14	54.82	2.6093547	2.6393752	192.30
HHACS	72	1	19.25	2.6040123	2.6053343	14.62

Where, new added HHPSO is an improved heuristic PSO algorithm which using the same improvement method proposed by this study. The experiment results show that our algorithm is also the best by comparing the other four algorithms and the HHPSO is also achieving a good performance.

Figure 9 and 10 give the situation of randomly generated agents and targets and average iterative processes based on Fig. 9, respectively. The distance between the platforms (or agents) and the targets is 30 km, targets are randomly generated in the range of 5-145 km (x-axis) and 80-130 km (y-axis), platforms are randomly generated in the range of 10-140 km (x-axis) and 20-50 km (y-axis).

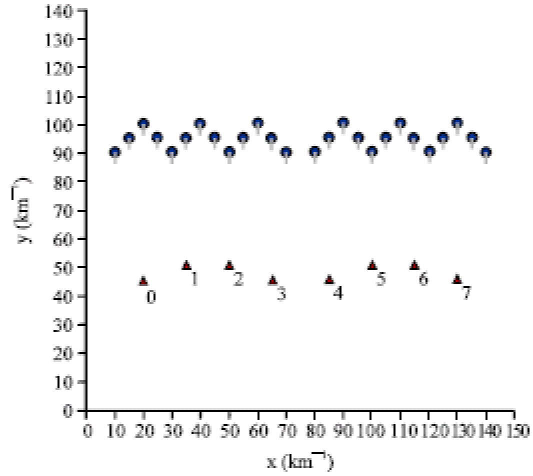

Fig. 7: The situation between agents and targets (8 platforms: 26 targets)

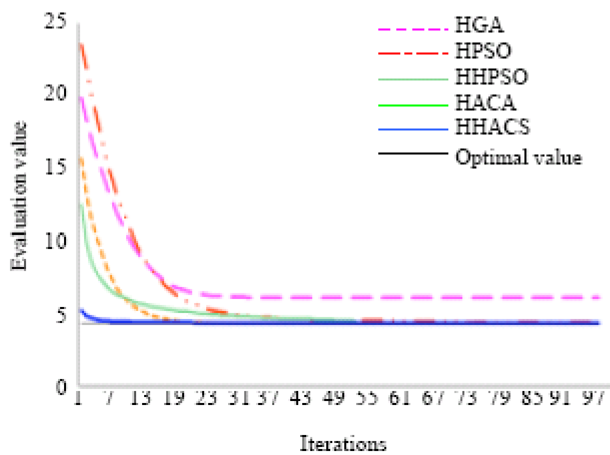

Fig. 8: Iterative processes of the algorithms based on situation of Fig. 7

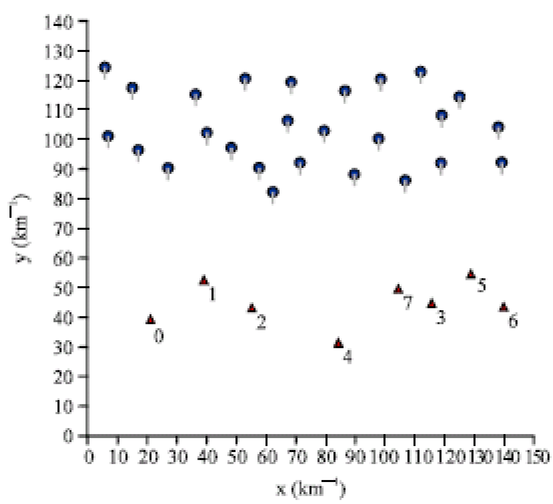

Fig. 9: The situation between randomly generated agents and targets (8 platforms: 26 targets)

And the experiments which platforms and targets are complete randomly generated in the range of 0-150 km (x-axis) and 0-140 km (y-axis) were done. Figure 11 and 12 give the situation and the results, respectively.

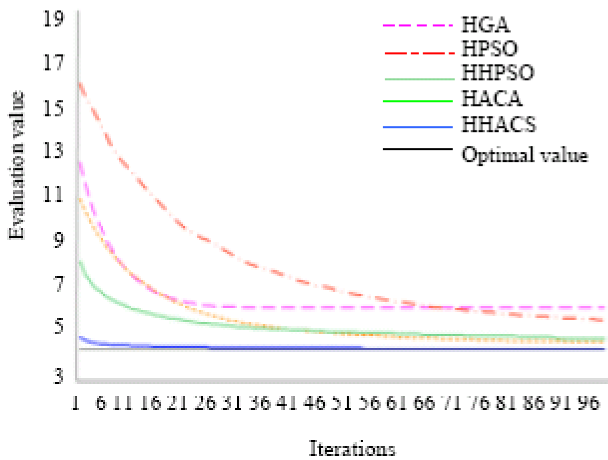

Fig. 10: Iterative processes of the algorithms based on situation of Fig. 9

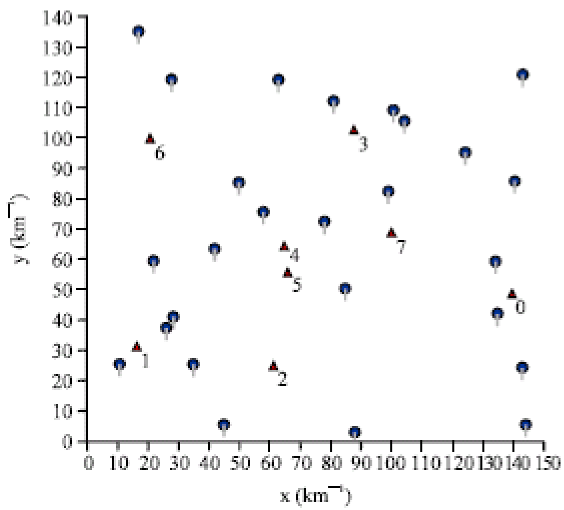

Fig. 11: The situation between completely randomly generated agents and targets (8 platforms: 26 targets)

From the Fig. 10 and 12, the experiment results show that our algorithm is the only algorithm that can converge to the global optimum in 100 iterations. And it has the best stability.

Finally, we investigate the performance of the algorithms on a larger scale where there are 15 platforms and 50 targets and every platform carries 4 agents. Figure 13 gives the situation where agents and targets are completely randomly generated in the range of 0-150 km (x-axis) and 0-140 km (y-axis) and Fig. 14 gives the average iterative processes based on Fig. 13.

From the Fig. 14, the experiment results show that for the large scale assignment of agent-to-target, all algorithms cannot converge to the global optimum in a limited number of iteration. But this algorithm also has the best performance by comparing the other four algorithms. It can reach the global optimum of 98% in 100 iterations and have the best stability.

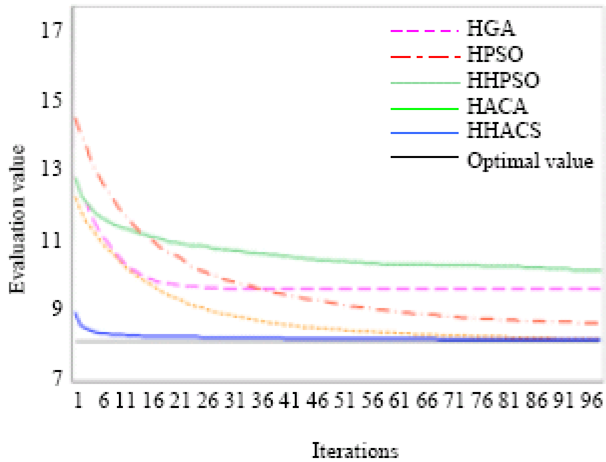

Fig. 12: Iterative processes of the algorithms based on situation of Fig. 10

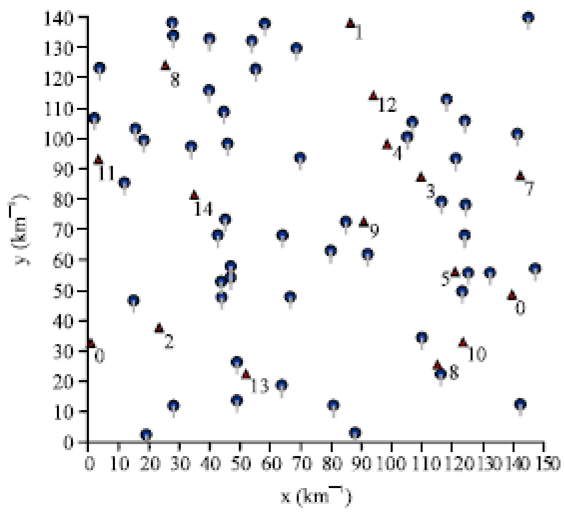

Fig. 13: The situation between completely randomly generated agents and targets (15 platforms: 50 targets)

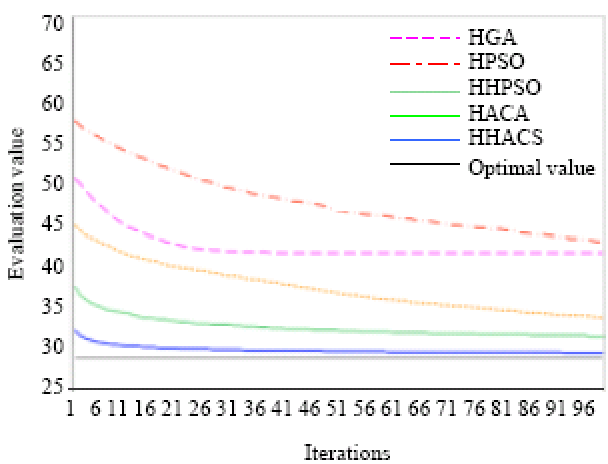

Fig. 14: Iterative processes of the algorithms based on situation of Fig. 13

All results show that present algorithm is an effective algorithm; it can converge to the global optimum more stably and faster, while the scalability of the algorithm has been improved by comparing the other four algorithms.

CONCLUSIONS

One of most important aspects in the design of coordinated multi-agent systems is the assignment of tasks among the agents in an effective manner. To address it, this study presents a new hybrid algorithm that combines heuristic ant colony system and decentralized cooperative auction. The proposed algorithm reduced the numbers of blind search by using decentralized cooperative auction to generate original ants` solutions and then the original solutions are improved to increase the system stability by a new heuristic approach in which the effect of coordinated work of the agents is considered. To compare the original methods, present methods can converge to the global optimum more stable and faster and have the better scalability.

ACKNOWLEDGMENTS

This research was supported by the national Natural Science Foundation of China under grant No. 60673024 and also supported by the 11th-Five-Year Preliminary Research Project of China under grant No. 282660008 and No. 570851613.