SCIENCE CHINA
Technological Sciences

Special Topic: Planning and Control for Specialized Robotics
•Article•

May 2023 Vol.66 No.5: 1235–1244
https://doi.org/10.1007/s11431-022-2302-6

# An adaptive distributed auction algorithm and its application to multi-AUV task assignment

WANG Yu[1], LI HuiPing[1*] & YAO Yao[2]

[1] School of Marine Science and Technology, Northwestern Polytechnical University, Xi'an 710072, China;
[2] Jiangsu Automation Research Institute, Lianyungang 222061, China

Received October 8, 2022; accepted December 28, 2022; published online April 18, 2023

The task assignment of multi-agent system has attracted considerable attention; however, the contradiction between computational complexity and assigning performance remains to be resolved. In this paper, a novel consensus-based adaptive optimization auction (CAOA) algorithm is proposed to greatly reduce the computation load while attaining enhanced system payoff. A new optimization scheme is designed to optimize the critical control parameter in the price update role of auction algorithm which can reduce the searching complexity in obtaining a better bidding price. With this new scheme, the CAOA algorithm is designed. Then the developed algorithm is applied to the multi-AUV task assignment problem for underwater detection mission in complex environments. The simulation and comparison studies verify the effectiveness and advantage of the CAOA algorithm.

**task assignment, multi-agent systems, consensus-based adaptive optimization auction, intelligent algorithm, multiple AUVs**

**Citation:** Wang Y, Li H P, Yao Y. An adaptive distributed auction algorithm and its application to multi-AUV task assignment. Sci China Tech Sci, 2023, 66: 1235–1244, https://doi.org/10.1007/s11431-022-2302-6

## 1 Introduction

Robots and/or agent systems find more and more important applications in civilian and defense areas [1]. The task assignment which dominates the system performance of such systems is essential to the development of intelligent robotic systems. The basic task assignment problem also known as the linear assignment problem can be described as follows. Given a set of agents and tasks, with each agent obtaining some payoff (or incurring some cost) for each task, find a one-to-one assignment of agents to tasks so that the overall payoff of all the agents is maximized [2].

Recently, the development and implementation of multiple robotic systems such as multiple mobile robots, unmanned aerial vehicles (UAV) [3,4], and autonomous underwater vehicles (AUVs) become a technology trend, and the multi-agent systems (MASs) task assignment has attracted growing attention [5,6]. The existing MASs task assignment algorithms mainly include the centralized algorithms and distributed ones with a shared memory [7]. The centralized task assignment methods can solve the problem globally, but have demanding communication bandwidth requirements [8,9]. Compared with the centralized one, the distributed algorithms are less dependent on the communication center, which are more robust for the overall system [10]. Meanwhile, the distributed algorithm can be executed via parallel computation in real time and is suitable for highly dynamic environments [11]. In the literature, the most typical distributed task assignment methods mainly include the behavioral incentive-based method [12], the vacancy chains method [13,14], the swarm intelligence-based method [5,7] and the market-based method.

*Corresponding author (email: lihuiping@nwpu.edu.cn)

The market-based method has been widely utilized due to its outstanding advantages. In particular, this type of method has the following features: (1) It can maintain the balance between task assignment and resource consumption, i.e., it not only requires each individual to bid for the task which benefits it most, but also needs to fully consider the current consumption of the agent. (2) It can accommodate the agent heterogeneity, which can be adopted for most application scenarios. (3) It is very efficient in solution searching due to the fact that each agent only needs to consider the maximization of its own payoffs, and it is not necessary to enumerate all assignment situations to find the optimal solution.

Nowadays, the auction algorithm is one of the most popular assignment algorithms based on market-based method [15]. Because of its scalability, the auction algorithm is suitable for distributed robot systems, and it has been proven that the bidders can obtain the optimal assignment theoretically [16]. Nevertheless, the communication cost of this algorithm is rather heavy, which requires multiple cycles for the optimal solution to converge. A resource-oriented distributed auction algorithm is proposed in ref. [17] to solve the problem of multi-robot task assignment, where each robot generates the bid price in a probabilistic way based on the remaining resources of the tasks, and it assigns the task to the robot who takes the least time to complete the task. Qiao et al. [18] solve dynamic task assignment problem of multi-UAV with multiple constraints, where all the system constraints are fulfilled when calculating the bidding cost.

Nevertheless, the above studies only focus on whether agents are fully assigned, and the update step size of the price update rule is fixed, which fails to take into consideration the balance between the computational complexity and the system total payoff in the assignment process. In fact, the update step size plays a critical role. If it is minor, the computational complexity will increase exponentially, otherwise, the system payoff will be poor, and the optimal assignment cannot be obtained. Therefore, how to design a reasonable price update rule is a key issue to be solved.

Motivated by this fact, in this paper, the consensus-based adaptive optimization auction (CAOA) algorithm is proposed, which designs a reasonable price update rule based on the particle swarm optimization (PSO) algorithm. Furthermore, the CAOA algorithm is applied to the multi-AUV task assignment, which improves the accuracy and optimization efficiency of task assignment in complex ocean environment. The major contributions are summarized as follows.

(1) A novel adaptive distributed auction algorithm is proposed. Firstly, an improved PSO algorithm is proposed, which prevents the optimization algorithm from falling into the dilemma of local optimum. Then, the PSO-based optimization scheme is proposed to adaptively tune the step size of the price update rule in the distributed auction algorithm. This results in a new auction algorithm called CAOA which can reduce computation complexity while attaining better payoff.

(2) The CAOA algorithm is applied to multi-AUV navigation positions assignment in complex and changeable underwater environment with high communication cost. In particular, the communication between AUVs and motion energy consumption is considered and integrated into the cost of fitness function, and the optimal assignment scheme is obtained via the CACO algorithm. The comparison studies verify the effectiveness and advantage of the proposed task assignment algorithm.

## 2 Preliminaries

### 2.1 Problem description

Let us first introduce the multi-agent task assignment problem considered in this paper and underlying assumptions.

#### 2.1.1 *Problem formulation*

Suppose that there exist $M$ agents, $U=\{u_1, u_2, \ldots, u_M\}$, which need to be assigned to $M$ different tasks, which is $T=\{t_1, t_2, \ldots, t_M\}$. The overall objective is to assign all tasks to the agents and the system payoff is maximized, where the system payoff refers to the sum of individual payoffs in agent set. The multi-agent task assignment problem can formally be stated as follows:

$$\varphi^* = \max(\varphi) = \max\sum_{i=1}^{M}\left(\sum_{j=1}^{M}(c_{ij}-p_{ij})x_{ij}\right), \quad \forall u_i \in U, \forall t_j \in T, \tag{1}$$

$$x_{ij} \in \{0,1\}, \forall u_i \in U, \forall t_j \in T, \tag{2}$$

$$\sum_{i=1}^{M} x_{ij} = 1, \ \forall u_i \in U, \tag{3}$$

$$\sum_{j=1}^{M} x_{ij} = 1, \ \forall t_j \in T, \tag{4}$$

where the system payoff $\varphi$ can be considered as the difference between assignment benefit $c_{ij}$ and cost $p_{ij}$. $x$ is a binary decision variable which indicates whether $u_i$ is assigned to $t_j$. Eqs. (3) and (4) state that an agent can only perform one task and each task must be assigned to exactly one agent.

#### 2.1.2 *Communication topology*

The graph $G=(M, L)$ is utilized to characterize the communication network between agents in the distributed system, where $M$ is the topology node set and $L$ is the topology edge set. If there exists a directed communication link between two agents, namely, $(\alpha, \beta) \subseteq L$, then these two agents $\alpha$ and $\beta$ are adjacent. The number of agents in the network is $M$ and the adjacent matrix $A$ of $G$ can be determined as

$$a_{\alpha,\beta}=\begin{cases}1,\ (\alpha,\beta)\subseteq L,\\ 0,\ (\alpha,\beta)\nsubseteq L.\end{cases} \tag{5}$$

$a_{\alpha,\beta}$ is the element of the matrix $A$, which is a binary decision variable indicating whether there exists a communication link between the two nodes $\alpha$ and $\beta$. In order to execute the consensus algorithm, it is assumed that the communication graph is undirected, and it is connected.

### 2.2 Auction algorithm

In the auction algorithm, it is supposed that agent $u_i$ pays $c_{ij}$ for task $t_j$ uppermost, which means that agent $u_i$ will have no gains if the price of task $t_j$ is higher than $c_{ij}$ and it does not bid for it. If agent $u_i$ pays $p_j$ for task $t_j$, the payoff is $c_{ij}-p_j$. Each agent pursues the maximum payoff, thus the payoff function is established as follows:

$$\pi_i=\max_{t_j\in T}(c_{ij}-p_j), \tag{6}$$

where $T$ is the set of tasks. If eq. (6) is satisfied for all agents, we say that the assignment and the set of prices are at equilibrium. At this time, such an equilibrium assignment provides the maximum system payoff for the whole, that is, the overall optimum is achieved. The system payoff $\varphi$ can be defined as

$$\varphi=\sum_{i=1}^{M}\pi_i. \tag{7}$$

Otherwise, proceed to the next round of auction based on the price update rule. Firstly, the optimal payoff for agent $u_i$ this round is calculated according to eq. (8):

$$\pi_i=g_{ij^*}=\max_{t_j\in T}(c_{ij}-p_j),t_{j^*}\in T, \tag{8}$$

where $\pi_i$ the optimal payoff of agent $u_i$ and $j^*$ is the task number corresponding to the optimal payoff. Then, calculate the suboptimal payoff $g_{ij_1^*}$ for agent $u_i$:

$$g_{ij_1^*}=\max(c_{ij}-p_j),\ t_j\in T,\ t_j\neq t_{j^*}, \tag{9}$$

where $j_1^*$ is the task number corresponding to the suboptimal payoff. Finally, the next round offer can be obtained according to eq. (10).

$$p_{ij^*}=b_{j^*}+g_{ij^*}-g_{ij_1^*}+\varepsilon, \tag{10}$$

where $p_{ij^*}$ is the next round offer of agent $u_i$ to task $t_{j^*}$, $b_{j^*}$ is the highest offer for task $t_{j^*}$ in the previous round of auction, and $\varepsilon$ is the minimum incremental constant which represents the increment of each auction, i.e., the update step size.

In the consensus-based auction algorithm, each agent is unable to obtain the global price and assignment result of each task accurately, and can only update the auction status through local information, which is different from the classical auction algorithm. The algorithm consists of iterations between two phases: the auction and the consensus phase. In the auction phase, each agent judges whether the task needs to be assigned firstly, and if so, calculates the payoff of the task and further bids for the task with the highest payoff [19]. The price update rule here is the same as auction algorithm. The consensus phase is a consensus algorithm that is used to make the bidding process to converge on a winning bids list. By iterating between the two steps, the distributed auction algorithm is executed.

### 2.3 Problem analysis

In the consensus-based auction algorithm, each agent needs to offer prices for tasks in the task set separately, and bids on the highest payoff item by increasing the task price. Therefore, the price update rule is the core of the auction algorithm. Each agent identifies the task that benefits itself most by solving the bidding payoff equation, and further increases the price of the task, where the minimum incremental constant represents the increment of each auction, i.e., the update step size.

There are several points to be noticed here for the auction algorithm. Firstly, different values of the update step size will produce different results, namely, the multi-agent system might obtain different payoffs and assignment schemes. Secondly, the update step size plays a crucial role in the assignment, which can prevent the assignment optimization from falling into an infinite loop. Finally, it is also the key factor affecting the computational complexity. That is, if the update step size is not set properly, the computation load will increase exponentially, which stops the optimal task assignment to be obtained. Hence, a reasonable update step size, i.e., the price update rule is of great significance to multi-agent task assignment. In this paper, we develop a new scheme to optimize the update step size adaptively to resolve this issue.

## 3 Methodology

This section presents the adaptive PSO and CAOA algorithms which optimize the update step size of the price update rule to bring a better balance between the system payoff and computation consumption.

### 3.1 Algorithm framework

The framework of CAOA algorithm mainly consists of two parts, including the distributed task assignment and the price update rule optimization, which is shown in Figure 1. In the task assignment phase, the distributed auction algorithm is adopted [20]. That is, each agent bids for the task in the task set separately, chooses the task which benefits it most and further increases the price according to the price update rule. When all agents obtain tasks which benefit them most, this

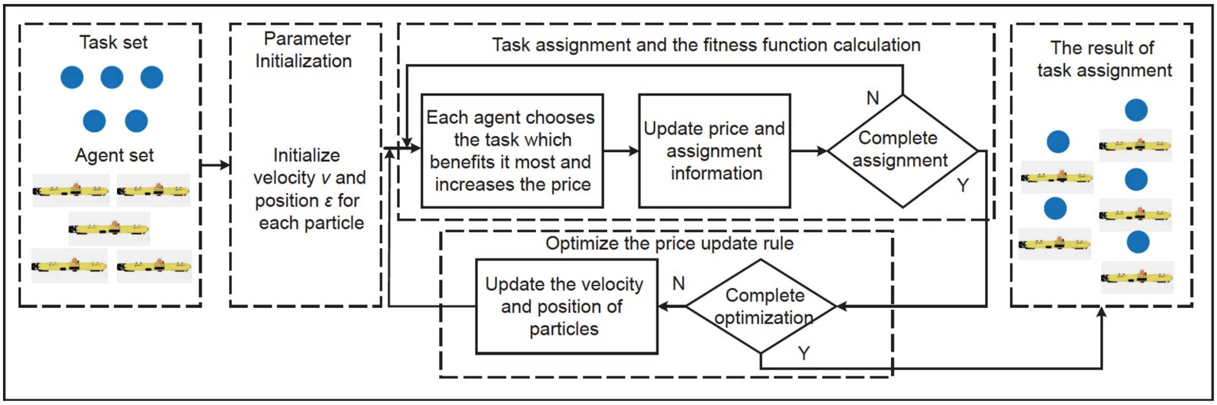


**Figure 1** (Color online) Architecture of the proposed adaptive distributed auction algorithm.

round of task assignment ends, and the system payoff (i.e., fitness function) is calculated. In the price update rule optimization phase, an adaptive PSO algorithm is proposed to optimize the price update rule according to the resulting fitness function, and then the next round of task assignment is carried out. These two phases iterate until a criterion is met.

## 3.2 Design of adaptive PSO algorithm

### *3.2.1 PSO algorithm*

The key idea of PSO algorithm comes from the simulation of the social behavior of individuals belonging to a swarm. Each member of the swarm is called a particle, and it corresponds to one potential solution to the optimization problem. The particles share their individual information to decide the further movement. The velocity and position vectors are associated with each swarm particle. The fitness value for each particle is determined using an objective function $\varphi(\varepsilon_\zeta)$, which decides the position and velocity of each agent in each iteration. The particle will find the optimal solution after a certain number of iterations. The dynamics of the particles are written as

$$v_\zeta(k+1) = wv_\zeta(k) + c_1r_1\big(p_\zeta(k) - \varepsilon_\zeta(k)\big) + c_2r_2\big(p_g(k) - \varepsilon_\zeta(k)\big), \tag{11}$$

$$\varepsilon_\zeta(k+1) = \varepsilon_\zeta(k) + v_\zeta(k+1), \tag{12}$$

where $v_\zeta(k)$ and $\varepsilon_\zeta(k)$ represents the velocity and position of the $\zeta$th particle at time $k$. $w$ is the inertia weight, $c_1$ is the cognition component, and $c_1$ is the social component. $r_1$ and $r_2$ stand for weight values. $P_\zeta(k)$ and $p_g(k)$ is the personal optimal position of $\zeta$th particle and the global best. Moreover, the initial condition $\varepsilon_\zeta(0)$ is set at random values and $\varepsilon_\zeta(k)$ corresponds to a solution for the optimization problem.

### *3.2.2 Adaptive PSO algorithm*

It is worth noting that classical versions of the PSO lacked inertia components, which makes the local search feature less useful. In this paper, an adaptive PSO algorithm is proposed, which changes values of key parameters through adaptive update strategy to improve the searching ability of the algorithm in different evolutionary periods. The adaptive update strategy is presented as follows.

Inertia weight factor $w^*$: it determines the influence of the speed of the predecessor particle on the speed update of the contemporary particle, a reasonable value of which can improve the global convergence speed. A larger $w^*$ is beneficial to improve the global convergence speed, which makes the algorithm have a strong global search ability during initial stage. A smaller $w^*$ is good for improving the local search ability, accelerating the speed of the algorithm to converge to the optimal value. Nevertheless, $w^*$ always takes a fixed value in the conventional PSO algorithm, which causes the switching lack smoothness between the two search modes and is unable to balance the process of global exploration and local development during particle search iteration. Therefore, the adaptive inertia weight factor is designed as

$$w^* = w_o - w_v, \tag{13}$$

$$w_o = w_{\max} - \frac{w_{\max} - w_{\min}}{k_{\max}} \times k, \tag{14}$$

$$w_v = \frac{\varphi(p_g(k-1))}{\varphi(p_g(k))}, \tag{15}$$

where $w^*$ consists of two parts $w_o$ and $w_v$. The term $w_o$ indicates the reference value of linear variation, which ensures the overall trend of $w$ is gradually decreasing. $w_{\max}$ and $w_{\max}$ represent the upper and lower bound of $w$. On this basis, we define the evolution rate index $w_v$ considering the task assignment problem described in sect. 2.1, where $\varphi(p_g(k))\geq \varphi(p_g(k-1))$, namely, $0<w_v<1$. Obviously, the smaller $w_v$ is, the faster the particle swarm evolves, and the algorithm should keep a large range of global optimization. On the contrary, it should be more inclined to a small area of local search to find the optimal solution faster. Summing up the above, $w^*$ designed in this paper fluctuates around the reference value according to the evolution rate of the particle swarm, so that the convergence of the algorithm can be controlled flexibly.

Acceleration constant $c_1^*$ and $c_2^*$: The values of $c_1^*$ and $c_2^*$ are modified using a linear function that changes with the number of iterations $k$.

$$c_1^* = c_{1\max} - \frac{c_{1\max} - c_{1\min}}{k_{\max}} \times k, \tag{16}$$

$$c_2^* = c_{2\min} + \frac{c_{2\max} - c_{2\min}}{k_{\max}} \times k. \tag{17}$$

The dynamics of the particles are designed as

$$\begin{aligned} v_\zeta^*(k+1) = {} & w^* v_\zeta(k) + c_1^* r_1\big(p_\zeta(k) - \varepsilon_\zeta(k)\big) \\ & + c_2^* r_2\big(p_g(k) - \varepsilon_\zeta(k)\big), \end{aligned} \tag{18}$$

$$\varepsilon_\zeta^*(k+1) = \varepsilon_\zeta(k) + v_\zeta^*(k+1). \tag{19}$$

Algorithm 1 describes the pseudocode for implementing the adaptive PSO algorithm.

### 3.3 Design of CAOA algorithm

With the improved PSO algorithm shown in Algorithm 1 to optimize the price update rule in the consensus-based distributed auction procedure, the detailed CAOA algorithm is presented in Algorithm 2, which mainly includes the following steps and Table 1 is the symbol description.

Step 1 (Parameters initialization): Combining with the characteristics of adaptive PSO algorithm, the required parameter, i.e., the update step size is represented by the position $\varepsilon_\zeta$ of each particle, and the dimension is equal to the number of required parameters. The key parameters of the adaptive PSO algorithm have been determined in Sect. 3.2. According to the payoff and energy consumption models, the bidding prices, the agent group participating in the task assignment, the task group and the task group that each agent can bid for will be initialized. In this paper, all tasks can be bidden and the termination condition of auction is the optimal system payoff. This step is shown in Algorithm 2 (Line 1).

**Table 1** Symbol description of CAOA algorithm

| Symbol | Description |
|---|---|
| $\zeta$ | Number of particles |
| $d$ | Dimensions of the particle motion |
| $c_1$, $c_2$ | Acceleration constants |
| $k$ | Particles velocities and positions are updated in the $k$ iteration |
| $\delta$ | Agent $\delta$ is the neighbor of agent $i$ |
| $x_i$ | The task list of agent $i$ |
| $y_i$ | The winning bids list of agent $i$ |
| $\mathbb{I}(\cdot)$ | $\mathbb{I}(\cdot)$ is the indicator function that is unity if the argument is true and zero otherwise |
| $G$ | The multi-agent distributed communication topology |

Step 2 (Calculate the individual fitness value): It refers to the value of $\varphi(\varepsilon)$, when $\varepsilon=\varepsilon_\zeta$. In this paper, $\varepsilon_\zeta$ is the update step size, and $\varphi(\varepsilon_\zeta)$ is the system payoff after completing a round of task assignment, which is the sum of the payoff obtained by each agent shown in eqs. (2) and (3). The computational process consists of iterations between the auction phase (Lines 5–15) and the consensus phase (Lines 16–22) in Algorithm 2.

Step 3 (Update the individual and global optimal fitness value): Compared the current fitness value of each particle with its experienced optimal one, if it is better, update it as the optimal fitness value; otherwise, keep the current state. The specific calculation process is shown in Algorithm 2 (Lines 24–30).

Step 4 (Update the position and velocity): Each particle updates coordinate $\varepsilon_\zeta^*$ and velocity $v_\zeta^*$ of the next iteration based on evolution eqs. (18) and (19), which implies the motion tendency of the particle in this step (Line 31).

Step 5 (Inspection termination condition): The iteration termination condition is usually set to the preset accuracy or maximum iteration. In the context of the multi-agent task assignment, we set maximum iteration as the iteration termination condition. When the termination condition is reached, the iteration ends and the value of $\varphi(\varepsilon^*)$ and $\varepsilon^*$ are obtained; otherwise it returns to the Step 3 until assignment ends.

## 4 Application to multi-AUV task assignment

In this section, the designed CAOA algorithm is applied to the multi-AUV task assignment under complex underwater environment. Firstly, the problem of multi-AUV task assignment is presented. Then the experimental settings are

Algorithm 1 Adaptive PSO algorithm

```
for each particle ζ do
    initialize velocity v_ζ and position ε_ζ for particles
    evaluate particle ζ, ie, calculate φ(ε_ζ) and set.
    the best previous personal position p_ζ = ε_ζ.
end for
initialize the best system personal position φ* = φ(ε*), φ(ε*)
  = max[φ(ε_ζ)], ζ = 1, 2, ..., m, p_g = ε*
while not stop do
    for ζ = 1 to m do
      update the position and velocity for each particle
      ζ according to eqs. (18) and (19)
      evaluate particle ζ
      if φ(ε_ζ) > φ(p_ζ) then
        p_ζ = ε_ζ
      end if
      if φ(p_ζ) > φ(p_g) then
        p_g = p_ζ
      end if
    end for
end while
```

introduced. Finally, two groups of experiments have been conducted to verify effectiveness of the CAOA algorithm.

### 4.1 Multi-AUV task assignment description

We consider the scenario that there are $M$ AUVs, which need to be assigned to $M$ different points to complete the detection mission. The overall objective is to assign all tasks to AUVs and the system payoff is maximized, where the system payoff is the sum of the local payoff of all AUVs. The multi-AUV benefit and energy consumption models for underwater detection mission can be stated as follows.

The benefit from AUVs task completion mainly includes remaining energy and working efficiency, which can be calculated as follows:

$$c_{ij}^{*} = e_{ij} + v_i, \tag{20}$$

where $c_{ij}$ represents the benefit of AUV $u_i$ after completing task $t_j$, $e_{ij}$ is the remaining energy, which is closely related to the cruising range of $u_i$, and $v_i$ denotes the velocity of $u_i$.

Energy consumption of AUVs is mainly composed of communication between AUVs and their movement. On one hand, AUVs communicate mainly by means of sound waves, which consumes much energy and is affected by transmission frequency, transmission speed, path and transmission wastage. By considering all these factors, the acoustic communication energy consumption model can be established as

$$p_{ij}^{1} = nP\xi^{d}d^{k}, \tag{21}$$

where $p_{ij}$ is the energy consumed by one AUV to transmit $n$ bits of data at distance $d$, $P$ is the energy required to transmit 1bit of data, and $k$ is the communication consumption index, which can choose a value between 1 and 3, usually 1.5. $\xi$ is the absorption coefficient in dB/km.

The value of $\xi$ can be calculated:

$$\xi = 10^{\alpha(f)/10}, \tag{22}$$

$$\alpha(f) = 0.003 + \frac{0.11f^{2}}{1+f^{2}} + 2.75\times10^{-4}f^{2} + \frac{44f^{2}}{4100}, \tag{23}$$

where $f$ denotes the frequency of signal transmission.

On the other hand, AUVs are subjected to different degrees of resistance when the state of motion changes underwater, such as acceleration, steering and uniform motion. Since the study focuses on the strategy of task assignment, the motion of AUVs is simplified to uniform motion. According to the knowledge of fluid mechanics, the fluid resistance experienced by AUVs can be defined as

$$F = 0.5sC\rho v^{2}, \tag{24}$$

where $s$ is the cross-sectional area of the AUV, $C$ is the hydrodynamic coefficient, the value of which is affected by media characteristics, the shape of the AUV, the area of flow attack and so on. There is no exact value of $C$, which is equal

Algorithm 2 CAOA algorithm

**Require:** the agent set $U = \{u_1, u_2, ..., u_M\}$, the task set $T = \{t_1, t_2, ..., t_N\}$, the initial price $p_{ij}$
**Ensure:** the update step size $\varepsilon_\zeta$, the optimal payof $\varphi^*$, and the assignment result $x_{ij}$

initialize $v_\zeta$, $\varepsilon_\zeta$, $w$, $c_1$, $c_2$, $r_1$ and $r_2$ for particles
**while** not stop **do**
  **for** each particle $\zeta$ **do**
    **while** assignment is not ends **do**
      $x_i(t) = x_i(t-1)$
      $y_i(t) = y_i(t-1)$
      **if** $\Sigma_j x_{ij}(t) = 0$ **then**
        $h_{ij} = \mathbb{I}(c_{ij} > y_{ij}(t)), \forall j\in T$
        **if** $h_j \neq 0$ **then**
          $J_i = \mathrm{argmax}_j\ h_{ij}\cdot c_{ij}$
          $x_i, J_i(t) = 1$
          $y_i, J_i(t) = c_i, J_i$
          update the task price according to eq. (10)
        **end if**
      **end if**
      send $y_i$ to $\delta$ with $G_{i\delta}(\tau) = 1$
      receive $y_\delta$ from $\delta$ with $G_{i\delta}(\tau) = 1$
      $y_{ij}(t) = \max_\delta G_{i\delta}(\tau) y_{\delta j}(t), \forall j\in T$
      $z_i, J_i = \mathrm{argmax}_\delta\ G_{i\delta}(\tau)\cdot y_\delta, J_i(t)$
      **if** $z_i, J_i \neq i$ **then**
        $x_i, J_i(t) = 0$
      **end if**
    **end while**
    calculate $\varphi(\varepsilon_\zeta)$ according to eq. (7)
    **if** $\varphi(\varepsilon_\zeta) > \varphi(p_\zeta)$ **then**
      $p_\zeta = \varepsilon_\zeta$
    **end if**
    **if** $\varphi(p_\zeta) > \varphi(p_g)$ **then**
      $p_g = p_\zeta$
    **end if**
    update the position and velocity for each particle $\zeta$ according to eqs. (18) and (19)
  **end for**
**end while**

to 0.7 generally according to experience. $\rho$ is the density of the water medium, the value of which is equal to the average density of seawater generally, namely $\rho$=1.025 g/cm$^3$. $v$ is the speed of the AUV. Ignoring the component heating and power consumption other than thrusters, the AUV energy consumption is mainly caused by work done against the resistance of the water, which can be calculated by the following equation.

$$p_{ij}^{2} = F\cdot v\cdot t_{ij} = 0.5sC\rho v^{3}t_{ij}, \tag{25}$$

where $t_{ij}$ refers to the time required for the AUV $u_i$ to complete task $t_j$. The total energy consumption of the AUV system is

$$p_{ij} = p_{ij}^{1} + p_{ij}^{2}. \tag{26}$$

### 4.2 Experimental settings

#### *4.2.1 Experimental parameters*

The significant parameters of the proposed adaptive PSO algorithm is determined as follows: $w_{\max}\in\mathbb{R}^{+}$ and $w_{\min}\in\mathbb{R}^{+}$ are the upper and lower bounds for $w$, respectively, where

$w_{max}$=1.2 and $w_{min}$=0.9. $m$ is the number of particles. A large $m$ will lead to excellent search capability, but it will increase the computation time exponentially. Therefore, the algorithm has a swarm of $m$=20 particles with a maximum number of iterations $k_{max}$=200 according to the actual situation in this paper. The acceleration parameters $c_1$ and $c_2$ are the cognitive coefficient of the individual particles and the social coefficient of all particles, respectively. The range of them is between 0 and 2 in normal cases.

*4.2.2 Experimental installation*

The configuration of the computer used in the laboratory is as follows. The processor is Intel core CPU i5-8400 at 2.8 GHz and 8 GB RAM onboard, the operating system is Windows 10 (64 bit), and the programming simulation environment is Python 3.6.

*4.2.3 Evaluation metrics*

The AUV system total payoff can be obtained through the improved task assignment algorithm CAOA, and the performance of AUV task assignment is evaluated by the evaluation metrics $\eta$, which is shown as

$$\eta(\varepsilon)=\frac{\varphi^{\mathrm{opt}}-\varphi(\varepsilon)}{\varphi^{\mathrm{opt}}}\times 100\%, \tag{27}$$

where $\varphi^{\mathrm{opt}}$ is the optimal system payoff and $\varphi(\varepsilon)$ is the system payoff when the price update step size is $\varepsilon$.

### 4.3 Simulation studies

To validate the necessity and efficiency of the proposed CAOA algorithm, two aspects of validation have been studied.

*4.3.1 Verification of necessity*

Firstly, the effectiveness of different values of the update step size $\varepsilon$ on the computation time and system total payoff, i.e., different $\varphi(\varepsilon)$ of the distributed auction algorithm for multi-AUV task assignment are tested. Figure 2 shows the variation of $\varphi(\varepsilon)$ as $\varepsilon$ changes, where the number of tasks is 4. It can be seen from Figure 2 that $\varphi(\varepsilon)$ decreases with the increase of $\varepsilon$. But until $\varepsilon$ decreases to a certain extent, $\varphi(\varepsilon)$ no longer changes greatly. To see it more clearly, the change of $\varphi(\varepsilon)$ when $\varepsilon$ between 0 and 0.1 is enlarged in the upper right corner of Figure 2. When $\varepsilon$<0.009, $\varphi(\varepsilon)$ is unchanged essentially. When $\varepsilon$<0.007, if $\varepsilon$ is further reduced, it just increases the computation time, but $\varphi(\varepsilon)$ stays the same, and $\varphi(\varepsilon)$ obtained at this time is basically equal to the optimal one. When $\varepsilon$=1, the error between the system total payoff and optimal one is about 11%.

In order to further verify the impact of the update step size $\varepsilon$ to the AUV task assignment result, we increase the number of tasks and AUVs respectively, which is set to 10, and the relationship between them is shown in Figure 3. Same as

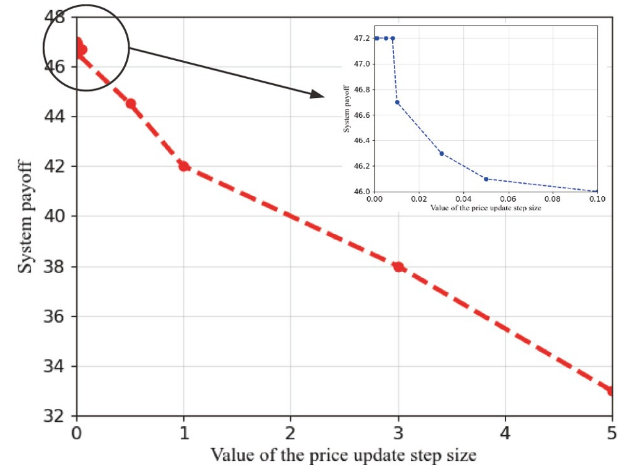


**Figure 2** (Color online) The system payoff under different values of the price update step size ($M$ = 4).

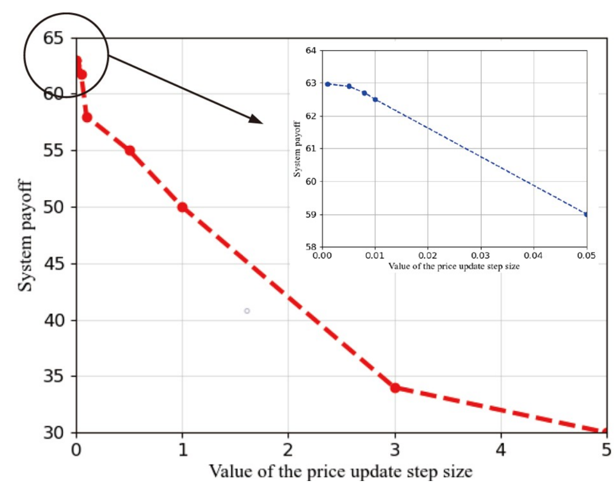


**Figure 3** (Color online) The system payoff under different values of the price update step size ($M$ = 10).

Figure 2, the system total payoff increases with the decrease of $\varepsilon$, and when $\varepsilon$ decreases to a certain extent, the system total payoff is basically unchanged. However, specific attention should be paid to the fact that if $\varepsilon$ decreases continuously, the computation time will increase by about 10 times. Therefore, it is necessary to optimize the update step size $\varepsilon$, which maximizes the system total payoff and saves computation resources. Furthermore, with different number of tasks and AUVs, the optimal update step size is also different. Hence, it is significant to optimize the price update rule.

Tables 2 and 3 show the simulation results of multi-AUV task assignment with different task groups for different $\varepsilon$. When $M$=4 and $\varepsilon$=0.001, the optimal payoff $\varphi(\varepsilon)$ of the AUV system is 47.2. Similarly, when $M$=10 and $\varepsilon$=0.0001, $\varphi(\varepsilon)$ is 62.997.

*4.3.2 Verification of advancement*

After illustrating the necessity of optimizing the update step size for the task assignment algorithm, the validity and su-

**Table 2** Mechanical and thermal properties of the specimens ($M$ = 4)

| $\varepsilon$ | System payoff | Computation time (s) |
|---|---|---|
| 0.0001 | 47.2 | 0.02 |
| 0.001 | 47.2 | 0.009 |
| 0.005 | 47.1 | 0.002 |
| 0.01 | 46.7 | 0.0006 |
| 0.05 | 46.1 | 0.0002 |
| 0.1 | 46.0 | 0.001 |

**Table 3** Mechanical and thermal properties of the specimens ($M$ = 10)

| $\varepsilon$ | System payoff | Computation time (s) |
|---|---|---|
| 0.0001 | 62.997 | 0.04 |
| 0.001 | 62.97 | 0.01 |
| 0.01 | 62.5 | 0.009 |
| 0.1 | 57 | 0.023 |
| 0.5 | 52 | 0.001 |
| 1 | 50 | 0.001 |

periority of the CAOA proposed are verified next. The results are shown in Figures 4–6, the horizontal axis of which represents tasks to be assigned to AUVs, and the vertical axis of which represents the AUVs participating in the task assignment. Numbers in the heat map represent the individual payoff assigned to the corresponding AUV for each task. The parts highlighted in the box are the results of task assignment.

In this section, two groups of multi-AUV task assignment are carried out. Figure 4(a) shows the results of task assignment utilizing CAOA algorithm ($M$ = 4). It can be seen from Figure 4(a) that AUV 1 is assigned to Task 3, AUV 2 is assigned to Task 2, AUV 3 is assigned to Task 4, and AUV 4 is assigned to Task 1, which indicates that all tasks have been assigned and each AUV has obtained the task with the maximal payoff for itself. The AUV system payoff is 47.06, the optimized update step size $\varepsilon$=0.007, and the calculation time is 0.002 s. Figure 4(b) is the result of task assignment using the method of distributed auction algorithm. The value of control parameter $\varepsilon$ is set to 0.001, because the previous verification shows that when $\varepsilon$=0.001, the optimal system payoff can be obtained. What can be seen from Figure 4(b) is that AUV 1, AUV 2, AUV 3 and AUV 4 are assigned to Task 3, Task 2, Task 4 and Task 1, respectively, and the system payoff is 47.2. According to eq. (27), it can be calculated that the error between the result of CAOA and the optimal result is 0.4%. Therefore, the method CAOA proposed in this paper has the effect of more reasonable task assignment and saving calculation time.

The result of another group of multi-AUV task assignment is shown in Figure 5, and the number of tasks is 10. From Figure 5, we see that the results of task assignment obtained by two methods are the same, and each AUV obtains the task which benefits it most. The system total payoff obtained by CAOA algorithm is 62.314, the optimal system payoff is 62.997, and the error between the result of CAOA and the optimal result is 1%. The result of the third group of assignment is shown in Figure 6. As can be seen from it, each AUV has obtained the task that benefits it most and the result is reasonable.

To sum up, compared with the conventional distributed auction algorithm, the CAOA algorithm is more efficient in computation time, and attains better system payoff.

## 5 Conclusions

Aiming at solving the problem existing in auction algorithm, this paper proposes a novel CAOA algorithm, which not only obtains better payoff of the AUV system in the process of

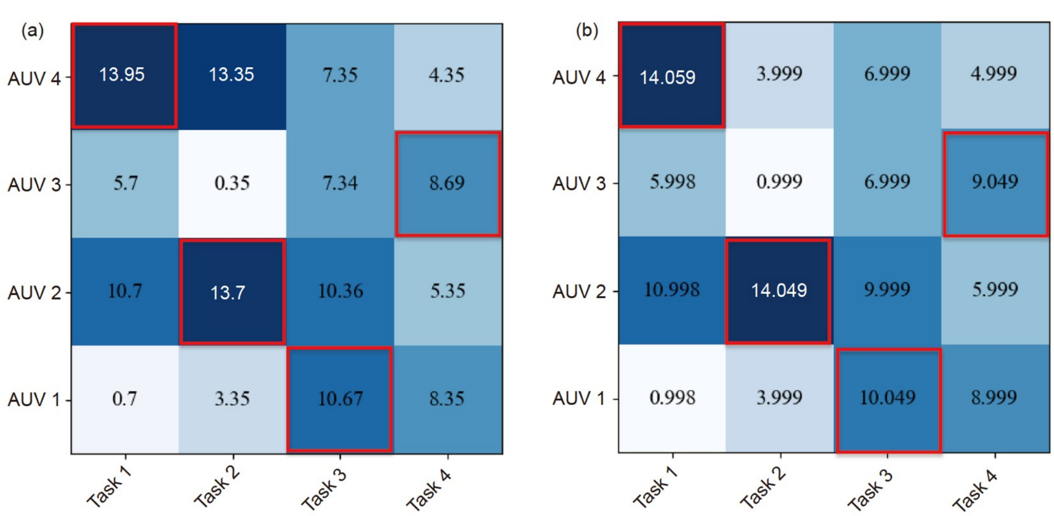


**Figure 4** (Color online) The task assignment scheme using CAOA (a) and auction (b) algorithms ($M$ = 4).

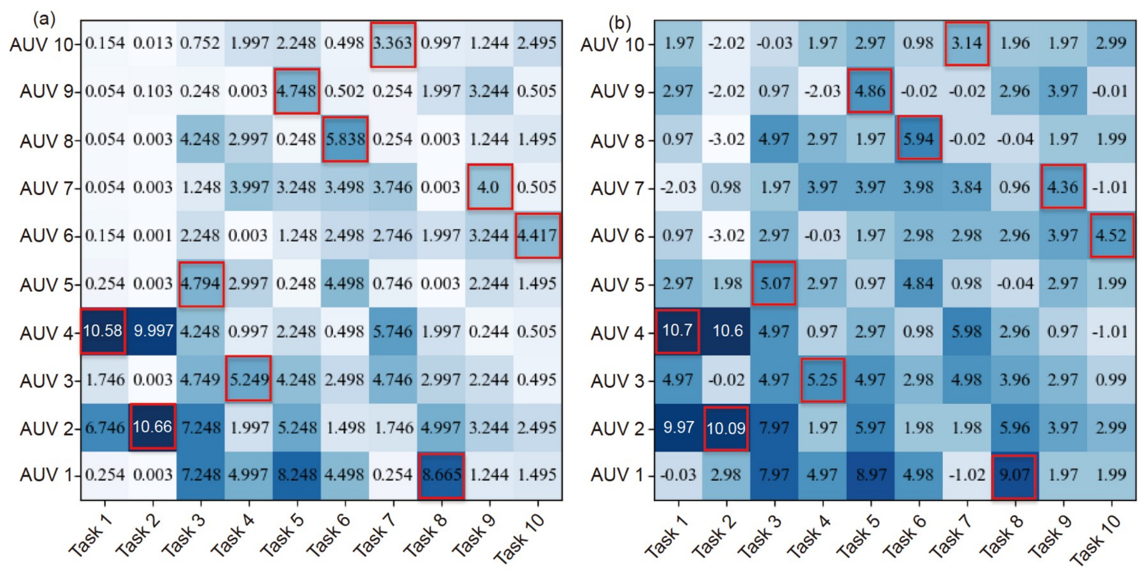


**Figure 5** (Color online) The task assignment scheme using CAOA (a) and auction (b) algorithms ($M$ = 10).

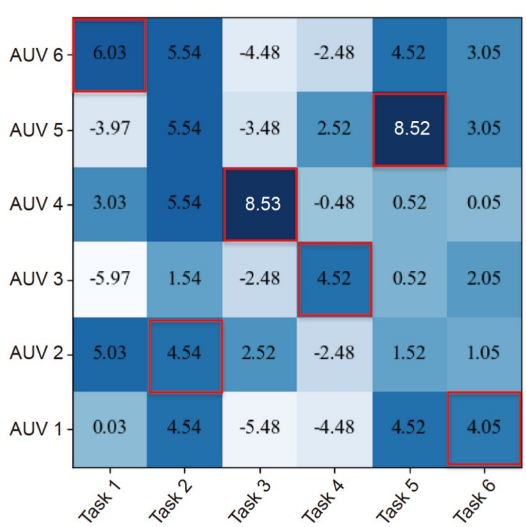


**Figure 6** (Color online) The task assignment scheme using CAOA algorithm ($M$ = 6).

task assignment, but also reduces the computational energy consumption and time in comparison with conventional auction algorithm. The necessity and superiority of the algorithm are verified through two groups of simulation and comparison studies. This paper provides a useful tool for solving the distributed task assignment problem.

For the future studies, some possible improvements of the proposed work might need to be considered: (1) develop multi-task assignment considering temporal effects, (2) redesign the algorithm for dynamic task assignment which react quickly to changes in surrounding environments, and (3) propose the multi-task assignment framework considering underwater communication constraints.

*This work was supported by the National Natural Science Foundation of China (Grant Nos. 62273281, U22B2039, and 61922068).*

1 Zhu D, Zhou B, Yang S X. A novel algorithm of multi-AUVs task assignment and path planning based on biologically inspired neural network map. IEEE Trans Intell Veh, 2020, 6: 333–342
2 Zhu D, Huang H, Yang S X. Dynamic task assignment and path planning of multi-AUV system based on an improved self-organizing map and velocity synthesis method in three-dimensional underwater workspace. IEEE Trans Cybern, 2013, 43: 504–514
3 Shi W, Song S, Wu C, et al. Multi pseudo Q-learning-based deterministic policy gradient for tracking control of autonomous underwater vehicles. IEEE Trans Neural Netw Learn Syst, 2019, 30: 3534–3546
4 Zhu D, Liu Y, Sun B. Task assignment and path planning of a multi-auv system based on a Glasius bio-inspired self-organising map algorithm. J Navigation, 2018, 71: 482–496
5 Qu G, Brown D, Li N. Distributed greedy algorithm for multi-agent task assignment problem with submodular utility functions. Automatica, 2019, 105: 206–215
6 Wang G. Distributed control of higher-order nonlinear multi-agent systems with unknown non-identical control directions under general directed graphs. Automatica, 2019, 110: 108559
7 Yao W, Qi N, Wan N, et al. An iterative strategy for task assignment and path planning of distributed multiple unmanned aerial vehicles. Aerospace Sci Tech, 2019, 86: 455–464
8 Poudel S, Moh S. Task assignment algorithms for unmanned aerial vehicle networks: A comprehensive survey. IEEE Trans Cogn Dev Syst, 2022, 35: 100469
9 Chen X, Zhang P, Du G, et al. A distributed method for dynamic multi-robot task allocation problems with critical time constraints. Robotics Autonomous Syst, 2019, 118: 31–46
10 Liu C H, Ma X, Gao X, et al. Distributed energy-efficient multi-uav navigation for long-term communication coverage by deep reinforcement learning. IEEE Trans Mobile Comput, 2020, 19: 1274–1285
11 Hou K, Yang Y, Yang X, et al. Distributed cooperative search algorithm with task assignment and receding horizon predictive control for multiple unmanned aerial vehicles. IEEE Access, 2021, 9: 6122–6136
12 Gong J, Ma Y, Jiang B, et al. Distributed adaptive fault-tolerant for-

mation control for heterogeneous multiagent systems under switching directed topologies. J Franklin Institute, 2022, 359: 3366–3388
13 Yu Z, Qu Y, Zhang Y. Distributed fault-tolerant cooperative control for multi-uavs under actuator fault and input saturation. IEEE Trans Contr Syst Technol, 2019, 27: 2417–2429
14 Kamel M A, Yu X, Zhang Y. Fault-tolerant cooperative control design of multiple wheeled mobile robots. IEEE Trans Contr Syst Technol, 2018, 26: 756–764
15 Morgan D, Subramanian G P, Chung S J, et al. Swarm assignment and trajectory optimization using variable-swarm, distributed auction assignment and sequential convex programming. Int J Robotics Res, 2016, 35: 1261–1285
16 Bertsekas D P. The auction algorithm: A distributed relaxation method for the assignment problem. Ann Oper Res, 1988, 14: 102–123
17 Lee D H, Zaheer S A, Kim J H. A resource-oriented, decentralized auction algorithm for multirobot task allocation. IEEE Trans Automat Sci Eng, 2015, 12: 1469–1481
18 Qiao C, Dong Y, Jian Y, et al. An auction-based multiple constraints task allocation algorithm for multi-UAV system. In: Proceedings of the 2016 International Conference on Cybernetics, Robotics and Control (CRC). Hong Kong, 2016
19 Choi H L, Brunet L, How J P. Consensus-based decentralized auctions for robust task allocation. IEEE Trans Robot, 2008, 25: 912–926
20 Chen M, Zhu D. A novel cooperative hunting algorithm for inhomogeneous multiple autonomous underwater vehicles. IEEE Access, 2018, 6: 7818–7828