

Figure 1. The typical competition field of the volleyball match and the interaction involved

In recent years, an important trend related to computational intelligence has been to establish a new search mechanism based on sports competitions (Boucekara et al., 2018; Chagwiza et al., 2016; Jaramillo et al., 2016a; Kashan, 2014a; Khattab, Sharieh, and Mahafzah, 2019; Moghdani and Salimifard, 2018; Purnomo and Wee, 2015), through which effective optimization algorithms can be designed. The CVA proposed in this paper is a population-based algorithm, which generates candidate solutions by imitating the adjustment strategies adopted by teams and players in the volleyball league.

The rest of this article is organized as follows. Section 2 describes the CVA algorithm in detail. The upper part of Section 3 analyzes the impact of several important parameters on the performance of the algorithm. The second half of this section uses 23 test functions to test the CVA, and compares the results with the results of the other 9 famous algorithms mentioned above. Finally, Section 4 concludes this article and prospects related research in the future.

2. The CV algorithm

CVA is used to find the global optimal solution in the continuous search space, and a better solution is obtained by using a variety of search mechanisms in a season.

Every season, the teams will try their best to win the championship. At the same time, every player wants to be the best player (BP) in the team. At the end of the league, the judges choose the best player (MVP). The coach should not only guide the players before the game, but also arrange the players in the field, and adjust the training program according to the results of the game, so the coach's ability plays an important role in a team. Volleyball team generally consists of two coaches, six players and six substitutes. The formation of the two teams on the field is shown in Fig.1.

Before describing the process of the algorithm designed, the following assumptions are stated:

- Each athlete is graded by using the fitness function (objective function) to calculate the fitness value.
- The success or failure of the game depends on the average fitness value of the players in each team.
- The outcome of each game is unknown. In other words, a weaker team may beat a stronger team.
- During recovery, each team adjusts its strategy for the next game only based on last week's results.
- The result of the game is only win or lose, no draw.
- Each team can replace one or more players at any time in a game.

The basic steps of competitive volleyball algorithm can be represented by the schematic flow chart shown in Fig.2. In the following sections, we will introduce the main modules of CVA, especially the pre-match reinforcement, the judgment of winners and losers, and strategies of winners and losers after the competition.

2.1. Solution representation

Each volleyball team consists of coaches (head coach and assistant coach), players on the court and substitutes. These three types of members of a team constitute the representation structure of the solution. Every player

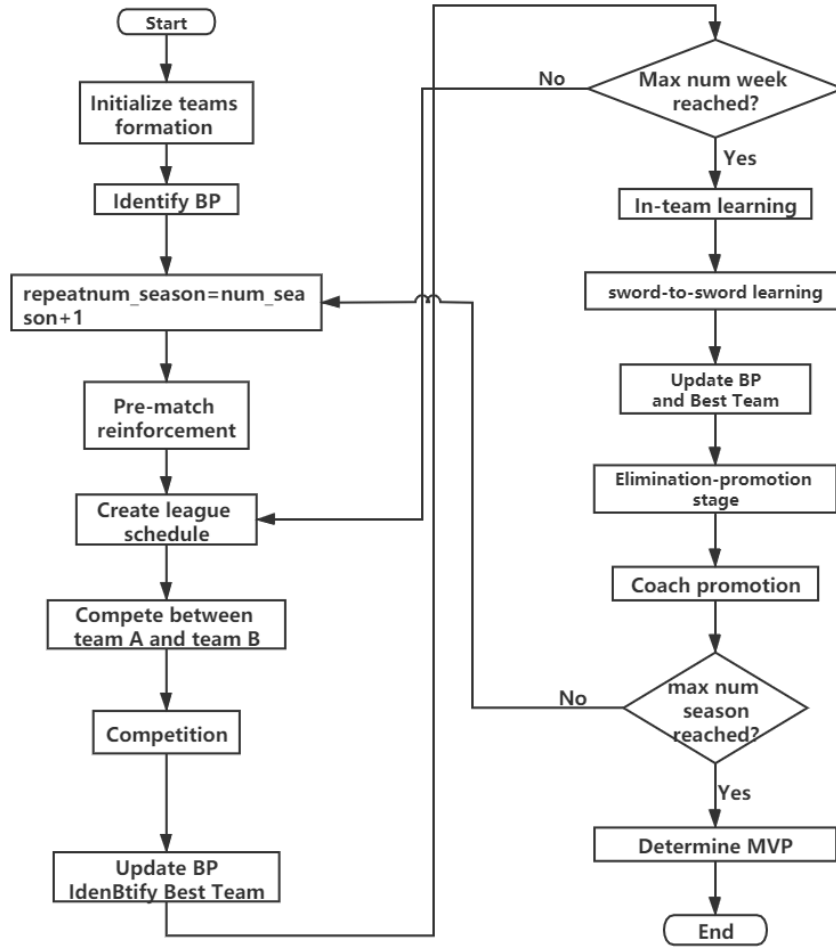


Figure 2. of the Competitive Volleyball Algorithm (CVA)

on the court is the main part of finding the best solution, and the key object in the process of training and competition. Coaches and substitutes are the coordination part, and some adjustment strategies are inspired by this part, such as coach learning strategies and strategies for using substitutes. The rest of this section discusses only the players on the court who represent the solution. The relationship between team composition and solution is shown in Fig.3.

In CVA, the number of seasons is used to represent the number of iterations. A team represents a group of synchronized vector solutions, and each participant represents a vector solution. The composition of each vector solution (decision variable) represents the capabilities of the player. The team, players and abilities are represented as follows:

$$Team_i = \begin{bmatrix} Coach_1 \\ Coach_2 \\ Player_1 \\ Player_2 \\ \dots \\ Player_{PlayerSize} \\ Substitute_1 \\ Substitute_2 \\ \dots \\ Substitute_{SubstituteSize} \end{bmatrix} \quad (1)$$

$$Player_j = [Ab_1 \ Ab_2 \ Ab_3 \ \dots \ Ab_{ProblemSize}] \quad (2)$$

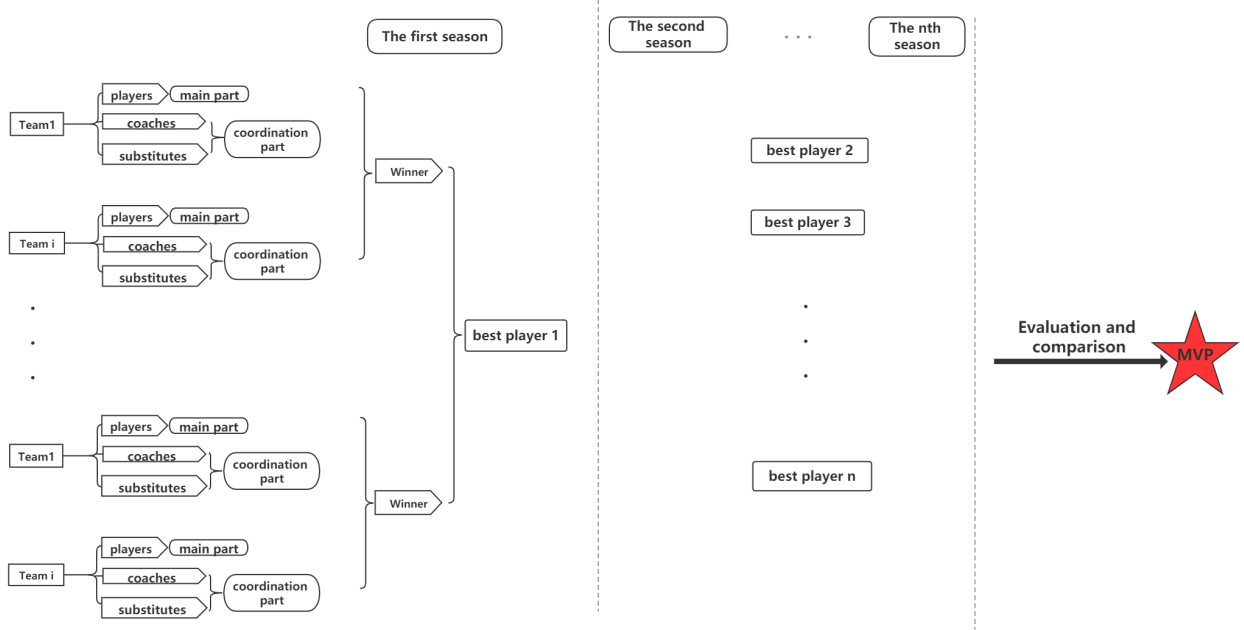


Figure 3. Relationships between team structure and the solution representation

$$f(Player_j) = f([Ab_1 \ Ab_2 \ Ab_3 \ \dots \ Ab_{ProblemSize}]) \quad (3)$$

where $PlayerSize$ is the number of players on a team, $SubstituteSize$ is the number of substitutes on a team, $Ab_{ProblemSize}$ stands for a player's ability, $ProblemSize$ is the dimension of the problem, and $f(Player_j)$ is the objective function value (fitness value) of each player.

2.2. Initialization

In the initialization phase, the total number of members of the team is randomly generated within the search area given by the problem. Let $M_{i,d}^l$ be the l ability of the k member of the i team:

$$M_{i,d}^l = lb_l + r(ub_l - lb_l) \quad (4)$$

where lb_l and ub_l are the lower bound and upper bound of the variable l , respectively, r is a uniformly distributed random number between 0 and 1.

2.3. Formation of team

Here, we assume that the comprehensive strength of the coaches is better than the players in the team. From a mathematical perspective, according to the results of the test function, the randomly generated initial solutions are distributed according to the results of the test function, and the best result is defined as the head coach. the second is deputy coach, and the rest of the initial solutions are randomly assigned to players and substitutes. When forming a team, the coaches need N_c , the players and the substitutes need N_p and N_s respectively, and the number is equal. The following expression can be used to calculate the number of team members:

$$N_c = 2 \quad (5)$$

$$N_p = N_s = \frac{N - 2}{2} \quad (6)$$

where N is the total number of team members.

The main parts of the initial solutions are demonstrated in a matrix P_i , where the numbers in the lower corner represent the team code and player code, respectively, while the number in the upper corner represents

a certain attribute or ability (dimension of the problem) of a player. Similarly, the matrix of coordination part (coaches and substitute) represents the same meaning as the matrix of players on the field. Eqs.7 - 9 represent the coaches, players and substitutes of the team i , respectively,

$$P_i = \begin{bmatrix} X_{i,1}^1 & X_{i,1}^2 & \cdots & X_{i,1}^d \\ X_{i,2}^1 & X_{i,2}^2 & \cdots & X_{i,2}^d \\ \vdots & \vdots & \ddots & \vdots \\ X_{i,N_P}^1 & X_{i,N_P}^2 & \cdots & X_{i,N_P}^d \end{bmatrix} \quad (7)$$

$$C_i = \begin{bmatrix} Y_{i,1}^1 & Y_{i,1}^2 & \cdots & Y_{i,1}^d \\ Y_{i,2}^1 & Y_{i,2}^2 & \cdots & Y_{i,2}^d \\ \vdots & \vdots & \ddots & \vdots \\ Y_{i,N_C}^1 & Y_{i,N_C}^2 & \cdots & Y_{i,N_C}^d \end{bmatrix} \quad (8)$$

$$S_i = \begin{bmatrix} Z_{i,1}^1 & Z_{i,1}^2 & \cdots & Z_{i,1}^d \\ Z_{i,2}^1 & Z_{i,2}^2 & \cdots & Z_{i,2}^d \\ \vdots & \vdots & \ddots & \vdots \\ Z_{i,N_S}^1 & Z_{i,N_S}^2 & \cdots & Z_{i,N_S}^d \end{bmatrix} \quad (9)$$

Players on each team are a major part of the solution. All players in the league, i.e., all candidate solutions generated at the same time, can be represented by composite matrix P , as follows:

$$P = [P_1 \ P_2 \ \dots \ P_{NT}]^T \quad (10)$$

where NT is the total number of teams in the league.

2.4. Pre-match reinforcement

As the game is played under the conditions of high intensity and high speed, it is impossible for volleyball players to win without physical strength. At the same time, the atmosphere of the game is often tense, so how the player performs under pressure is a key to winning the game. This entails having the right physical and psychological qualities.

The activity that makes athletes perform better on the field by improving the overall quality is called pre-match reinforcement strategy. We define team i 's player on the court as $X_{i,j}$ and the substitute as $Y_{i,k}$. The mathematical expressions of this strategy are as follows:

$$phy_{i,j} = e^{\left| 1 - \frac{\sqrt{2X_{i,j}^2(t)}}{\pi} \right|} \quad (11)$$

$$psy_{i,j} = \sin(X_{i,j}(t)) \times \cos(X_{i,j}(t)) \quad (12)$$

$$X_{i,j}(t+1) = -|phy_{i,j} \times psy_{i,j}| \quad (13)$$

$$Y_{i,k}(t+1) = -|phy'_{i,k} \times psy'_{i,k}| \quad (14)$$

where t indicates the current iteration, $phy_{i,j}$ indicates the effect of physical fitness training of the j player of team i , $psy_{i,j}$ indicates the effect of psychological guidance of the j player of team i , $phy'_{i,k}$ indicates the effect of physical fitness training of the k substitute of team i , and $psy'_{i,k}$ indicates the effect of psychological guidance of the k substitute of team i .

The pseudocode of pre-match reinforcement is expressed in Algorithm 1.

Algorithm 1 pseudo-code of pre-match reinforcement

```

for  $i = 1$  to  $NT$  do
  for  $j = 1$  to  $nP$  do
    Update physical ability of player  $j$  using Eq (11)
    Update psychological quality of player  $j$  using Eq (12)
    Update physical ability of substitute  $j$  using Eq (11)
    Update psychological quality of substitute  $j$  using Eq (12)
  end for
end for

```

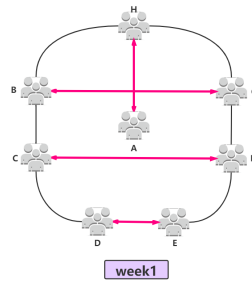


Figure 4. First week of SSR method

2.5. Match schedule

In the single round robin (SSR)(Jaramillo et al., 2016a; Kashan, 2014a; Moghdani and Salimifard, 2018) of each season, each team plays each other only once, and the final rankings are determined by the average strength of each team. In this paper, the SSR method of scheduling, also known as the counterclockwise rotation method, can maintain the excitement of the championship game.

If there are NT teams in the league, then each team will play $(NT - 1)$ games. Then every season, the league has a total of $(NT - 1)NT/2$ games. Whether the number of participating teams is even or odd, it should be arranged according to an even number. If it is an odd number, you can add a team with the serial number of "0". When a team is scheduled to play against team "0", it takes a break.

To demonstrate how the SSR works, we drew a circle of eight teams in the league, keeping team A in the center of the circle. As mentioned above, the eight teams will play seven rounds (weeks) in a season (iteration). As shown in Fig.4, in the first week, team A competes with team H, and the other six teams, which are connected by horizontal lines, compete in pairs (B to G, C to F, and D to E). When the seven teams on the ring rotate in counterclockwise order, this becomes the next week's schedule. The competitive arrangement of the eight teams from the second week to the seventh week is shown in Fig.5 .

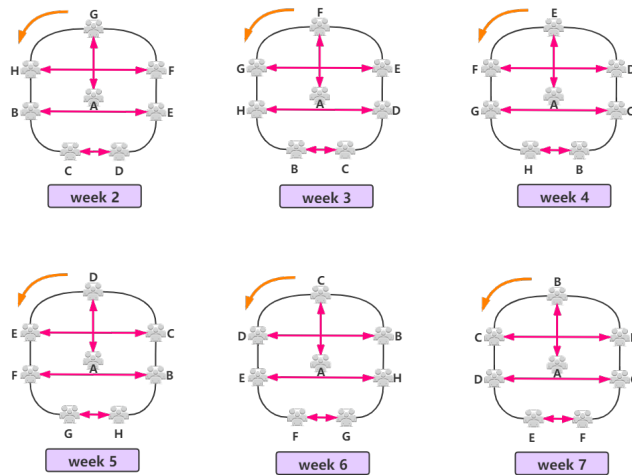


Figure 5. Making league schedule for weeks 2–7

Table 1. A usual league schedule

Week 1	Week 2	Week 3	Week 4	Week 5	Week 6	Week 7
B-G	B-E	B-C	B-H	B-F	B-G	C-H
C-F	C-D	D-H	C-G	C-E	C-F	D-G
D-E	F-H	E-G	D-F	G-H	D-E	E-F

To better illustrate the competitive arrangement, corresponding to Fig.5, the league schedule is shown in Table.1.

2.6. Competition

The aim of the proposed CVA is to solve the optimization problem through the analogy of volleyball competition, which is that the goal of every team is to win the championship, and that every player wants to be the best player (BP) or MVP. It should be stated that the BP is the best player on each team (a local optimum), and MVP is the best player in the league (a global optimum).

Usually, each team plays only one game a week. In order to quantify the comprehensive strength of the team every week and determine the winning team, we use the average strength of the players on the court to represent the strength of the entire team (assumption 2). The mathematical expression is

$$T(i) = \frac{\sum_{j=1}^{N_p} f(X_{i,j})}{N_p} \quad (15)$$

where $T(i)$ indicates the strength of the team i , $f(X_{i,j})$ is the fitness value of player j of team i , and N_p is the number of players of team i .

If there is no significant difference between the two $T(i)$ values, the probability of individual selection is almost the same, which may weaken the CVA selection function. To ensure that the value of winning or losing is not very large, and that there is a certain gap between the two teams, we use regularization technology to calibrate $T(i)$. The specific method is as follows:

$$\phi(i) = \frac{T(i) - T_{min} + \varepsilon}{T_{max} - T_{min} + \varepsilon} \quad (16)$$

$$\varepsilon = 0.1^t \quad (17)$$

where $\phi(i)$ indicates the strength index for team i , and T_{min} and T_{max} represent the lowest and highest strength of the team in a season, respectively. ε is a disturbance (a very small number) so that even the worst team in the league still has a chance to win the game (assumption 3). This chance of winning the game decreases as the number of iterations increases, and indicates that the process of ε change can be represented by Eq.17. It is worth noting that the constant 0.1 is selected after many experiments, and good results are obtained for the optimization problems used in the experiments. However, for other specific instances, the constant 0.1 may not be the most suitable one. It is implied by Eq.(18) that the $\phi(i)$ value of each team is proportional to its fitness value, so the stronger the team, the higher the $\phi(i)$ value. Assuming that team k and team q play each other and their strengths are $T(k)$ and $T(q)$, respectively, the strength index values of the two teams can be calculated as follows:

$$\phi(k) = \frac{T(k) - T_{min} + \varepsilon}{T_{max} - T_{min} + \varepsilon} \quad (18)$$

Let $p(k, q)$ denotes the probability that team k beats team q . The specific formula is as follows:

$$p(k, q) = \frac{\phi(k)}{\phi(k) + \phi(q)} \quad (19)$$

It follows that:

$$p(k, q) + p(q, k) = 1 \quad (20)$$

It can be deduced from Eqs.19 and 20:

$$p(q, k) = 1 - p(k, q) \quad (21)$$

To determine the winner (or loser) of each game, a random number $r \in [0, 1]$ is introduced. If $r \leq p(k, q)$, team k wins; otherwise, team q wins. The manner of determining the winner/loser is consistent with assumption 3 and assumption 5 at the beginning of this section.

After the result of the competition, in order to better prepare for the next week's competition, the winning and losing sides adopt different strategies and training. The pseudocode of competition is shown in Algorithm 2.

The pseudo-code of competition is shown in Algorithm 2.

Algorithm 2 pseudo-code of competition

```

function COMPETITION( $k, q$ )
  Calculate  $T(i)$  and  $\phi(i)$  using Eqs.(15) and (16)
  Calculate  $p(k, q)$  using Eq.(19)
  Generate  $r \in [0, 1]$ 
  if  $r \leq p(k, q)$  then
    team  $k$  is the winner and team  $q$  is the loser
  else
    team  $q$  is the winner and team  $k$  is the loser
  end if
  apply winner promotion strategy for winner,
  apply loser strategy for loser,
end function

```

2.7. Winner promotion strategy

Every member of the winning team places high demands on themselves and wants to be the league MVP. Therefore, they try to learn from the MVP candidate in the league. We know that the learning ability of each participant is different, and so we introduce two evenly distributed random numbers, r_1 and r_2 , into this strategy. The overall strengths of player j and substitute k are defined as $X_{i,j}(t)$ and $Y_{i,k}(t)$, respectively. $X^*(t)$ is the MVP (Boucekara et al., 2018; Boucekara, 2020) candidate for the t season. The overall ability value of each member of the winning team after implementing the training strategy of the winner can be calculated by the following formulas:

$$X_{i,j}(t+1) = (X^*(t) - X_{i,j}(t)) + r_1 \times X_{i,j}(t) \quad (22)$$

$$Y_{i,k}(t+1) = (X^*(t) - Y_{i,k}(t)) + r_2 \times Y_{i,k}(t) \quad (23)$$

where r_1 and r_2 are uniformly distributed random numbers between 0 and 1.

The pseudo-code of winner promotion strategy is shown in Algorithm 3.

Algorithm 3 pseudo-code of winner pseudo-code of winner promotion strategy

```

for  $j = 1$  to  $N_p$  do
  Update the comprehensive strength of player  $j$  Eq.(22)
end for
for  $k = 1$  to  $N_s$  do
  Update the comprehensive strength of substitute  $k$  using Eq.(23)
end for

```

2.8. Loser strategy

In order to achieve better results in their next game, the losing team should employ additional strategies and methods to improve its overall strength. The coach's role is very important, because he not only strategizes to help his players improve their skills but also understands the skills of the best players on the other teams in the league and can formulate a strategy to overcome them.

2.8.1. Coach training strategy

The technical performance level of players is strongly influenced by the guidance of the coaches. The head coach and assistant coach may individually as well as cooperatively train the players and substitutes. In the implementation of this strategy, the head coach and assistant coach guide the players on the court as well as the substitutes. The coaches should not only take into account each player's individual deficiencies but also consider the shortcomings common to all players.

Each of the players is instructed by the head or assistant coach. This process is called coach training, which may help losing teams perform better. The mathematical expressions of this strategy are

$$X_{i,j}(t+1) = X_{i,j}(t) + (ub - lb) \times (Z_{i,1or2}(t) - X_{i,j}(t)) \quad (24)$$

$$Y_{i,k}(t+1) = Y_{i,k}(t) + (ub - lb) \times (Z_{i,1or2}(t) - Y_{i,k}(t)) \quad (25)$$

where ub and lb are the lower and upper bound, respectively, of the variable vector.

The pseudocode for the coach's training strategy is shown in Algorithm 4.

Algorithm 4 pseudo-code of coach training strategy

```

for  $j = 1$  to  $N_p$  do
  Update the comprehensive strength of player  $j$  using Eq.(24)
end for
for  $k = 1$  to  $N_s$  do
  Update the comprehensive strength of substitute  $k$  using Eq.(25)
end for

```

2.8.2. Optimal strategy

Under the guidance of the coach, the confidence of the players increases. Next, the players will improve their physical and psychological abilities. Generally speaking, other athletes should learn from the best players of the current season. At the same time, the best players in the league also serve as examples for the other players. Because the core idea of this strategy is to let players learn from the excellent players to achieve their goal of self-improvement, this strategy is called the optimal strategy.

Members of the losing team need to find the gap between themselves and others in order to improve. Let the gap between the player and the best player in the group be $\Delta X_{i,j}^{BP}(t)$, and the gap between the player and the league's best player be $\Delta X_{i,j}^{CMVP}(t)$ in the t season:

$$\Delta X_{i,j}^{BP}(t) = X_i^*(t) - X_{i,j}(t) \quad (26)$$

$$\Delta X_{i,j}^{CMVP}(t) = X^*(t) - X_{i,j}(t) \quad (27)$$

In the t season, the gap between a particular substitute and the best player on the team i is defined as $\Delta Y_{i,k}^{BP}(t)$, and the gap between that substitute and the best player in the league is defined as $\Delta Y_{i,k}^{CMVP}(t)$:

$$\Delta Y_{i,k}^{BP}(t) = X_i^*(t) - Y_{i,k}(t) \quad (28)$$

$$\Delta Y_{i,k}^{CMVP}(t) = X^*(t) - Y_{i,k}(t) \quad (29)$$

To search for different values, we introduce a common nonlinear activation function Tanh (Emad H.M. Zahran, 2016; Frédéric Godin, 2018) (the range is $(-1, 1)$), so that the nonlinear part is introduced into the whole search space, and the search ability of the algorithm is improved. Next, in order to map $\Delta X_{i,j}^{BP}(t)$, $\Delta X_{i,j}^{CMVP}(t)$, $\Delta Y_{i,k}^{BP}(t)$ and $\Delta Y_{i,k}^{CMVP}(t)$ to the appropriate range, we improve Tanh by introducing the parameters aa, b and d . Here, aa is calculated from the constant a and the upper and lower bounds of variables, as shown in Eq.30:

$$aa = a \times (ub - lb) \quad (30)$$

where a is constant, and ub and lb are the upper and lower bounds of the variable vector.

We use the gap between the players on the losing team and the best players on the team ($\Delta X_{i,j}^{BP}(t)$) and in the league ($\Delta X_{i,j}^{CMVP}(t)$) as the independent variable of the Tanh function. The specific equations are

$$\eta(\Delta X_{i,j}^{BP}(t)) = \frac{b \times e^{aa \times (\Delta X_{i,j}^{BP}(t) - c)} - e^{-aa \times (\Delta X_{i,j}^{BP}(t) - c)}}{e^{aa \times (\Delta X_{i,j}^{BP}(t) - c)} + e^{-aa \times (\Delta X_{i,j}^{BP}(t) - c)}} + d \quad (31)$$

$$\eta(\Delta X_{i,j}^{CMVP}(t)) = \frac{b \times e^{aa \times (\Delta X_{i,j}^{CMVP}(t) - c)} - e^{-aa \times (\Delta X_{i,j}^{CMVP}(t) - c)}}{e^{aa \times (\Delta X_{i,j}^{CMVP}(t) - c)} + e^{-aa \times (\Delta X_{i,j}^{CMVP}(t) - c)}} + d \quad (32)$$

where $\eta(\Delta X_{i,j}^{BP}(t))$ and $\eta(\Delta X_{i,j}^{CMVP}(t))$ are coefficient values, and b , c and d are constant. In the next section, we will discuss the settings of the important parameters in the CVA.

The gap between the losing team's substitutes and the MVP is also mapped to the appropriate range through a variant of the activation function Tanh.

The overall abilities of the players are improved after adopting this strategy, and the skills and methods previously mastered will be partially retained, so in order to calculate the weight w of the retained skills in the improved overall abilities, w_{max} and w_{min} are used in the following formula:

$$w = w_{max} - (w_{max} - w_{min}) \quad (33)$$

where w_{max} refers to the maximum weight of previous ability, while w_{min} refers to the minimum weight of retention. Using the various parameters discussed above, the equation of the overall abilities of the members of each losing team after training is defined as

$$X_{i,j}(t+1) = w \times X_{i,j}(t) + r_1 \times \Delta X_{i,j}^{BP}(t) \times \eta(\Delta X_{i,j}^{BP}(t)) + r_2 \times \Delta X_{i,j}^{CMVP}(t) \times \eta(\Delta X_{i,j}^{CMVP}(t)) \quad (34)$$

where t denotes the current iteration, and r_1 and r_2 are two uniformly distributed random numbers between 0 and 1.

A losing team can use this strategy to calculate the gap between its players with the team's best player and the best player in the league, and then use the variation of Tanh function to map it to the appropriate interval. Finally, combined with the items and weights discussed in this section, the overall ability value of the players after training according to this strategy is obtained. The pseudocode of the optimal strategy is shown in Algorithm 5.

Algorithm 5 pseudo-code of optimal strategy

for $j = 1$ to $N_p(N_s)$ **do**

 Calculate the gap between the player j of team i and the best player in the team $\Delta X_{i,j}^{BP}(t)$ and the league $\Delta X_{i,j}^{CMVP}(t)$ respectively, using Eqs.(26)-(27)

 Calculate the gap between the substitute k of the team i and the best player in the team $\Delta Y_{i,k}^{BP}(t)$ and the $\Delta Y_{i,k}^{CMVP}(t)$ respectively, using Eqs.(28)-(29)

 Map the gap values to the appropriate range of values, using Eqs.(31)-(32)

 Update the comprehensive strength of player j and substitute k , using Eq.(34)

end for

2.8.3. Substitute strategy

When the team is behind in the score, or if one or more of the starting players are injured, the coach should consider putting in a substitute or substitutes (Purnomo and Wee, 2015). If necessary, the coach will choose the substitutes according to the opposing team's playing style and the substitutes' skills, so that the substitutes and the other starting players can form a new overall balance. This strategy is shown in Fig.1 In this paper, there is no limit to the number of substitutes (assumption 6).

We will use this strategy for the losing team and define δ as the team's substitution rate. The following formula calculates the number of team substitutions during the competition:

$$N_{sub} = \delta \times N_p \quad (35)$$

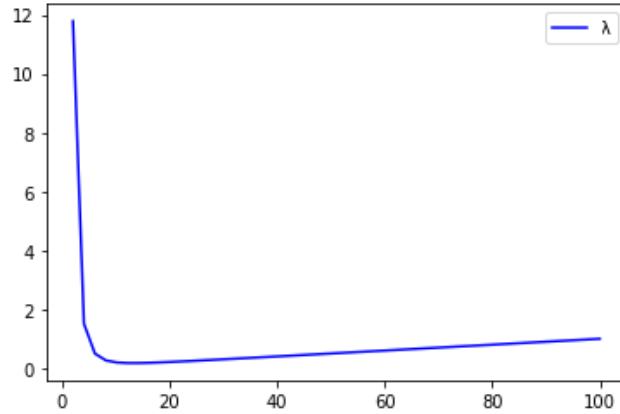


Figure 6. λ value, $t \in [1, 100]$

where N_{sub} denotes the number of substitutes. Suppose the coach puts substitute $Y_{i,k}$ in because starting player $X_{i,j}$ is injured or tired. We define two variables A and B as intermediate variables, and assign the properties of $X_{i,j}$ and $Y_{i,k}$ to A and B , respectively. The strategy is expressed by the following equations:

$$X_{i,j}, Y_{i,k} = Y_{i,k}, X_{i,j} \quad (36)$$

The pseudocode of the strategy for using substitutes is shown in Algorithm 6.

Algorithm 6 pseudo-code of substitute strategy

```

for  $j = 1$  to  $N_{sub}$  do
  Randomly select a player  $j$  and a substitute  $k$ 
  Reverse two members  $j$  and  $k$  using Eq.(36)
end for

```

2.9. One-to-one guidance strategy

Prior to the next week's game, the head coach adopted a one-on-one guidance strategy for the players on the court. In general, the coach has the most experience and the deepest understanding of the game, so the best solution in a group represents the head coach. To show that the ability of players will improve as the season goes on, we introduce a dynamic parameter λ , which changes with the number of iterations. The dynamic equation is

$$\lambda = \frac{T}{t^3} + \frac{t}{T} \quad (37)$$

where t represents the current number of iterations, and T represents the maximum number of iterations. The function is depicted in Fig.6. With the increase in the number of iterations, the value rapidly decreases to the lowest value, and then gradually increases with the increase in the number of iterations. The change in the value of λ can accelerate the convergence process of the solutions and avoid falling into local optimum.

The improvement of a player's ability depends not only on the guidance of the coach but also on the continuing efforts of the player to improve. At this stage, it is expressed by

$$X_{i,j}(t+1) = r \times Z_{i,1}(t) + \lambda X_{i,j}(t) \quad (38)$$

where $Z(i, 1)$ denotes the head coach of team i , and r represents a uniformly distributed random number with values in the range of $[-1, 1]$.

The pseudocode of the one-to-one guidance strategy is shown in Algorithm 7.

Algorithm 7 Pseudo-code of one-to-one guidance strategy

```

for  $i = 1$  to  $NT$  do
  for  $j = 1$  to  $N_p$  do
    Update the comprehensive strength of player  $j$  using Eq.(38)
  end for
end for

```

2.10. Inter-team learning strategy

The main idea of the inter-team learning strategy is that before the game, each team should learn from the other team. The coach should discuss the characteristics of the strongest team in the league to the players so that they can learn why that team is the strongest. For the purposes of this paper, the best team is the one with the highest average overall ability. Players from other teams should learn from the best team. To represent learning from the best team, we define the following equations:

$$\theta = hmr_1 - m \quad (39)$$

$$\nu = hr_2 \quad (40)$$

where θ and ν are coefficient values, h is constant, and r_1 and r_2 are uniformly distributed random numbers from 0 to 1. m is a linearly decreasing coefficient, as shown in Fig.7. Eq.41 is defined to determine the value of m :

$$m = \alpha - \left(t \frac{\alpha}{T}\right) \quad (41)$$

where α is constant, t represents current iteration, and T denotes the maximum number of iterations. Using the parameters proposed above, the equation for describing this strategy is

$$X_{i,j}(t+1) = X_{best,j}(t) - \theta (|\nu(X_{best,j}(t) - X_{i,j}(t))|) \quad (42)$$

From Eq.42, it can be seen that the players on other teams learn from the players in corresponding positions on the best team. As can be seen from Fig.7, as the number of iterations increases, the value of m gradually changes from α to 0, and each player of X gradually tends to the corresponding best player.

Algorithm 8 pseudo-code of Inter-team learning strategy

```

Selected the best team in the league this season.
for  $i = 1$  to  $NT$  do
  for  $j = 1$  to  $N_p$  do
    Update the comprehensive strength of player  $j$  using Eq.(42)
  end for
end for

```

The pseudocode that describes this process is shown in the Algorithm 8.

2.11. Promotion and demotion mechanism

After the competition stage, the top Q teams of the league took part in the next season, and the $(NT - Q)$ teams were demoted. In order to keep the number of teams in each season unchanged, teams with similar strength to those entering the next season will be promoted (Moghdani et al., 2020; Purnomo et al., 2017; Qiao et al., 2020).

If the number of eliminated teams N_{ep} is calculated according to the elimination rate ξ , the number of eliminated teams is calculated by Eq.43. To make the strength of the N_{ep} advancing teams equal to that of the existing teams, the method of this algorithm is to randomly select coaches, players and substitutes from the team that advances to the second stage of competition:

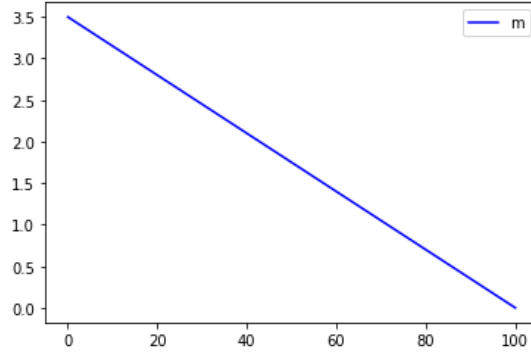


Figure 7. m value, $t \in [1, 100]$

$$N_{ep} = \xi NT \quad (43)$$

The pseudocode explaining the promotion and demotion strategy is shown in Algorithm 9.

Algorithm 9 pseudo-code of promotion and demotion strategy

```

Remove  $N_{ep}$  worst teams of league
Define  $N_{ep}$  empty teams  $PT$  with three properties: coaches, players and substitutes
for  $i = 1$  to  $NT$  do
  for  $j = 1$  to  $N_p$  do
     $s =$  randomly selected from current available teams ( $NT - N_{ep}$ )
     $pl =$  randomly selected from the players of the  $s$  team
     $sub =$  randomly selected from the substitutes of the  $s$  team
     $co =$  randomly selected from the coaches of the  $s$  team
     $X_{N_{ep},j} = X_{s,pl}$ 
     $Y_{N_{ep},j} = Y_{s,sub}$ 
     $Z_{N_{ep},1} = Z_{s,co}$ 
     $Z_{N_{ep},2} = Z_{s,co}$ 
  end for
end for
Add  $N_{ep}$  teams to league

```

2.12. Coach improvement strategy

One of the key factors in improving a team's level of play is that the team have coaches who are proficient in coaching and highly knowledgeable about volleyball skills. Therefore, the self-improvement of coaches is very important.

Coaches should have a full understanding not only of the ability of their players but also of the playing methods, tactics and habits of the players on the other teams in the league. These coaching skills are accumulated over a period of time. This process can be expressed mathematically as

$$Z_{i,m}(t+1) = X^*(t) + r_1(X^*(t) - r_2 X_{i,j}(t)) \quad (44)$$

where r_1 and r_2 represent uniformly distributed random numbers with values in the range of $[0, 1]$. $Z_{i,m}$ represents the head coach or assistant coach of team i , where m is 1 or 2. X^* is a candidate for MVP, and $X_{i,j}$ denotes any player j of team i .

The procedure of a coach improvement strategy is as shown in Algorithm 10.

Algorithm 10 pseudo-code of coach promotion strategy

Selected the candidate for MVP in the league this season
for $i = 1$ to NT **do**
 for $m = 1$ to 2 **do**
 Update the comprehensive strength of coach m using Eq.(44)
 end for
end for

Table 2. A comparison of various features among VPL, SLC, LCA and CVA

features	SCL	LCA	VPL	CVA
Solution structure	Includes just formation property	Includes just formation property	Includes both formation and substitution	Includes formations , substitutions and coaches
Population	League size	League size	League size	League size
Iteration	Week	Week	Season	Season
Initialization	Randomly initializes the team formations according to the league size and the number of variables	Randomly initializes the team formations according to the league size and the number of variables	Randomly initializes the team formations according to the league size and the number of variables	Randomly initializes the team formations according to the league size and the number of variables Applying this operator improved the overall quality of formations and substitutions of each team
Pre-match reinforcement	Not included	Not included	Not included	Using SRR
League schedule	Using SRR	Using SRR	Using SRR	Using SRR
Match analysis	Based on team power	Using SWOT ¹ analysis and yield four equation to find result of match	Based and probability rules and team power index and a new linear equation	Based and probability rules and normalized team power index with a disturbance and a new linear equation
Strategy for winner team	Imitation and provocation	Applying SWTO equations	Winner strategy	Winner promotion strategy
Strategy for loser team	Not included	Applying SWTO equations	Knowledge sharing and repositioning	Coach training strategy, Optimal strategy and Substitute strategy
One-to-one guidance	Not included	Not included	Not included	Using this operator to make athletes meet the coach's requirement
Inter-team learning	Not included	Not included	This operator moves teams toward the top3 best teams	Applying this operator moves teams toward the best team
Promotion and demotion	Promotion and demotion	Promotion and demotion	Promotion and demotion	Promotion and demotion
Coach improvement	Not included	Not included	Not included	This operator allows the coach to improve himself and it is applied at the end of each iteration(season)
Season transfer	Not available	The number of dimension is randomly selected for this operate and applied at the end of each season	Applying this operator based on the parameter and it is applied at the end of each iteration(season)	This operator based on the parameter ξ ,and it is applied at the end of each iteration (season)
Super star player or best team in the league	Super star player	Not included	Best team	MVP candidate, best player of each team and best team

¹ Strengths, Weaknesses, Opportunities and Threats (SWTO)**2.13. Comparison between algorithms inspired by sports tournaments**

In order to better understand our proposed algorithm, compare CVA with VPL(Moghdani and Salimifard, 2018),SLC(Jaramillo et al., 2016b) and LCA(Alizadeh and Kashan, 2019; Kashan, 2014b).Table 2 shows the similarities and differences of various features between the four algorithms.

3. Experiments

To evaluate the performance of CVA, we used the CEC(Behzad Javidy, 2015) for testing purposes. We compared the performance of the CVA with that of nine other classical metaheuristic algorithms(Moghdani and Salimifard, 2018). The convergence speed, global optimization ability and search accuracy of the algorithm were verified by 23 test functions, as shown in Table 3. These test functions have different characteristics.

$f_1 \sim f_7$ are unimodal functions, which can test the optimization accuracy and performance of the algorithm. $f_8 \sim f_{23}$ are multimodal functions with multiple local minima, which can be used to test the ability of the algorithm to jump out of the local optimal solution. Multimodal functions are either high-dimensional or low-dimensional. $f_8 \sim f_{13}$ are high-dimensional multimodal functions, and $f_{14} \sim f_{23}$ are low-dimensional, allowing the search ability of the algorithm to be verified in different dimensions.

The algorithm was tested by CEC in MATLAB. To make the maximum number of function evaluations equal, we set it to 100,000, and to reduce the impact of accidents on the accuracy of the results, we repeated each test experiment 30 times for each algorithm. Finally, the performance of the algorithm was evaluated through the experimental results.

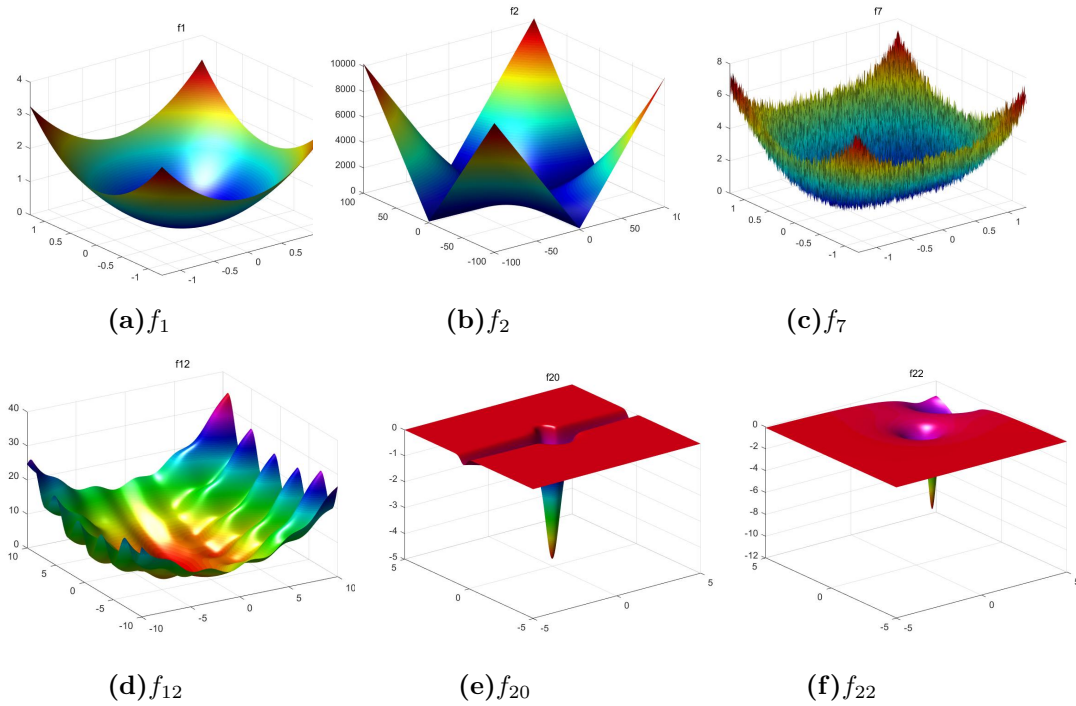


Figure 8. 3-D diagrams of benchmark functions.

3.1. Effects of various parameter settings in CVA

In order to test and evaluate the proposed algorithm, we performed the empirical study of selecting six benchmark functions from three types of test functions ($f_1, f_2, f_7, f_{12}, f_{20}, f_{22}$) (Andrea Tangherloni, 2019).

f_1 is a Sphere Model function with a unique global minimum, which is mainly used to test the optimization accuracy of the algorithm. f_2 is the Schwefel 2.22 function, and its independent variable is epistatic, so its gradient direction will not change along the axis, which means it is difficult to optimize. f_7 is a Quartic noise function with a uniformly distributed random variable in a variation range of $[0,1)$. This function is used to measure the performance of the optimization algorithm when dealing with unimodal functions that are mixed with many noises. f_{12} is a Generalized Penalized Function, which is used to detect the ability of an algorithm to jump out of the local optimum. f_{20} is the Hartmann 6-D function, with six local minimums. f_{22} is the Shekel function, which has seven local minimums. Fig.8 shows the graphs of these six benchmark functions.

Since all the parameters in the algorithm cannot be adjusted at the same time, the method of controlling variables is used when adjusting the parameter values, that is, keeping the other parameter values unchanged and adjusting only the studied parameter values. Results for the optimal values of the different benchmark functions were obtained under different parameter values.

3.2. Impact of ξ value

Elimination rate ξ refers to the proportion of eliminated teams in the league. To maintain the number of teams in the league, we make the elimination rate equal to the promotion rate. At the same time, in order to make the strength of the team members from the promoted team equal to remaining teams in the next season, we randomly select members from other teams to form the promoted team. Therefore, the size of the promotion rate determines the number of promoted teams, which indirectly affects the quality of candidate solutions. To determine the parameter value, we select six equations from three benchmark functions and test each test function with a different set of values for ξ . The results are shown in Table 4.

To better illustrate how the value of parameter ξ is determined, we look at the data in Table 4 (as shown in visual format in Fig.3.5) to see which parameter value is the best. Our rankings are based on the optimal solutions obtained by the different parameter values. The better the solution, the higher the ranking.

As shown in Table 4, most values for ξ enable CVA to find the global optimal solution efficiently. As can be seen from Fig.3.5, when $\xi = 0.5$, the ranking of each function is the highest, so it shows that $\xi = 0.5$ is the most appropriate and that the accuracy of obtaining the optimal solution is the highest.

Table 3 Test functions in experiments

Function	DimRange	f_n
$f_1(x) = \sum_{i=1}^n x_i^2$	30 [-100,100]	0
$f_2(x) = \sum_{i=1}^n x_i + \prod_{i=1}^n x_i $	30 [-10,10]	0
$f_3(x) = \sum_{i=1}^n \left(\sum_{j=1-1}^i x_j \right)^2$	30 [-100,100]	0
$f_4(x) = \max_i \{ x_i , 1 \leq i \leq n \}$	30 [-100,100]	0
$f_5(x) = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i)^2 + (x_i - 1)^2]$	30 [-30,30]	0
$f_6(x) = \sum_{i=1}^n [(x_i + 0.5)]^2$	30 [-100,100]	0
$f_7(x) = \sum_{i=1}^n ix_i^4 + \text{random}[0, 1]$	30 [-1.28, 1.28]	0
$f_8(x) = \sum_{i=1}^n -x_i \sin \sqrt{ x_i }$	30 [-100,100]	0
$f_9(x) = \sum_{i=1}^n [x_i^2 - 10 \cos(2\pi x_i) + 10]$	30 [-10,10]	0
$f_{10}(x) = -20 \exp \left(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2} \right) - \exp \left(\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i) \right) + 20 + e$	30 [-100,100]	0
$f_{11}(x) = \frac{1}{4000} - \prod_{i=1}^n \cos \left(\frac{x_i}{\sqrt{i}} \right) + 1$	30 [-100,100]	0
$f_{12}(x) = \frac{\pi}{n} \left\{ \sin(\pi y_1) + \sum_{i=1}^{n-1} (y_i - 1)^2 [1 + \sin^2(\pi y_{i+1})] + (y_n - 1)^2 \right\} + \sum_{i=1}^n u(x_i, 10, 100, 4)$	30 [-30,30]	0
$y_i = 1 + \frac{x_i+1}{4}$		
$u(x_i, 10, 100, 4) = \begin{cases} k(x_i - a)^m & \text{for } x_i > a \\ 0 & \text{for } -a < x_i < a \\ k(-x_i - a)^m & \text{for } x_i < -a \end{cases}$		
$f_{13}(x) = 0.1 \left\{ \sin^2(3\pi y_1) + \sum_{i=1}^n (x_i - 1)^2 [1 + \sin^2(3\pi y_i + 1)] + (x_n - 1)^2 [1 + \sin^2(2\pi x_n + 1)] \right\} + \sum_{i=1}^n u(x_i, 5, 100, 4)$	30 [-100,100]	0
$f_{14}(x) = \left(\frac{1}{500} + \sum_{j=1}^{25} \frac{1}{j + \sum_{i=1}^n (x_i - a_{ij})^6} \right)^{-1}$	2 [-65,65]	1
$f_{15}(x) = \sum_{i=1}^{11} \left[a_i - \frac{x_i(b_i^2 + b_i x_2)}{b_i^2 + b_i x_3 + x_4} \right]^2$	4 [-5,5]	0.00030
$f_{16}(x) = 4x_i^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4$	2 [-5,5]	-1.0316
$f_{17}(x) = (x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6)^2 + 10(1 - \frac{1}{8}\pi) \cos(x_1) + 10$	2 [-5,5]	0.398
$f_{18}(x) = [1 + (x_1 + x_2 + 1)^2(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)][30 + (2x_1 - 3x_2)^2(18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)]^2$	[-2,2]	3
$f_{19}(x) = - \sum_{i=1}^4 C_i \exp \left(- \sum_{j=1}^3 a_{ij}(x_j - p_{ij})^2 \right)$	3 [1,3]	-3.86
$f_{20}(x) = - \sum_{i=1}^4 C_i \exp \left(- \sum_{j=1}^6 a_{ij}(x_j - p_{ij})^2 \right)$	6 [0,10]	-3.32
$f_{21}(x) = - \sum_{i=1}^5 [(X - a_i)(X - a_i)^T + C_i]^{-1}$	4 [0,10]	-10.1532
$f_{22}(x) = - \sum_{i=1}^7 [(X - a_i)(X - a_i)^T + C_i]^{-1}$	4 [0,10]	-10.4028

Table 4 Specific values

ξ	$f_1(x)$	$f_2(x)$	$f_7(x)$	$f_{12}(x)$	$f_{20}(x)$	$f_{22}(x)$
0.05	4.11E-235	1.25E-146	0.00056412	1.44E-26	-3.3224	-5.0877
0.1	1.23E-243	8.14E-148	0.00061347	3.17E-26	-3.3224	-5.0877
0.15	2.44E-253	5.07E-155	0.00052086	3.13E-30	-3.3224	-10.4029
0.2	4.72E-248	9.85E-157	0.00026888	5.71E-31	-3.3224	-10.4029
0.25	1.83E-259	1.06E-161	0.00034192	2.86E-33	-3.3224	-10.4029
0.3	4.22E-270	1.43E-159	0.00050521	1.57E-33	-3.3224	-10.4029
0.35	6.02E-279	1.15E-165	0.0001021	1.57E-33	-3.3224	-10.4029
0.4	7.03E-272	6.72E-167	0.00031891	1.57E-33	-3.3224	-10.4029
0.45	7.59E-278	2.84E-169	0.00081723	1.57E-33	-3.3224	-10.4029
0.5	6.06E-281	4.97E-174	4.23E-05	1.57E-33	-3.3224	-10.4029

3.3. Impact of g value

In order to analyze the influence of g value setting on CVA, different parameter values are introduced into the same test function, and the results are shown in Table.5 . As can be seen from this table, it is not that the larger the g , the better the results. In other words, the performance of CVA is not proportional to the g value. As can be seen from Fig.3.5, when $g = 3.5$, except for the third in f_1 and f_2 , the ranking is the first in all other test functions, and the average ranking is the highest when $g = 3.5$. So, $g = 3.5$ is the most appropriate one. It can also be seen from Fig.3.5 that the value of g has no effect on the results of two low-dimensional multimodal functions f_{20} and f_{22} .

Table 5 Impact of g

g	$f_1(x)$	$f_2(x)$	$f_7(x)$	$f_{12}(x)$	$f_{20}(x)$	$f_{22}(x)$
0.5	4.13E-159	2.55E-94	0.00044394	2.62E-29	-3.3224	-5.0877
1	4.71E-165	7.35E-98	0.00042002	6.73E-33	-3.3224	-10.4029
1.5	4.19E-241	4.40E-142	0.00053771	6.77E-31	-3.3224	-10.4029
2	6.48E-264	7.12E-169	0.00026056	1.57E-33	-3.3224	-10.4029
2.5	3.11E-275	3.54E-179	0.00033893	1.17E-30	-3.3224	-10.4029
3	4.78E-266	6.61E-182	0.00033846	4.67E-32	-3.3224	-10.4029
3.5	1.43E-265	3.35E-177	0.00024002	1.57E-33	-3.3224	-10.4029
4	1.05E-261	4.62E-172	0.00041067	2.22E-32	-3.3224	-10.4029
4.5	1.82E-259	8.54E-163	0.00079109	7.48E-28	-3.3224	-10.4029
5	4.55E-250	4.50E-164	0.00035712	1.07E-31	-3.3224	-10.4029

3.4. Impact of league size NT

After adjusting the parameters ξ and g , we can adjust and test the number of teams. For this parameter, we specify a maximum number of iterations of 1000 to ensure that the maximum number of iterations for each test (the product of NT and iterations) is equal. The values are set as 10, 20, 30, 40, and 50. The results of each function from different NT values are recorded in Table 6. It can be seen that when $NT \geq 20$, the optimal values of unimodal and multimodal functions can be searched.

For each test function, the ranking of different values for NT is shown in Fig.3.5. To balance the relationship between the number of iterations in each test and the number of groups, NT is set to 20. We can also see that when $NT = 20$, the optimal value of most of the functions can be found with high accuracy, except for in f_2 , when it is in fourth place.

Table 6 Impact of NT

NT	$f_1(x)$	$f_2(x)$	$f_7(x)$	$f_{12}(x)$	$f_{20}(x)$	$f_{22}(x)$
10	1.97E-265	9.11E-166	0.00065735	6.04E-31	-3.3224	-10.4029
20	0	0	0.00039346	1.57E-33	-3.3224	-10.4029
30	0	0	0.00015304	1.57E-33	-3.2032	-10.4029
40	0	0	3.77E-05	1.57E-33	-3.3224	-10.4029
50	0	0	1.11E-05	1.57E-33	-3.3224	-10.4029

3.5. Impact of w_{max} value

The last value to be examined is w_{max} . To determine whether different values of w_{max} affect the performance of CVA, we set the values for the parameter as 1, 2, 3, 4 and 5. The results obtained, as shown in Table 7, show that after determining the three parameters discussed above, w_{max} has little influence on the performance of the algorithm, and there are only some differences in the accuracy of the optimal solutions. According to the ranking of different w_{max} values shown in Fig.3.5, the value is $w_{max} = 4$.

Now we summarize the parameter settings: $\xi = 0.5, g = 3.5, NT = 20$ and $w_{max} = 4$. The control variable method used in parameter settings in this paper may not be perfect, and more in-depth research may be needed to determine these parameters, but this is not the focus of this paper.

Table 7 Impact of w_{max}

w_{max}	$f_1(x)$	$f_2(x)$	$f_7(x)$	$f_{12}(x)$	$f_{20}(x)$	$f_{22}(x)$
1	0	0	0.00011952	1.57E-33	-3.3224	-10.4029
2	0	0	0.00011952	1.57E-33	-3.3224	-10.4029
3	0	0	0.00012512	1.57E-33	-3.2032	-10.4029
4	0	0	5.23E-05	1.57E-33	-3.3224	-10.4029
5	0	0	5.23E-05	1.57E-33	-3.3224	-10.4029

3.6. Results from benchmark functions

Using the same test functions as above, we compared CVA with Genetic Algorithm (GA), Differential Evolution (DE), Harmony Search (HS), Artificial Bee Colony (ABC), Particle Swarm Optimization (PSO), Sine Cosine Algorithm (SCA), Soccer League Competition (SLC), League Championship Algorithm (LCA) and Volleyball Premier League (VPL). To ensure the fairness of the comparative experiment, the number of teams of the algorithm was set at 20, the number of iterations in a single test was 50, and the other algorithms also ensured that the product of the number of groups and the number of iterations was 1000.

In Table 3, $f_1 \sim f_7$ are unimodal functions with only a strict local minimum (peak) in the considered interval. Table 8 shows the statistical results of these 10 algorithms on unimodal functions. In Fig. 10, the bar chart represents the average ranking of algorithm results in $f_1 \sim f_7$, and the line chart is the average ranking of algorithm standard deviation in $f_1 \sim f_7$. Both the average ranking of the results and the average ranking of standard deviation have the highest rankings.

The multimodal functions have many local extremes within the constraint range, which can be used to investigate the ability of jumping out of the local optimum and searching efficiency. Table 9 lists the best, worst, and average optimal solutions, as well as the variance of these 10 algorithms in 30 iterations (Chen, Wang, and Zhao, 2020; Gupta and Deep, 2020). It can be seen that except for f_{10} , the average optimal solution of the algorithm proposed ranks in the top three. The same conclusion can be drawn from Fig. 11.

The performance of the algorithms in low dimensional multimodal functions is analyzed. As shown in Table 10, the average values of the CVA optimal solution rank first among the 10 algorithms, and the variance is the smallest. This shows that the algorithm has good optimization ability and stable performance, and offers advantages in solving many optimization problems. The results in Table 10 are shown in visual format in Fig. 12.

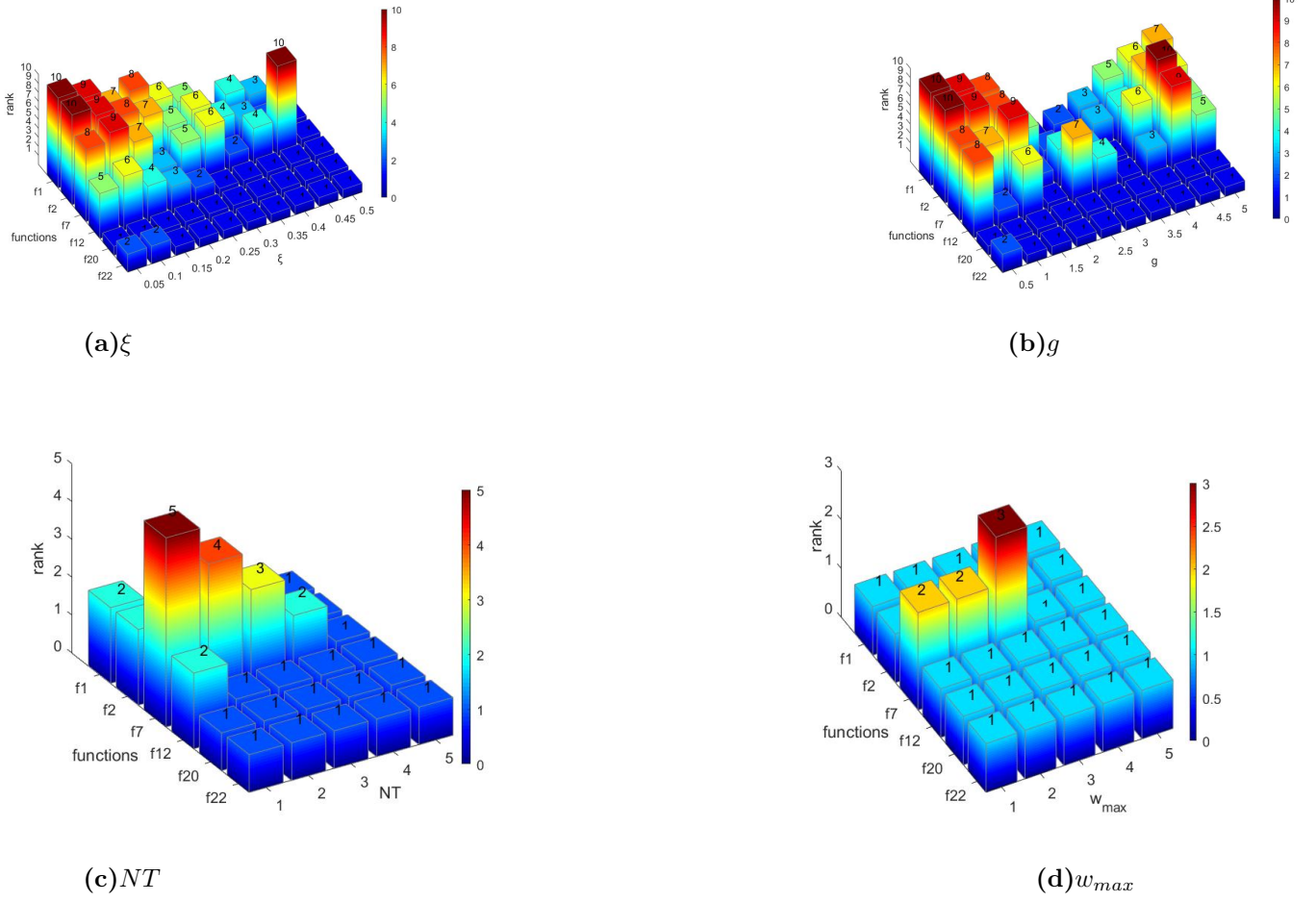


Figure 9. 3-D diagrams of benchmark functions.

Table 8 Results of $f_1 \sim f_7$

Function		GA	DE	HS	ABC	PSO
f_1	Best	1.47E+04	8.08E+02	2.14E+04	8.46E+03	1.12E+02
	Worst	3.87E+04	2.83E+03	3.31E+04	2.52E+04	7.23E+02
	Mean	2.60E+04	1.65E+03	2.73E+04	1.73E+04	2.89E+02
	Std	6.23E+03	8.08E+02	2.89E+03	4.79E+03	1.64E+02
f_2	Best	5.11E+01	9.75E+00	5.51E+01	9.92E+01	3.56E+00
	Worst	2.24E+02	1.98E+01	7.76E+01	1.31E+07	1.53E+01
	Mean	8.80E+01	1.40E+01	6.63E+01	7.65E+05	6.85E+00
	Std	2.80E+01	9.75E+00	5.84E+00	2.53E+06	2.55E+00
f_3	Best	3.47E+04	4.21E+04	3.99E+04	6.90E+04	2.34E+03
	Worst	8.83E+04	9.57E+04	7.99E+04	1.55E+05	1.11E+04
	Mean	5.64E+04	6.61E+04	5.71E+04	9.99E+04	5.74E+03
	Std	1.42E+04	4.21E+04	9.22E+03	2.43E+04	2.30E+03
f_5	Best	6.99E+01	5.00E+01	6.20E+01	6.43E+01	1.32E+01
	Worst	8.83E+01	7.48E+01	7.28E+01	9.11E+01	3.11E+01
	Mean	8.11E+01	6.17E+01	6.73E+01	7.95E+01	2.02E+01
	Std	4.94E+00	5.00E+01	2.41E+00	6.35E+00	4.47E+00
f_6	Best	1.71E+07	8.83E+04	2.10E+07	4.37E+07	1.89E+03
	Worst	1.23E+08	5.15E+06	8.96E+07	1.89E+08	5.26E+04
	Mean	5.60E+07	9.80E+05	5.39E+07	1.06E+08	1.59E+04
	Std	2.19E+07	8.83E+04	1.15E+07	3.51E+07	1.19E+04

f_6	Best	1.23E+04	7.40E+02	1.78E+04	6.90E+03	5.53E+01
	Worst	3.44E+04	6.99E+03	3.33E+04	2.83E+04	7.50E+02
	Mean	2.52E+04	2.05E+03	2.66E+04	1.85E+04	2.51E+02
	Std	5.70E+03	7.40E+02	3.78E+03	4.62E+03	1.51E+02
f_7	Best	6.23E+00	5.23E-01	1.34E+01	6.90E+03	1.05E-01
	Worst	7.11E+01	3.06E+00	3.74E+01	1.02E+02	6.62E-01
	Mean	2.58E+01	1.02E+00	2.42E+01	4.16E+01	3.14E-01
	Std	1.18E+01	5.23E-01	5.26E+00	1.78E+01	1.43E-01
Function		SCA	SLC	LCA	VPL	CVA
f_1	Best	2.67E+02	1.90E-166	1.41E-48	0.00E+00	0.00E+00
	Worst	1.57E+04	1.26E-85	1.25E-45	2.34E-130	0.00E+00
	Mean	6.39E+03	4.40E-160	3.25E-46	7.81E-132	0.00E+00
	Std	3.98E+03	3.91E-80	1.79E-46	4.20E-131	0.00E+00
f_2	Best	1.30E+00	1.12E-125	4.58E-25	1.12E-102	0.00E+00
	Worst	1.98E+01	1.05E-109	5.86E-24	2.85E-89	0.00E+00
	Mean	8.72E+00	8.85E-06	9.79E-25	1.13E-90	0.00E+00
	Std	4.45E+00	9.84E+03	1.49E-24	5.13E-90	0.00E+00
f_3	Best	1.58E+04	2.58E-25	3.26E+03	1.93E-33	0.00E+00
	Worst	7.14E+04	8.16E-01	7.88E+03	1.53E-02	0.00E+00
	Mean	4.18E+04	2.11E-02	1.12E+03	8.16E-04	0.00E+00
	Std	1.33E+04	1.33E+01	6.06E+03	2.85E-03	0.00E+00
f_4	Best	5.64E+01	8.96E-30	1.49E+00	0.00E+00	0.00E+00
	Worst	8.99E+01	9.97E-01	3.47E+00	1.63E-28	0.00E+00
	Mean	7.59E+01	3.07E-04	5.09E-01	1.54E-29	0.00E+00
	Std	7.78E+00	1.03E+00	2.63E+00	3.96E-29	0.00E+00
f_5	Best	1.91E+06	3.96E+01	9.86E-03	2.58E+01	7.58E-07
	Worst	1.32E+08	3.00E+00	2.87E+00	2.67E+01	1.77E-02
	Mean	3.04E+07	3.00E+01	8.42E-01	2.62E+01	6.03E-03
	Std	3.28E+07	3.32E-01	7.15E-01	2.76E-01	4.85E-03
f_6	Best	7.20E+02	1.02E+01	0.00E+00	1.82E-05	0.00E+00
	Worst	2.54E+04	1.04E+01	0.00E+00	2.34E-03	0.00E+00
	Mean	7.72E+03	1.05E+01	0.00E+00	4.09E-04	0.00E+00
	Std	5.51E+03	1.12E-01	0.00E+00	5.33E-04	0.00E+00
f_7	Best	6.95E-01	3.43E-01	1.56E-02	4.67E-05	9.51E-07
	Worst	2.94E+01	3.34E-01	5.91E-02	4.81E-03	3.58E-05
	Mean	1.06E+01	3.76E-06	9.48E-03	1.93E-03	1.46E-05
	Std	7.19E+00	1.60E+01	3.44E-02	1.36E-03	9.08E-06

Table 9 Results of $f_8 \sim f_{13}$

Function		GA	DE	HS	ABC	PSO
f_8	Best	-8.24E+03	-6.18E+04	-7.93E+03	-1.51E+21	-7.79E+03
	Worst	-5.40E+03	-1.59E+04	-5.39E+03	-5.74E+11	-3.84E+03
	Mean	-6.81E+03	-2.68E+04	-6.14E+03	-9.46E+19	-6.21E+03
	Std	6.80E+02	-6.18E+04	5.14E+02	2.96E+20	7.65E+02
f_9	Best	1.36E+02	1.49E+02	2.41E+02	2.79E+02	3.24E+01
	Worst	2.67E+02	2.26E+02	2.89E+02	4.01E+02	1.51E+02
	Mean	1.98E+02	1.90E+02	2.64E+02	3.45E+02	8.96E+01
	Std	3.17E+01	1.49E+02	1.17E+01	2.83E+01	2.58E+01
f_{10}	Best	1.63E+01	8.05E+00	1.80E+01	1.66E+01	3.54E+00
	Worst	1.95E+01	1.43E+01	1.94E+01	2.05E+01	7.88E+00

f_{11}	Mean	1.89E+01	1.04E+01	1.88E+01	1.90E+01	5.57E+00
	Std	6.12E-01	8.05E+00	2.75E-01	1.16E+00	9.84E-01
	Best	1.37E+02	8.49E+00	1.81E+02	7.54E+01	1.55E+00
	Worst	3.07E+02	4.67E+01	3.13E+02	2.92E+02	8.55E+00
f_{12}	Mean	2.26E+02	1.77E+01	2.51E+02	1.48E+02	3.60E+00
	Std	3.65E+01	8.49E+00	3.25E+01	4.55E+01	1.92E+00
	Best	1.84E+07	4.28E+01	3.25E+07	7.23E+07	5.36E+00
	Worst	2.43E+08	1.07E+07	1.17E+08	7.40E+08	3.72E+01
f_{13}	Mean	9.48E+07	4.79E+05	7.99E+07	2.31E+08	1.41E+01
	Std	6.08E+07	4.28E+01	2.50E+07	1.33E+08	6.68E+00
	Best	4.06E+07	6.38E+04	1.04E+08	2.48E+08	2.45E+01
	Worst	4.15E+08	6.22E+06	2.84E+08	8.64E+08	3.48E+03
	Mean	1.39E+08	1.48E+06	2.03E+08	4.94E+08	4.87E+02
	Std	7.88E+07	6.38E+04	4.78E+07	1.72E+08	1.00E+03
Function		SCA	SLC	LCA	VPL	CVA
f_8	Best	-3.51E+03	-1.33E-25	-3.72E+03	-1.19E+112	-1.91E+03
	Worst	-2.44E+03	4.47E-04	-1.06E+03	-7.38E+90	-1.91E+03
	Mean	-2.95E+03	-7.07E+03	7.53E+02	-4.68E+110	-1.91E+03
	Std	3.19E+02	1.24E+02	2.21E+03	2.15E+111	4.48E-06
f_9	Best	3.97E+01	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	Worst	3.22E+02	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	Mean	1.63E+02	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	Std	6.38E+01	0.00E+00	0.00E+00	0.00E+00	0.00E+00
f_{10}	Best	7.59E+00	7.08E-16	2.22E-14	8.88E-16	-1.07E+13
	Worst	2.06E+01	5.18E-12	4.35E-14	8.88E-16	-1.07E+13
	Mean	1.80E+01	7.05E-14	4.93E-15	8.88E-16	-1.07E+13
	Std	3.96E+00	1.86E-29	3.74E-14	9.86E-32	3.97E-03
f_{11}	Best	4.36E+00	0.00E+00	1.28E-13	0.00E+00	0.00E+00
	Worst	1.66E+02	0.00E+00	2.22E-02	0.00E+00	0.00E+00
	Mean	5.71E+01	0.00E+00	5.84E-03	0.00E+00	0.00E+00
	Std	4.19E+01	0.00E+00	2.65E-03	0.00E+00	0.00E+00
f_{12}	Best	4.92E+04	1.04E+01	1.57E-32	1.11E-06	6.26E-08
	Worst	2.51E+08	1.05E+01	1.57E-32	6.56E-05	1.86E-06
	Mean	9.92E+07	0.00E+00	1.09E-47	2.58E-05	9.31E-07
	Std	7.19E+07	8.46E-82	1.57E-32	1.74E-05	4.53E-07
f_{13}	Best	9.70E+05	7.61E-01	1.35E-32	2.63E-05	4.77E-02
	Worst	5.21E+08	9.20E-01	1.35E-32	2.31E-03	4.80E-02
	Mean	1.48E+08	1.00E+00	5.47E-48	4.18E-04	4.77E-02
	Std	1.24E+08	3.09E-01	1.35E-32	4.84E-04	7.79E-05

Table 10 Results of $f_{14} \sim f_{23}$

Function		GA	DE	HS	ABC	PSO
f_{14}	Best	9.98E-01	9.98E-01	9.98E-01	1.03E+00	1.99E+00
	Worst	2.99E+00	1.17E+01	9.98E-01	6.77E+00	1.74E+01
	Mean	1.28E+00	3.49E+00	9.98E-01	2.76E+00	8.90E+00
	Std	6.20E-01	9.98E-01	1.25E-05	1.33E+00	4.40E+00
f_{15}	Best	1.80E-03	1.16E-03	9.15E-04	1.02E-03	6.73E-04
	Worst	1.02E+00	1.59E-02	5.39E-03	2.53E-03	2.74E-02
	Mean	7.97E-02	4.22E-03	1.91E-03	1.38E-03	3.25E-03
	Std	1.86E-01	1.16E-03	1.04E-03	3.62E-04	5.97E-03

f_{16}	Best	-1.03E+00	-1.03E+00	-1.03E+00	-1.03E+00	-1.03E+00
	Worst	3.56E+00	-1.03E+00	-1.03E+00	-1.03E+00	-1.03E+00
	Mean	-2.39E-01	-1.03E+00	-1.03E+00	-1.03E+00	-1.03E+00
	Std	9.85E-01	-1.03E+00	1.33E-03	7.68E-08	3.20E-09
f_{17}	Best	3.98E-01	3.98E-01	3.98E-01	3.98E-01	3.98E-01
	Worst	4.02E-01	3.98E-01	3.99E-01	4.34E-01	3.98E-01
	Mean	3.99E-01	3.98E-01	3.98E-01	4.07E-01	3.98E-01
	Std	1.13E-03	3.98E-01	1.65E-04	9.46E-03	1.08E-10
f_{18}	Best	3.00E+00	3.00E+00	3.00E+00	3.02E+00	3.00E+00
	Worst	9.42E+01	8.40E+01	3.02E+00	6.68E+00	3.00E+00
	Mean	2.44E+01	5.70E+00	3.00E+00	3.82E+00	3.00E+00
	Std	2.89E+01	3.00E+00	4.77E-03	8.27E-01	4.24E-09
f_{19}	Best	-3.00E-01	-3.86E+00	-3.00E-01	-3.86E+00	-3.00E-01
	Worst	-3.00E-01	-9.19E-01	-2.99E-01	-3.60E+00	-3.00E-01
	Mean	-3.00E-01	-3.69E+00	-3.00E-01	-3.79E+00	-3.00E-01
	Std	0.00E+00	-3.86E+00	2.46E-04	6.26E-02	0.00E+00
f_{20}	Best	-3.32E+00	-3.32E+00	-3.31E+00	-3.31E+00	-3.32E+00
	Worst	-3.19E+00	-3.20E+00	-3.15E+00	-3.08E+00	-3.20E+00
	Mean	-3.28E+00	-3.30E+00	-3.24E+00	-3.22E+00	-3.31E+00
	Std	5.28E-02	-3.32E+00	4.93E-02	5.99E-02	4.04E-02
f_{21}	Best	-1.02E+01	-1.02E+01	-1.01E+01	-9.21E+00	-1.02E+01
	Worst	-2.60E+00	-2.54E+00	-2.42E+00	-1.70E+00	-2.63E+00
	Mean	-4.74E+00	-6.90E+00	-4.58E+00	-5.51E+00	-6.24E+00
	Std	2.98E+00	-1.02E+01	3.07E+00	2.16E+00	3.51E+00
f_{22}	Best	-1.04E+01	-1.04E+01	-1.04E+01	-9.60E+00	-1.04E+01
	Worst	-2.66E+00	-2.22E+00	-2.71E+00	-2.84E+00	-2.75E+00
	Mean	-6.32E+00	-6.77E+00	-6.49E+00	-5.79E+00	-6.44E+00
	Std	3.31E+00	-1.04E+01	3.33E+00	1.62E+00	-3.53E+00
f_{23}	Best	-1.05E+01	-1.05E+01	-1.04E+01	-9.90E+00	-1.05E+01
	Worst	-1.67E+00	-2.42E+00	-2.29E+00	-2.36E+00	-1.86E+00
	Mean	-4.23E+00	-6.25E+00	-7.33E+00	-6.24E+00	-6.45E+00
	Std	3.11E+00	-1.05E+01	3.01E+00	2.09E+00	3.86E+00

Function		SCA	SLC	LCA	VPL	CVA
f_{14}	Best	9.98E-01	1.06E-04	9.98E-01	9.98E-01	9.98E-01
	Worst	1.36E+01	7.44E+02	9.98E-01	9.98E-01	9.98E-01
	Mean	5.70E+00	0.00E+00	3.33E-16	9.98E-01	9.98E-01
	Std	4.00E+00	9.86E-32	9.98E-01	2.32E-13	1.36E-13
f_{15}	Best	9.80E-04	0.00E+00	9.95E-04	2.45E-05	2.60E-20
	Worst	6.85E-03	1.44E-24	3.35E-03	1.82E-03	2.30E-15
	Mean	2.61E-03	2.22E-03	4.84E-04	1.25E-03	3.99E-16
	Std	1.56E-03	6.66E-16	1.29E-03	3.08E-04	6.23E-16
f_{16}	Best	-1.03E+00	2.92E-04	-1.01E+00	-1.03E+00	-1.03E+00
	Worst	-1.00E+00	0.00E+00	-9.56E-04	-1.03E+00	-1.03E+00
	Mean	-1.03E+00	4.49E-04	3.26E-01	-1.03E+00	-1.03E+00
	Std	8.03E-03	1.14E-06	4.50E-01	2.56E-06	1.23E-09
f_{17}	Best	3.98E-01	0.00E+00	3.98E-01	3.98E-01	3.98E-01
	Worst	5.17E-01	5.82E-02	3.98E-01	3.98E-01	3.98E-01
	Mean	4.22E-01	1.78E-15	1.11E-16	3.98E-01	3.98E-01
	Std	2.76E-02	0.00E+00	3.98E-01	2.69E-06	3.09E-08
f_{18}	Best	3.00E+00	1.93E-14	3.00E+00	3.00E+00	3.00E+00
	Worst	3.16E+00	4.02E-160	3.00E+00	3.00E+00	3.00E+00
	Mean	3.02E+00	4.81E-82	9.33E-07	3.00E+00	3.00E+00
	Std	3.47E-02	2.58E-152	3.00E+00	7.58E-05	2.32E-08
f_{19}	Best	-3.00E-01	-4.10E-77	-1.96E-01	-3.85E+00	-3.86E+00

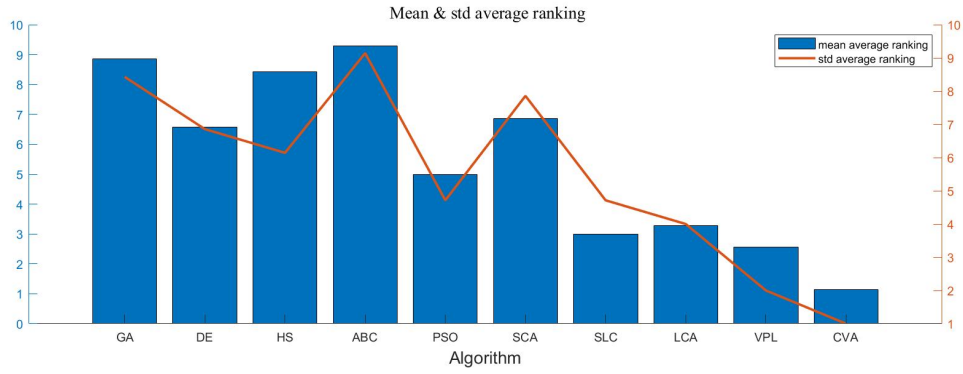


Figure 10. Performances for $f_1 \sim f_7$

	Worst	-3.00E-01	1.42E+01	-6.21E-12	-3.49E+00	-3.86E+00
	Mean	-3.00E-01	-1.36E-26	4.96E-02	-3.77E+00	-3.86E+00
	Std	0.00E+00	1.27E-04	2.81E-02	9.37E-02	2.58E-05
f_{20}	Best	-3.01E+00	-8.60E+03	-3.00E+00	-3.32E+00	-3.32E+00
	Worst	-1.15E+00	0.00E+00	-5.18E-01	-3.20E+00	-3.20E+00
	Mean	-2.56E+00	8.88E-16	5.89E-01	-3.28E+00	-3.26E+00
	Std	5.20E-01	0.00E+00	-1.54E+00	5.41E-02	6.15E-02
f_{21}	Best	-4.30E+00	3.00E-25	-3.40E+00	-1.02E+01	-1.02E+01
	Worst	-3.51E-01	5.67E-04	-2.23E-01	-5.06E+00	-1.02E+01
	Mean	-1.06E+00	9.97E-01	5.67E-01	-9.30E+00	-1.02E+01
	Std	9.39E-01	8.89E-04	5.23E-01	1.90E+00	6.17E-04
f_{22}	Best	-5.34E+00	-1.03E+00	-2.09E+00	-1.04E+01	-1.04E+01
	Worst	-3.74E-01	3.98E-01	-2.91E-01	-5.09E+00	-1.04E+01
	Mean	-1.59E+00	3.00E+00	3.63E-01	-8.99E+00	-1.04E+01
	Std	1.26E+00	3.00E-01	7.00E-01	2.35E+00	3.01E-04
f_{23}	Best	-4.26E+00	-3.27E+00	-2.06E+00	-1.05E+01	-1.05E+01
	Worst	-5.54E-01	-1.02E+01	-4.70E-01	-3.57E+00	-1.05E+01
	Mean	-1.74E+00	-1.04E+01	4.22E-01	-9.40E+00	-1.05E+01
	Std	1.02E+00	1.05E+01	9.31E-01	2.28E+00	4.81E-04

3.7. Convergence property of the algorithm

To explore the convergence of CVA, we selected two benchmark functions from each group. The convergence of different functions is shown in Fig.13.

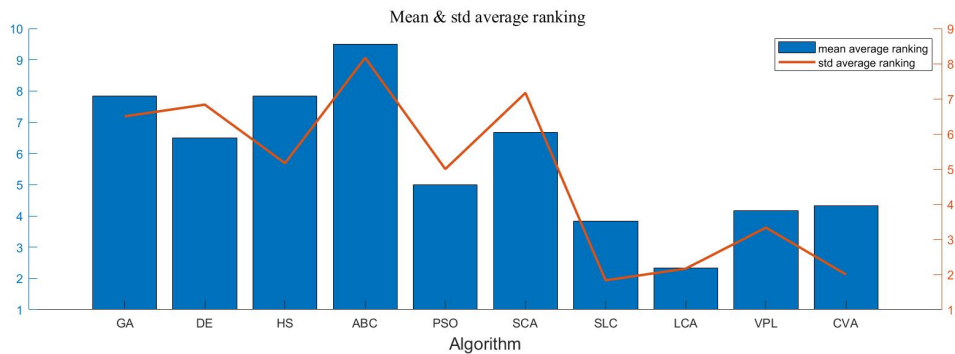


Figure 11. Performances for $f_8 \sim f_{13}$

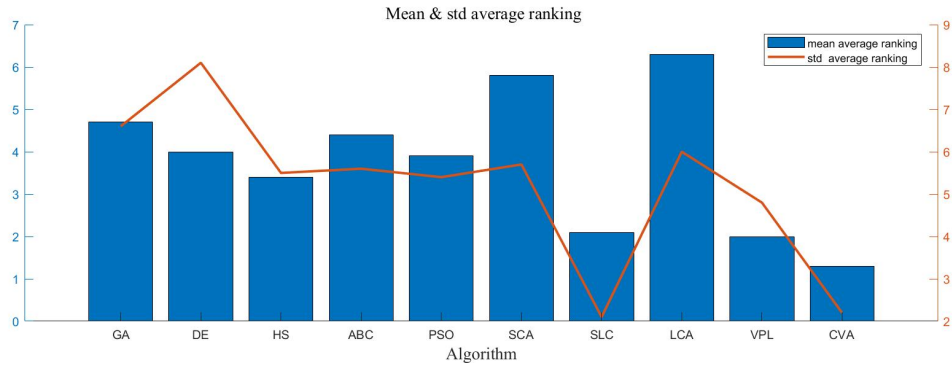


Figure 12. Performances for $f_{14} \sim f_{23}$

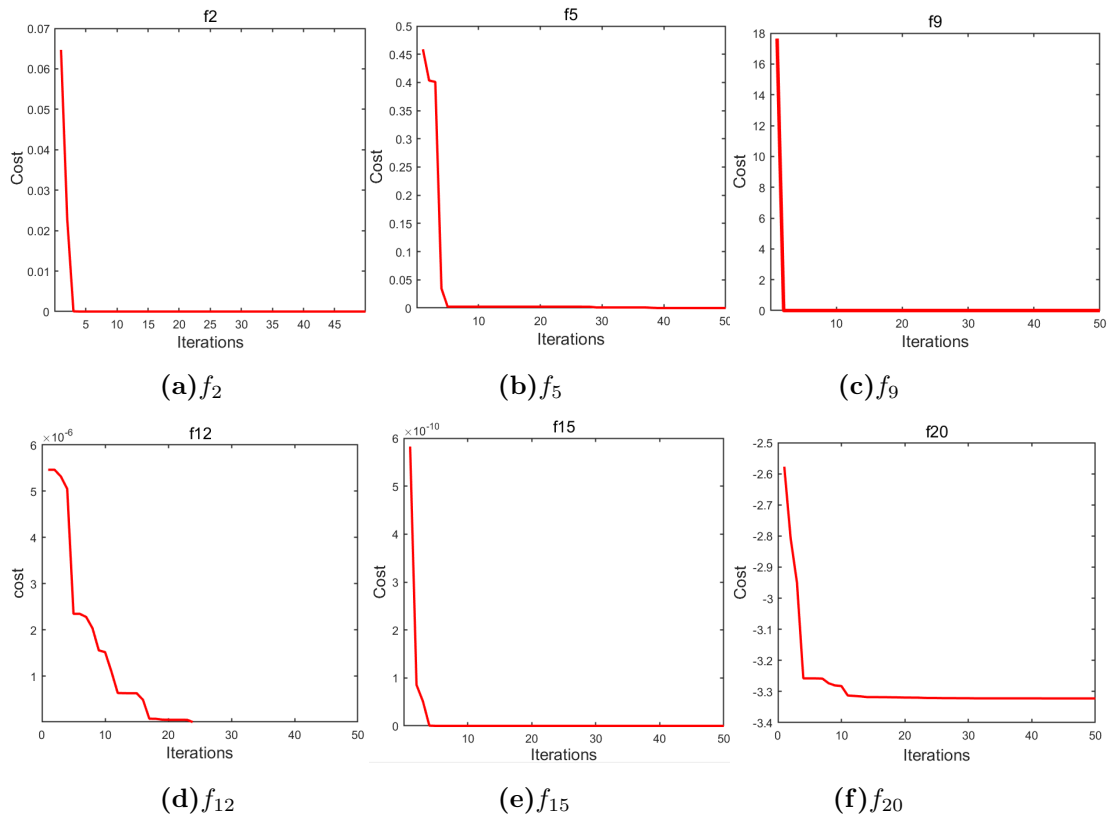


Figure 13. The convergence property of CVA.

4. Application of CVA in Engineering Optimization

This section is dedicated to evaluating the performance of CVA in practical engineering applications, including design of a pressure vessel, minimization of the weight of a tension / compression string and Himmelblau’s non-linear optimization problem. The results are compared with the results calculated by other classical meta-heuristic algorithms.

In this paper, the penalty function is used to solve the constrained nonlinear programming problem, and the appropriate penalty function is made by using the constraint functions in the problem, from which the objective function with parameters is constructed, and the problem is transformed into an unconstrained nonlinear programming problem as follows (Coello, 2002):

$$\text{Minimize } f(x) \tag{45}$$

$$s.t. \begin{cases} g_i(x) \leq 0, i = 1, \dots, r, \\ h_j(x) \geq 0, j = 1, \dots, s, \\ k_m(x) = 0, m = 1, \dots, t \end{cases}$$

$$F(x, M) = f(x) + M \sum_{i=1}^r \max(g_i(x), 0) - M \sum_{i=1}^s \max(h_i(x), 0) + M \sum_{i=1}^t |k_i(x)| \quad (46)$$

Where M is a sufficiently large positive number, $g_i(x)$, $h_i(x)$ and $k_i(x)$ denote the violation from constraint i , the sum of r , s and t is the total number of constraints.

4.1. Design of a pressure vessel

In this problem, there is a special cylindrical container with hemispherical heads at both ends. The purpose of this problem is to minimize the total cost, including material cost, forming cost and welding cost (Kashan (2011)). A schematic diagram of the problem and its characteristics is shown in Fig.14.

There are four design variables in this problem: shell thickness (x_1), container head thickness (x_2), inner diameter (x_3) and cylindrical section length (x_4). The engineering optimization problem can be expressed as follows:

$$\text{Minimize } f(x) = 0.62224x_1x_3x_4 + 1.7781x_2x_3^2 + 3.1661x_4x_1^2 + 19.84x_3x_1^2 \quad (47)$$

$$g_1(x) = -x_1 + 0.0193x_3 \leq 0 \quad (48)$$

$$g_2(x) = -x_3 + 0.00954x_3 \leq 0 \quad (49)$$

$$g_3(x) = -\pi x_3^2 x_4 - \frac{4}{3}\pi x_3^3 + 1296000 \leq 0 \quad (50)$$

$$g_4(x) = x_4 - 240 \leq 0 \quad (51)$$

$$0 \leq x_1 \leq 99, \quad (52)$$

$$0 \leq x_2 \leq 99, \quad (53)$$

$$10 \leq x_3 \leq 200, \quad (54)$$

$$10 \leq x_4 \leq 200 \quad (55)$$

In the above formula, the objective function Eq.47 means to minimize the total cost. Eqs.48-51 are all the constraints of the problem, and the range of variables can be expressed by Eqs.52-55.

Various methods have been applied to solve the Pressure vessel design problem. These methods include , improved ACO (Kaveh and Talatahari, 2010a), CSS (Kaveh and Talatahari, 2010b), different genetic algorithms (Coello, 2000; Coello and Montes, 2002), CPSO (He and Wang, 2007), improved HS (Shayanfar and Gharehchopogh, 2018), DE (Li et al., 2007a), ES (Mezura-Montes and Coello Coello, 2008a), and branch and bound method (Sandgren, 1990). The results of the best solutions found using different methods are shown in Table 11. Compared with other algorithms, CVA has found a better solution.

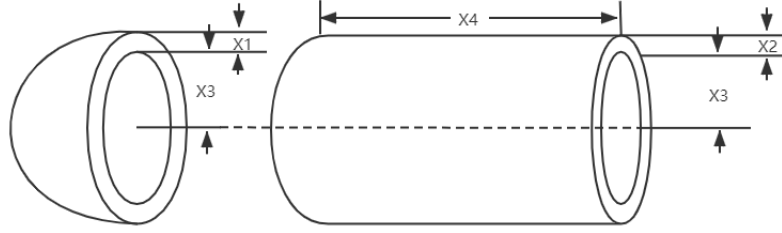


Figure 14. Pressure vessel design and its features.

Table 11. Comparison of CVA results with other methods for Pressure vessel design problem.

Algorithm	Optimum variables				Optimum cost	Feasibility (Yes/No)
	x_1	x_2	x_3	x_4		
CVA	0.812900	0.420200	42.088500	176.950000	6012.6710	Yes
ACO	0.812500	0.437500	42.098353	176.637751	6059.7258	Yes
ALM	1.125000	0.625000	58.2910	43.69	7198.0428	No
CSS	0.812500	0.437500	42.103624	176.572656	6059.0888	No
GAS	0.937500	0.500000	48.329	112.679	6410.3811	Yes
GA(Coello)	0.812500	0.437500	40.323900	200.000000	6288.7445	Yes
GA(Coello and Montes)	0.937500	0.437500	42.097398	176.654050	6059.9463	Yes
GA(Deb and Gene)	0.812500	0.437500	48.329000	112.679000	6410.3811	No
CPSO	0.812500	0.437500	42.091266	176.746500	6061.0777	Yes
WOA	0.812500	0.437500	42.0982699	176.638998	6059.7410	Yes
VPL	0.815200	0.426500	42.0912541	176.742314	6044.9565	Yes
Improved HS	1.125000	0.625000	58.29015	43.69268	7197.730	No
ES	0.812500	0.437500	42.098087	176.640518	6059.7456	No
DE	0.812500	0.437500	42.098411	176.637690	6059.7340	Yes
Branch and Bound	1.125000	0.625000	47.7000	117.7010	8129.1036	No

4.2. Minimization of the weight of a tension / compression string

In this problem, the aim is to minimize the weight of the tension / compression spring, which is subject to minimum deflection, shear stress, surge frequency and limits on outside diameter. The problem consists of three variables: wire diameter (x_1), average coil diameter (x_2) and number of movable coils (x_3) (Coello Coello, 2000). The schematic diagram is shown in Fig.15. The problem can be stated as follows:

$$\text{Minimize } f(x) = (x_3 + 2)x_2x_1^2 \quad (56)$$

$$g_1(x) = 1 - \frac{x_2^3x_3}{71785x_1^4} \leq 0 \quad (57)$$

$$g_2(x) = \frac{4x_2^2 - x_1x_2}{12566(x_2x_1^3 - x_1^4)} + \frac{1}{5108x_1^2} - 1 \leq 0 \quad (58)$$

$$g_3(x) = 1 - \frac{140.45x_1}{x_2^2x_3} \leq 0 \quad (59)$$

$$g_4(x) = \frac{x_1 + x_2}{1.5} - 1 \leq 0 \quad (60)$$

$$0.05 \leq x_1 \leq 2.00, \quad (61)$$

$$0.25 \leq x_2 \leq 1.30, \quad (62)$$

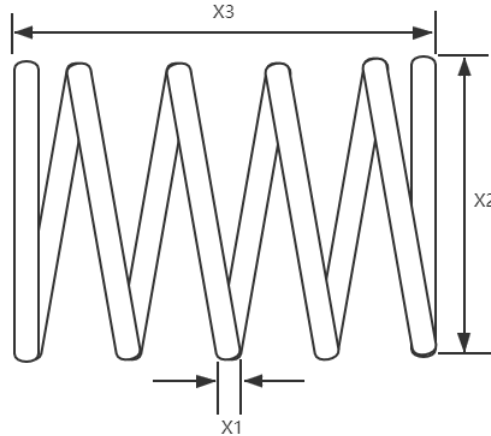


Figure 15. Tension / compression string design and its features.

Table 12. Comparison of CVA results with other methods for tension / compression string design.

Algorithm	Optimum variables			Optimum cost	Feasibility (Yes/No)
	x_1	x_2	x_3		
CVA	0.050600	0.336100	11.182900	0.0113000	Yes
GA (Coello)	0.051480	0.351661	11.632201	0.0127048	Yes
GA(Montes and Coello)	0.051643	0.355360	11.397926	0.0126698	Yes
PSO	0.051728	0.357644	11.244543	0.0126747	Yes
ACO	0.051865	0.361500	11.000000	0.0126432	Yes
DE	0.051609	0.354714	11.410831	0.0126702	Yes
ES	0.051989	0.363965	10.890522	0.0126810	Yes
MCSS (Share et al. 2013)	0.051645	0.356496	11.271529	0.0126192	No
WOA	0.051207	0.345215	12.004032	0.0126763	Yes
VPL	0.0501910	0.331680	12.834269	0.0123947	Yes
IHS	0.051154	0.349871	12.076432	0.0126706	Yes
RO	0.051370	0.349096	11.76279	0.0126788	No

$$2.00 \leq x_3 \leq 15.00, \tag{63}$$

In the above formulas, Eq.56 is the objective function of the problem, Eqs.57-60 are the limiting conditions of the problem, and Eqs. 61-63 define the range of variables.

In the literature, various methods have been used to solve this classical engineering design optimization problem, such as Coello (Coello Coello, 2000), Belegundu [117], Coello and Montes(Coello Coello and Mezura Montes, 2002), He and Wang (He and Wang, 2007), Montes and Coello (Mezura-Montes and Coello Coello, 2008b), Kaveh and.Talathari(Kaveh and Talatahari, 2010a), Mahdavi (Mahdavi, Fesanghary, and Damangir, 2007), Li (Li et al., 2007b), and Kaveh (Kaveh and Khayatazad, 2012),(Moghdani and Salimifard, 2018). The optimal solution obtained by using these methods is shown in Table 12.Compared with other methods, CVA has found a very good solution.

4.3. Himmelblau’s non-linear optimization problem

This question was originally raised by Him-melblau (Himmelblau, 1972). It has previously been used as a benchmark for several other GA-based algorithmsCoello Coello (2000). In this problem, there are five variables, six nonlinear inequality constraints and ten boundary conditions. This problem can be expressed in mathematical expressions as:

$$Minimizef(x) = 5.3578547x_3^2 + 0.8356891x_1x_5 + 37.29329x_1 - 40,792.141. \tag{64}$$

$$g_1(x) = 85.334407 + 0.0056858x_2x_5 + 0.00026x_1x_4 - 0.0022053x_3x_5 \tag{65}$$

$$g_2(x) = 80.51249 + 0.0071317x_2x_5 + 0.0029955x_1x_2 + 0.0021813x_3^2 \quad (66)$$

$$g_3(x) = 9.300961 + 0.0047026x_3x_5 + 0.0012547x_1x_3 + 0.0019085x_3x_4 \quad (67)$$

$$0 \leq g_1(x) \leq 92 \quad (68)$$

$$90 \leq g_2(x) \leq 110 \quad (69)$$

$$20 \leq g_3(x) \leq 25 \quad (70)$$

$$78 \leq x_1 \leq 102 \quad (71)$$

$$33 \leq x_2 \leq 45 \quad (72)$$

$$27 \leq x_3 \leq 45 \quad (73)$$

$$27 \leq x_4 \leq 45 \quad (74)$$

$$27 \leq x_5 \leq 45 \quad (75)$$

In order to solve the Himmelblau's non-linear optimization problem, the generalized reduced gradient method (GRG) has been used, Gen and Cheng (Gen and Cheng) has used a GA based on both local global reference , and Calors A.Coello Coello introduced the notion of using co-evolution to adapt the penalty factors of a fitness function incorporated in GA to solve this problem(Coello Coello, 2000). The results are summarized in Table 13. From the comparison results, CVA ranks first among all the competing algorithms and is the most effective algorithm.

5. Conclusions

This paper presents a swarm-based optimization algorithm. The main idea behind the algorithm is to simulate that all the teams in a volleyball league want to win the championship, and every player wants to be the best player on their team, or even the MVP of the league. A team's strength is defined as the average of the strengths of its players. In order to improve the overall ability of the team and the ability of each individual player, every team and every team member should employ strategies for improving their ability. Each team uses the same pre-game training method, and in a single round robin tournament, the post-match training used will depend on whether the team won or lost. Lastly, each team takes part in a final series of intensive training prior to the next week's game.

The highlight of this algorithm is that both teams and players are constantly improving themselves, and this interactive process of improvement makes the candidate solutions tend to the global optimal solution. In addition, coaches not only lead the team, but also continue to accumulate experience in every season to improve their teaching ability, so that the abilities of players are also constantly being improved. From a mathematical perspective, the algorithm uses the variant of the activation function to strengthen the global search ability of the algorithm. In addition, the algorithm also uses self-adaptability to make the algorithm jump out of the local optimum. Like other sports-inspired algorithms, this algorithm also uses a single round robin system in the competition phase and a promotion and demotion mechanism at the end of each season. These common processes in the field of sports can improve the performance of the algorithm.

Through a series of experimental tests, the value of the important parameters of the algorithm is set, which makes the algorithm more suitable for the specific optimization problem to be solved. The CVA and the other nine algorithms are tested by three groups of benchmark functions (unimodal, multimodal and fixed-dimensional multimodal functions), and then compared against each other. The results show that the CVA algorithm obviously outperforms the other nine algorithms.

Table 13. Comparison of CVA results with other methods for Himmelblau's non-linear optimization problem.

Algorithm	Optimum variables					Optimum cost	Feasibility (Yes/No)
	x_1	x_2	x_3	x_4	x_5		
CVA	78.0000	33.0000	27.0610	45.00000	45.0000	-31,021.477	Yes
Calors A.Coello Coello	78.0495	33.0070	27.0810	45.0000	44.9400	-31,020.859	Yes
Gen and Cheng	81.4900	34.0900	31.2400	42.2000	34.3700	-30,183.576	Yes
Homaifar	78.0000	33.0000	29.9950	45.0000	36.7760	-30,665.609	Yes
GRG	78.6200	33.4400	31.0700	44.1800	35.2200	-30,373.949	Yes

Acknowledgments This work was supported by the project of the humanities and social sciences of the Ministry of Education of China (No.16YJA630037) and the key project of soft science research in science and technology innovation action plan of Shanghai (No.18692110500) , as well as the philosophy and social science planning project of Shanghai (No. 2019BGL014)

Declarations

Conflict of interest The author declares that he has no conflict of interest.

Ethical approval This article does not contain any studies with human participants or animals performed by any authors.

References

- Alizadeh, N., Kashan, A.H.: 2019, Enhanced grouping league championship and optics inspired optimization algorithms for scheduling a batch processing machine with job conflicts and non-identical job sizes. *Applied Soft Computing Journal* **83**.
- Amari, S.I.: 1971, Characteristics of randomly connected threshold-element networks and network systems. *Proceedings of the IEEE* **59**, 35. DOI. [⟨GotoISI⟩://INSPEC:227161](https://doi.org/10.1109/INSPEC.227161).
- Andrea Tangherloni, P.C.D.B.M.S.N. Simone Spolaor: 2019, Biochemical parameter estimation vs. benchmark functions: A comparative study of optimization performance and representation design. *Applied Soft Computing* **81**, 105494. DOI. <http://www.sciencedirect.com/science/article/pii/S1568494619302649>.
- Behzad Javidy, S.M. Abdolreza Hatamlou: 2015, Ions motion algorithm for solving optimization problems. *Applied Soft Computing* **32**, 72. DOI <http://www.sciencedirect.com/science/article/pii/S1568494615001933>.
- Ben Abdellafou, K., Hadda, H., Korbaa, O.: 2019, An improved tabu search meta-heuristic approach for solving scheduling problem with non-availability constraints. *Arabian Journal for Science and Engineering* **44**, 3369. DOI. [⟨GotoISI⟩://WOS:000462305100033](https://doi.org/10.1016/j.aesce.2019.04.003).
- Boucekara, H.R.E.H.: 2020, Most valuable player algorithm: a novel optimization algorithm inspired from sport. *Operational Research* **20**, 139. DOI. [⟨GotoISI⟩://WOS:000512993700007](https://doi.org/10.1016/j.or.2020.05.007).
- Boucekara, H.R.E.-H., Orlandi, A., Al-Qdah, M., de Paulis, F.: 2018, Most valuable player algorithm for circular antenna arrays optimization to maximum sidelobe levels reduction. *Ieee Transactions on Electromagnetic Compatibility* **60**, 1655. DOI. [⟨GotoISI⟩://WOS:000443149000007](https://doi.org/10.1109/TEMC.2018.2788007).
- Chagwiza, G., Jaison, A., Masamha, T., Zhang, H.: 2016, Parameter improvement of the soccer league competition algorithm by introducing stubborn players: Application to water distribution network. *Mathematical Problems in Engineering* **2016**.
- Chekanin, V.A., Kulikova, M.Y.: 2017, Adaptive adjustment of parameters of the genetic algorithm. *Vestnik MGTU Stankin*, 85. [⟨GotoISI⟩://RSCI:29987024](https://doi.org/10.26907/2542-2463.2017.29987024).
- Chen, H., Wang, M., Zhao, X.: 2020, A multi-strategy enhanced sine cosine algorithm for global optimization and constrained practical engineering problems. *Applied Mathematics and Computation* **369**. DOI. [⟨GotoISI⟩://WOS:000500918200004](https://doi.org/10.1016/j.amc.2020.107004).
- Coello, A.C.A.C.: 2000, Use of a self-adaptive penalty approach for engineering optimization problems. *Computers in Industry* **41**, 113. DOI. [http://refhub.elsevier.com/S1568-4946\(17\)30706-8/sbref0580](http://refhub.elsevier.com/S1568-4946(17)30706-8/sbref0580).
- Coello, C.A.C.: 2002, Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: a survey of the state of the art. *Computer Methods in Applied Mechanics and Engineering* **191**, 1245. DOI. [⟨GotoISI⟩://WOS:000173304000009](https://doi.org/10.1016/S0045-7825(02)00009-9).
- Coello, C.A.C., Montes, E.M.: 2002, Constraint-handling in genetic algorithms through the use of dominance-based tournament selection. *Advanced Engineering Informatics* **16**, 193. DOI. [⟨GotoISI⟩://WOS:000182977100003](https://doi.org/10.1016/S1538-4304(02)00003-3).
- Coello Coello, C.A.: 2000, Use of a self-adaptive penalty approach for engineering optimization problems. *Computers in Industry* **41**, 113. DOI. <http://www.sciencedirect.com/science/article/pii/S0166361599000469>.
- Coello Coello, C.A., Mezura Montes, E.: 2002, Constraint-handling in genetic algorithms through the use of dominance-based tournament selection. *Advanced Engineering Informatics* **16**, 193. DOI. <http://www.sciencedirect.com/science/article/pii/S15474034602000113>.

- Duan, M.-y.: 2012, Study of network survivability based on immune clonal simulated annealing algorithm. *Computer Engineering and Design* **33**, 4436. [\(GotoISI\)://INSPEC:13650881](#).
- Emad H.M. Zahran, M.M.A.K.: 2016, Modified extended tanh-function method and its applications to the bogoyavlenskii equation. *Applied Mathematical Modelling* **40**, 1769 . DOI. <http://www.sciencedirect.com/science/article/pii/S0307904X15005429>.
- Frédéric Godin, J.D.W.D.N. Jonas Degraeve: 2018, Dual rectified linear units (drelus): A replacement for tanh activation functions in quasi-recurrent neural networks. *Pattern Recognition Letters* **116**, 8 . DOI. <http://www.sciencedirect.com/science/article/pii/S0167865518305646>.
- Gen, M., Cheng, R.: *Genetic algorithms and engineering design*, Wiley.
- Gupta, S., Deep, K.: 2020, Hybrid sine cosine artificial bee colony algorithm for global optimization and image segmentation. *Neural Computing & Applications* **32**, 9521. DOI. [\(GotoISI\)://WOS:000544784200044](#).
- He, Q., Wang, L.: 2007, An effective co-evolutionary particle swarm optimization for constrained engineering design problems. *Engineering Applications of Artificial Intelligence* **20**, 89. DOI. [\(GotoISI\)://WOS:000243647100008](#).
- Himmelblau, D.M.: 1972, *Applied nonlinear programming*, McGraw-Hill.
- Jalali, S.M.J., Ahmadian, S., Khosravi, A., Mirjalili, S., Mahmoudi, M.R., Nahavandi, S.: 2020, Neuroevolution-based autonomous robot navigation: A comparative study. *Cognitive Systems Research* **62**, 35. DOI. [\(GotoISI\)://WOS:000539725100004](#).
- Jaramillo, A., Crawford, B., Soto, R., Misra, S., Olguin, E., Rubio, A.G., Salas, J., Villablanca, S.M.: 2016a, In: Gervasi, O., Murgante, B., Misra, S., Rocha, A., Torre, C.M., Tanier, D., Apduhan, B.O., Stankova, E., Wang, S. (eds.) *An Approach to Solve the Set Covering Problem with the Soccer League Competition Algorithm*, *Lecture Notes in Computer Science* **9786**, 373. ISBN 978-3-319-42085-1. DOI. [\(GotoISI\)://WOS:000330569000014](#).
- Jaramillo, A., Gomez, A., Mansilla, S., Salas, J., Crawford, B., Soto, R., Olguin, E.: 2016b, In: Rocha, A., Reis, L.P., Cota, M.P., Suarez, O.S., Goncalves, R. (eds.) *Using the Soccer League Competition Algorithm to solve the Set Covering Problem*, *Iberian Conference on Information Systems and Technologies*. ISBN 978-989-98434-6-2. [\(GotoISI\)://WOS:000382923300083](#).
- Kashan, A.H.: 2011, An efficient algorithm for constrained global optimization and application to mechanical engineering design: League championship algorithm (lca). *Computer-Aided Design* **43**, 1769. DOI. [\(GotoISI\)://WOS:000299134900020](#).
- Kashan, A.H.: 2014a, League championship algorithm (lca): An algorithm for global optimization inspired by sport championships. *Applied Soft Computing* **16**, 171 . DOI. <http://www.sciencedirect.com/science/article/pii/S1568494613004250>.
- Kashan, A.H.: 2014b, League championship algorithm (lca): An algorithm for global optimization inspired by sport championships. *Applied Soft Computing* **16**, 171. DOI. [\(GotoISI\)://WOS:000330569000014](#).
- Kaveh, A., Khayatazad, M.: 2012, A new meta-heuristic method: Ray optimization. *Computers & Structures* **112**, 283. DOI. [\(GotoISI\)://WOS:000311595500009](#).
- Kaveh, A., Talatahari, S.: 2010a, An improved ant colony optimization for constrained engineering design problems. *Engineering Computations* **27**, 155. DOI. [\(GotoISI\)://WOS:000277296900008](#).
- Kaveh, A., Talatahari, S.: 2010b, Optimal design of skeletal structures via the charged system search algorithm. *Structural and Multidisciplinary Optimization* **41**, 893. DOI. <https://doi.org/10.1007/s00158-009-0462-5>.
- Khattab, H., Shari'eh, A., Mahafzah, B.A.: 2019, Most valuable player algorithm for solving minimum vertex cover problem. *International Journal of Advanced Computer Science and Applications (IJACSA)* **10**.
- Kim, M.-J., Kim, J.S., Kim, S.J., Kim, M.-j., Ahn, C.W.: 2020, Genetic state-grouping algorithm for deep reinforcement learning. *Expert Systems with Applications* **161**, 113695. DOI. <http://www.sciencedirect.com/science/article/pii/S0957417420305194>.
- Leite, N., Melício, F., Rosa, A.C.: 2019, A fast simulated annealing algorithm for the examination timetabling problem. *Expert Systems with Applications* **122**, 137. DOI. <http://www.sciencedirect.com/science/article/pii/S0957417418308169>.
- Leong, L.-Y., Hew, T.-S., Ooi, K.-B., Dwivedi, Y.K.: 2020, Predicting trust in online advertising with an sem-artificial neural network approach. *Expert Systems with Applications*, 113849. DOI. <http://www.sciencedirect.com/science/article/pii/S0957417420306618>.
- Li, L.J., Huang, Z.B., Liu, F., Wu, Q.H.: 2007a, A heuristic particle swarm optimizer for optimization of pin connected structures. *Computers & Structures* **85**, 340. DOI. [\(GotoISI\)://WOS:000245778800002](#).
- Li, L.J., Huang, Z.B., Liu, F., Wu, Q.H.: 2007b, A heuristic particle swarm optimizer for optimization of pin connected structures. *Computers & Structures* **85**, 340. DOI. [\(GotoISI\)://WOS:000245778800002](#).
- Mahdavi, M., Fesanghary, M., Damangir, E.: 2007, An improved harmony search algorithm for solving optimization problems. *Applied Mathematics and Computation* **188**, 1567. DOI. <http://www.sciencedirect.com/science/article/pii/S0096300306015098>.
- Martin, E., Cervantes, A., Saez, Y., Isasi, P.: 2020, Iacs-hcsp: Improved ant colony optimization for large-scale home care scheduling problems. *Expert Systems with Applications* **142**, 112994. DOI. <http://www.sciencedirect.com/science/article/pii/S0957417419307110>.
- Mezura-Montes, E., Coello Coello, C.A.: 2008a, An empirical study about the usefulness of evolution strategies to solve constrained optimization problems. *International Journal of General Systems* **37**, 443. DOI. [\(GotoISI\)://WOS:000257753200005](#).
- Mezura-Montes, E., Coello Coello, C.A.: 2008b, An empirical study about the usefulness of evolution strategies to solve constrained optimization problems. *International Journal of General Systems* **37**, 443. DOI. [\(GotoISI\)://WOS:000257753200005](#).
- Moghdani, R., Salimifard, K.: 2018, Volleyball premier league algorithm. *Applied Soft Computing* **64**, 161. DOI. [\(GotoISI\)://WOS:000426011800012](#).
- Moghdani, R., Salimifard, K., Demir, E., Benyettou, A.: 2020, Multi-objective volleyball premier league algorithm. *Knowledge-Based Systems* **196**, 22. DOI. [\(GotoISI\)://WOS:000527301700018](#).
- Mohammed, A.M., Duffuaa, S.O.: 2020, A tabu search based algorithm for the optimal design of multi-objective multi-product supply chain networks. *Expert Systems with Applications* **140**, 112808. DOI. <http://www.sciencedirect.com/science/article/pii/S0957417419305044>.
- Nino-Ruiz, E.D., Yang, X.-S.: 2019, Improved tabu search and simulated annealing methods for nonlinear data assimilation. *Applied Soft Computing* **83**, 105624. DOI. <http://www.sciencedirect.com/science/article/pii/S1568494619304041>.
- Niu, X.-z., She, K., Lu, G., Zhou, M.-t.: 2007, Overview of new progress and prospect on ant colony algorithm. *Application Research of Computers* **24**, 12. [\(GotoISI\)://INSPEC:9524636](#).
- Pandey, H.M.: 2020, Secure medical data transmission using a fusion of bit mask oriented genetic algorithm, encryption and steganography. *Future Generation Computer Systems* **111**, 213 . DOI. <http://www.sciencedirect.com/science/article/pii/S0167739X20303848>.
- Purnomo, H.D., Wee, H.-M.: 2015, Soccer game optimization with substitute players. *Journal of Computational and Applied Mathematics* **283**, 79. DOI. [\(GotoISI\)://WOS:000351645000007](#).

-
- Purnomo, H.D., Fibriani, C., Somya, R., Wee, H.-M., Ieee: 2017, *Soccer game optimization for travelling salesman problem, 2017 International Conference on Innovative and Creative Information Technology*. ISBN 978-1-5386-4046-3. [⟨GotoISI⟩://WOS:000454494000018](#).
- Qiao, Y., Dao, T.-K., Pan, J.-S., Chu, S.-C., Nguyen, T.-T.: 2020, Diversity teams in soccer league competition algorithm for wireless sensor network deployment problem. *Symmetry-Basel* **12**. DOI. [⟨GotoISI⟩://WOS:000525824300120](#).
- Sandgren, E.: 1990, Nonlinear integer and discrete programming in mechanical design optimization. *Journal of Mechanical Design* **112**, 223. DOI. <https://doi.org/10.1115/1.2912596>.
- Shayanfar, H., Gharehchopogh, F.S.: 2018, Farmland fertility: A new metaheuristic algorithm for solving continuous optimization problems. *Applied Soft Computing* **71**, 728. DOI. [⟨GotoISI⟩://WOS:000445126100048](#).
- Srichandum, S., Pothiya, S.: 2020, Multiple plants multiple sites ready mixed concrete planning using improved ant colony optimization. *International Journal of Geomate* **19**, 88. DOI. [⟨GotoISI⟩://WOS:000522671800014](#).
- Zhang, H., Liu, F., Zhou, Y., Zhang, Z.: 2020, A hybrid method integrating an elite genetic algorithm with tabu search for the quadratic assignment problem. *Information Sciences* **539**, 347 . DOI. <http://www.sciencedirect.com/science/article/pii/S0020025520306228>.