Advances in Computer, Signals and Systems (2023)
Clausius Scientific Press, Canada

DOI: 10.23977/acss.2023.070703
ISSN 2371-8838 Vol. 7 Num. 7

A Self-Organizing Multimodal Multi-Objective Coati Optimization Algorithm

Waixing Deng[1,a], Yuanbin Mo[2,b,*], Liang Deng[3,c]

[1]School of Artificial Intelligence, Guangxi Mingzu University, Nanning, 530006, China
[2]Guangxi Laboratory of Hybrid Computation and IC Design Analysis, Guangxi Minzu University, Nanning, 530006, China
[3]School of Information Science, Guangdong University of Finance and Economics, Guangzhou, 510000, China
[a]dwx1656471864@163.com, [b]moyuanbin2020@gxmzu.edu.cn, [c]Dengliang_66@outlook.com
[*]Corresponding author

Keywords: Coati Optimization Algorithm, Multimodal multi-objective, Self-organizing speciation, Niching technique

Abstract: The Coati Optimization Algorithm (COA) has emerged as a prominent evolutionary algorithm renowned for its efficacy in addressing real-world problems. Its wide-ranging applicability across diverse domains is a testament to its exceptional performance and versatility. Compared to other evolutionary algorithms, COA has been proven to possess excellent global and local search capabilities. This paper introduces a novel self-organizing multimodal multi-objective Coati Optimization Algorithm (MMOCOA) designed specifically to tackle multimodal multi-objective problems. The proposed algorithm aims to effectively handle the complexities associated with such problems by incorporating self-organizing mechanisms into the Coati optimization framework. Primarily, MMOCOA utilizes a self-organizing speciation method as its primary approach to identify the Pareto optimal solutions. This speciation tactic can establish stable niches and continually updates them to actively search for and preserve the optimal Pareto solutions. Furthermore, an improved self-organization mechanism is proposed to enhance the generation speed of the niches. Additionally, MMOCOA incorporates a non-dominated sorting method and a specialized crowding distance technique to effectively preserve the diversity of both the decision and objective space. To assess the effectiveness of MMOCOA, this study presents a comprehensive evaluation using eleven multimodal multi-objective test functions. Additionally, MMOCOA is benchmarked against five state-of-the-art multimodal multi-objective optimization algorithms. The experimental results highlight the superior performance of MMOCOA, as it demonstrates the capability to discover a larger number of Pareto solutions compared to the other algorithms under consideration.

1. Introduction

Multi-objective problems have been widely used for various real-life problems. In contrast to single-objective optimization problems, multi-objective optimization problems entail the consideration of multiple conflicting objectives, thus requiring a trade-off when attempting to optimize one objective: improving the value of one might affect the value of another. For generality, a minimum description for such multi-objective problems can be expressed as follows.

$$\begin{aligned} &M\,in\; f(x) = (f_1(x), f_2(x), ..., f_m(x)) \\ &s.t.\quad g_s(x) \le 0 \qquad (s = 1, 2, ..., k) \\ &\qquad h_t(x) = 0 \qquad (t = 1, 2, ..., l) \end{aligned} \tag{1}$$

Where $f = (f_1, f_2, ..., f_m) \in Y \subset R^m$ Indicates variables in the objective space and m represents the dimensionality of the objective vector, $x = (x_1, x_2, ..., x_n) \in X \subset R^n$ denotes variables in decision space, n represents the dimensionality of the in decision space. $g_s(x) \le 0(s = 1, 2, ..., l)$ and $h_t(x)=0\,(t = 1, 2, ..., k)$ indicate the imitations that the solution must satisfy in addition to optimizing the objective functions. If a solution x satisfies all the specified constraints, it is referred to as a feasible solution. Different feasible solutions can be evaluated using dominance relations In multi-objective optimization problems, for feasible solution x_1 and feasible solution x_2 , x_1 dominates x_2 requiring two conditions to be satisfied.1)The feasible solution x_1 is better than or equal to x_2 in all objective functions, $i.e. f_i(x_1) \le f_i(x_2)\;\; (i = 1, ..., n)$ .2)The feasible solution x_1 is superior to x_2 at least one objective function, $i.e. f_i(x_1) < f_i(x_2)\;\; (i \in 1, ..., n)$.

A feasible solution x is deemed as a non-dominated solution if none of the other feasible solutions dominate it. The collection of non-dominated solutions is commonly referred to as the Pareto-optimal set (PS) and the collection of vectors that correspond to PS in the objective space is referred to as the Pareto front (PF).

There may be two or more detached Pareto-optimal sets (PSs) of which correspond to the same PF in the multi-objective optimization problem, Liang et al [1] defines this type of problems as a multimodal multi-objective problems (MMOPs). Having multiple PSs is beneficial for decision makers as it provides them with a range of feasible solutions to choose from. Although a single PS is generally sufficient for solving a problem, finding several PSs can have advantages such as increased efficiency for decision makers. Therefore, when working on MMOPs, the opportunity for obtaining numerous PSs in the decision space must be taken into account.

Due to its capacity to address multi-objective problem challenges, multi-objective optimization algorithms have garnered a lot of attention in recent years. As the outcome, numerous unique multi-objective optimization techniques comprised of sophisticated optimization algorithms for various groups have been proposed. Schaffer et al. (1985) [2] presented the Vector Evaluation Genetic approach (VEGA). However, this approach is prone to converge quickly to certain regions. Kalyanmoy Deb et al (1994) [3]. Introduced non-dominated sorting and niches approaches in the genetic algorithm (NSGA) to locate multiple PSs, this algorithm excludes the disadvantage of premature convergence in some regions of VEGA and enables the populations to be distributed over the entire Pareto optimal region. The high computational complexity, non-elitism policy, and requirement for common parameters are a few cons of NSGA, though. Kalyanmoy et al. (2002)[4] developed a quick and exclusive multi-objective genetic algorithm (NSGA-ll) to handle these problems. This algorithm implements a selection operator that combines parents and children to generate a mating pool, then selects the N greatest feasible solutions from this pool. NSGA-ll utilizes a fast non-dominated sorting procedure, an elite-preserving mechanism, and a parameter niching operator, and employs the crowded comparison technique instead of the shared function technique. Consequently, the algorithm's temporal complexity will be reduced, and there won't be any parameters to specify. However, the elite preserving technique may be incapable of storing plenty of non-dominated solutions, and the computing cost of the crowded distance method is expected to increase as the amount of objective functions expands. Kalyanmoy et al. (2013) [5] developed an Evolutionary Many-Objective Optimization Algorithm based on the Reference-Point-Based Non-dominated Sorting Approach (NSGA-lll). This approach prioritizes the size of non-dominated populations, increases population variety, and is well-suited for handling multi-objective optimization problems. However, all of the algorithms discussed above are concerned with the diversity, feasibility, and convergence of solutions in the objective space, instead of only a handful are focused with the allocation of feasible solutions in the decision space. Liang et al. (2016)[1] proposed the decision space Base on niching approach multi-objective evolutionary algorithm (DN-NSGAII), which not only notices solutions in the objective space but also finds the majority of solutions in the decision space. Obtaining all Pareto optimal solutions in the decision space, on the other hand, remains a significant difficulty. Liu et al. (2019) [6] proposed a novel multimodal multi-objective evolutionary algorithm that uses a density-based one-by-one update strategy to estimate the overall density of solutions in the decision space while attempting to maintain the population's diversity in the decision space.

Mohammad Dehghani et al. [7] introduced the Coati Optimization Algorithm (COA) In 2022, a brand-new meta-heuristic method for optimization problems that models two of the Coati's natural behaviors: attacking and hunting iguanas and evading predators. According to simulation studies, COA is superior to existing swarm intelligence optimization algorithms in terms of both global and local search capabilities. It has also shown to be more successful in solving real-world optimization issues.

To efficiently solve MMOPs, a self-organizing multimodal multi-objective Coati optimization algorithm (MMOCOA) is put forth in this study. This study's main contributions include describing the Coati optimization algorithm's implementation process, introducing a self-organized speciation method to create stable niches (sub-populations), exploring and maintaining the Pareto optimal solution set continuously, using a special crowding distance technique and non-dominated sorting methods to maintain diversity in the objective and decision spaces, and evaluating MMOCOA's performance on eleven multimodal multi-objective test functions against five other multi-objective optimization techniques.

The remaining parts of this paper are structured as follows: Chapter 2 introduces the Coati optimization algorithm and the MMOPs; Chapter 3 describes the MMOCOA's detailed implementation process; Chapter 4 presents the paper's experimental findings and analysis; and Chapter 5 offers pertinent conclusions and a look ahead to future research.

2. Related work

2.1 Coati optimization algorithm

The Coati optimization algorithm mimics two of the coatis' natural behaviors: attacking and capturing iguana behavior and escaping from predators behavior. Firstly, Half of the coatis go to intimidate the iguana, and the mathematical simulation of the process is illustrated in equation (2).

$$X_i^{P1} : x_{i,j}^{P1} = x_{i,j} + r \cdot (Iguana_j - I \cdot x_{i,j}),\ i = 1,2,\ldots,\left\lfloor \frac{N}{2} \right\rfloor and\ j = 1,2,\ldots,m, \tag{2}$$

Iguana are deemed to occupy the ideal place within the population. Then, the iguana jumps to a random position and the other half of the coatis attempt to capture it; the mathematical simulation of this process is illustrated in equations (3) and (4).

$$Iguana^G : Iguana_j^G = lb_j + r \cdot (ub_j - lb_j), \quad j = 1, 2, ..., m, \tag{3}$$

$$X_i^{P1} : x_{i,j}^{P1} = \begin{cases} x_{i,j} + r \cdot (Iguana_j^G - I \cdot x_{i,j}), & F_{Iguana^G} < F_i, \\ x_{i,j} + r \cdot (x_{i,j} - Iguana_j^G), & else, \end{cases}$$

$$for \ i = \left\lfloor \tfrac{N}{2} \right\rfloor + 1, \left\lfloor \tfrac{N}{2} \right\rfloor + 2, ..., N \ and \ j = 1, 2, ..., m \tag{4}$$

where X_i^{P1} is the updated location of the i th coati, $X_{i,j}^{P1}$ indicates the value of the j th dimension of X_i^{P1}, F_i^{P1} is the objective value corresponding to X_i^{P1} after the i th update, r is a number chosen at random from [0, 1], $Iguana$ denotes the position of iguana in the search space, which represents the best position in the population, and $Iguana_j$ denotes $Iguana$'s j th dimension. $Iguana^G$ denotes the random position of iguana jump and $Iguana_j^G$ signifies the value of $Iguana^G$'s j th dimension. I is a random number in {1, 2} and $\lfloor \cdot \rfloor$ is an upward rounding symbol.

The content that follows is an in-depth explanation of the process of escaping from the predator: when the predator attacks the coatis. The coatis flee the present spot in search of a safer area closer to it. The mathematical simulation of the process is illustrated in equations (5) and (6).

$$lb_j^{local} = \frac{lb_j}{t}, ub_j^{local} = \frac{ub_j}{t}, where \, t = 1, 2, ..., T \tag{5}$$

$$X_i^{P2} : x_{i,j}^{P2} = x_{i,j} + (1 - 2r) \cdot (lb_j^{local} + r \cdot (ub_j^{local} - lb_j^{local})), \\ i = 1, 2, ..., N, \, j = 1, 2, ..., m, \tag{6}$$

Where X_i^{P2} indicates most recently location the i th coati, $x_{i,j}^{P2}$ is the value of the j th dimension of the X_i^{P2} variable, F_i^{P2} is the objective value corresponding to X_i^{P2} , t is the total number of COA updates., lb_j and ub_j are the decision space's maximum as well as minimum values, respectively.

2.2 Multimodal multi-objective problems

The coupling relationship between the decision space and the objective space, particularly the PS to PF mapping relationship, is at the heart of the problem for MMOPSs. The former determines the search difficulty, and the latter determines the value of the PS. The mapping of the relationship between PS and PF of the MMOPs is generally one-to-one, while there may be circumstances where it is multiple-to-one.Liang et al.(2016)[1] investigated numerous PSs that corresponded to the same PF problem and noted define it as a multimodal multi-objective problem. Then Tanabe et al.(2017)[8] presented a decomposition-based evolutionary algorithm for MMOPs. A group of complex multimodal multi-objective test functions were developed by Liu et al. (2019), and they additionally suggested an efficient MMODE (multimodal multi-objective algorithm combined with differential evolution) algorithm [9]. Liang et al[10] developed a collection of multimodal multi-objective evaluation questions with various characteristics based on the prior research. such as issues with various PS and PF forms, issues with the coexistence of local and global PS, and issues with scalable PS quantities, decision factors, and objectives.

There are multi-modal multi-objective optimization issues with numerous PSs corresponding to the same PF, as illustrated in Figure 1, which shows two PSs corresponding to the same PF.

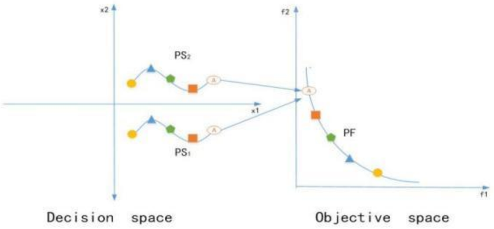

Figure 1: Example of multimodal multi-objective problem.

3. MMOCOA

3.1 The self-organized speciation

The multi-population method is often used to address multimodal problems [11-12]. The size of each sub-population is usually preset when dividing the population into sub-populations [13], which can sometimes lead to inappropriate allocation [14], lowering algorithm performance or resulting in an incomplete PS. Currently, multiple swarming tactics have been employed in MMOPs, but the existing methods still have plenty of shortcomings. For example, MO_Ring_PSO_SCD [15] constructs several sub-populations using a ring topological framework. However, it selects the neighborhood based on an individual index, which occasionally fails to reflect the true distribution in the decision space, potentially influencing the exploration effectiveness of subpopulations. SSMOPSO [16] used a self-organizing mechanism approach to form subpopulations. While this approach improved the effectiveness of the algorithm and reduced the number of overlapping individuals between subpopulations, the seed of the subpopulation may not be the best Pareto optimal solution which can lead to individuals in this subpopulation moving towards local optimal solutions, thus making it less conducive to obtaining all PSs.

To alleviate the drawbacks of the previously described population techniques, this paper proposes an improved self-organized mechanism. This approach uses a self-organized mechanism to form subpopulations, ensuring that the seed of the said subpopulation is a non-dominated individual instead of a local Pareto optimal solution. Since all individuals in the subpopulation will then move toward this non-dominated individual, this enhanced approach improves the overall algorithm performance and its global search ability. The details of this improved self-organizing mechanism can be found in the following section.

3.2 The improved self-organized mechanism

We present a superior self-organizing mechanism that produces subpopulations using a self-organizing technique in the paper. To begin, all solutions that are not dominated in the population are retrieved and saved in the variable P .The subpopulation's radius R is then determined. Following that, a seed for the subpopulation is chosen based on the non-dominated rank of its individuals, individuals within the radius R from the seed are assigned to establish the subpopulation depends on the Euclidean distance, and the individuals assigned to the subpopulation are then removed from the initial population. The assignment step is stopped once every member of the population has been assigned to the subpopulation. Last but not least, the method verifies whether all seeds in the subpopulations are non-dominated solutions; if not, it adds the subpopulation to the subpopulation with the closest non-dominated solution seeds. As indicated in Figure 2, triangles A and G represent non-dominated individuals in the initial population, whereas circle D represents the dominated individual. Triangle A is selected as the seed of the first subpopulation because it is the most excellent non-dominated individual in the initial population. Those within R of the seed A are assigned to the identical subpopulation, resulting in the first subpopulation containing A, B, and C. G is the most excellent non-dominated individual among the remaining individuals in the population and picked as the seed of the 2nd subpopulation. Individuals H and I are included inside the radius R, resulting in the second subpopulation of individuals G, H, and I. Individual D is then the best of the remaining individuals in the population, with individuals E and F falling inside the radius R. As a result, the third population includes those D, E, and F. Since Individual D is not a non-dominated individual in the original population, and the closest non-dominated one from individual D is A, therefore the third subpopulation is integrated into the first. The seed of the subpopulation is the greatest one of the original population, guiding its members to better place. As the subpopulations evolve, they are capable of finding several PSs in the decision space.

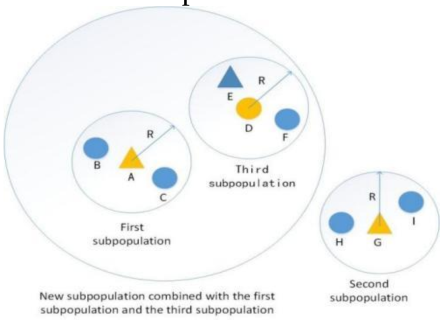
R
E
D
F
Third
subpopulation
R
B
A
C
First
subpopulation
R
I
H
G
Second
subpopulation
New subpopulation combined with the first
subpopulation and the third subpopulation

Figure 2: The fundamentals of the improved self-organized mechanism

Table 1: Pseudo-code for the self-organized speciation

```
individuals are arranged in $P_{\text{sort}}$ according to increasing rank value
Produce $P_{\text{sort}}$
$NP\_sorted$ = the quantity of individuals in $P_{\text{sort}}$
while $NP\_sort \neq 0$
//decide the specie seed
$subpopulation\ seed(j)$ =The first one in $P_{\text{sort}}$
//Speciation based on the improving self-organized mechanism
    for i=1: $NP\_sort$
        if dis($subpopulation\ seed(j)$, $P_{sort}(i)$) $\leq R$
                $P_{sort}(i)$ belongs to $subpopulation(j)$
        end if
    end for
Upgrade $P_{\text{sort}}$ by eliminating those who have been assigned to $subpopulation(j)$
    $NP\_sort$ and $j$
endwhile
$Q$ = The dominated individual in the $species\ seed$
Update $species\ seed$ by removing the individuals in $species\ seed$ that is assigned to $Q$ ;
for k=1:length( $Q$ )
    $t$ =Index of non-dominated individuals in $species\ seed$ closest to $Q(k)$.
    Add the subpopulation to which $Q(k)$ belongs to the population which $species\ seed(t)$ belongs
endfor
Output the set of sub-populations
```

The implementation process of the self-organizing speciation method based on the improved self-organizing mechanism is outlined in Table 1. The population is first sorted by the non-dominated sorting method employing special crowding distances [1] and saved in P_{sort} in order of ascending. The first one in P_{sort} is then chosen as the subpopulation's seed. The Euclidean proximity of the population to the seed is computed, and individuals who fall within the Euclidean distance R from the seed are segregated into the same subpopulation. Individuals designated to subpopulations are subsequently removed from the initial population. The aforementioned processes are repeated until the entire population has been assigned to a subpopulation. Lastly, the algorithm checks whether the seeds in all subpopulations are non-dominated individuals; if not, it adds the subpopulation to which the seed belongs to the subpopulation of the nearest non-dominated seed.

3.4 Procedure of MMOCOA

The primary purpose of Multimodal Multi-Objective Optimization Algorithms is to identify and retain several PSs in the decision space. To achieve this, the improved self-organizing mechanism method is utilized to construct the subpopulation and locate multiple PSs, while Non-dominated-SCD-sort is used to preserve them. Thus, by combining the Non-dominated-SCD-sort with the self-organized speciation method, this strategy has sufficient advantages to solve MMOPs.

Table 2: MMOCOA

```
Initialize the population $POP(0)$
Evaluate ($POP(0)$)
Initialize $Fes$ =1
while $Fes$ <MaxFes do
    Sort every individual in the $POP$ using the Non-dominated-SCD-sort t.
    Formulating subpopulations using the Algorithm 1.
    //SpeciesNumber is the number of subpopulations.
    for j = 1: SpeciesNumber
        if j< SpeciesNumber/2
            for i = 1: length($POP_j(t)$)
                $Iguana_j$ = the first individual in the $POP_j(t)$.
                Upgrade $POP_j(t)$ to $POP_j(t+1)$ according to equation(1) and (4).
            endfor
        else
            for i = 1:length($POP_j(t)$)
                $Iguana_j^G$ = The random individual in the decision space.
                Ugrade $POP_j(t)$ to $POP_j(t+1)$ according to equation (2), (3) and (4).
            endfor
        endif
    endfor
    Upgrade $POP_j(t)$ to $POP_j(t+1)$ according to equation(6) and(7)
    Fes=Fes+1
endwhile
Output the non-dominated individuals in the $POP$
```

After constructing the subpopulations, all individuals in the subpopulation move towards their respective seeds. The process of implementing MMOCOA is outlined in Table 2, where POP

represents the whole population, $POP_j^k(t)$ reflects the value of the j th individual in the k th subpopulation in the t th iteration, and $POP_j(t)$ denotes the t th cycle of the j th subpopulation. $Iguana_j$ denotes the seed of the j th subpopulation, and $Iguana_j^G$ signifies the decision space's random position in the j th subpopulation. The MMOCOA algorithm's process is explained in the following manner. The population as a whole is initialized first, and then the members of the population are ordered using Non-dominated-SCD-sort. Following sorting, several subpopulations are created using the self-organizing speciation technique mentioned in Algorithm 1. Next, the seed in the j th subpopulation is selected as $Iguana_j$ and the random position in the decision space is chosen as $Iguana_j^G$ .Finally, the population $POP_j(t)$ is updated to $POP_j(t+1)$ in accordance with equations (2), (3), (4), and equations (6), (7). Until the termination criteria is met, this process is stopped.

4. Experiment settings

4.1 Test evaluation functions

Eleven test functions are used in this paper's evaluation of MMOCOA's performance. These test functions are MMF1-MMF2 [1], MMF3-MMF8 (suggested by Yue et al. [15]), SYM-PART simple and SMY-PART rotational, Omin-test function (n=3), and two complicated benchmark functions. There are two PSs for each of these functions (MMF1, MMF2, MMF3, and MMF7), while there are four PSs for the remaining MMF test functions. The Omin-test function (n=3) contains 27 PSs, but the SYM-PART issues have nine PSs.

4.2 Performance indicators

In this research paper, we analyze the performance of MMOCOA using Inverted Generational Distance (IGDX) [18], and Pareto Sets Proximity (PSP) [15]. IGDX is the average Euclidean distance between the reference points (true PS) and the obtained PS in the decision space. $S*$ stands for a collection of reference points that are evenly spaced along the actual PS, and K stands for a collection of final population solutions that are not dominated. The average distance in the decision space from $S*$ to K is then used to determine IGDX, as shown below:

$$IGDX(K,S*)=\frac{\sum_{v\in S*}\min\_d(v,K)}{|S*|} \tag{7}$$

where v indicates a true PS in $S*$ and $\min\_d(v,K)$ represents the shortest Euclidean distance between v and the point in K . The IGDX value represents the diversity and convergence of solutions in the decision space; the lower the value, the closer the obtained PS are to the genuine reference points (true PS), and the better the algorithm's performance. IGDX evaluates the quality of the obtained solutions in the decision space.The PSP indicator reflects the similarity between the obtained PSs and true PSs, and the PSP is

$$PSP=\frac{CR}{IGDX} \tag{8}$$

Where IGDX is the Inverted Generational Distance in the decision space, and CR exhibits the overlap ratio between the true PSs and the obtained PSs. The equation for CR is as follows:

$$CR = (\prod_{i=0}^{n} \sigma_1)^{1/2n} \tag{9}$$

$$\sigma_1 = \begin{cases} 1 & V_l^{\max} = V_l^{\min} \\ 0 & v_l^{\min} \geq V_l^{\max} \,\|\, v_l^{\max} \leq V_l^{\min} \\ (\frac{\min(v_l^{\max} - V_l^{\max}) - \max(v_l^{\min}, V_l^{\min})}{V_1^{\max} - V_1^{\min}}) & otherwise \end{cases} \tag{10}$$

where n is the dimension of the decision space, and $v_l^{\min}$ and $v_l^{\max}$ represent the minimum and maximum values in the l dimension of the obtained PSs, and $V_l^{\max}$,$V_l^{\min}$ represent the maximum and minimum values of the l dimension of the true PSs, respectively. The larger PSP values are desirable.

4.3 Comparison with other algorithms

To demonstrate the performance of MMOCOA, this paper compares the effectiveness of MMOCOA with five cutting-edge algorithms: MO_Ring_PSO_SCD [15], SSMOPSO [16], DN-NSGAll [1], NSGA-ll[4], and Omni_Opt [19]. The other five comparison algorithms' parameters are identical to those listed in the original research.The 11 benchmark test functions are subjected to all experiments individually for a total of 30 times, with the average value being used to determine the experiment's ultimate outcome. All algorithms' iteration counts are set to 1000, while the population sizes are set to 100. The experiments are carried out using MATLAB 2021 on a computer running the 64-bit version of Windows 11 and powered by an Intel(R) Core(TM) i7-8750H processor clocked at 2.20 GHz and 2.21 GHz and 8 GB of RAM.

5. Experimental simulation and analysis

5.1 The results of PSP and IGDX on algorithms

The averages and standard deviations of the PSP and IGDX indicator values that the six comparison algorithms produced for each test function are displayed in Tables 3 and 4, respectively. Wilcoxon's rank sum test was utilized to show any notable differences between MMOCOA and the comparative algorithms at a level of significance of 0.05 after thirty independent runs on each test function. The findings of MMOCOA are markedly better, worse, and similar to the comparison algorithm, respectively, by the symbols "+", "-," and "=".

On the 11 multimodal multi-objective test functions, the performance of the proposed MMOCOA in this paper is much superior than the other five multimodal multi-objective algorithms. As shown in Table 3, MMOCOA has the highest average PSP values on MMF1, MMF4, MMF5, MMF6, and MMF7 test functions. Furthermore, MMOCOA obtains weaker average PSP values than SSMOPSO on MMF2 test functions. Additionally, MMOCOA obtains weaker average PSP values than SSMOPSO and MO_Ring_PSO_SCD on MMF8 and MMF3 test function, yet still higher than DN-NGSAll, Omin_Opt, and NSGA-ll. These outcomes show that the suggested MMOCOA algorithm works well on part of all of the benchmark functions, whereas SSMOPSO and MO_Ring_PSO perform well on the other part; however, these benchmark functions only involve 2 and 4 PSs, respectively. The number of PSs for SYM-PART simple, SYM-PART rotated, and Omni-test is relatively big for the remaining three benchmark functions. When compared to other algorithms, MMOCOA's PSP average value performed the best on SYM-PART simple and SYM-PART rotated test functions. Even while the average PSP values achieved by MMOCOA were lower than the averaged PSP values acquired by MO_Ring_PSO_SCD and SSMOPSO on the Omin-test test function, they were still higher than the average PSP values obtained by the other comparison algorithms.

According to the statistical data, the suggested MMOCOA earned the best average PSP values on all seven multimodal multi-objective test functions. On the MMF1, MMF5, MMF6, SYM-PART simple, and SMY-PART rotated test functions, there was no significant difference between MMOCOA and SSMOPSO. However, the average value acquired by MMOCOA was much lower than that obtained by SSMOPSO in the MMF2 and MMF3 test functions, and the average PSP value obtained by MMOCOA on the Omin-test test function was weaker than that obtained by SSMOPSO and MO_Ring_PSO_SCD. Overall, the findings show that MMOCOA performs well on all benchmark functions, despite poor performance on some test functions, such as MMF2 and Omin-test.

Table 3: Comparison of PSP values of MMOCOAs and five comparison algorithms

		NSGA-11	DN-NSGAll	Omni_Opt	SSMOPSO	MO_Ring_PSO _SCD	MMOCOA
MMF1	Mean	5.63E+00	1.01E+01	7.82E+00	1.18E+01	1.01E+01	1.30E+01
	Std	8.78E-01	9.93E-01	1.77E+00	1.03E+00	8.80E-01	1.07E+00
		+	+	+	+	+	
MMF2	Mean	7.60E+00	1.10E+01	2.09E+01	6.06E+01	4.91E+01	5.13E+01
	Std	4.37E+00	6.69E+00	2.17E+01	8.58E+00	7.37E+00	1.07E+01
		+	+	+	-	+	
MMF3	Mean	7.82E+00	1.37E+01	1.20E+01	5.56E+01	5.38E+01	4.90E+01
	Std	3.27E+00	9.31E+00	1.81E+00	9.55E+00	6.68E+00	9.80E+00
		+	+	+	-	-	
MMF4	Mean	3.10E+00	5.17E+00	4.73E+00	1.66E+01	1.65E+01	1.81E+01
	Std	4.75E-01	1.42E+00	1.18E+00	2.18E+00	1.92E+00	2.26E+00
		+	+	+	+	+	
MMF5	Mean	3.08E+00	4.92E+00	3.61E+00	7.26E+00	6.55E+00	7.77E+00
	Std	4.30E-01	6.16E-01	7.20E-01	4.88E-01	6.05E-01	3.34E-01
		+	+	+	+	+	
MMF6	Mean	3.58E+00	5.61E+00	4.79E+00	8.73E+00	7.53E+00	8.82E+00
		5.83E-01	5.34E-01	8.87E-01	6.57E-01	6.15E-01	5.98E-01
		+	+	+	+	+	
MMF7	Mean	7.82E+00	1.84E+01	1.59E+01	1.43E+01	1.66E+01	2.14E+01
		1.63E+00	1.79E+00	2.88E+00	1.69E+00	1.72E+00	1.68E+00
		+	+	+	+	+	
MMF8	Mean	3.30E-01	1.66E+00	1.62E+00	5.35E+00	7.71E+00	5.14E+00
	Std	1.90E-01	6.94E-01	7.17E-01	1.39E+00	1.15E+00	1.15E+00
		+	+	+	-	-	
SYM-PART simple	Mean	1.47E-02	1.45E-01	1.17E-01	8.47E+00	6.54E+00	8.61E+00
	Std	2.33E-02	5.83E-02	6.96E-02	1.64E+00	6.34E-01	1.18E+00
		+	+	+	+	+	
SMY-PART rotated	Mean	4.83E-02	1.04E-01	8.30E-02	7.68E+00	7.02E+00	8.33E+00
	Std	2.03E-02	4.50E-02	5.14E-02	9.10E-01	7.36E-01	2.17E+00
		+	+	+	+	+	
Omin-test	Mean	2.24E-01	3.98E-01	3.94E-01	3.34E+00	2.31E+00	1.60E+00
	Std	9.57E-02	9.97E-02	1.26E-01	6.11E-01	3.78E-01	3.22E-01
		+	+	+	-	-	
+\=\-		11\0\0	11\0\0	11\0\0	7\0\4	9\0\2	

Table 4: Comparison of IGDX values of MMOCOAs and five comparison algorithms

		NSGA-11	DN-NSGAll	Omni_Opt	SSMOPSO	MO_Ring_PSO_SCD	MMOCOA
MMF1	Mean	1.73E-01	9.88E-02	1.22E-01	8.45E-02	9.81E-02	7.77E-02
	Std	2.63E-02	9.30E-03	2.26E-02	7.60E-03	8.40E-03	5.20E-03
		+	+	+	+	+	
MMF2	Mean	1.62E-01	1.17E-01	1.26E-01	1.67E-02	2.04E-02	2.11E-02
	Std	9.04E-02	7.27E-02	1.15E-01	2.40E-03	3.30E-03	9.40E-03
		+	+	+	-	-	
MMF3	Mean	1.27E-01	1.01E-01	9.99E-02	1.84E-02	1.84E-02	2.14E-02
	Std	3.73E-02	5.82E-02	4.86E-02	3.50E-03	2.30E-03	6.20E-03
		+	+	+	-	-	
MMF4	Mean	3.12E-01	2.07E-01	2.16E-01	5.95E-02	6.06E-02	5.53E-02
	Std	4.44E-02	5.14E-02	4.72E-02	7.20E-03	7.50E-03	5.50E-03
		+	+	+	+	+	
MMF5	Mean	3.12E-01	2.04E-01	2.57E-01	1.37E-01	1.51E-01	1.28E-01
	Std	4.58E-02	2.51E-02	4.56E-02	8.90E-03	1.39E-02	9.10E-03
		+	+	+	+	+	
MMF6	Mean	2.71E-01	1.79E-01	2.01E-01	1.14E-01	1.31E-01	1.12E-01
		3.52E-02	1.73E-02	3.82E-02	8.80E-03	1.02E-02	6.90E-03
		+	+	+	+	+	
MMF7	Mean	1.21E-01	5.47E-02	6.22E-02	6.98E-02	5.90E-02	4.67E-02
		1.53E-02	5.60E-03	1.23E-02	8.60E-03	5.70E-03	4.70E-03
		+	+	+	+	+	
MMF8	Mean	1.94E+00	6.97E-01	6.87E-01	1.98E-01	1.30E-01	2.06E-01
	Std	4.88E-01	2.47E-01	3.07E-01	5.17E-02	1.95E-02	5.73E-02
		+	+	+	-	-	
SYM-PART simple	Mean	1.10E+01	6.86E+00	7.21E+00	1.23E-01	1.54E-01	1.17E-01
	Std	2.30E+00	2.47E+00	2.00E+00	2.52E-02	1.52E-02	1.29E-02
		+	+	+	+	+	
SMY-PART rotated	Mean	8.92E+00	6.80E+00	7.79E+00	1.30E-01	1.42E-01	1.26E-01
	Std	2.18E+00	2.14E+00	2.38E+00	1.52E-02	1.53E-02	3.15E-02
		+	+	+	+	+	
Omin-test	Mean	2.51E+00	2.16E+00	2.13E+00	3.08E-01	4.38E-01	6.53E-01
	Std	3.64E-01	2.37E-01	3.06E-01	5.65E-02	7.62E-02	1.36E-01
		+	+	+	-	-	
+\=\-		11\0\0	11\0\0	11\0\0	7\0\4	7\0\4	

Lower IGDX values[17] are preferable, and SSMOPSO delivers the smallest IGDX values among the MMF2, MMF3, MMF8, and Omin-test algorithms. Similarly, MMOCOA achieves the smallest IGDX values on MMF1, MMF4-MMF7, and the SYM-PART simple test function, while MO_Ring_PSO_SCD has the lowest IGDX value on MMF8. The IGDX values gained from SSMOPSO and MMOCOA are similar on the SYM-PART simple, but MMOCOA performs worse than SSMOPSO on MMF2, MMF3, and MMF8. Furthermore, MMOCOA is less efficient than MO_Ring_PSO_SCD on MMF8. Notably, MMOCOA outperforms NSGA-11, DN-NSGAll, and Omni_Opt in terms of IGDX values on all MMF test functions. Consequently, MMOCOA performed satisfactorily when tested against five other multimodal multi-objective algorithms on the MMF test tasks. However, only 2 and 4 PSs were considered for these benchmark tests. For the remaining three problems with a higher number of PSs, Table 4 shows that the IGDX values obtained by MMOCOA are weaker than those obtained by SSMOPSO and MO_Ring_PSO_SCD on the Omin-test test function, similar to those obtained by SSMOPSO on the SYM-PART test functions, weaker than those obtained by MO_Ring_PSO_SCD on the SMY-PART rotated test function, and outperformed DN_NSGAII, Omni_Opt, and NSGA in three test functions. According to the Wilcoxon rank sum test, the IGDX values obtained by MMOCOA are significantly superior than those obtained by NSGA, DN_NSGAII, Omni_Opt, SSMOPSO, and MO_Ring_PSO_SCD in 11, 11, 11, 7, and 7 of 11 comparisons, respectively. Overall, the MMOCOA method produces excellent outcomes for the MMF problems, but it performs poorly for situations with a large number of PSs. It reveals that the MMOCOA method is appropriate for handling issues with fewer PSs.

6. Conclusion

To solve MMOPs, this paper proposes a self-organizing multimodal multi-objective Coati optimization algorithm (MMOCOA). The speciation technique is employed in MMOCOA to produce several enduring niches/subpopulations, and individuals in the subpopulations advance in parallel towards a large number of PSs. In the meantime, a novel self-organized method has been introduced to increase the original speciation strategy's effectiveness and reliability.

The results of the experiment reveal that MMOCOA outperforms other algorithms in terms of performance and capacities to keep more PSs in the decision space. However, MMOCOA performs poorly on multimodal multi-objective evaluations with irregular PS or numerous PSs, such as MMF3, MMF8, and Omin-test.In the future, we will increase MMOCOA's performance so that it can solve test functions with irregular PSs or a high number of PSs, and it will be used to solve real-world MMOPs.

Acknowledgements

This work was supported by the National Natural Science Foundation of China [grant numbers 21466008]; the Guangxi Natural Science Foundation, China [grant numbers 2019GXNSFAA185017]; the Scientific Research Project of Guangxi Minzu University [grant numbers 2021MDKJ004].

References

[1] Liang J, Yue C T, Qu B Y. Multimodal multi-objective optimization: A preliminary study[C]//2016 IEEE Congress on Evolutionary Comp utation (CEC). IEEE, 2016: 2454-2461.

[2] Schaffer J. D. (1985). Multiple objective optimization with vector evaluated genetic algorithms. In Grefenstette, J. J., editor, Proc. First Int. Conf. on Genetic Algorithms, pp. 93–100. Lawrence Erlbaum.

[3] Srinivas N, Deb K. Multiobjective Optimization Using Nondominated Sorting in Genetic Algorithms [J]. Evolutionary Computation, 1994, 2(3):221-248.

[4] Deb K, Pratap A, Agarwal S, et al. A fast and elitist multiobjective genetic algorithm: NSGA-II [J]. IEEE transactions on evolutionary computation, 2002, 6(2): 182-197.

[5] Deb K, Jain H. An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part I: solving problems with box constraints [J]. IEEE transactions on evolutionary computation, 2013, 18(4): 577-601.

[6] Shi R, Lin W, Lin Q, et al. Multimodal multi-objective optimization using a density-based one-by-one update strategy [C]//2019 IEEE Congress on Evolutionary Computation (CEC). IEEE, 2019: 295-301.

[7] Dehghani M, Montazeri Z, Trojovská E, et al. Coati Optimization Algorithm: A new bio-inspired metaheuristic algorithm for solving optimization problems [J]. Knowledge-Based Systems, 2023, 259: 11 0011.

[8] R. Tanabe, H. Ishibuchi, A Decomposition-Based Evolutionary Algorithm for Multi-modal Multi-objective Optimization, International Conference on Parallel Problem Solving from Nature, pp. 249-261, 2018.

[9] Liang J, Xu W, Yue C, et al. Multimodal multiobjective optimization with differential evolution[J]. Swarm and evolutionary computation, 2019, 44: 1028-1059.

[10] Liang J J, Qu B Y, Gong D W, et al. Problem definitions and evaluat ion criteria for the CEC 2019 special session on multimodal multiobjective optimization[J]. Computational Intelligence Laboratory, Zhengzhou University, 2019.

[11] Li Y, Chen Y, Zhong J, et al. Niching particle swarm optimization with equilibrium factor for multi-modal

optimization [J]. Information Sciences, 2019, 494: 233-246.

[12] Deb K, Tiwari S. Omni-optimizer: a procedure for single and multi-objective optimization. In: Proceedings of International Conference onEvolutionary Multi-Criterion Optimization, Guanajuato, 2005. 47–61

[13] Qu B Y, Suganthan P N, Liang J J. Differential evolution with neighborhood mutation for multimodal optimization [J]. IEEE transactions on evolutionary computation, 2012, 16(5): 601-614.

[14] Qu B, Li C, Liang J, et al. A self-organized speciation based multi-objective particle swarm optimizer for multimodal multi-objective problems [J]. Applied Soft Computing, 2020, 86: 105886.

[15] Yue C, Qu B, Liang J. A multiobjective particle swarm optimizer using ring topology for solving multimodal multiobjective problems [J]. IEEE Transactions on Evolutionary Computation, 2017, 22(5): 805-817.

[16] Liang J, Guo Q, Yue C, et al. A self-organizing multi-objective particle swarm optimization algorithm for multimodal multi-objective problems[C]//Advances in Swarm Intelligence: 9th International Conference, ICSI 2018, Shanghai, China, June 17-22, 2018, Proceedings, Part I 9. Springer International Publishing, 2018: 550-560.

[17] Rudolph G, Naujoks B, Preuss M. Capabilities of EMOA to detect and preserve equivalent Pareto subsets [C]// Evolutionary Multi-Criterion Optimization: 4th International Conference, EMO 2007, Matsushima, Japan, March 5-8, 2007. Proceedings 4. Springer Berlin Heidelberg, 2007: 36-50.

[18] Zhou A, Zhang Q, Jin Y. Approximating the set of Pareto-optimal solutions in both the decision and objective spaces by an estimation of distribution algorithm[J]. IEEE transactions on evolutionary computation, 2009, 13(5): 1167-1189.

[19] Yue C, Qu B, Yu K, et al. A novel scalable test problem suite for multimodal multiobjective optimization [J]. Swarm and Evolutionary Computation, 2019, 48: 62-71.