MOMPA: a high performance multi-objective optimizer based on marine predator algorithm

Long Chen
College of Computer Science & Artificial Intelligence
Wenzhou University, Wenzhou, China

Xuebing Cai
College of Computer Science & Artificial Intelligence
Wenzhou University, Wenzhou, China

Kezhong Jin
College of Computer Science & Artificial Intelligence
Wenzhou University, Wenzhou, China

Zhenzhou Tang*
College of Computer Science & Artificial Intelligence
Wenzhou University, Wenzhou, China

ABSTRACT

We propose a novel and effective multi-objective marine predator algorithm (MOMPA) to solve multi-objective optimization (MOO) problems. MOMPA incorporates the non-dominated sorting approach and the reference point strategy to select elite individuals and ensures the diversity of the Pareto optimal solution sets. Also, the Gaussian perturbation mechanism is leveraged to further improve the population diversity and global search ability in MOMPA. The performance of MOMPA is evaluated and comprehensively compared with benchmark functions. The results show that MOMPA is very competitive.

CCS CONCEPTS

• **Computing methodologies** → *Concurrent computing methodologies*;

KEYWORDS

Multi-objective optimization, swarm intelligent optimization algorithms, marine predator algorithm

ACM Reference Format:

Long Chen, Xuebing Cai, Kezhong Jin, and Zhenzhou Tang. 2021. MOMPA: a high performance multi-objective optimizer based on marine predator algorithm. In *2021 Genetic and Evolutionary Computation Conference Companion (GECCO '21 Companion), July 10–14, 2021, Lille, France.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3449726.3459581

1 INTRODUCTION

Multi-objective swarm intelligent optimization algorithms (MO-SIOAs) have been proved to be promising approaches to solve multi-objective optimization (MOO) problems since they are capable of obtaining a set of Pareto optimal solutions in one single simulation run for quite complicated MOO problems even if they are NP-hard, discontinuous, non-convex. Considering that the performance of a MO-SIOA highly depends on the core optimizer (typically is a single objective SIOA) and the elites selection (ES) method, this paper proposes a novel MO-SIOA, called MOMPA, which takes the marine predator algorithm (MPA)[5] as the basic optimizer for its outstanding global searchability and the reference point strategy[3] as the ES method for the remarkable ability to achieve excellent solution diversity and distribution. As far as we know, this is the first MPA-based MOO algorithm. Besides, the Gaussian perturbation mechanism is introduced to MOMPA to further enhance the population diversity as well as the global searchability. The source code is available on https://github.com/da-da-chen/MOMPA.

*Corresponding author

GECCO '21 Companion, July 10–14, 2021, Lille, France

ACM ISBN 978-1-4503-8351-6/21/07.
https://doi.org/10.1145/3449726.3459581

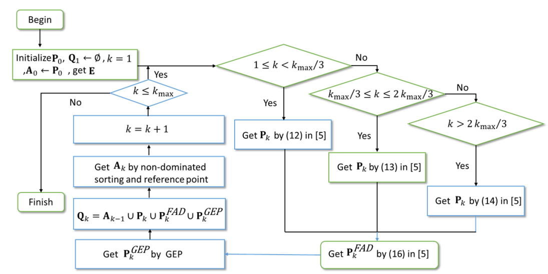

Figure 1: The flow chart of MOMPA

2 METHODOLOGY

Definition 1. *Archive* ($\mathbf{A}$). *The archive* $\mathbf{A}$ *is a set that is used to store the* $N_a = \binom{m+p-1}{p}$ *best solutions obtained in each iteration where* m *is the number of optimization objectives and* p *is the number of divisions on each objective in the normalized hyperplane.*

Definition 2. *File* ($\mathbf{Q}$). *The file* $\mathbf{Q}$ *is a set that is used to store all individuals generated in each iteration. Individuals in* $\mathbf{Q}$ *are candidates for elite selection.*

The main steps of MOMPA can be summarized as follows.

Step1: Initialize population $\mathbf{P}_0$, $\mathbf{Q}_1 \leftarrow \emptyset$, $\mathbf{A}_0 \leftarrow \mathbf{P}_0$.

Step2: Randomly select an individual from $\mathbf{A}_0$ duplicated Na times to construct predator matrix $\mathbf{E}$ (refer to (10) in [5]).

Step3: In the kth iteration, $\mathbf{P}_k$ is generated from $\mathbf{P}_{k-1}$ according to the three-stage evolution proposed in MPA.

Step4: Generate $\mathbf{P}_k^{\mathrm{FAD}}$ from $\mathbf{P}_k$ fish aggregating devices (FADs) elitism effect (refer to (16) in [5]).

Step5: For each individual in $\mathbf{P}_k^{\mathrm{FAD}}$, randomly select a dimension j and recalculate its value as $X_{i,j}^{\mathrm{GEP}} \leftarrow X_{i,j}^{\mathrm{FAD}} + \left(X_{\max,j} - X_{\min,j}\right) G$, where $G \sim N(0, 1)$, $X_{\max,j}$ and $X_{\min,j}$ are the upper and lower

Table 1: Comparison results

Algrithm	ZDT1 Avg.	ZDT1 Std.	ZDT1 Rank	ZDT2 Avg.	ZDT2 Std.	ZDT2 Rank	ZDT3 Avg.	ZDT3 Std.	ZDT3 Rank
NSGAII	4.82E-03	1.75E-04	4	4.86E-03	1.94E-04	4	7.35E-03	7.40E-03	4
NSGAIII	3.91E-03	1.20E-05	2	3.86E-03	2.78E-05	2	7.03E-03	5.23E-03	3
MOEAD	1.19E-02	8.49E-03	6	2.60E-02	2.70E-02	6	3.03E-02	2.10E-02	6
PESAII	1.15E-02	3.44E-03	5	1.14E-02	1.83E-03	5	2.06E-02	1.42E-02	5
CMOPSO	4.19E-03	8.96E-05	3	4.13E-03	8.86E-05	3	**4.64E-03**	6.08E-05	1
NSLS	2.34E-01	2.42E-02	7	4.05E-01	5.69E-02	7	2.27E-01	3.81E-02	7
MOMPA	**3.90E-03**	7.58E-05	1	**3.80E-03**	8.92E-06	1	6.30E-03	3.16E-04	2
	ZDT4			ZDT6			WFG4		
Algrithm	Avg.	Std.	Rank	Avg.	Std.	Rank	Avg.	Std.	Rank
NSGAII	5.41E-03	8.81E-04	2	3.72E-03	1.13E-04	4	2.73E-01	9.96E-03	6
NSGAIII	1.28E-02	1.73E-02	3	3.21E-03	2.75E-04	3	2.21E-01	3.05E-05	3
MOEAD	2.09E-02	1.24E-02	5	7.00E-03	1.16E-03	6	2.47E-01	2.09E-03	4
PESAII	1.34E-02	3.10E-03	4	7.43E-03	7.99E-04	7	2.93E-01	1.57E-02	7
CMOPSO	2.60E-01	2.54E-01	6	3.11E-03	2.76E-05	2	2.60E-01	4.05E-03	5
NSLS	8.26E-01	2.35E-01	7	6.23E-03	1.87E-03	5	**2.13E-01**	2.40E-03	1
MOMPA	**3.90E-03**	5.87E-05	1	**3.00E-03**	1.01E-05	1	2.14E-01	8.59E-04	2
	WFG5			WFG6			WFG7		
Algrithm	Avg.	Std.	Rank	Avg.	Std.	Rank	Avg.	Std.	Rank
NSGAII	2.80E-01	9.50E-03	7	3.03E-01	1.80E-02	6	2.83E-01	1.16E-02	6
NSGAIII	2.30E-01	9.22E-06	3	2.34E-01	6.99E-03	2	2.21E-01	1.67E-05	2
MOEAD	2.47E-01	1.76E-03	4	2.68E-01	1.11E-02	4	2.44E-01	1.61E-03	4
PESAII	2.78E-01	9.47E-03	6	3.11E-01	1.58E-02	7	2.87E-01	1.36E-02	7
CMOPSO	2.50E-01	5.05E-03	5	2.37E-01	4.58E-03	3	2.33E-01	4.65E-03	3
NSLS	**2.16E-01**	2.24E-03	1	**2.15E-01**	3.04E-03	1	2.70E-01	7.57E-03	5
MOMPA	2.21E-01	3.50E-03	2	2.70E-01	6.02E-02	5	**2.14E-01**	7.35E-04	1
	WFG8			WFG9			DTLZ1		
Algrithm	Avg.	Std.	Rank	Avg.	Std.	Rank	Avg.	Std.	Rank
NSGAII	3.74E-01	1.04E-02	6	2.76E-01	1.39E-02	7	2.74E-02	1.34E-03	6
NSGAIII	**2.78E-01**	3.42E-03	1	2.21E-01	5.77E-04	3	2.06E-02	2.71E-06	2
MOEAD	2.97E-01	2.00E-03	4	2.48E-01	2.03E-02	5	**2.06E-02**	6.79E-07	1
PESAII	3.78E-01	1.61E-02	7	2.76E-01	2.69E-02	6	2.47E-02	1.43E-03	5
CMOPSO	3.31E-01	5.59E-03	5	2.19E-01	3.35E-03	2	2.07E-02	3.74E-04	3
NSLS	2.84E-01	4.49E-03	2	2.40E-01	5.68E-03	4	2.39E-01	1.69E-01	7
MOMPA	2.87E-01	4.10E-03	3	**2.12E-01**	5.32E-04	1	2.09E-02	8.83E-04	4
	DTLZ2			DTLZ3			DTLZ4		
Algrithm	Avg.	Std.	Rank	Avg.	Std.	Rank	Avg.	Std.	Rank
NSGAII	6.94E-02	2.62E-03	7	6.83E-02	2.90E-03	3	6.82E-02	2.58E-03	3
NSGAIII	5.45E-02	7.63E-07	3	**5.45E-02**	8.71E-06	1	1.20E-01	1.68E-01	5
MOEAD	5.45E-02	5.07E-08	3	5.45E-02	1.50E-05	2	5.45E-02	7.92E-06	2
PESAII	6.73E-02	3.70E-03	6	7.18E-02	9.97E-03	4	9.44E-02	1.61E-01	4
CMOPSO	5.76E-02	9.24E-04	5	3.66E+00	3.92E+00	7	2.09E-01	3.35E-01	7
NSLS	5.42E-02	7.03E-04	2	2.63E+00	1.21E+00	5	1.54E-01	1.09E-01	6
MOMPA	**5.25E-02**	2.32E-05	1	2.80E+00	9.99E-01	6	**5.25E-02**	3.64E-05	1

bounds of the j-dimensional variables, respectively. The new population is denoted as $\mathbf{P}_k^{\mathrm{FAD}}$.

Step6: Generate $\mathbf{Q}_k \leftarrow \mathbf{A}_{k-1} \cup \mathbf{P}_k \cup \mathbf{P}_k^{\mathrm{FAD}} \cup \mathbf{P}_k^{\mathrm{GEP}}$.

Step7: Generate $\mathbf{A}_k$ on the basis of $\mathbf{Q}_k$ by the non-dominated sorting approach [4] and the reference point strategy [3].

Step8: Repeat from step 3 to 7 until the maximum number of iterations is reached.

Figure.1 shows the flow chart of MOMPA.

Next, we analyze the computational complexity of MOMPA. We use n to represent the size of the population, d to represent the dimension of the decision variable, and m to represent the number of objectives. In Step3, the three-stage evolutionary complexity of the population is $O(n \cdot d)$. In Step4, the complexity of the FADs effect is $O(n \cdot d)$. In Step5, Gaussian disturbance complexity is $O(n)$. In Step7, the complexity of the non-dominated sorting and the reference point-based elites selecting are $O(m \cdot n^2)$. We use cof_1 to represent the computational complexity of the fitness function, so the computational complexity of MOMPA is $O(k_{max}(n \cdot d + n + cof_1 \cdot n + m \cdot n^2))$.

Table 2: Score ranking and Wilcoxon signed-rank results

Algorithms	MOMPA	NSGAII	NSGAIII	MOEAD	PESAII	CMOPSO	NSLS
Score	32	75	38	62	85	60	67
Ranking	1	6	2	4	7	3	5
Wilcoxon signed-rank test(+\=\-)		13\1\1	11\2\2	13\2\0	14\1\0	11\3\1	10\2\3

3 NUMERICAL RESULTS

We compared MOMPA with NSGA-II [4], NSGA-III [3], MOEA/D [7], PESA-II [2], CMOPSO [8], NSLS [1]. The parameters of the comparison algorithms follow the default settings in [6]. All the algorithms were implemented in MATLAB 2019b and the performance evaluations were conducted on a server with an Intel Xeon E5-2620 3.0 GHz CPU, 64 GB RAM, and a Windows Server 2019 operating system. The population size is 100, the maximum number of iterations for the ZDT suite is 300, and that for the other suites is 3000. Table 1 shows the comparison results on the inverted generational distance (IGD). Table 2 summarizes the results on the total scores, the final ranking, and the Wilcoxon signed-rank test, respectively. An algorithmś score is derived from its ranking on each benchmark function. In the results of the Wilcoxon signed-rank test, the number of plus signs, minus signs, and equal signs represent the number of benchmark functions in which MOMPA is superior to, inferior to, and non-significantly different from the counterpart MO-SIOAs, respectively. It can be observed that MOMPA outperforms all the others.

ACKNOWLEDGMENTS

This work was supported in part by the Zhejiang Provincial Natural Science Foundation of China under Grants No. LZ20F010008, in part by Fundamental Scientific Research Project of Wenzhou City under Grants No. G20180008.

REFERENCES

[1] Bili Chen, Wenhua Zeng, Yangbin Lin, and Defu Zhang. 2014. A new local search-based multiobjective optimization algorithm. *IEEE transactions on evolutionary computation* 19, 1 (2014), 50–73.

[2] David W Corne, Nick R Jerram, Joshua D Knowles, and Martin J Oates. 2001. PESA-II: Region-based selection in evolutionary multiobjective optimization. In *Proceedings of the 3rd annual conference on genetic and evolutionary computation.* 283–290.

[3] Kalyanmoy Deb and Himanshu Jain. 2013. An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part I: solving problems with box constraints. *IEEE transactions on evolutionary computation* 18, 4 (2013), 577–601.

[4] Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and TAMT Meyarivan. 2002. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE transactions on evolutionary computation* 6, 2 (2002), 182–197.

[5] Afshin Faramarzi, Mohammad Heidarinejad, Seyedali Mirjalili, and Amir H Gandomi. 2020. Marine Predators Algorithm: A nature-inspired metaheuristic. *Expert Systems with Applications* 152 (2020), 113377.

[6] Ye Tian, Ran Cheng, Xingyi Zhang, and Yaochu Jin. 2017. PlatEMO: A MATLAB platform for evolutionary multi-objective optimization [educational forum]. *IEEE Computational Intelligence Magazine* 12, 4 (2017), 73–87.

[7] Qingfu Zhang and Hui Li. 2007. MOEA/D: A multiobjective evolutionary algorithm based on decomposition. *IEEE Transactions on evolutionary computation* 11, 6 (2007), 712–731.

[8] Xingyi Zhang, Xiutao Zheng, Ran Cheng, Jianfeng Qiu, and Yaochu Jin. 2018. A competitive mechanism based multi-objective particle swarm optimizer with fast convergence. *Information Sciences* 427 (2018), 63–76.