Expert Systems With Applications 189 (2022) 116126

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

Guided Manta Ray foraging optimization using epsilon dominance for multi-objective optimization in engineering design

Djaafar Zouache [a,b,*], Fouad Ben Abdelaziz [c]

[a] Department of Computer Science, University of Mohamed El Bachir El Ibrahimi, Bordj Bou Arreridj, Algeria
[b] LRIA Laboratory, University of Science and Technology Houari Boumediene, Algiers, Algeria
[c] Neoma Business School, Rouen Campus, France

ARTICLE INFO

Keywords:
Multi-objective optimization
Pareto set approximation
Epsilon-dominance
Manta Ray foraging optimization
Engineering design

ABSTRACT

In recent decades, metaheuristics have proven their effectiveness in solving large-scale real-world problems with multiple objectives. However, we still need to design robust algorithms capable of converging and approximating efficiently the true Pareto set. In this paper, we extend the recently Manta Ray foraging optimization (MOMRFO) to the multiobjective case. MOMRFO uses a population archive to store the non-dominated solutions generated so far by the exploration process. The leader's solutions are selected from the population archive to guide the Manta Rays population towards promising search regions. We use crowding distance and ϵ-dominance to provide a good compromise between diversity and convergence of the obtained potential Pareto set.

The proposed algorithm is validated on five bi-objective test functions, seven three objective test functions, and is applied to structural design problems such as four-bar truss design, speed reduced design, welded beam design, and disk brake design. The algorithm is compared with four well-known multi-objective meta-heuristics. The experimental results show that the MOMRFO algorithm outperforms against the selected multiobjective meta-heuristics by providing better convergence behaviour with a better diversity of solutions.

1. Introduction

When addressing a complex engineering problem, like structural design, the modeller is often facing multiple objectives and non-linear constraints. These objectives are often contradictory but need to be optimized simultaneously. In these situations, there is no single optimal solution but a set of non dominated solutions called Pareto set or Pareto Front. The Pareto set contains solutions with the best possible trade-offs between the objectives in a way that no objective can be improved without deteriorating the other objectives.

To solve multi-objective optimization problems (MOP), there are three main approaches according to the preferences of the decision-maker (DM), namely, a priory, a posteriory, and interactive approaches. In a priory approach, the DMs set their preferences before the start of the optimization process. These preferences are modelled by a utility function that aggregates all objectives in one. The MOP can be then solved as a single-objective optimization problem. We can cite in this the weighted sum method for example. The drawbacks of the priory approach for solving multi-objective optimization problems are the difficulty of explicating the preferences of the DM.

In the interactive approach, the DM chooses a compromise solution among locally/partially generated Pareto set. This category includes all interactive methods, such as STEM or Steuer and Choo methods. These approaches can be based on initial estimates and derivative calculations. They suffer from stagnation in local optima, and they are not suitable for solving a wide variety of MOP.

In a posteriory approach, all the Pareto set (PS) is tentatively generated. It can be proven that any rational DM with an increasing utility function would select a solution from the PS. However, The generation of the whole PS is highly demanding in terms of computational time. For the 3 last decades, meta-heuristics has been used to overcome the complexity of MOP as they are capable of providing a good approximation of the PS within a reasonable amount of time. Metaheuristics often mimic successful characteristics of nature, particularly biological systems and can provide acceptable solutions to complex optimization problems.

In the literature of multi-objective meta-heuristics, there are mainly three categories of methods: (1) Pareto dominance-based meta-heuristics, (2) indicator-based meta-heuristics and (3) Decomposition-based meta-heuristics. In the first category, the focus is on relaxing or modifying the Pareto dominance relation to guarantee a high convergence towards the Pareto set. For instance, the NSGAII 'Non-dominated

* Corresponding author at: Department of Computer Science, University of Mohamed El Bachir El Ibrahimi, Bordj Bou Arreridj, Algeria.
E-mail addresses: djaafarzouache@yahoo.fr (D. Zouache), fouad.ben.abdelaziz@neoma-bs.fr (F.B. Abdelaziz).

https://doi.org/10.1016/j.eswa.2021.116126
Received 29 April 2021; Received in revised form 20 September 2021; Accepted 17 October 2021
Available online 30 October 2021
0957-4174/

List of Abbreviations

MOP	Multi-objective Optimization Problems
DM	Decision-Maker
MRFO	Manta Ray foraging optimization
MOMRFO	Multi-objective Manta Ray foraging optimization
PS	Pareto set
MOEAs	Multi-Objective Evolutionary Algorithms
P(t)	Current population
t	Current iteration
Ub	Uper bound
Lb	Lower bound
S	Somersault factor
T	Maximal number iteration
Xbest	Best solution
Tmax	Maximum archive size
MOEA/D	Multi-objective evolutionary algorithm based on decomposition
MOGWO	Multi-Objective Grey Wolf Optimizer
MOPSO	Multi-objective Particle Swarm Optimization
MSSA	Multi-objective Salp Swarm Algorithm
ZDTn	Bi-objective test functions with n=1 to 6
DTLZn	Three-objective test functions with n=1 to 7
IGD	Inverted Generational Distance
HV	Hyper-Volume

sorting genetic algorithm' uses the Pareto dominance relation to select the elite solutions and uses the crowding distance to preserve the diversity of selected solutions (Deb, Pratap, Agarwal and Meyarivan, 2002). Several extensions have been proposed in the area of multi-objective meta-heuristics based on Pareto dominance relation [see for example Guo, Cheng, Luo, Gong, & Xue, 2017; Guo, Yang, Chen, Cheng and Gong, 2019; Guo, Zhang, Cheng, Wang, & Gong, 2018; Guo, Zhang, Gong, Zhang and Yang, 2019]. Other meta-heuristics modify the dominance relation to improve the convergence and the diversity of solutions. The first proposed modification to the dominance relation is the epsilon-dominance relation which extends the dominance region of a candidate solution x by modifying the objective values by $f_i(x) - \epsilon$ (Laumanns, Thiele, Deb, & Zitzler, 2002). This dominance relation is used in several multi-objective meta-heuristic such as the evolutionary algorithm based Epsilon-dominance to update and avoid the explosion of the archive, improve the diversity of the solutions (Cheng, Jin, & Hu, 2009; Deb, Mohan, & Mishra, 2005; Got, Moussaoui, & Zouache, 2020; Zouache, Abdelaziz, Lefkir and Chalabi, 2019; Zouache, Arby, Nouioua and Abdelaziz, 2019; Zouache, Moussaoui, & Abdelaziz, 2018). In the same way, we can cite alpha-dominance (Liu et al., 2017), cone-dominance (Batista, Campelo, Guimarães, & Ramírez, 2011), $(1-k)$-dominance (Farina & Amato, 2004), grid dominance (Yang, Li, Liu, & Zheng, 2013), fuzzy dominance (He, Yen, & Zhang, 2013), the controlling dominance area of solutions (CDAS) (Sato, Aguirre, & Tanaka, 2007), and Theta-dominance (Yuan, Xu, Wang, & Yao, 2015) methods. However, most of the proposed dominance relations aim to improve the convergence of MOEAs algorithms and may not achieve a good balance between convergence and diversity when solving MOPs.

The second category is the indicator-based meta-heuristics. It uses the performance metrics of diversity and convergence, such as IGD indicator, hypervolume indicator (HV) and S-metric, in the optimization process, to provide a good compromise between diversity and convergence. HV metric, for example, is widely used as a guide in the optimization process to converge towards the Pareto front. This metric is used in (SMS-EMOA) (Beume, Naujoks, & Emmerich, 2007; Brockhoff & Zitzler, 2007), HypE (Bader & Zitzler, 2011) and a new hypervolume-based Evolutionary Algorithm for many-objective optimization (Shang & Ishibuchi, 2020). Recently, some meta-heuristics have used other metrics, that seem to be effective in guiding towards the Pareto front, such as the meta-heuristics based IGD metric (Sun, Yen, & Yi, 2018), MOEAs based distance indicators (Wagner & Neumann, 2013) and MOEAs based R-metric (Brockhoff, Wagner, & Trautmann, 2015). The advantage of these methods is again the good balance between diversity and convergence.

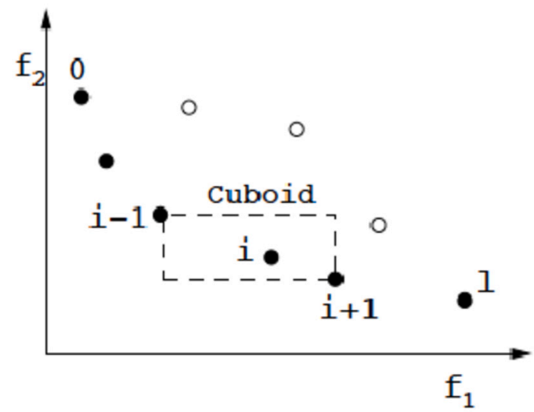

Fig. 1. The crowding distance calculation.

Table 1
Properties of multi-objective test functions.

Test function	Properties
Bi-objective test function	
ZDT1	Convex front
ZDT2	Non-convex front
ZDT3	Discontinuous front
ZDT4	This test function has 2^{21} local Pareto-optimal fronts and therefore is highly multi-modal.
ZDT6	This test function has a non-uniform search space
Three-objective test functions	
Test function	Characteristics
DTLZ1	Linear Pareto-optimal front
DTLZ2	Spherical Pareto-optimal front
DTLZ3	Many Pareto-optimal fronts
DTLZ4	Pareto-optimal front has dense set of solutions to exist near the $f_M - f_1$
DTLZ5	This problem is used to assess the ability of MOEA to converge to a degenerated curve.
DTLZ6	This problem has 2^{M-1} disconnected Pareto-optimal front.
DTLZ7	This problem has Pareto-optimal front which is a combination of a straight line and a hyper-plane.

The third category is the meta-heuristics based on decomposition where the main problem is decomposed into several scalar optimization sub-problems and simultaneously optimized using a set of weight vectors and scalarizing functions. The scalarizing functions commonly used are the Tchebychef method, the weighted sum, the boundary intersection, and vector angle distance scaling. MOEA/D is one of the most popular MOEAs algorithms based on decomposition. It explores the search space by a set of scalar subproblems (Zhang & Li, 2007). MOEAs was followed by several algorithms, MOEA/DD (Li, Deb, Zhang, & Kwong, 2014), NSGA-III (Deb & Jain, 2013), MOEA/D-DE (Li & Zhang, 2008), MOEA/D-M2M (Liu, Gu, & Zhang, 2013), etc. The meta-heuristics based on decomposition are the most effective for solving MOPs. However, all these methods are sensitive to the choice of the scalarizing function.

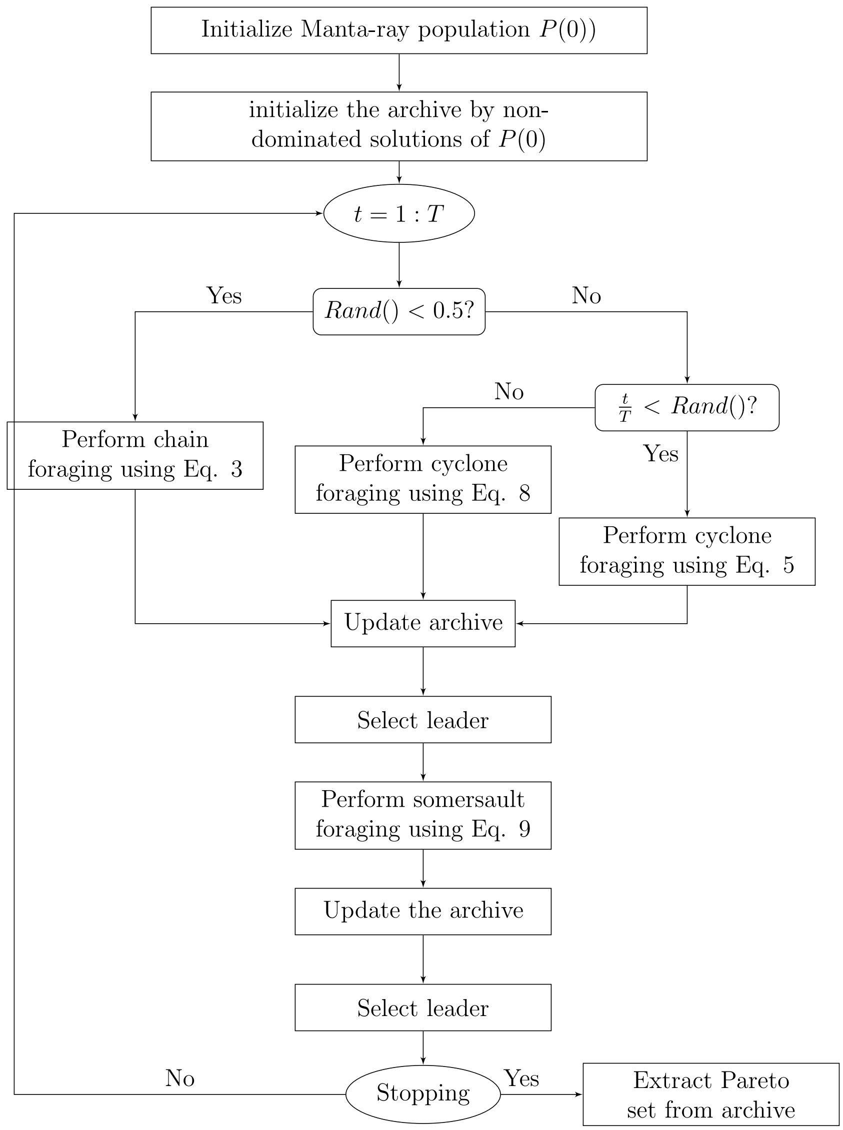

Fig. 2. Flowchart of the proposed MOMRFO algorithm.

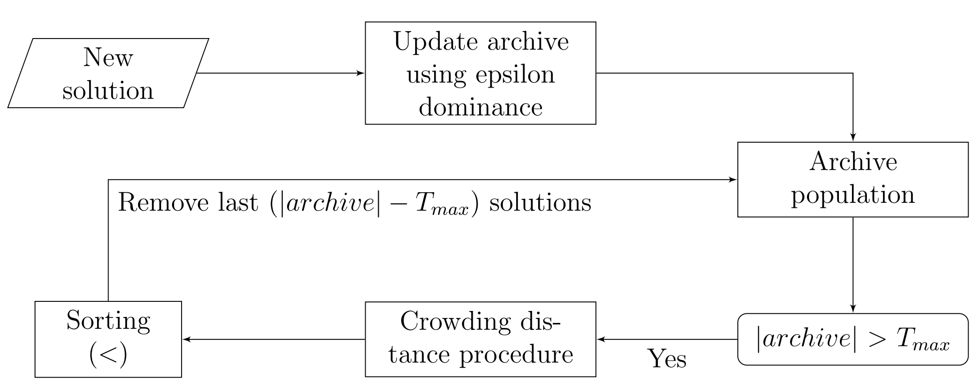

Fig. 3. Flowchart of the archive update strategy.

This study presents a new multi-objective version of the recently swarm intelligence algorithm called Manta-Ray foraging optimization (MRFO) aiming at providing a good compromise between diversity and convergence when generating the Pareto set. MRFO is a recent population-based metaheuristic that simulates the behaviour of Manta-Rays for solving single-objective problems. Despite its novelty, MRFO algorithm has attracted the attention of several researchers and it was applied with high success to various single-objective problems.

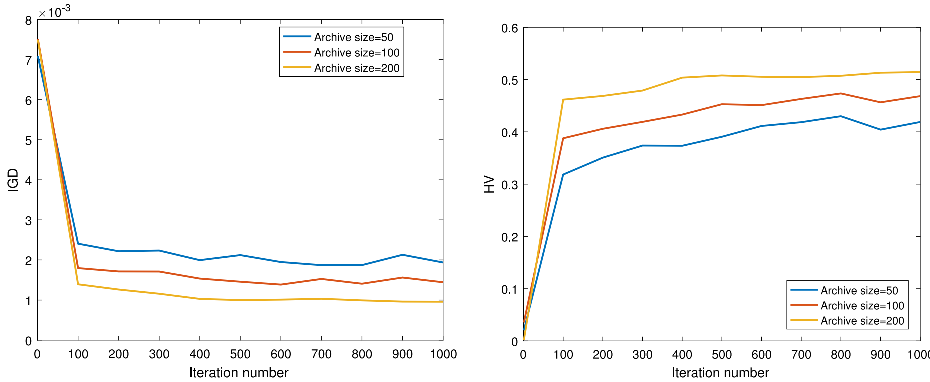

Fig. 4. Variation of IGD metric and HV metric and the IGD metric for DTLZ2 with three different archive sizes.

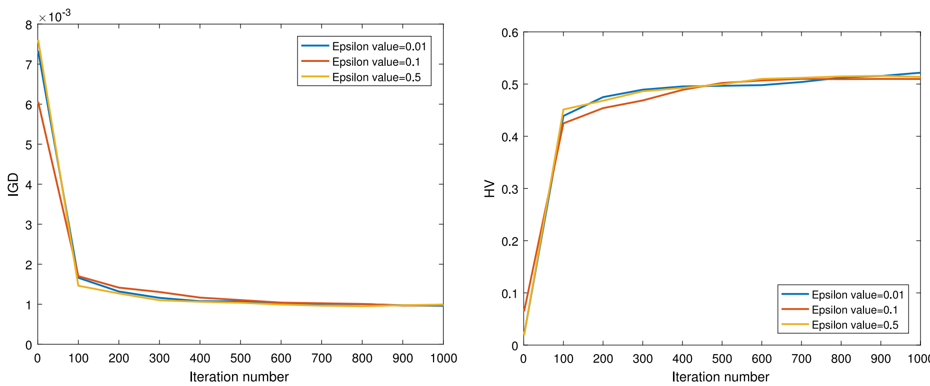

Fig. 5. Variation of IGD metric and HV metric and the IGD metric for DTLZ2 with three different epsilon values.

Table 2
Statistical results of IGD metric on ZDT test functions.

Test function	Statistical results	MOMRFO	MOGWO	MOEAD	MOPSO	MSSA
ZDT1	Best	**9,27E−05**	5,89E−04	5,41E−04	2,87E−04	3,15E−03
	Worst	**1,25E−04**	6,10E−03	3,14E−03	1,21E−03	7,20E−03
	Average	**1,01E−04**	1,50E−03	1,15E−03	4,22E−04	4,73E−03
	STD	**7,09E−06**	9,63E−04	7,65E−04	1,99E−04	1,06E−03
	Wilcoxon test	+ + ++	− +	− + ++	− + −+	− −
ZDT2	Best	**9,24E−05**	7,93E−04	3,99E−04	3,20E−04	3,65E−03
	Worst	**1,28E−04**	2,31E−02	6,24E−02	2,31E−02	1,16E−02
	Average	**1,03E−04**	6,45E−03	2,20E−02	1,94E−02	5,30E−03
	STD	**7,38E−06**	9,16E−03	1,54E−02	8,50E−03	1,43E−03
	Wilcoxon test	+ + ++	− + ++	− −	− −	− ++
ZDT3	Best	7,94E−03	**3,06E−04**	6,03E−03	1,81E−03	4,50E−03
	Worst	8,28E−03	**3,26E−03**	2,39E−02	1,41E−02	1,16E−02
	Average	8,13E−03	**9,66E−04**	1,04E−02	7,49E−03	7,18E−03
	STD	1,29E−04	**5,93E−04**	3,07E−03	3,14E−03	1,78E−03
	Wilcoxon test	− + −	+ + ++	+ − +−	−−	+ − +
ZDT4	Best	**9,66E−05**	2,89E−02	2,03E−01	2,14E−02	6,93E−02
	Worst	**1,12E−04**	7,19E−01	2,56E+00	6,51E−01	5,01E−01
	Average	**1,02E−04**	2,88E−01	1,16E+00	1,78E−01	1,94E−01
	STD	**3,95E−06**	2,09E−01	6,88E−01	1,33E−01	9,52E−02
	Wilcoxon test	+ + ++	− + −	− + +−	−−	− +
ZDT6	Best	**6,73E−05**	2,88E−04	3,52E−04	2,48E−04	6,68E−04
	Worst	**1,45E−04**	2,99E−03	1,69E−01	7,09E−04	4,21E−03
	Average	**8,09E−05**	1,45E−03	3,35E−02	3,54E−04	2,05E−03
	STD	**1,45E−05**	8,96E−04	4,86E−02	1,11E−04	1,00E−03
	Wilcoxon test	+ + ++	− ++	− + ++	− −	− −

Best results are marked in bold.

However, it has not been applied yet in multi-objective optimization problems. Furthermore, the intelligent foraging strategies used by MRFO algorithm can improve its ability to explore the search space and increase the chance to get well-distributed solutions, which is one of the challenges in multi-objective optimization. Hence, the major's contributions of this study are as follows:

Table 3
Statistical results of HV metric on ZDT test functions.

Test function	Statistical results	MOMRFO	MOGWO	MOEAD	MOPSO	MSSA
ZDT1	Best	**7,21E−01**	7,09E−01	7,11E−01	7,14E−01	5,96E−01
	Worst	**7,22E−01**	6,48E−01	5,90E−01	7,05E−01	4,83E−01
	Average	**7,21E−01**	6,93E−01	6,85E−01	7,11E−01	5,48E−01
	STD	**1,00E−04**	1,09E−02	3,73E−02	2,37E−03	2,58E−02
	Wilcoxon test	+ + ++	− +	− + −+	− + ++	− −
ZDT2	Best	**7,18E−01**	4,23E−01	4,38E−01	4,37E−01	3,13E−01
	Worst	**4,46E−01**	9,09E−02	0,00E+00	9,09E−02	1,91E−01
	Average	**4,55E−01**	3,39E−01	7,52E−02	1,46E−01	2,61E−01
	STD	**4,88E−02**	1,37E−01	1,04E−01	1,28E−01	2,59E−02
	Wilcoxon test	+ + ++	− + ++	− −	− + −	− ++
ZDT3	Best	**8,04E−01**	6,00E−01	5,80E−01	7,80E−01	6,46E−01
	Worst	**6,96E−01**	5,48E−01	1,30E−01	6,13E−01	4,70E−01
	Average	**8,00E−01**	5,81E−01	4,22E−01	7,43E−01	5,55E−01
	STD	**1,93E−02**	1,07E−02	8,87E−02	4,95E−02	4,57E−02
	Wilcoxon test	+ + ++	− + −+	− −	− + ++	− + −
ZDT4	Best	**7,21E−01**	9,09E−02	0,00E+00	8,20E−02	0,00E+00
	Worst	**7,02E−01**	0,00E+00	0,00E+00	0,00E+00	0,00E+00
	Average	**7,20E−01**	5,86E−03	0,00E+00	3,01E−03	0,00E+00
	STD	**3,41E−03**	2,27E−02	0,00E+00	1,48E−02	0,00E+00
	Wilcoxon test	+ + ++	− −	− −	− −	− −
ZDT6	Best	**3,90E−01**	3,85E−01	3,83E−01	3,86E−01	3,76E−01
	Worst	**1,93E−02**	3,26E−01	0,00E+00	3,78E−01	2,52E−01
	Average	**3,78E−01**	3,59E−01	2,08E−01	3,84E−01	3,38E−01
	STD	**6,66E−02**	1,95E−02	1,76E−01	1,88E−03	3,19E−02
	Wilcoxon test	+ + ++	− + −+	− −	− + ++	− −

Best results are marked in bold.

Table 4
Statistical results of IGD metric on DTLZ test functions.

Test function	Statistical results	MOMRFO	MOGWO	MOEAD	MOPSO	MSSA
DTLZ1	Best	**2,59E−03**	3,49E−02	3,71E−02	2,07E−02	6,01E−03
	Worst	**1,03E−01**	2,01E−01	2,68E−01	1,21E−01	2,70E−01
	Average	**3,74E−02**	1,25E−01	1,58E−01	6,77E−02	6,39E−02
	STD	**2,48E−02**	3,96E−02	5,84E−02	2,49E−02	8,12E−02
	Wilcoxon test	+ + +−	− + −	− + +−	− −	− + ++
DTLZ2	Best	**9,06E−04**	6,69E−03	1,32E−03	2,87E−03	5,28E−03
	Worst	**1,02E−03**	1,03E−02	1,82E−03	5,02E−03	8,21E−03
	Average	**9,62E−04**	7,88E−03	1,58E−03	4,06E−03	7,32E−03
	STD	**2,92E−05**	8,98E−04	1,31E−04	5,46E−04	6,18E−04
	Wilcoxon test	+ + ++	− −	− + −+	− + ++	− + −
DTLZ3	Best	**9,58E−02**	1,70E+00	4,08E−01	4,61E−01	1,01E−01
	Worst	**8,36E−01**	2,97E+00	1,71E+00	2,35E+00	2,37E+00
	Average	**4,15E−01**	2,78E+00	1,16E+00	1,47E+00	1,56E+00
	STD	**2,18E−01**	2,22E−01	3,62E−01	5,01E−01	8,08E−01
	Wilcoxon test	+ + ++	− −	− + −	− + ++	− + −
DTLZ4	Best	**1,20E−03**	1,76E−03	1,36E−03	1,34E−03	4,57E−03
	Worst	**2,68E−03**	2,80E−03	1,43E−02	1,39E−02	1,02E−02
	Average	**1,87E−03**	2,20E−03	5,52E−03	7,19E−03	6,80E−03
	STD	**4,05E−04**	2,62E−04	4,34E−03	4,06E−03	1,27E−03
	Wilcoxon test	+ + ++	− + −+	− +	− −	− −
DTLZ5	Best	**8,62E−05**	8,71E−04	2,01E−04	1,67E−04	6,53E−04
	Worst	**1,78E−04**	2,40E−03	7,83E−04	4,81E−04	3,72E−03
	Average	**1,14E−04**	1,50E−03	4,41E−04	2,58E−04	1,55E−03
	STD	**2,24E−05**	4,23E−04	1,59E−04	9,25E−05	6,05E−04
	Wilcoxon test	+ + ++	− −	− + ++	− + −+	− −
DTLZ6	Best	**4,35E−05**	7,58E−04	2,49E−04	5,19E−02	6,17E−04
	Worst	**7,80E−05**	7,31E−03	1,09E−03	9,70E−02	5,75E−03
	Average	**5,11E−05**	2,19E−03	5,43E−04	6,94E−02	1,84E−03
	STD	**6,39E−06**	1,44E−03	1,83E−04	1,38E−02	1,26E−03
	Wilcoxon test	+ + ++	− + −	− −	− + ++	− + −
DTLZ7	Best	**6,27E−04**	2,71E−03	2,69E−03	1,09E−02	6,72E−03
	Worst	**8,08E−04**	9,95E−03	6,52E−02	1,17E−02	2,52E−02
	Average	**7,28E−04**	5,58E−03	2,42E−02	1,13E−02	1,12E−02
	STD	**5,07E−05**	2,07E−03	1,52E−02	2,07E−04	3,76E−03
	Wilcoxon test	+ + ++	− + ++	− + −	− −	− +

Best results are marked in bold.

Table 5
Statistical results of HV metric on DTLZ test functions.

Test function	Statistical results	MOMRFO	MOGWO	MOEAD	MOPSO	MSSA
DTLZ1	Best	**4,22E−01**	0,00E+00	0,00E+00	0,00E+00	1,36E−02
	Worst	**0,00E+00**	0,00E+00	0,00E+00	0,00E+00	0,00E+00
	Average	**4,54E−02**	0,00E+00	0,00E+00	0,00E+00	4,37E−04
	STD	**1,17E−01**	0,00E+00	0,00E+00	0,00E+00	2,44E−03
	Wilcoxon test	+ + ++	− −	− −	− −	− −
DTLZ2	Best	**5,21E−01**	1,75E−01	5,03E−01	3,41E−01	1,90E−01
	Worst	**5,05E−01**	1,15E−01	4,48E−01	2,80E−01	1,07E−01
	Average	**5,15E−01**	1,30E−01	4,79E−01	3,08E−01	1,35E−01
	STD	**4,37E−03**	1,33E−02	1,31E−02	1,56E−02	2,16E−02
	Wilcoxon test	+ + ++	− −	− + ++	− + −+	− −
DTLZ3	Best	0,00E+00	0,00E+00	0,00E+00	0,00E+00	0,00E+00
	Worst	0,00E+00	0,00E+00	0,00E+00	0,00E+00	0,00E+00
	Average	0,00E+00	0,00E+00	0,00E+00	0,00E+00	0,00E+00
	STD	0,00E+00	0,00E+00	0,00E+00	0,00E+00	0,00E+00
	Wilcoxon test	− −	− −	− −	− −	− −
DTLZ4	Best	**5,28E−01**	4,50E−01	5,00E−01	5,12E−01	3,99E−01
	Worst	**4,67E−01**	3,65E−01	3,72E−02	9,09E−02	1,01E−02
	Average	**4,99E−01**	4,18E−01	3,85E−01	3,48E−01	1,90E−01
	STD	**1,37E−02**	1,95E−02	1,14E−01	1,55E−01	1,02E−01
	Wilcoxon test	+ + ++	− +	− +	− − +	− −
DTLZ5	Best	**1,99E−01**	1,74E−01	1,96E−01	1,97E−01	1,90E−01
	Worst	**1,97E−01**	1,26E−01	1,73E−01	1,70E−01	1,05E−01
	Average	**1,98E−01**	1,50E−01	1,90E−01	1,90E−01	1,44E−01
	STD	**5,14E−04**	1,38E−02	4,70E−03	6,72E−03	2,08E−02
	Wilcoxon test	+ + ++	− −	− + −+	− + −+	− −
DTLZ6	Best	**2,01E−01**	1,81E−01	1,95E−01	0,00E+00	1,82E−01
	Worst	**2,01E−01**	1,51E−02	1,82E−01	0,00E+00	9,91E−02
	Average	**2,01E−01**	1,20E−01	1,92E−01	0,00E+00	1,60E−01
	STD	**1,51E−04**	3,82E−02	2,64E−03	0,00E+00	2,31E−02
	Wilcoxon test	+ + ++	− + −	− + ++	− −	− + −+
DTLZ7	Best	**2,79E−01**	2,08E−01	1,42E−01	1,98E−01	1,40E−01
	Worst	**2,73E−01**	3,34E−02	0,00E+00	1,89E−01	0,00E+00
	Average	**2,76E−01**	9,56E−02	1,33E−02	1,94E−01	5,36E−02
	STD	**1,39E−03**	6,25E−02	3,34E−02	2,68E−03	3,07E−02
	Wilcoxon test	+ + ++	− + −+	− −	− + ++	− − +−

Best results are marked in bold.

Table 6
Statistical results of IGD metric for engineering design problems.

Test function	Statistical results	MOMRFO	MOGWO	MOEAD	MOPSO	MSSA
WBD	Best	**6,18E−03**	1,67E−02	6,90E−01	1,18E−02	2,41E−02
	Worst	**6,72E−03**	1,08E−01	9,51E−01	3,28E−02	3,03E−01
	Average	**6,40E−03**	4,47E−02	8,35E−01	2,04E−02	9,88E−02
	STD	**1,34E−04**	2,17E−02	7,03E−02	5,13E−03	7,85E−02
	Wilcoxon test	+ + ++	− ++	− + ++	− −	− +
SRD	Best	**1,03E+01**	1,03E+01	1,16E+01	9,61E+00	1,16E+01
	Worst	**1,27E+01**	1,34E+01	4,92E+10	2,35E+01	2,04E+01
	Average	**1,11E+01**	1,15E+01	2,83E+09	1,26E+01	1,62E+01
	STD	**5,57E−01**	7,23E−01	1,10E+10	3,75E+00	2,79E+00
	Wilcoxon test	+ − ++	− ++	− ++	− −	− +
Disk	Best	**1,31E−03**	5,05E−03	2,25E−02	4,16E−03	1,10E−02
	Worst	**1,70E−03**	1,22E−02	1,29E−01	2,26E−02	5,98E−02
	Average	**1,44E−03**	6,63E−03	6,76E−02	8,78E−03	2,34E−02
	STD	**9,06E−05**	1,56E−03	2,89E−02	5,74E−03	1,04E−02
	Wilcoxon test	+ + ++	− ++	++ −	− −	− +
Bar	Best	**7,84E−02**	2,36E−01	8,87E+00	2,35E−01	6,39E−01
	Worst	**9,85E−02**	5,63E−01	2,04E+01	1,27E+00	2,02E+00
	Average	**8,66E−02**	3,52E−01	1,67E+01	5,76E−01	1,02E+00
	STD	**5,27E−03**	8,67E−02	2,52E+00	3,24E−01	3,31E−01
	Wilcoxon test	+ + ++	− + ++	− ++	− −	− +

Best results are marked in bold.

- A Guided Archive population Manta-Ray foraging optimization based epsilon-dominance is proposed for large-scale multi-objective optimization.
- An archive population is introduced into the basic version of MRFO to store and update the so far generated non-dominated solutions.
- The leader solutions and archive population is used to guide the Manta Rays solutions toward the Pareto front.

Table 7
Statistical results of HV metric for engineering design problems.

Test function	Statistical results	MOMRFO	MOGWO	MOEAD	MOPSO	MSSA
WBD	Best	**8,44E−01**	8,30E−01	5,26E−01	8,40E−01	8,30E−01
	Worst	**8,42E−01**	8,14E−01	0,00E+00	8,22E−01	7,71E−01
	Average	**8,43E−01**	8,22E−01	5,59E−02	8,36E−01	8,17E−01
	STD	**3,09E−04**	3,94E−03	1,41E−01	3,73E−03	1,21E−02
	Wilcoxon test	+ + ++	− + −+	− −	− + ++	− + −
SRD	Best	**2,63E−01**	2,63E−01	2,63E−01	2,65E−01	2,61E−01
	Worst	**2,61E−01**	2,60E−01	0,00E+00	2,52E−01	2,56E−01
	Average	**2,62E−01**	2,62E−01	2,11E−01	2,63E−01	2,59E−01
	STD	**4,99E−04**	5,88E−04	8,15E−02	2,79E−03	1,78E−03
	Wilcoxon test	− + −+	− + −+	− −	+ + ++	− + −
Disk	Best	**7,66E−01**	7,61E−01	7,57E−01	7,61E−01	7,55E−01
	Worst	**7,65E−01**	7,56E−01	7,01E−01	7,56E−01	7,25E−01
	Average	**7,66E−01**	7,59E−01	7,37E−01	7,58E−01	7,43E−01
	STD	**7,28E−05**	1,03E−03	1,40E−02	1,23E−03	7,17E−03
	Wilcoxon test	+ + ++	− + ++	− −	− ++	− −
Bar	Best	**2,84E−01**	2,82E−01	1,68E−01	2,82E−01	2,72E−01
	Worst	**2,84E−01**	2,78E−01	0,00E+00	2,75E−01	2,61E−01
	Average	**2,84E−01**	2,81E−01	5,36E−02	2,79E−01	2,69E−01
	STD	**1,37E−05**	1,22E−03	4,21E−02	1,71E−03	2,41E−03
	Wilcoxon test	+ + ++	− + ++	− −	− ++	− + −

Best results are marked in bold.

Table 8
Runtime of algorithms for one execution (s).

	MOEA-D	MOGWO	MSSA	MOPSO	MOMRF0
Zdt1	440.0093	668.2665	**89.9722**	177.9549	162.5103
Zdt2	374.8618	67.2408	**55.2313**	205.3638	219.1736
Zdt3	371.5014	448.3240	**92.0169**	114.7108	180.7861
Zdt4	208.1556	76.3337	**43.3857**	218.5095	54.5312
Zdt6	370.6754	713.5653	**49.8646**	141.6082	137.0437
DTLZ1	405.8341	186.3638	56.8196	**42.5678**	51.5171
DTLZ2	444.8980	543.7021	222.9273	330.8720	**106.2344**
DTLZ3	433.9067	688.6034	78.0878	92.7621	**72.2486**
DTLZ4	428.1579	447.3627	150.7616	197.2002	**97.9203**
DTLZ5	424.8031	437.7445	**153.5962**	291.6511	180.7067
DTLZ6	453.2583	211.5742	**28.6178**	174.9252	186.4520
DTLZ7	292.8392	482.3534	**83.7363**	205.2021	119.2009
BAR	409.3450	409.8567	**85.9099**	177.3356	114.6190
DISK	562.8505	402.8977	**82.7433**	162.5823	121.8596
SRD	264.2190	197.1676	**90.5698**	123.1173	125.6093
WBD	567.3919	468.3748	**63.8661**	154.0572	123.9263

- The archive population is updated using the epsilon dominance to ensure a good diversity to the solutions population, obtain a good approximation of the Pareto set, avoid the explosion of the archive size and reduce the execution time of the MOMRFO algorithm.
- Leader's selection procedure based on crowding distance is used to improve the diversity of solutions population.
- The performance of MOMRFO is validated by different simulations on five bi-objective test functions, seven three-objective test functions and four real engineering applications.

The novelty and strength of the MOMRFO lie in the intelligent exploration of search space by using the foraging strategies of Manta Rays and the management of the objective space through the archive population, the epsilon dominance, the solution leaders and the crowding distance.

Following the introduction in 1, the remainder of the paper is organized as follows. Section 2 presents the fundamental concepts of MOP and briefly overviews the basics of MRFO. Section 3 presents the MOMRFO algorithm. The experiments and results are introduced in Section 4. Section 5 concludes the present study and provides some recommendations for future studies.

2. Background

2.1. Multi-objective optimization

In general, a multi-objective optimization problem (MOP) consists in optimizing (minimizing or maximizing) a set of M objective functions under a set of J inequality constraints and a set K of equality. the general form of a MOP is defined as follows:

$$\begin{array}{lll} Minimize: & f_m(x), & m = 1,2,\ldots,M \\ Subject\ to: & g_j(x) \geq 0, & j = 1,2\cdots,J \\ & h_k(x) = 0, & k = 1,2,\ldots,K \\ & L_i \leq x_i \leq U_i & i = 1,2,\ldots,n \end{array} \quad (1)$$

where x is a solution of n decision variables: $x = (x_1, x_2, \ldots, x_n)$, which satisfies J inequality constraints and K equality constraints. M is the objective number functions. L_i and U_i correspond respectively to the lower and upper limits of the decision variable. the set of all feasible solutions is denoted by S. The resolution of a MOP aims to generate the Pareto set (PS) or Pareto Front (PF).

Definition 1 (*Dominance Relation, Minimization Case*)**.** For two solutions $x^{(i)}$ and $x^{(j)}$, $x^{(i)}$ is said to Pareto dominates $x^{(j)}$ (denoted as $x^{(i)} \prec x^{(j)}$), if and only if:

- $\forall m \in \{1,\ldots,M\} : f_m(x^{(i)}) \leq f_m(x^{(j)})$

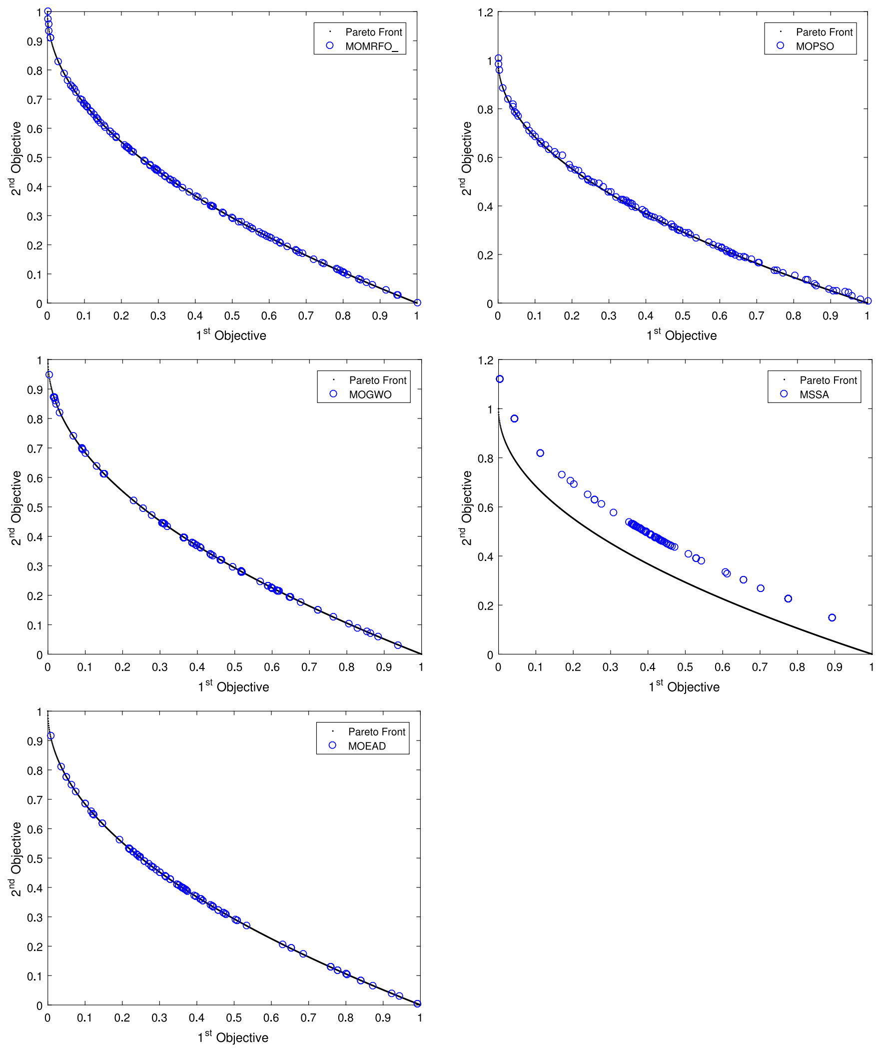

Fig. 6. Pareto optimal fronts obtained by MOMRFO, MOPSO, MOGWO, MSSA and MOEA/D for ZDT1 test function.

- $\exists m \in \{1, \ldots, M\} : f_m(x^{(i)}) < f_m(x^{(j)})$

Definition 2 (*ε−Dominance*)**.** For two solutions $x^{(i)}$ and $x^{(j)}$, $x^{(i)}$ is said to $\varepsilon-$dominate $x^{(j)}$, denoted as $x^{(i)} \prec_{\varepsilon} x^{(j)}$, if:

$$\forall k \in 1, \ldots, m : (1-\varepsilon_k)x_k^{(i)} \leq x_k^{(j)} \tag{2}$$

Definition 3 (*Non-Dominated Set*)**.** Given a solution set A, the set of non-dominated solutions A', where $A' \subseteq A$, is a set of all the solutions that are not dominated by any element of the solution set A.

Definition 4 (*Pareto-Optimal Set*)**.** The non-dominated set of the entire feasible decision space is called the Pareto-optimal set (Pareto front)

2.2. Crowding distance estimation

To maintain the distribution of generated solutions over the Pareto front, Deb et al. propose the crowding estimator named Crowding Distance (Deb, Pratap et al., 2002). The crowding distance of solution i, estimates the size of the largest cuboid containing the solution i without including any other solution. Firstly, The distance of each solution is set to 0. For each objective m, the solutions are sorted in ascending order according to objective function values m. The crowding distance value of each solution is the distance to its two nearest neighbours of the solution i. The extreme solutions which have the lowest and highest objective function values are assigned to an infinite distance value so that they are always selected. The crowding distance values of the solution i corresponding to each objective function are summed to obtain the final crowding distance value of i, as explained in Fig. 1

2.3. Manta Ray foraging optimization: A brief overview

Manta Ray foraging optimization (MRFO) is a recent swarm intelligence algorithm proposed by Zhao, Zhang, and Wang (2020) for solving single-objective optimization problems. MRFO mimics the three intelligent foraging strategies from Manta Rays, including chain, cyclone, and somersault foraging to solve single optimization problems. The main features of the Manta Rays are

- Manta Rays are one of the largest known marine creatures.
- Manta rays are mostly made up of microscopic water animals
- Manta Rays feed on plankton.

The Manta Rays have three intelligent foraging strategies.

(1) Chain foraging strategy
Manta rays observe the plankton position and swim towards it by forming an orderly line. So, the plankton that is missed by the previous manta rays will be devoured by the following manta rays. The higher the concentration of plankton in a position, the better that position is.

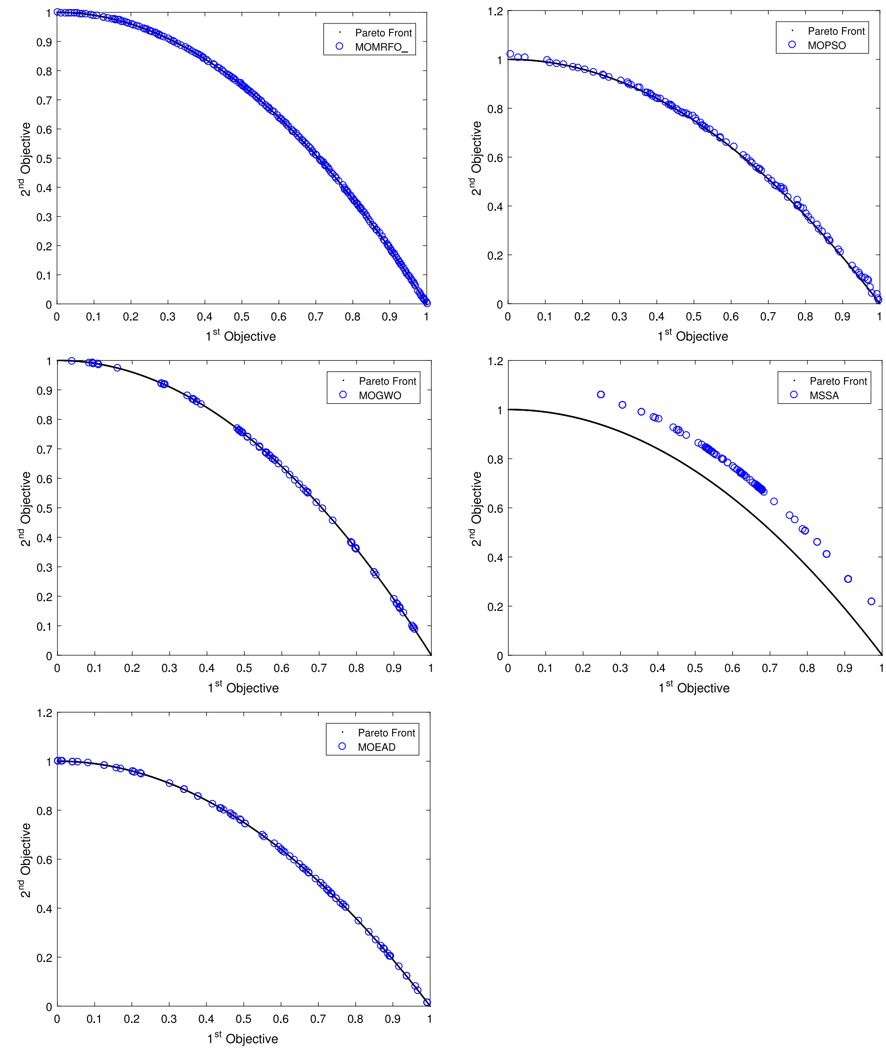

Fig. 7. Pareto optimal fronts obtained by MOMRFO, MOPSO, MOGWO, MSSA and MOEA/D for ZDT2 test function.

Manta Rays line up, one behind another, forming an orderly line to start the foraging. Smaller male Manta Rays are carried over the female to swim above her back to match the beats of pectoral fins of the female. Consequently, the plankton that has been missed by the previous Manta Rays will harvest it from behind them. In this way, they can hunt as many planktons as possible and improve their nutritional reward.

(2) **Cyclone foraging strategy**
The second foraging strategy is cyclone foraging. The Manta Rays gather together on the plankton group and their tail ends are connected with a helical head to generate a spiralling vortex in the eye of the cyclone, which causes the filtered water to be pushed towards the surface. With this, the planktons are pulled into the mouths of the Manta Rays.

(3) **Somersault foraging strategy**
The final foraging strategy is somersault foraging. Manta Rays make a series of backward somersaults when they find a food source, which helps optimize their food intake.

From an optimization point of view, MRFO is a population-based meta-heuristic with population $P(t)$ of N Manta Ray (individual) is evolved during each iteration t. The Manta Ray X_i is represented by a vector of D dimensions, where each dimension corresponds to a decision variable of the optimization problem. The new population $P(t+1)$ is updated by considering three foraging strategies of Manta Ray behaviour. Thus, each new position of the Manta Ray is updated by the flowing foraging strategies. The chain foraging behaviour significantly used to improve the ability of exploitation search. For the cyclone foraging behaviour is used to enhance the ability of the exploration search. Regarding, the somersault foraging behaviour is used to enhance the ability exploitation search and improve the convergence rate.

In chain foraging strategy, the Manta rays observe the plankton positions and move towards it by forming a foraging chain. Except the first Manta ray, the Manta rays swim towards both the best solution obtained so far and the solution in front of it. The mathematical model of this movement strategy is defined as follows:

$$x_i^k(t+1) = \begin{cases} x_i^k(t) + rand(x_{best}^k(t) - x_i^k(t)) \\ \quad +\alpha(x_{best}^k(t) - x_i^k(t)) & , i = 1 \\ x_i^k(t) + rand(x_{i-1}^k(t) - x_i^k(t)) \\ \quad +\alpha(x_{best}^k(t) - x_i^k(t)) & , i = 2, \dots, N \end{cases} \tag{3}$$

$$\alpha = 2.r.\sqrt{|log(r)|} \tag{4}$$

where, $x_i^k(t)$ is the position of ith individual in iteration t at kth dimension, r is a random number within the range of $[0,1]$, α is a weight coefficient, $x_{best}^k(t)$ is the position with highest plankton concentration.

In cyclone foraging, when a set of Manta Rays finds a plankton with high concentration, they will form a long foraging chain and move

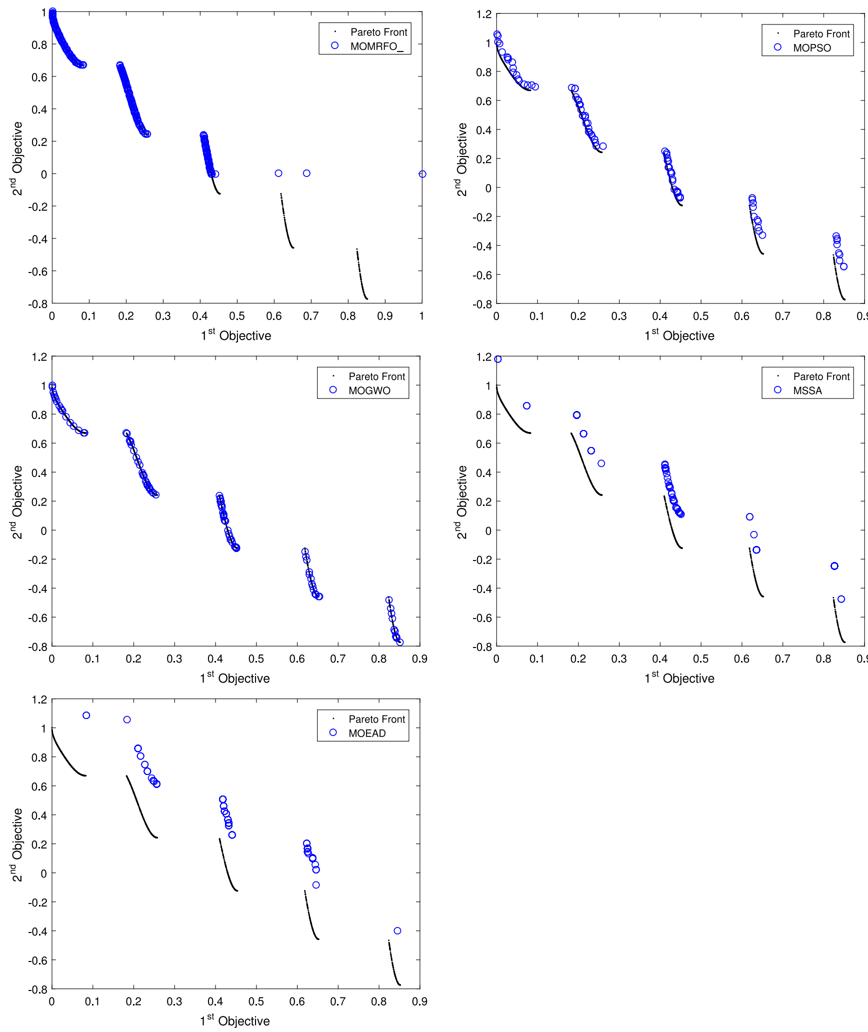

Fig. 8. Pareto optimal fronts obtained by MOMRFO, MOPSO, MOGWO, MSSA and MOEA/D for ZDT3 test function.

towards the food by a spiral movement. In this formation, each Manta-Ray moves towards the plankton position as well as the one in front of it. The mathematical movement cyclone foraging of Manta Rays can be defined as:

$$x_i^k(t+1) = \begin{cases} x_{best}^k(t) + r.(x_{best}^k(t) - x_i^k(t)) \\ \quad +\beta.(x_{best}^k(t) - x_i^k(t)) & , i = 1 \\ x_i^{best}(t) + r.(x_{i-1}^k(t) - x_i^k(t)) \\ \quad +\beta.(x_{best}^k(t) - x_i^k(t)) & , i = 2, \dots, N \end{cases} \tag{5}$$

$$\beta = 2e^{r\frac{T-t+1}{T}}.sin(2\pi r) \tag{6}$$

where β is the weight coefficient, T is the maximum number of iterations, and r_1 is the random number in $[0,1]$.

All Manta Rays randomly move towards the best plankton position, so this strategy paves the way to intensively explore fertile regions around the best position and provide a good exploitation capacity of the algorithm. The cyclone's foraging strategy also provides good exploration capabilities and helps guide the population of individuals to explore unvisited regions of the search space. In this strategy, we can force individuals to move to random positions which should be far from the best plankton position and their current position. The mathematical model of the exploration mechanism can be modelled as follows:

$$x_{rand}^k(t) = Lb^k + rand.(Ub^k - Lb^k) \tag{7}$$

$$x_i^k(t+1) = \begin{cases} x_{rand}^k(t) + r.(x_{rand}^k(t) - x_i^k(t)) \\ \quad +\beta.(x_{rand}^k(t) - x_i^k(t)) & , i = 1 \\ x_i^{rand}(t) + r.(x_{i-1}^k(t) - x_i^k(t)) \\ \quad +\beta.(x_{rand}^k(t) - x_i^k(t)) & , i = 2, \dots, N \end{cases} \tag{8}$$

where x_{rand}^k is a random position, Lb^k and Ub^k are the lower and upper limits of the kth dimension,respectively.

In the somersault foraging behaviour, each individual tends to swim to and from around the pivot and somersault to a new position. Therefore, they always update their positions around the best position x_{best} found so far. The somersault foraging behaviour can be modelled as follows:

$$x_i^k(t+1) = x_i^k(t) + S.(r_1.x_{best}^k(t) - r_2.x_i^k(t)), i = 1, \dots, N \tag{9}$$

where S is the somersault factor that decides the somersault range of Manta Rays and $S = 2$, r_1 and r_2 are two random numbers in $[0,1]$.

3. Guided archive population Manta-Ray foraging optimization for multi-objective optimization

In this section, we present the main phases of MOMRFO to solve the multi-objective optimization problems. Before going into the details of the proposed algorithm, we first present the key points and the intuitions behind our method.

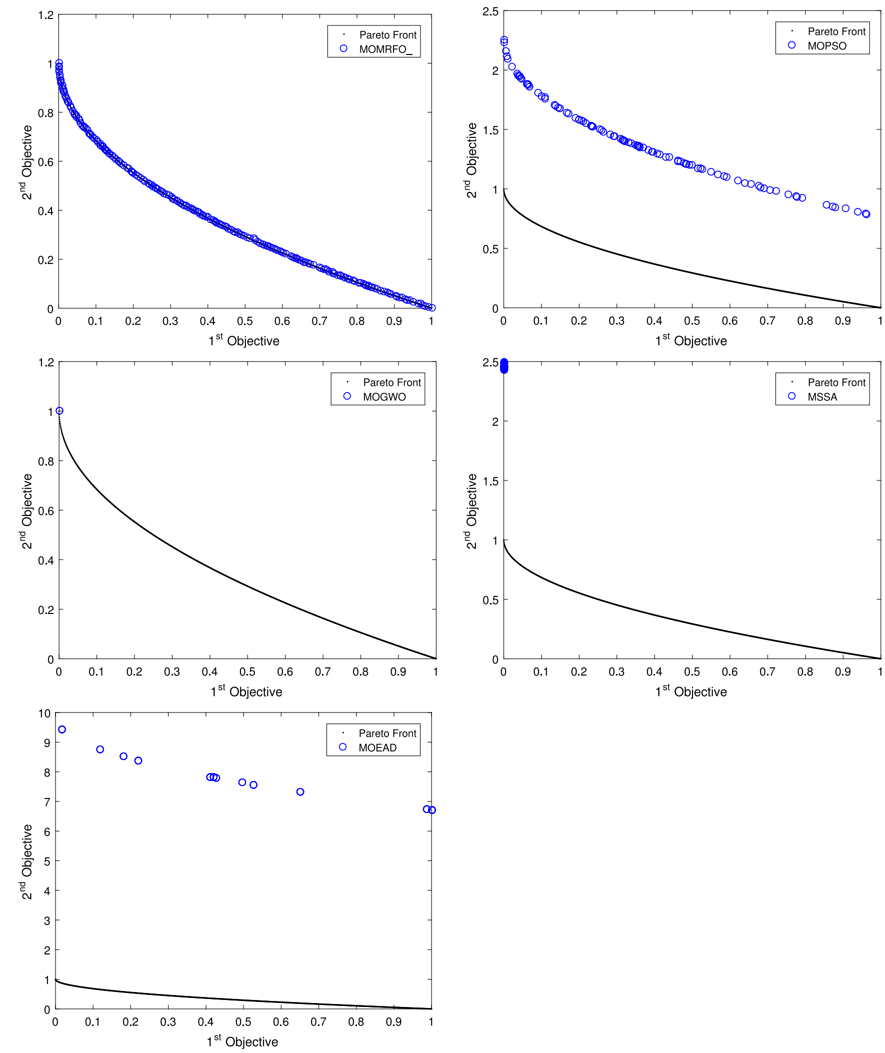

Fig. 9. Pareto optimal fronts obtained by MOMRFO, MOPSO, MOGWO, MSSA and MOEA/D for ZDT4 test function.

- We integrate a population archive to store and update the obtained non-dominated solutions during the exploration search processes. Also, the population archive uses to guide the Manta Rays population to converge toward the Pareto front.
- We use the epsilon-dominance to update the archive population, avoid the explosion of the size of archive population, improve the convergence of solutions and enhance the diversity of solutions on the whole Pareto front during the exploration of search space.
- We use the crowding distance to limit the archive size by deleting the most crowding solutions and we select the leader solutions from the archive population based on crowding distance to provide a good compromise between convergence and diversity.

The MOMRFO algorithm starts with the initialization of n Manta-Ray population in the search space, where each Manta-Ray presents a potential solution to multi-objective optimization problem (MOP). The solutions are then evaluated with M objective functions. Thereafter, the archive population is initialized by the non-dominated solutions of the initial population using the epsilon-dominance and the Manta Ray leader is selected from the population archive to guide the population of Manta Rays solutions to converge towards the Pareto front. After the initialization step, the MOMRFO applies the three foraging strategies to explore the space search of MOP. At each iteration, each Manta-Ray X_i updates its position with respect to the position of the one in front of it (X_{i-1}) the current position, and the leader Manta-Ray position.

To explore the space search, firstly the MOMRFO switch between the cyclone foraging behaviour and the chain foraging behaviour based on a randomly produced number (when 0.5‘$rand$, it applies the chain foraging strategy, else, it applies the cyclone foraging strategy) to update the position of Manta Rays and finally improve the solutions found so far by the Somersault foraging behaviour.

In cyclone foraging strategy, when $t/T < rand$ (t: current iteration and T: maximal iteration), each Manta-Ray applies a random movement with respect to a random position which is considered as the best leader solution and the solution in front of it to improve the exploration of space search and covers the whole search space by the Manta Rays population at the start of the exploration process (using Eq. (8)). When $t/T > rand$, the MOMRFO updates the position Manta-Ray X_i with respect the leader Manta-Ray position and the solution in front of it to improve the exploitation of space search at the end of the exploration process (using Eq. (5)). Therefore, cyclone foraging provides a good balance between exploration and exploitation of space search. In the chain foraging behaviour, the Manta Rays line up head-to-tail to forms a foraging chain, where each Manta-ray updates its position by the leader solution and the solution in front of it (using Eq. (3)).

After, the update position of all Manta Rays by the two foraging strategies (cyclone and chain foraging behaviour), the MOMRFO updates the position of each Manta Ray by somersault foraging strategy to improve the convergence toward the Pareto front. In the somersault foraging strategy, each Manta-Ray updates its position around the leader

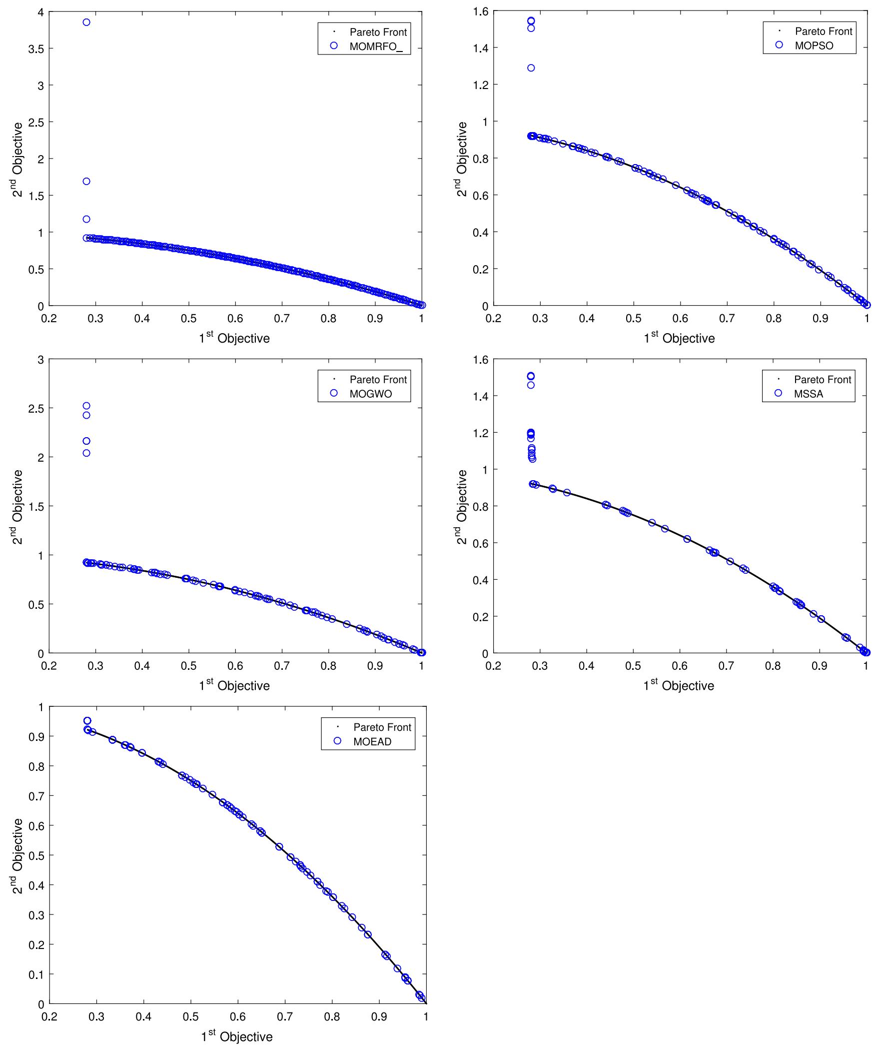

Fig. 10. Pareto optimal fronts obtained by MOMRFO, MOPSO, MOGWO, MSSA and MOEA/D for ZDT6 test function.

solution by swims to and from around the pivot and somersault to a new position (using Eq. (9)). As explained above, the update of each Manta-Ray position by the three foraging strategies is always guided by the leader solution which provides a good convergence towards the True front. Therefore, the MOMRFO algorithm dynamically updates the archive population and the solution leader in each movement of each Manta-Ray for good exploration efficiency of the search space. At each end of the iteration of the exploration process, in case the archive size is not limited by the epsilon dominance relation. MOMRFO uses the crowding distance to limit the size of the archive and keeps only the archive solutions with a large distance to improve diversity. Furthermore, the algorithm 3 and Fig. 2 outlines the pseudo-code and the flowchart of the proposed MOMRFO. At the end of the optimization process, MOMRFO extracts the Pareto solutions set from the archive population.

3.1. Manta-Ray leader selection

The Manta Ray leaders are the best solutions that guide the Manta Rays population to convergence towards the Pareto front set and improve the diversity of the solutions over the whole Pareto front, which makes the leader selection strategy a crucial step. In MOMRFO, the Manta Ray leader is selected from the population archive as follows:

- Compute the crowding distance of each solution in the archive population.
- Sort the archive solutions in descending order according to their crowding distance.
- Extract the higher part of the sorted archive which contains the less crowded solutions, where the size of the higher part which increases dynamically during the exploration process to improve the diversity of solutions.
- Select randomly a solution from the higher part of the sorted archive as a Manta Ray Leader to guide the Manta Rays population towards the least crowded space to improve the distribution of solutions along Pareto front.

3.2. Update archive population

The strategy of external archive update is a critical phase in the optimization process of the proposed algorithm to provide a good balance between the exploration and the exploitation of space search. In the MOMRFO algorithm, we use an external archive population with a limited size. This archive uses to keep the non-dominated solutions generated so far to contribute in the convergence of the population of Manta Rays toward the Pareto front. Initially, the archive initializes by the non dominated solutions of the initial Manta Rays population. At each iteration, the MOMRFO algorithm calls the update archive population after the update position of each Manta Ray by one of the three foraging strategies. The updated archive population uses the ϵ-dominance relation to accept or reject the new solution corresponding

Algorithm 1 Manta Ray foraging optimization

Require:
N : the size population;
T_{max}: the maximal number iteration;
Ensure:
X_{best} : the best solution;
Create an initial population of n Manta Ray within d-dimensional search: x_{ik}, $i = 1, \cdots, N$ and $k = 1, \cdots, d$;
Evaluate the fitness of the Manta Ray population;
while stopping criterion not met **do**
  for $i = 1$ to N **do**
    if $rand < 0.5$ **then**
      if $\frac{t}{T_{max}} < rand$ **then**
        Update the position of Manta Ray X_i using (7) and (8)
      else
        Update the position of Manta Ray X_i using (5) and (6)
      end if{ Chain foraging strategy}
    else
      Update the position of Manta X_i Ray using (3)
    end if
    Update the position of best solution X_{best};
  end for{ Somersault foraging strategy}
  for $i = 1$ to N **do**
    Update the position of Manta X_i Ray using (9)
    Update the position of best solution X_{best};
  end for
end while
Return the best solution found so far X_{best};

to the new position of Manta Ray by the archive population A. Furthermore, Fig. 3 and the algorithm 2 outlines the flowchart and the pseudo-code of the update archive population procedure.

For archive solutions $A(t)$ and the new solution $Xnew$ corresponding to the new position of Manta Ray solution, we associate an identification vector $B = (B_1, B_2, \ldots, B_M)^T$, where M denotes the objective function number of MOP, as follows:

$$B_j(f) = \lfloor \frac{\log(f_j)}{\log(\epsilon + 1)} \rfloor \tag{10}$$

where $\lfloor \cdot \rfloor$ is denotes the absolute value, f_j: the objective value jth of an archive solution and ϵ present the admissible error. The identification vector can divides the criteria space into hyper-boxes.

A new Manta-Ray solution is compared with all the solution of archive population using ϵ-dominance relation to decide if this solution is accepted into the archive population. More precisely, MOMRFO compares the new solution $Xnew$ with all solutions of archive population. If the identification vector $BXnew$ of the new solution dominates the subset identification vectors D of archive population, the new solution $Xnew$ is accepted and the subset of the solution is deleted from population archive. If the identification vector of the new solution $Xnew$ is dominated by the identification vector Ba of any archive solution a, then the new solution $Xnew$ is ϵ-dominated by the archive solution a and so the new solution is rejected.

If neither of the above two cases occurs, then the new solution $Xnew$ is equivalent to the archive solutions with ϵ-dominance relation. We can differentiate this into two cases:

1. If the new solution $Xnew$ and an archive solution a share the same identification vector B. If the new solution $Xnew$ dominates the archive solution or the new solutions $Xnew$ is equivalent to the archive solution but is closer to the identification B vector than the archive solution, then the new solution $Xnew$ is accepted.
2. In the event of a new solution $Xnew$ is not sharing the same B vector with any archive solution, the new solution $Xnew$ is accepted.

Algorithm 2 Update the archive population

Require: $Archive(t)$: the external archive population $Archive$ at iteration t, and $Xnew$: the new Manta Ray solution
compute the identification vector for new solution $Xnew$ and all solutions of the external archive population $A(t)$;
if $\exists X \in Archive(t) | B_X \geq B_X new$ **then**
  $Xnew$ is rejected;
end if
if $\exists X \in Archive(t) | B_X new \geq B_X$ **then**
  $Xnew$ replaces X in $Archive(t)$;
end if
if Neither of the above two cases occur **then**
  if $\exists a \in A(t) | B_c \sim B_a$ **then**
    if $c \sim a$ **then**
      Keep the solution with the smallest distance to the identification vector B;
    else
      Keep the solution that dominates the other;
    end if
  else
    Add the solution $Xnew$ in the external archive population $Archive(t)$;
  end if
end if

3.3. Limitation of the archive population size

In case the archive size exceeds the maximum size (Tmax). MOMRFO uses the crowding distance to limit the archive size as follows:

- Compute the crowding distance of each archive solution.
- The archive solutions are sorted in decreasing order according to their crowding distance.
- The less crowded solutions are kept and the most crowded solutions are removed from the archive population to limit the archive size to (Tmax).

The crowding distance uses to keep only the archive solutions with a large distance to improve the diversity of the archive solutions.

3.4. Complexity of MOMRFO algorithm

The time complexity of the proposed MOMRFO depends, obviously, on the original movements of MRFO algorithm, and on the new modules introduced in MRFO that deal with the multi-objective optimization problems. In the main loop, the proposed algorithm performs the three MRFO original movements, then performs the update of the population archive population and the leader solution selection. Therefore, the time complexity of the proposed algorithm can be estimated as follows:

O(MOMRFO) = Nmax(O(Movements of MRFO), O(Update archive population), O(Leader solution selection)), where N is the population size. The time complexity of three movements of MRFO is MDN, where, M: the number of objective functions, D: the number of decision variables, and N: the population size.

After the execution of the MRFO movements, the MOMRFO updates the archive with an $O(MNA)$ complexity, where NA is the number of individuals in the archive population. The last operation, during the main loop of MOMRFO, is the selection of the leader which is similar in

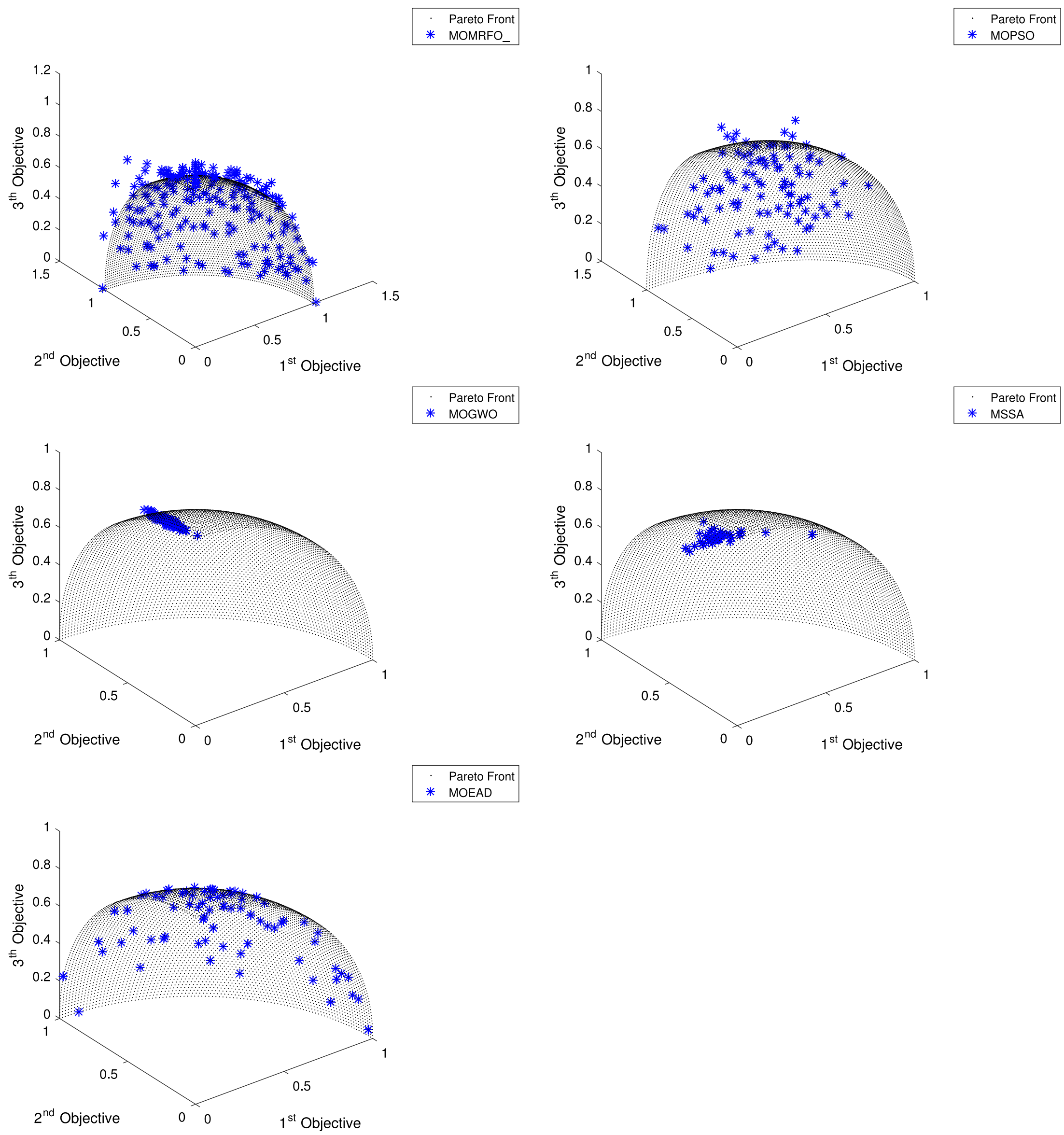

Fig. 11. Pareto optimal fronts obtained by MOMRFO, MOPSO, MOGWO, MSSA and MOEA/D for DTLZ2 test function.

terms of complexity to the update of the population archive. Therefore, the complexity of the proposed algorithm is given by:

$$O(MOMRFO) = N\max(O(MDN), O(MNA), O(MNA))$$

Hence, $O(MOMRFO) = NO(MNA) = O(MN^2)$

The complexity of our algorithm is better than some classical algorithms such as NSGA, and is similar to NSGA-II (Deb, Pratap et al., 2002), SPEA2 (Zitzler, Laumanns, Thiele, et al., 2001), MSSA (Mirjalili et al., 2017), and MOPSO (Coello, Pulido, & Lechuga, 2004). However, it looks is worse than the prominent MOEA/D (Zhang & Li, 2007).

4. Results and discussion

In this section, we discuss the performance of MOMRFO algorithm for solving multi-objective optimization problems. We begin by describing the benchmark tests and performance metrics used in our experimental study. Then, we present the state-of-the-art multi-objective metaheuristics used to validate the performance of the proposed algorithm. Next, we present the settings parameter used for the comparative studies of these algorithms. We also present the statistical results of these algorithms. Finally, we discuss the different obtained results. MOMRFO is compared with four state-of-the-art algorithms (MOEA/D (Zhang & Li, 2007), MOGWO (Mirjalili, Saremi, Mirjalili, & Coelho, 2016), MOPSO (Coello et al., 2004) and MSSA (Mirjalili et al., 2017)) on five bi-objective test functions, namely ZDT-series (Zitzler, Deb, & Thiele, 2000), and seven three-objective test functions, namely DTLZ-series (Deb, Thiele, Laumanns and Zitzler, 2002). The characteristics of bi-objective test functions and three-objective test functions are presented in Table 1. To confirm the performance of the proposed algorithm, the MOMRFO is tested on engineering design problems such as four-bar truss (FBT) design, welded beam deign (WB), disk brake design (DB), and speed reduced design (SR).

4.1. Comparison of multi-objective metaheuristics

In this comparison, we evaluated the performance of MOMRFO with four multi-objective metaheuristics which are briefly presented as follows:

- Multi-objective evolutionary algorithm based on decomposition (MOEA/D): which uses the reference points and the decomposition of criteria space to maintain the diversity and improve the convergence (Zhang & Li, 2007);

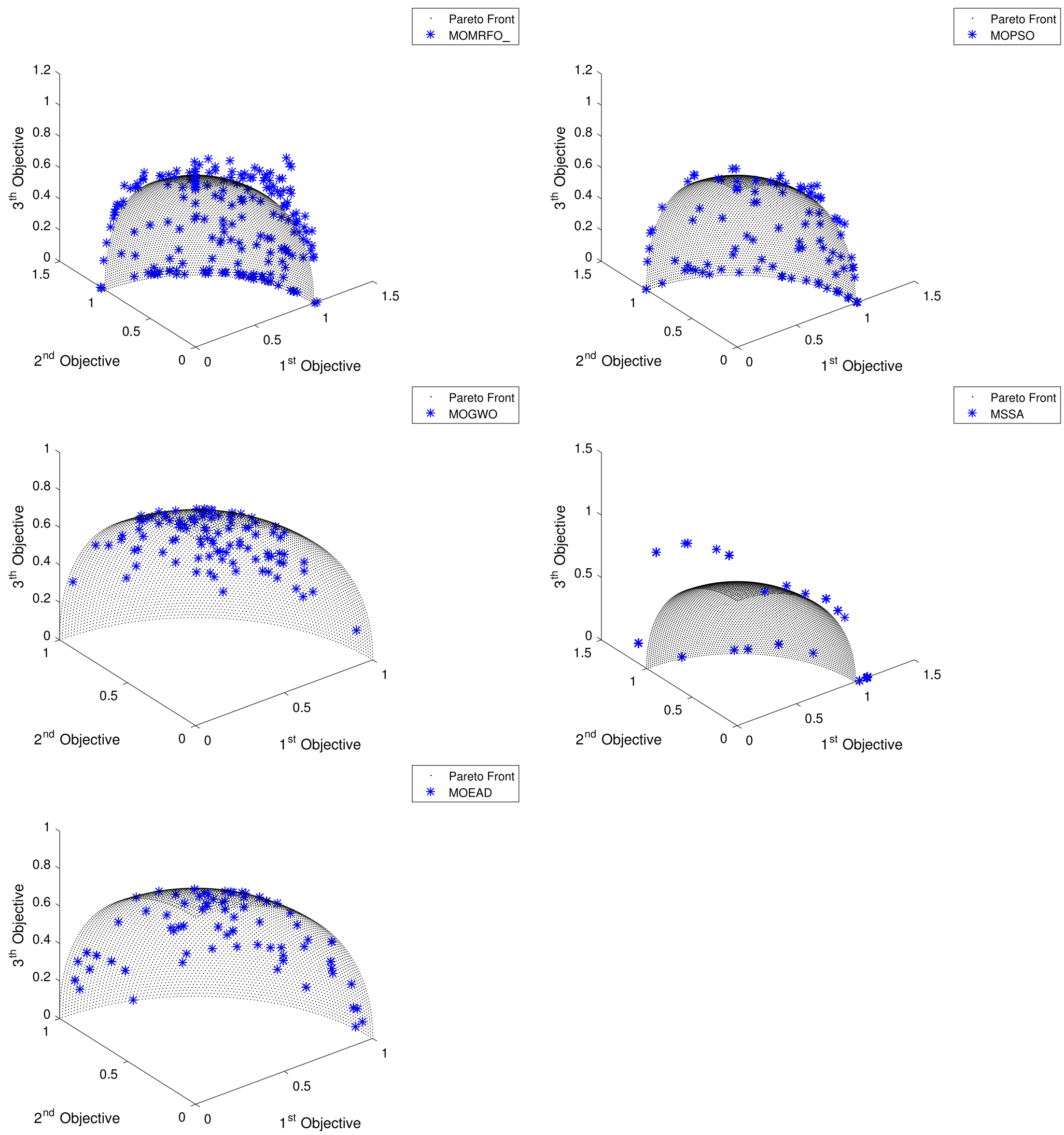

Fig. 12. Pareto optimal fronts obtained by MOMRFO, MOPSO, MOGWO, MSSA and MOEA/D for DTLZ4 test function.

- Multi-objective Grey Wolf Optimizer (MOGWO): which integrates a fixed-sized external archive and the Pareto dominance to converge toward the Optimal Pareto set; Mirjalili et al. (2016).
- Multi-objective Particle Swarm Optimization (MOPSO): which uses an external population and the leader solution to guide the particles solutions toward the Optimal Pareto set (Coello et al., 2004);
- Multi-objective Salp Swarm Algorithm (MSSA): which integrates an external repository to keep the non-dominated solutions obtained during the optimization process of Salp Swarm Algorithm (SSA) and the Pareto dominance relation to solve multi-objective problems (Mirjalili et al., 2017).

For the parameter setting of compared algorithms, we use the same parameters as the original papers (MOGWO Mirjalili et al., 2016, MOPSO Coello et al., 2004, MOEA/D Zhang & Li, 2007 and MSSA Mirjalili et al., 2017). The population size of all the algorithms used in this study is set to 100. Each test function was run 31 times. In each run, the maximal number of iterations for all algorithms is set to 1000.

4.2. Parameter's analysis

The performance of proposed algorithm depends on the parameters of the original MRFO algorithm and the parameters of the new introduced modules into MRFO. The later does not require any parameters during the optimization process, for that, the two parameters which can influence the performances of the proposed algorithm are: the size of archive population and the epsilon value of dominance relation.

In this study, we experiment the effect of the archives size and the epsilon value on the behaviour of the proposed algorithm. We modified the size of archive as well as the value of epsilon. We present in Fig. 4, the variation of the HV metric and the IGD metric during the 1000 iterations (one run) for DTLZ2 test function with three different size values of the archive population (50, 100, and 200), and we present in Fig. 5, the variation of the HV metric and the IGD metric during the 1000 iterations (one run) for DTLZ2 test function with three different epsilon values (0.01, 0.1 and 0.5). From Fig. 4, we notice that for DTLZ2, the IGD value is significantly minimized and the HV value is significantly maximized during the 1000 iterations with 200 archive size compared to 50 and 100 sizes. From Fig. 5, we notice that there is a small difference between the results of IGD metric or HV metric

Algorithm 3 Multi-objective Manta Ray foraging optimization

Require:
N : the size population;
T_{max}: the maximal number iteration;
Ensure:
X_{best} : the best solution;
1: Create an initial population of n Manta Ray within d-dimensional search: x_{ik}, $i = 1, \cdots, N$ and $k = 1, \cdots, d$;
2: Evaluate the Manta Ray population with M objective functions;
3: Safeguard the non-dominated solutions set of the initial population in archive population;
4: Select the leader Manta Ray from non-dominated solutions;
5: **while** stopping criterion not met **do**
6: **for** $i = 1$ to N **do**
7: **if** $rand < 0.5$ **then**
8: **if** $\frac{t}{T} < rand$ **then**
9: Move Manta Ray $x_i(t)$ towards the new position $x_i(t+1)$ according the cyclone foraging strategy using (8)
10: **else**
11: Move Manta Ray $x_i(t)$ towards the new position $x_i(t+1)$ according the cyclone foraging strategy using (5)
12: **end if**{ Chain foraging strategy}
13: **else**
14: Move Manta Ray $x_i(t)$ towards the new position $x_i(t+1)$ according the chain foraging strategy using (3)
15: **end if**
16: Update the archive population using Alg.2;
17: Update the leader Manta Ray using the phases of leader selection presented in Section 3.1;
18: **end for**{ Somersault foraging strategy}
19: **for** $i = 1$ to N **do**
20: Move Manta Ray $x_i(t)$ towards the new position $x_i(t+1)$ according the somersault foraging strategy using (9)
21: Update the archive population using Alg.2;
22: Update the leader Manta Ray using the phases of leader selection presented in Section 3.1;
23: **end for**
24: **end while**
25: Extract the Pareto set from the archive population;

for the three epsilon values. That is to say, there is an improvement in the performance of the proposed algorithm whenever the size of archive is increased. However, it should be noticed that increasing the size of the archive increases the number of evaluations, which affects the computational cost.

4.3. Performance metrics

To evaluate the performance of the different algorithms used in our study, we use the inverted generational distance (IGD) and the Hyper-Volume (HV) as performance metrics. Both can simultaneously measure the convergence and diversity of the obtained solution set. The IGD is one of the most widely used metrics. It can simultaneously measure the convergence and diversity of the obtained solution set, by calculating the minimum distance sum between each individual on the Pareto front P^* and the solutions set S obtained by a multi-objective algorithm (Sierra & Coello, 2004; While, Hingston, Barone, & Huband, 2006). IGD of S is computed as follows:

$$IGD(S, P^*) = \frac{\sum_{x \in P^*} dist(x, S)}{|P^*|} \tag{11}$$

where $dist(x; S)$ is the Euclidean distance between an individual x of P^* and its nearest neighbour in S, and $|P|$ is the cardinality of P. The IGD value is low, then the better convergence and diversity of S is obtained.

The HV is an indicator that measures the size of the hypercube dominated by the solutions in S (Zitzler & Thiele, 1999). the indicator HV can be computed as follows:

$$HV(S) = VOL(\bigcup_{x \in S} [f_1(x), r_1^*] \times \cdots [f_m(x), r_m^*]) \tag{12}$$

where VOL indicates the Lebesgue measure and $r = (r_1^*, r_2^*, \ldots, r_m^*)$ indicate a reference point in the objective space that is dominated by all the approximation S. The larger the HV value is, the better the convergence and diversity is obtained.

To verify if there is a significant difference between our MOMRFO and the considered algorithms, we use the 'Wilcoxon' non-parametric statistical test (Derrac, García, Molina, & Herrera, 2011). Given a significance level α, we say that there is no significant difference between algorithms if $p\_value \leq \alpha$ (i.e. the null hypothesis H_0 is validated), otherwise ($p\_value > \alpha$), there is a significant difference between algorithms (i.e. the alternative hypothesis H_1 is validated). The Wilcoxon test is applied on each couple of algorithms on the results of IGD and HV metrics, with a significance level equal to 5%. The hypotheses H_0 and H_1 are defined below, knowing that μ_1 and μ_2 represent the results of the first algorithm and the second algorithm, respectively.

For IGD metric, the hypotheses of the Wilcoxon test are defined as follows:

$H_0 : \mu_1 \geq \mu_2$
$H_1 : \mu_1 < \mu_2$

For HV metric, the hypotheses of Wilcoxon test are defined as follows:

$H_0 : \mu_1 \leq \mu_2$
$H_1 : \mu_1 > \mu_2$

The null hypothesis H_0 indicates that the IGD and HV of the first algorithm are similar or worst to the IGD and HV of the second algorithm as shown in the statistical results tables by (−); while the alternative hypothesis H_1 indicates that the IGD and HV of the first algorithm are better than the IGD and HV of the second algorithm, shown in the statistical results tables by (+).

4.4. Results on ZDT test functions

In this subsection, we investigate the performance of the proposed MOMRFO in the bi-objective test functions which are named ZDT functions. Table 2 reports the statically results of five compared algorithms in terms of IGD metric over 31 runs to ZDT test functions while Table 3 reports the statical results of five compared algorithms in terms HV metric over 31 runs to ZDT test functions. These statistical results of the IGD metric prove that in all ZDT test functions, where ZDT1 has a convex front, ZDT2 has a non-convex front, ZDT 3 has a discontinuous front and the ZDT4 function has a many Pareto local fronts 2^{19} and ZDT6 function has a non-uniform search space, the MOMRFO converges easily to the Pareto set. In terms of HV metric, The statistical results prove that the proposed MOMRFO outperforms the other algorithms on all ZDT function tests. Additionally, the visual observations in Figs. 6, 7, 8, 9 and 10 confirm that the non dominated solutions obtained by MOMRFO on all ZDT functions are well-distributed along the Pareto front.

4.5. Results on DTLZ test functions

In this subsection, we apply the MOMRFO algorithm to three-objective test functions which are called DTLZ functions and compare the results with MOGWO, MSSA, MOPSO and MOEA/D. The results of IGD and HV are given in Tables 4 and 5 respectively, whereas Figs. 11, 12, 13, 14 and 15 illustrate the Pareto set obtained by each

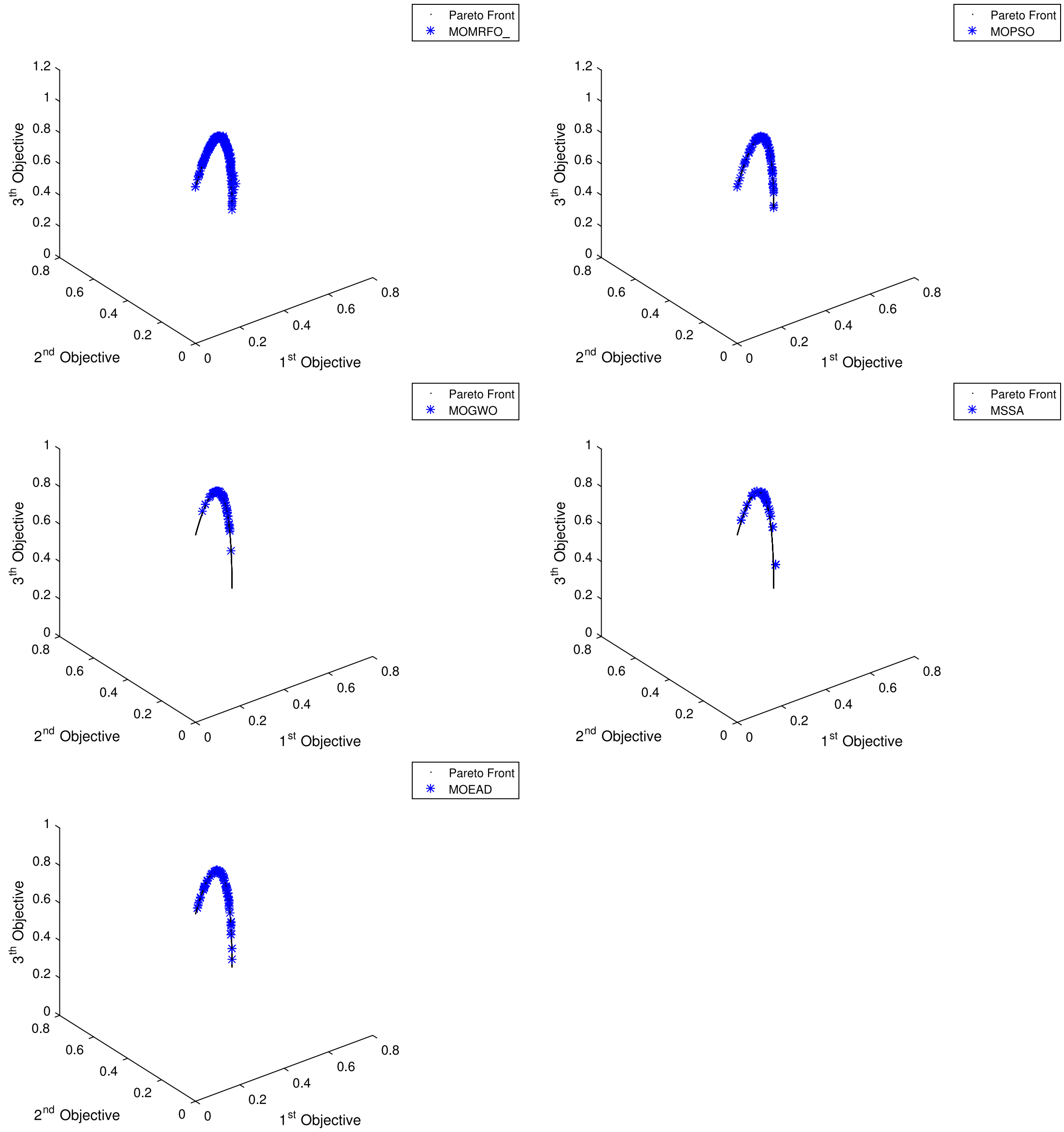

Fig. 13. Pareto optimal fronts obtained by MOMRFO, MOPSO, MOGWO, MSSA and MOEA/D for DTLZ5 test function.

algorithm on DTLZ test functions. According to the statistical results of IGD metric presented in Table 4, the proposed MOMRFO provides the best convergence towards the true optimal set compared to the other algorithms. More precisely, in all DTLZ test functions, the proposed MOMRFO ensures the better statistical results for IGD compared to other algorithms with a very low standard deviation which means that the MOMRFO algorithm is the most stable algorithm. In all DTLZ test functions, we can observe that MSSA and MOPSO algorithms obtain the lowest performance in terms of convergence among the five comparative algorithms.

Tables 5 present the statistical results of HV metric for DTLZ test functions. These results prove that the diversity of non-dominated solutions obtained by MOMRFO algorithm outperforms the other compared algorithms in all DTLZ test functions in terms of HV metric. From the statistical results presented in Tables 4 and 5, and the qualitative results illustrated in Figs. 11, 12, 13, 14 and 15, we confirm that the MOMRFO algorithm provides an excellent convergence behaviour with the best diversity of obtained non-dominated solutions set compared to the other algorithms in all DTLZ test functions.

4.6. Multi-objective engineering design problems

In this subsection, we apply the MOMRFO on four real engineering designs which are popular in the engineering design field (Askarzadeh, 2016; Got et al., 2020; Sadollah, Eskandar, Bahreininejad, & Kim, 2015). The engineering design problems are as follows: one problem without constraints (Four-bar truss design problem), and three problems with high constraints (Speed reduced design, Disk brake design problem, Welded beam design problem and).

- **Four bar truss problem:** This classical engineering design aims to minimize simultaneously the volume and displacement of a four-bar truss. This engineering problem has a highly constrained search space (Coello & Pulido, 2005). This problem can be formulated as:

$$minimize \begin{cases} f_1(x) = L(2x_1 + \sqrt{2x_2} + \sqrt{x_3} + x_4) \\ f_2(x) = \frac{FL}{E}(\frac{2}{x_2} + \frac{2\sqrt{2}}{x_2} - \frac{2\sqrt{2}}{x_3} + \frac{2}{x_4}) \end{cases} \tag{13}$$

where

$F = 10, E = 2e^5, L = 200$

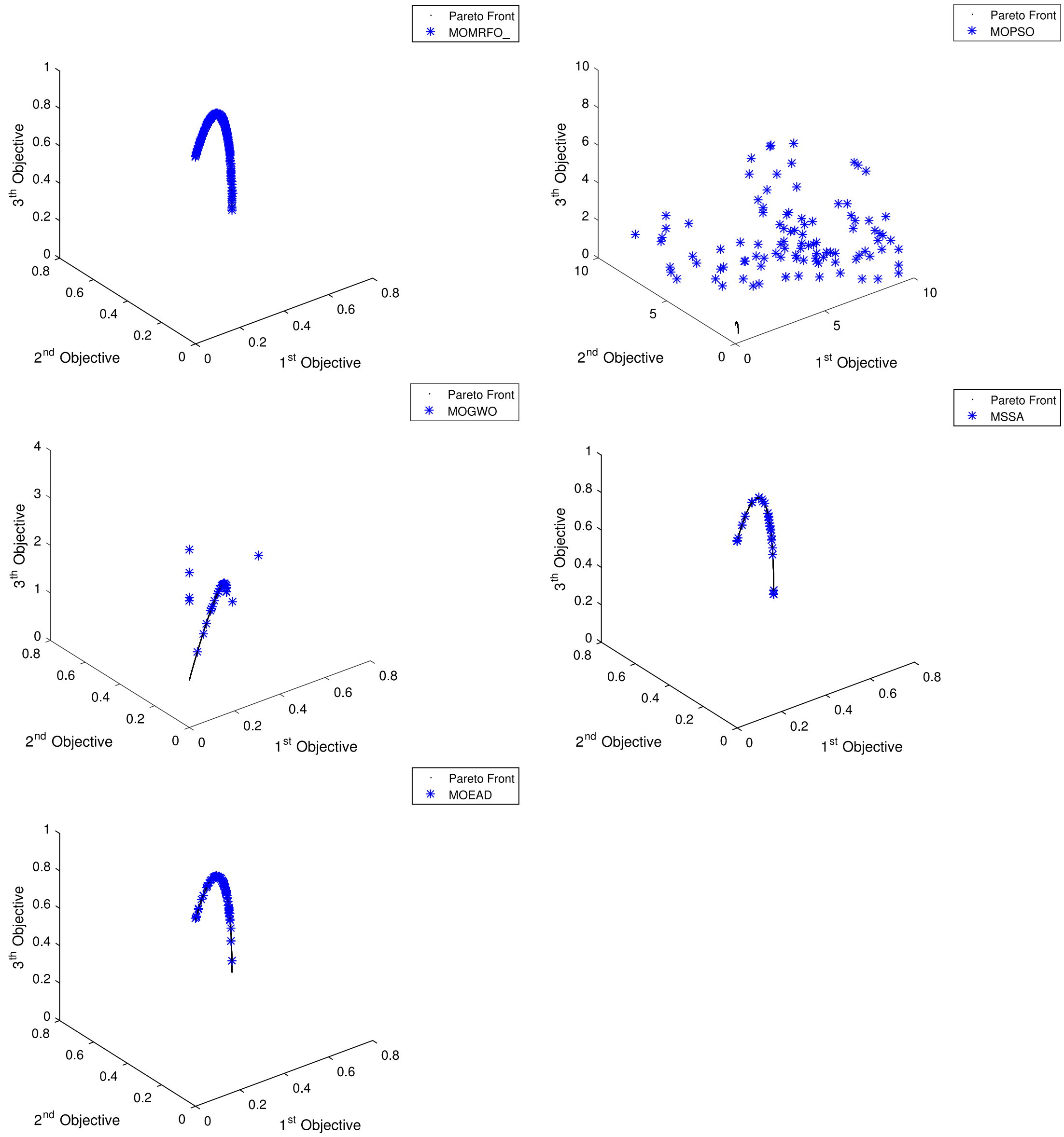

Fig. 14. Pareto optimal fronts obtained by MOMRFO, MOPSO, MOGWO, MSSA and MOEA/D for DTLZ6 test function.

$$1 \le x_1, x_4 \le 3,\ \sqrt{2} \le x_2, x_3 \le 3$$

- **Speed reduced design (SR):** The purpose of this design problem is to minimize both the weight of the gear assembly and the transverse deflection of the simultaneously optimized shaft. This problem under the design constraints such as the surfaces stress, transverse deflections of the shafts, bending stress of the gear teeth and stresses in the shafts. This design problem has seven design variables: the face width, a module of teeth, number of teeth in the pinion, length of the first shaft between bearings, length of the second shaft between bearings and the diameter of the first and second shafts (Coello & Pulido, 2005). This problem can be formulated as:

$$minimize \begin{cases} f_1(x) = 0.7854x_1x_2^2\left(3.3333x_3^2 + 14.9334x_3 - 43.0934\right) \\ -1.508x_1\left(x_6^2 + x_7^2\right) \\ +7.4777\left(x_6^3 + x_7^3\right) + 0.7854\left(x_4x_6^2 + x_5x_7^2\right) \\ f_2(x) = \dfrac{\sqrt{(\frac{745x_4}{x_2x_3})^2 + 16.9e6}}{110x_6^3} \end{cases} \quad (14)$$

Subject to:

$$g_1 = \frac{27}{x_1x_2^2x_3} - 1 \leqslant 0$$

$$g_2 = \frac{397.5}{x_1x_2^2x_2^2} - 1 \leqslant 0$$

$$g_3 = \frac{1.93x_4^3}{x_2x_6^4x_3} - 1 \leqslant 0$$

$$g_4 = \frac{1.93x_5^3}{x_2x_7^4x_3} - 1 \leqslant 0$$

$$g_5 = \frac{\left[\left(745x_4/x_2x_3\right)^2 + 16.9\times10^6\right]^{1/2}}{110x^3} - 1 \leqslant 0$$

$$g_6 = \frac{\left[\left(745x_5/x_2x_3\right)^2 + 157.5\times10^6\right]^{1/2}}{85x_7^3} - 1 \leqslant 0$$

$$g_7 = \frac{x_2x_3}{40} - 1 \leqslant 0$$

$$g_8 = \frac{5x_2}{x_1} - 1 \leqslant 0$$

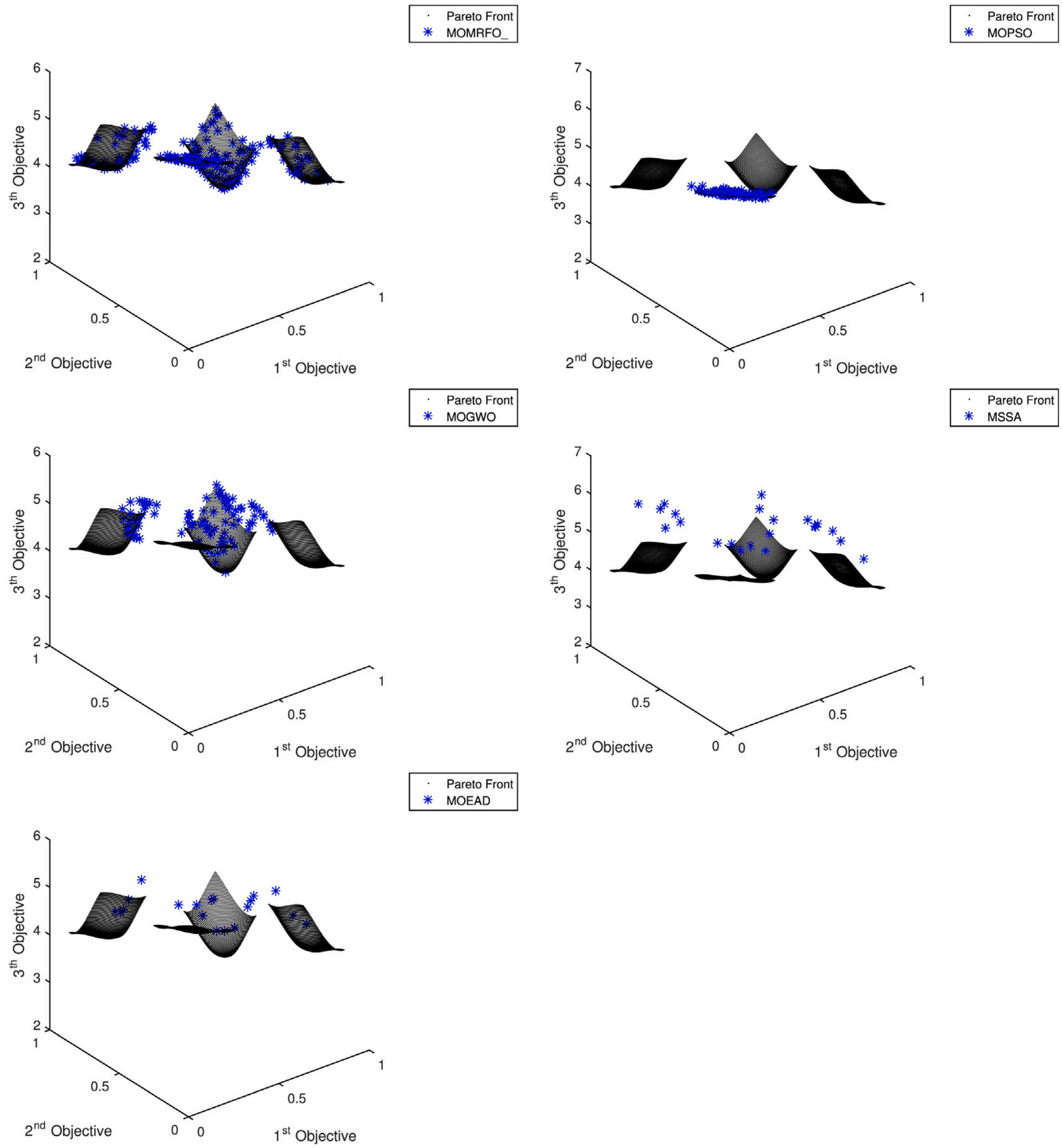

Fig. 15. Pareto optimal fronts obtained by MOMRFO, MOPSO, MOGWO, MSSA and MOEA/D for DTLZ7 test function.

$$g_9 = \frac{x_1}{12x_2} - 1 \leqslant 0$$
$$g_1 0 = \frac{1.5x_6 + 1.9}{x_4} - 1 \leqslant 0$$
$$g_1 1 = \frac{1.1x_7 + 1.9}{x_5} - 1 \leqslant 0$$

where

$$2.6 \leqslant x_1 \leqslant 3.6, 0.7 \leqslant x_2 \leqslant 0.8, 17 \leqslant x_3 \leqslant 28,$$
$$7.3 \leqslant x_4 \leqslant 8.3, 7.3 \leqslant x_5 \leqslant 8.3, 2.9 \leqslant x_6 \leqslant 3.9, 5.0 \leqslant x_7 \leqslant 5.5$$

- **Disk brake design problem:** The purpose of this design problem is to minimize simultaneously the overall mass and the braking time. The space search of this problem is defined on four design variables, the inner radius, outer radius, the engaging force and the number of friction surfaces. In addition, it is constrained by the torque, pressure,temperature, and length of the brake (Ray & Liew, 2002). The mathematical formulation used for modelling this problem is as follows:

$$minimize \begin{cases} f_1(x) = 4.9e - 5(x_2^2 - x_1^2)(x_4 - 1) \\ f_2(x) = (9.82e^6)\frac{x_2^2 - x_1^2}{x_3 x_4 (x_2^3 - x_1^3)} \end{cases} \tag{15}$$

Subject to:

$$g_1 = 20 + x_1 - x_2$$
$$g_2 = 2.5(x_4 + 1) - 30$$
$$g_3 = \frac{x_3}{3.14(x_2^2 - x_1^2)^2} - 0.4$$
$$g_4 = 2.22e - 3x_3 \frac{x_2^3 - x_1^3}{(x_2^2 - x_1^2)^2} - 1$$
$$g_5 = 900 - \frac{2.66e - 2x_3 x_4 (x_3^3 - x_1^3)}{x_2^2 - x_1^2}$$

where

$$\forall g_i \leq 0$$

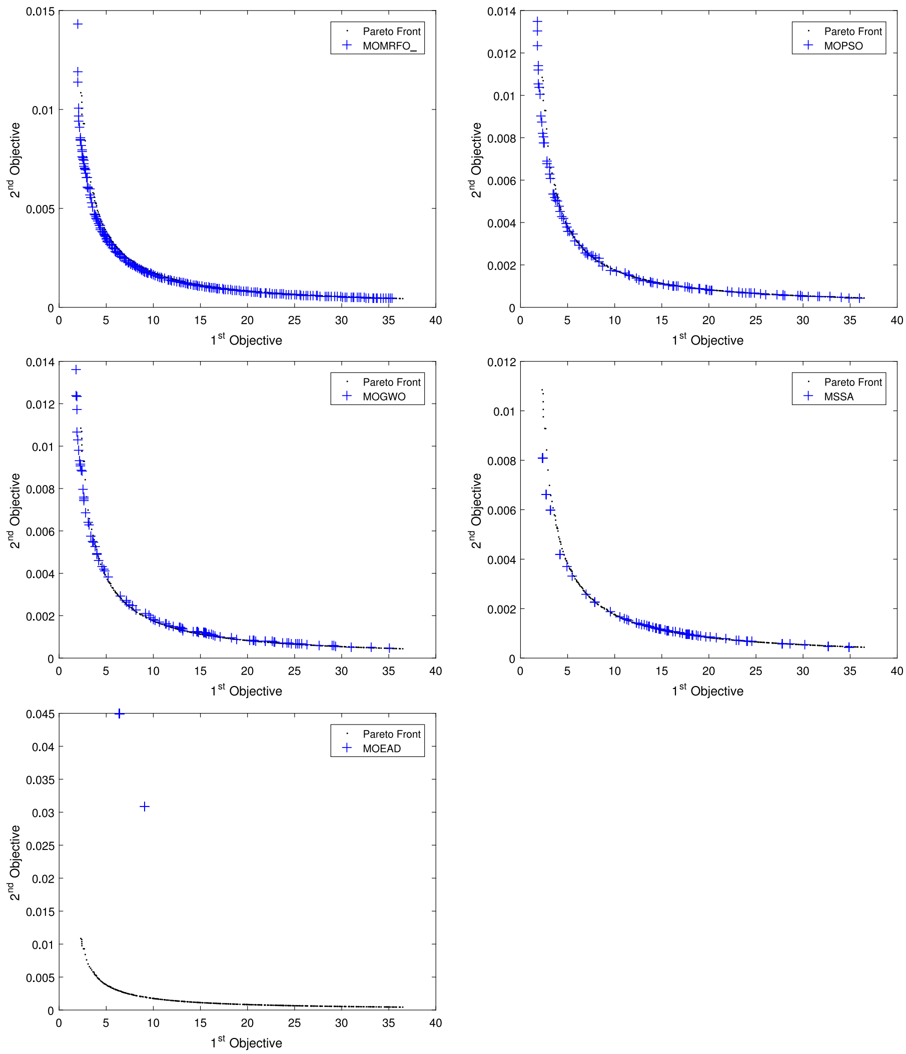

Fig. 16. Pareto optimal fronts obtained by MOMRFO, MOPSO, MOGWO, MSSA and MOEA/D for Welded beam design problem.

$55 \leq x_1 \leq 80, 75 \leq x_2 \leq 110, 1000 \leq x_3 \leq 3000, 2 \leq x_4 \leq 20$

- **Welded beam design problem:**
The purpose of this design problem is to minimize both the overall fabrication cost and the end deflection. This design problem subject to several constraints such as shear stress, bending stress, weld length and the buckling load. Also, this design problem has four design variables which are the height, the length of the welded joint, thickness, and the width of the beam (Deb, Pratap, & Moitra, 2000).

$$minimize \begin{cases} f_1(x) = 1.10471x_1^2x_2 + 0.04811x_3x_4(14 + x_2) \\ f_2(x) = del \end{cases} \tag{16}$$

Subject to:

$$g_1 = -(tau - taumax)$$
$$g_2 = -(sig - sigmax)$$
$$g_3 = -(x_1 - x_4)$$
$$g_4 = -(P - pc)$$

where,

$$pc = (\frac{4.013 * E\sqrt{\frac{x_3^2x_4^6}{36}}}{L^2})(1 - ((\frac{x_3}{2L})\sqrt{\frac{E}{4G}}))$$

$$del = \frac{4PL^3}{Ex_3^3x_4}$$

$$sig = \frac{6PL}{x_4x_3^2}$$

$$J = 2 * (\sqrt{2}x_1x_2((\frac{x_2^2}{12}) + (\frac{x_1 + x_3}{2})^2))$$

$$R = \sqrt{(\frac{x_2^2}{4}) + (\frac{x_1 + x_3}{2})^2}$$

$$M1 = P * (L + \frac{x_2}{2})$$

$$tau2 = \frac{M1R}{J}$$

$$tau1 = \frac{P}{\sqrt{2}x_1x_2}$$

$$tau = \sqrt{tau1^2 + 2.tau1.tau2.\frac{x_2}{2R} + tau2^2}$$

$$\forall g_i \geq 0$$

$$P = 6000, L = 14, E = 30e^6, G = 12e^6,$$

$$taumax = 13\,600, sigmax = 30\,000$$

$$0.125 \leq x_1, x_4 \leq 5, 0.1 \leq x_2, x_3 \leq 10$$

To deal with the constraints of these engineering problems and to explore the feasible space search, we use the static penalty method which

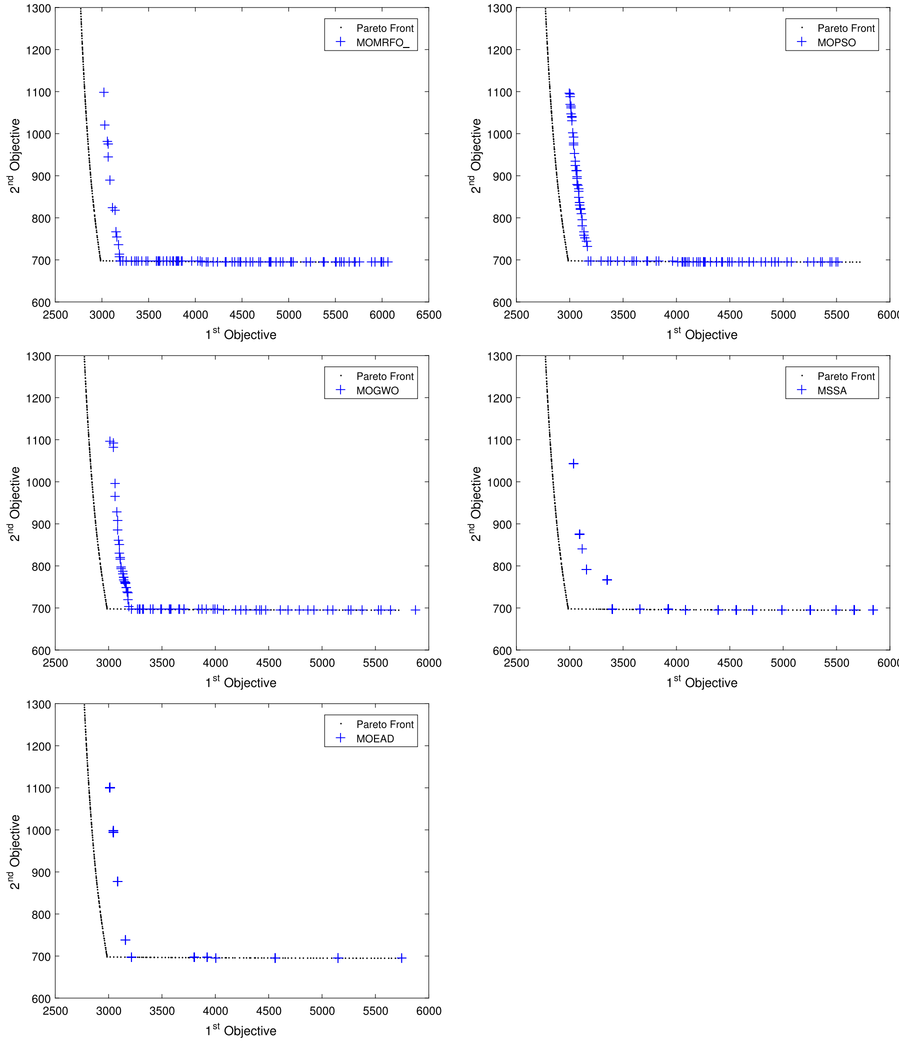

Fig. 17. Pareto optimal fronts obtained by MOMRFO, MOPSO, MOGWO, MSSA and MOEA/D for Speed reduced design.

transforms a constrained problem into an unconstrained problem (Rao, 2019). Therefore, if any constraint is violated, a penalty P_i is added to the objective function value $f_i(x)$ as follows:

$$f_m(x) = f_m(x) + \sum_{i=1}^{p} P_i.max(g_i(x), 0) + \sum_{i=1}^{K} P_i.max(|h_i(x)| - \delta, 0) \quad (17)$$

where

$f_m(x), m = 1, 2, \ldots, M$ are the objective number functions.

$g_i(x) \leq 0, i = 1, 2, \ldots, P$ are inequality constraints.

$h_i(x) = 0, i = 1, 2, \ldots, K$ are equality constraints.

P_i is the penalty factor and δ is the tolerance on the equality constraints to consider it as a feasible.

The severeness of the penalty function depends on the penalty factors P_i. A large penalty precludes exploring the unfeasible region. In contrast, a small penalty will explore the unfeasible regions; thus MOMRFO uses a big penalty factors P_i to avoid the exploration of unfeasible regions and ensure the feasibility of the Pareto solutions set. The results of IGD and HV for the four engineering design problems are given in Tables 6 and 7 respectively, whereas Figs. 16, 17, 18 and 19 illustrate the Pareto set obtained by each algorithm on engineering design problems. By observing of the IGD results obtained in Table 6, the proposed MOMRFO provides the best convergence towards the true optimal set compared to the other algorithms for all engineering design problems. More precisely, the proposed MOMRFO ensures the better statistical results of IGD for engineering design problems with high constraints (Disk brake design problem and welded beam design problem) or without constraints (Four-bar truss design problem and gear train problem). Also, MOMRFO algorithm has the best performance in terms of HV on all engineering design problems compared the other considered algorithms. According to these statistical results reported in Tables 6 and 7, and the Pareto set obtained for compared algorithms which are illustrated in Figs. 16, 17, 18 and 19, we can conclude that the MOMRFO algorithm has an excellent convergence and diversity behaviours for the considered engineering design problems.

According to the results of Wilcoxon statistical test reported in the Tables 2, 3, 4, 5, 6 and 7, we observe that almost in all the bi-objective test functions, three objectives test functions and engineering design problems, the hypothesis $H1$ is accepted for MOMRFO algorithm in terms of IGD or HV (MOMRFO minimizes the values of IGD compared to the other algorithms or MOMRFO maximize the values of HV compared to the other algorithms) with a significance level greater than 95%. Consequently, the Wilcoxon statistical test confirms that the MOMRFO algorithm converges better towards the Pareto front and improves the diversity of Pareto set with a significance level higher than 95%.

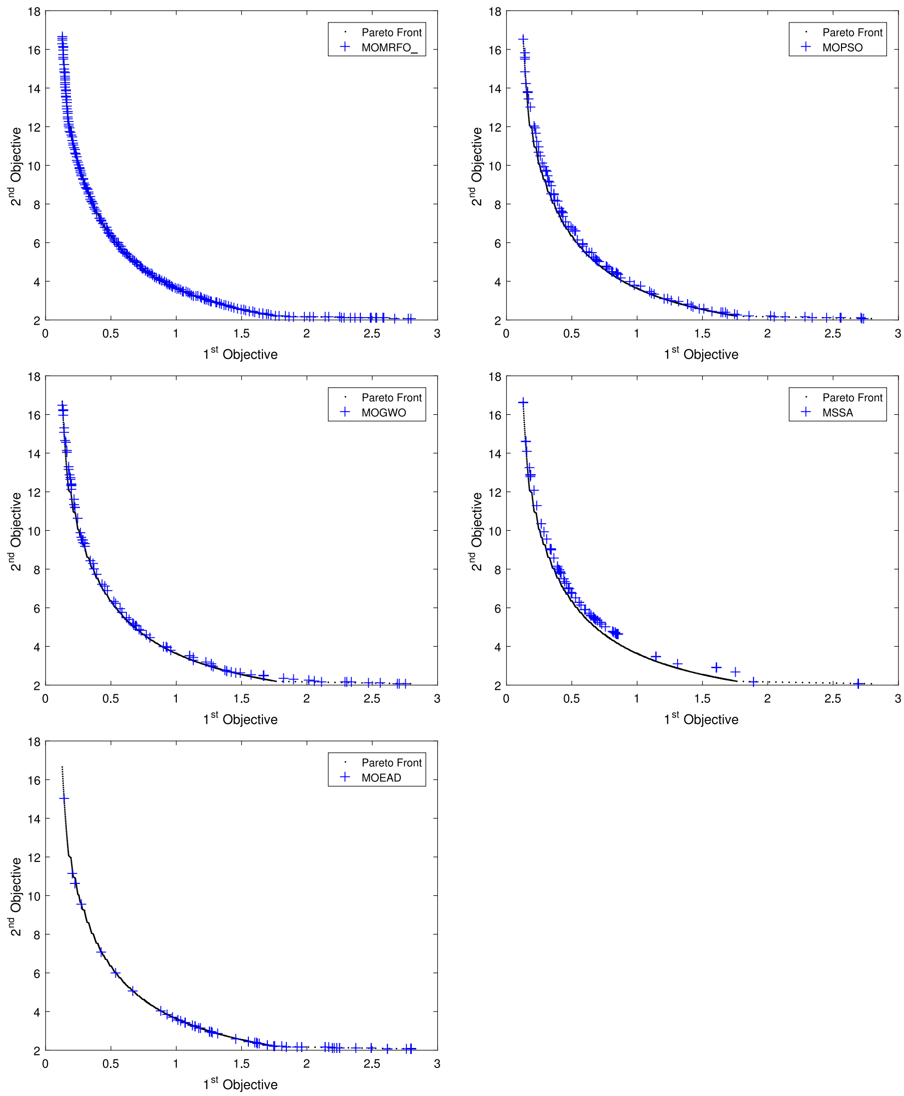

Fig. 18. Pareto optimal fronts obtained by MOMRFO, MOPSO, MOGWO, MSSA and MOEA/D for disk brake design problem test function.

4.7. Running time analysis

The running time is also taken into account in our simulation. For this reason, all algorithms are executed one time with 1000 iterations, and the results are reported in Table 8 in terms of seconds.

From Table 8, it can be seen that MSSA is the fastest algorithm for most test functions. It achieves the best time in eleven out of the sixteen test functions. The second fastest algorithm is MOMRFO, which obtain the best values for three DTLZ test functions (DTLZ2, DTLZ3 and DTLZ4), and achieves the best time after MSSA for the other test functions. In this study, we can see that the slowest algorithms with a great difference are MOGWO and MOEAD that consume a lot of execution time. Broadly speaking, the proposed algorithm significantly outperforms both MOGWO and MOEA/D in terms of computational time, and it was very competitive compared to MSSA algorithm.

4.8. Further discussion

In this study, the proposed MOMRFO algorithm is applied on different test functions such as ZDT test functions, DTLZ test functions and the real engineering application such as the disk brake design, welded beam design, four-bar truss design and speed reduced design. In comparison with other algorithms, the statistical results of evaluation metrics prove that the MOMRFO algorithm performs the almost all test functions (bi-objective test functions and three-objective test function) and almost all engineering design problems, and converges to Pareto set without suffering, while the other algorithms suffer to converge toward Pareto set. Moreover, the high performance of MOMRFO on engineering design problems confirmed in this study, helps in solving many real engineering applications. The reasons behind this performance can be described as follows:

- The high performance of MOMRFO is due to the good exploration strategies used by MRFO algorithm during the optimization process, where the Manta Rays population switch between the cyclone foraging strategy and the chain foraging strategy to improve the exploration of space search Manta Rays and uses at the end of the optimization process the somersault foraging strategy to improve the exploitation of space search.
- The relevant multi-objective optimization tools introduced in MRFO algorithm to solve the multi-objective optimization problems which are:
    1. The use of external archive population into the MOMRFO to keep the best non-dominated solutions obtained during the process optimization and guide the MOMRFO to explore the fertile regions of space search.
    2. The use of ϵ-dominance in updating the archive population allows to obtain a good approximation of the Pareto set,

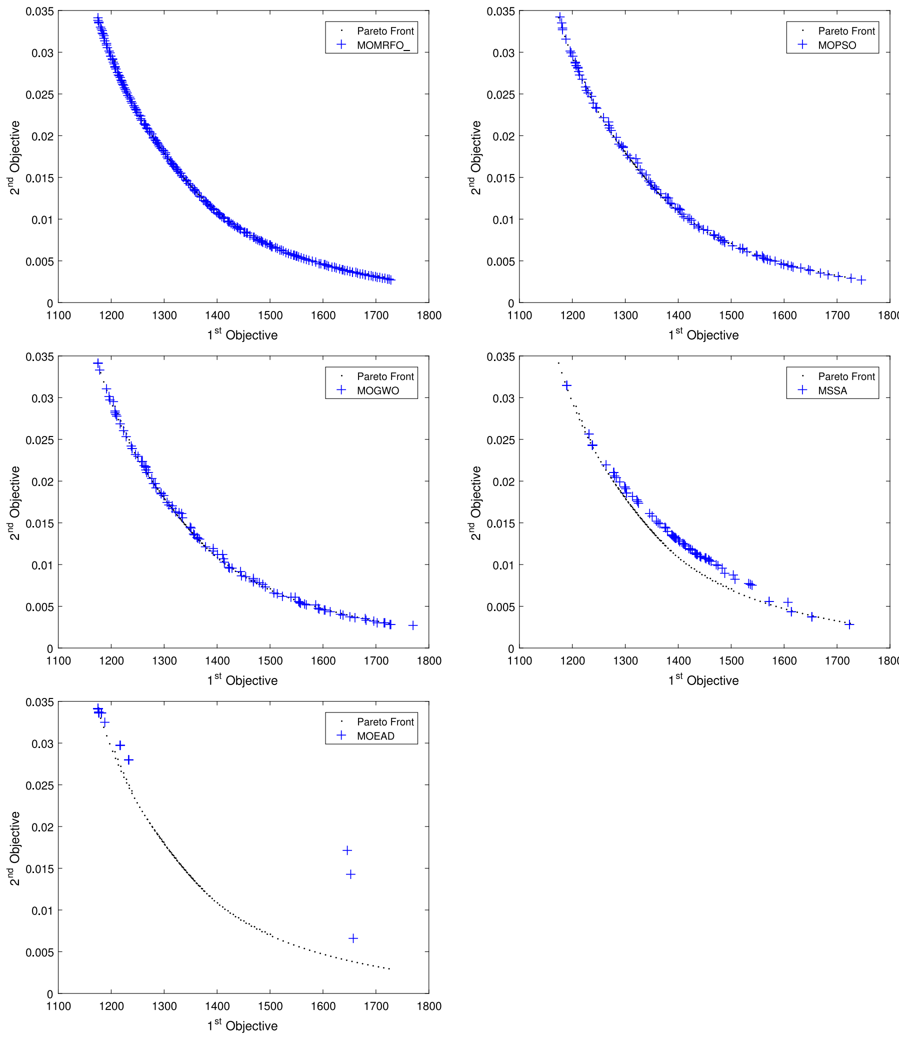

Fig. 19. Pareto optimal fronts obtained by MOMRFO, MOPSO, MOGWO, MSSA and MOEA/D for Four bar truss problem.

avoid the explosion of the archive size and reduce the execution time of the MOMRFO algorithm.

3. The selection of Leader's solutions from the archive population and the use of crowding distance in the selection of Leader solutions that allows a good balance between the exploration and the exploitation of the search space.

However, the proposed algorithm needs to be adapted to many-objective optimization problems (MoaPs):

- The use of distance crowding and the ϵ-dominance relation is very effective for a multi-objective optimization problem with 2 or 3 objectives because they can offer a good balance between convergence and diversity. However, in many-objective optimization with more than 5 objectives, these concepts fail to handle the explosion of non-dominated solutions. To extend the MOMRFO algorithm to solve the problems of MoaPs. it is necessary to use new dominance relations to discriminate between the non-dominated solutions.
- The proposed algorithm depends on the epsilon value to update the population archive. The bad choice of this parameter can influence negatively on the optimization performances of the algorithm MOMRFO. So, the MOMRFO algorithm requires new dominance relations to obtain the set of the Pareto front independently of the choice of epsilon.
- To improve the diversity of the population archive, the MOMRFO algorithm uses a crowding distance which favours following extreme points during the optimization process instead of checking solutions that present good compromise between the objectives. This problem can be solved by using other diversity estimation methods or by improving the used distance.

5. Conclusion

This paper presented a guided archive population Manta-Ray foraging algorithm (MOMRFO) for solving large-scale multi-objective optimization problems. The main ideas of our algorithm can be summarized as follows. Firstly, we integrate the population archive to store and update the obtained non-dominated solutions during the exploration search processes. Secondly, we used the leader's solution to guide the population towards promising regions of the search space. Finally, we introduced the epsilon dominance and crowding distance to update the population archive, avoid the explosion of the archive population size and enhance the solutions diversity. The proposed algorithm was tested and compared with four state-of-the-art algorithms on five bi-objective test functions, seven three-objective test functions, and the structural design problems such as 4-bar truss design, gear train problem, welded beam design, and disk brake design. The obtained experimental results have proven that the proposed algorithm is very efficient in solving of MOPs problems in terms of convergence and diversity. Our future work can be carried out in the following two aspects. Firstly, we are modifying the proposed algorithm to solve the many-objective optimization problem and secondly, we are looking for real-world applications with many objectives to prove the performance of our method.

CRediT authorship contribution statement

Djaafar Zouache: Writing – original draft, Writing – review & editing, Software, Conceptualization, Methodology, Formal analysis, Visualization. **Fouad Ben Abdelaziz:** Writing – review & editing , Methodology, Formal analysis, Investigation.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

Askarzadeh, A. (2016). A novel metaheuristic method for solving constrained engineering optimization problems: crow search algorithm. *Computers and Structures*, *169*, 1–12.

Bader, J., & Zitzler, E. (2011). Hype: An algorithm for fast hypervolume-based many-objective optimization. *Evolutionary Computation*, *19*(1), 45–76.

Batista, L. S., Campelo, F., Guimarães, F. G., & Ramírez, J. A. (2011). Pareto cone ε-dominance: improving convergence and diversity in multiobjective evolutionary algorithms. In *International conference on evolutionary multi-criterion optimization* (pp. 76–90). Springer.

Beume, N., Naujoks, B., & Emmerich, M. (2007). Sms-emoa: Multiobjective selection based on dominated hypervolume. *European Journal of Operational Research*, *181*(3), 1653–1669.

Brockhoff, D., Wagner, T., & Trautmann, H. (2015). 2 indicator-based multiobjective search. *Evolutionary Computation*, *23*(3), 369–395.

Brockhoff, D., & Zitzler, E. (2007). Improving hypervolume-based multiobjective evolutionary algorithms by using objective reduction methods. In *2007 ieee congress on evolutionary computation* (pp. 2086–2093). IEEE.

Cheng, Y., Jin, Y., & Hu, J. (2009). Adaptive epsilon non-dominated sorting multi-objective evolutionary optimization and its application in shortest path problem. In *2009 iccas-sice* (pp. 2545–2549). IEEE.

Coello, C. C., & Pulido, G. T. (2005). Multiobjective structural optimization using a microgenetic algorithm. *Structural and Multidisciplinary Optimization*, *30*(5), 388–403.

Coello, C. A. C., Pulido, G. T., & Lechuga, M. S. (2004). Handling multiple objectives with particle swarm optimization. *IEEE Transactions on Evolutionary Computation*, *8*(3), 256–279.

Deb, K., & Jain, H. (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), 577–601.

Deb, K., Mohan, M., & Mishra, S. (2005). Evaluating the ε-domination based multi-objective evolutionary algorithm for a quick computation of Pareto-optimal solutions. *Evolutionary Computation*, *13*(4), 501–525.

Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE Transactions on Evolutionary Computation*, *6*(2), 182–197.

Deb, K., Pratap, A., & Moitra, S. (2000). Mechanical component design for multiple ojectives using elitist non-dominated sorting ga. In *International conference on parallel problem solving from nature* (pp. 859–868). Springer.

Deb, K., Thiele, L., Laumanns, M., & Zitzler, E. (2002). Scalable multi-objective optimization test problems. In *Proceedings of the 2002 congress on evolutionary computation. cec'02 (cat. no. 02th8600), Vol. 1* (pp. 825–830). IEEE.

Derrac, J., García, S., Molina, D., & Herrera, F. (2011). A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. *Swarm and Evolutionary Computation*, *1*(1), 3–18.

Farina, M., & Amato, P. (2004). A fuzzy definition of" optimality" for many-criteria optimization problems. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, *34*(3), 315–326.

Got, A., Moussaoui, A., & Zouache, D. (2020). A guided population archive whale optimization algorithm for solving multiobjective optimization problems. *Expert Systems with Applications*, *141*, Article 112972.

Guo, Y.-N., Cheng, J., Luo, S., Gong, D., & Xue, Y. (2017). Robust dynamic multi-objective vehicle routing optimization method. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, *15*(6), 1891–1903.

Guo, Y., Yang, H., Chen, M., Cheng, J., & Gong, D. (2019). Ensemble prediction-based dynamic robust multi-objective optimization methods. *Swarm and Evolutionary Computation*, *48*, 156–171.

Guo, Y.-n., Zhang, P., Cheng, J., Wang, C., & Gong, D. (2018). Interval multi-objective quantum-inspired cultural algorithms. *Neural Computing and Applications*, *30*(3), 709–722.

Guo, Y.-N., Zhang, X., Gong, D.-W., Zhang, Z., & Yang, J.-J. (2019). Novel interactive preference-based multiobjective evolutionary optimization for bolt supporting networks. *IEEE Transactions on Evolutionary Computation*, *24*(4), 750–764.

He, Z., Yen, G. G., & Zhang, J. (2013). Fuzzy-based Pareto optimality for many-objective evolutionary algorithms. *IEEE Transactions on Evolutionary Computation*, *18*(2), 269–285.

Laumanns, M., Thiele, L., Deb, K., & Zitzler, E. (2002). Combining convergence and diversity in evolutionary multiobjective optimization. *Evolutionary Computation*, *10*(3), 263–282.

Li, K., Deb, K., Zhang, Q., & Kwong, S. (2014). An evolutionary many-objective optimization algorithm based on dominance and decomposition. *IEEE Transactions on Evolutionary Computation*, *19*(5), 694–716.

Li, H., & Zhang, Q. (2008). Multiobjective optimization problems with complicated pareto sets, moea/d and nsga-ii. *IEEE Transactions on Evolutionary Computation*, *13*(2), 284–302.

Liu, H.-L., Gu, F., & Zhang, Q. (2013). Decomposition of a multiobjective optimization problem into a number of simple multiobjective subproblems. *IEEE Transactions on Evolutionary Computation*, *18*(3), 450–455.

Liu, J., Wang, Y., Wang, X., Sui, X., Guo, S., & Liu, L. (2017). An alpha-dominance expandation based algorithm for many-objective optimization. In *2017 13th international conference on computational intelligence and security (cis)* (pp. 6–10). IEEE.

Mirjalili, S., Gandomi, A. H., Mirjalili, S. Z., Saremi, S., Faris, H., & Mirjalili, S. M. (2017). Salp swarm algorithm: A bio-inspired optimizer for engineering design problems. *Advances in Engineering Software*, *114*, 163–191.

Mirjalili, S., Saremi, S., Mirjalili, S. M., & Coelho, L. d. S. (2016). Multi-objective grey wolf optimizer: a novel algorithm for multi-criterion optimization. *Expert Systems with Applications*, *47*, 106–119.

Rao, S. S. (2019). *Engineering optimization: theory and practice*. John Wiley & Sons.

Ray, T., & Liew, K. (2002). A swarm metaphor for multiobjective design optimization. *Engineering Optimization*, *34*(2), 141–153.

Sadollah, A., Eskandar, H., Bahreininejad, A., & Kim, J. H. (2015). Water cycle algorithm for solving multi-objective optimization problems. *Soft Computing*, *19*(9), 2587–2603.

Sato, H., Aguirre, H. E., & Tanaka, K. (2007). Controlling dominance area of solutions and its impact on the performance of moeas. In *International conference on evolutionary multi-criterion optimization* (pp. 5–20). Springer.

Shang, K., & Ishibuchi, H. (2020). A new hypervolume-based evolutionary algorithm for many-objective optimization. *IEEE Transactions on Evolutionary Computation*.

Sierra, M. R., & Coello, C. A. C. (2004). *A new multi-objective particle swarm optimizer with improved selection and diversity mechanisms: Technical report of CINVESTAV-IPN*.

Sun, Y., Yen, G. G., & Yi, Z. (2018). Igd indicator-based evolutionary algorithm for many-objective optimization problems. *IEEE Transactions on Evolutionary Computation*, *23*(2), 173–187.

Wagner, M., & Neumann, F. (2013). A fast approximation-guided evolutionary multi-objective algorithm. In *Proceedings of the 15th annual conference on genetic and evolutionary computation* (pp. 687–694).

While, L., Hingston, P., Barone, L., & Huband, S. (2006). A faster algorithm for calculating hypervolume. *IEEE Transactions on Evolutionary Computation*, *10*(1), 29–38.

Yang, S., Li, M., Liu, X., & Zheng, J. (2013). A grid-based evolutionary algorithm for many-objective optimization. *IEEE Transactions on Evolutionary Computation*, *17*(5), 721–736.

Yuan, Y., Xu, H., Wang, B., & Yao, X. (2015). A new dominance relation-based evolutionary algorithm for many-objective optimization. *IEEE Transactions on Evolutionary Computation*, *20*(1), 16–37.

Zhang, Q., & Li, H. (2007). Moea/D: a multiobjective evolutionary algorithm based on decomposition. *IEEE Transactions on Evolutionary Computation*, *11*(6), 712–731.

Zhao, W., Zhang, Z., & Wang, L. (2020). Manta ray foraging optimization: An effective bio-inspired optimizer for engineering applications. *Engineering Applications of Artificial Intelligence*, *87*, Article 103300.

Zitzler, E., Deb, K., & Thiele, L. (2000). Comparison of multiobjective evolutionary algorithms: Empirical results. *Evolutionary Computation*, *8*(2), 173–195.

Zitzler, E., Laumanns, M., Thiele, L., et al. (2001). Spea2: Improving the strength pareto evolutionary algorithm. In *Eurogen, Vol. 3242* (pp. 95–100).

Zitzler, E., & Thiele, L. (1999). Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach. *IEEE Transactions on Evolutionary Computation*, *3*(4), 257–271.

Zouache, D., Abdelaziz, F. B., Lefkir, M., & Chalabi, N. E.-H. (2019). Guided moth–flame optimiser for multi-objective optimization problems. *Annals of Operations Research*, 1–23.

Zouache, D., Arby, Y. O., Nouioua, F., & Abdelaziz, F. B. (2019). Multi-objective chicken swarm optimization: A novel algorithm for solving multi-objective optimization problems. *Computers & Industrial Engineering*, *129*, 377–391.

Zouache, D., Moussaoui, A., & Abdelaziz, F. B. (2018). A cooperative swarm intelligence algorithm for multi-objective discrete optimization with application to the knapsack problem. *European Journal of Operational Research*, *264*(1), 74–88.