Applied Soft Computing 144 (2023) 110472

Contents lists available at ScienceDirect

Applied Soft Computing

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

A comparative study on evolutionary multi-objective algorithms for next release problem

Iman Rahimi [a], Amir H. Gandomi [a,b,*], Mohammad Reza Nikoo [c], Fang Chen [a]

[a] Faculty of Engineering & Information Technology, University of Technology Sydney, Australia
[b] University Research and Innovation Center (EKIK), Óbuda University, 1034 Budapest, Hungary
[c] College of Engineering, Department of Civil and Architectural Engineering, Sultan Qaboos University, Muscat, Oman

ARTICLE INFO

Article history:
Received 18 November 2022
Received in revised form 30 March 2023
Accepted 21 May 2023
Available online 30 May 2023

Keywords:
Multi-objective
Next release problem
Evolutionary algorithm
Software engineering

ABSTRACT

The next release problem (NRP) refers to implementing the next release of software in the software industry regarding the expected revenues; specifically, constraints like limited budgets indicate that the total cost corresponding to the next software release should be minimized. This paper uses and investigates the comparative performance of nineteen state-of-the-art evolutionary multi-objective algorithms, including NSGA-II, rNSGA-II, NSGA-III, MOEAD, EFRRR, tDEA, KnEA, MOMBIII, SPEA2, RVEA, NNIA, HypE, ANSGA-III, BiGE, GrEA, IDBEA, SPEAR, SPEA2SDE, and MOPSO, that can tackle this problem. The problem was designed to maximize customer satisfaction and minimize the total required cost. Three indicators, namely hyper-volume (HV), spread, and runtime, were examined to compare the algorithms. Two types of datasets, i.e., classic and realistic data, from small to large scale, were also examined to verify the applicability of the results. Overall, NSGA-II exhibited the best CPU run time in all test scales, and, also, the results show that the HV and spread values of 1st and 2nd best algorithms (NNIA and SPEAR), for which most HV values for NNIA are bigger than 0.708 and smaller than 1, while the HV values for SPEAR vary between 0.706 and 0.708. Finally, the conclusion and direction for future works are discussed.

1. Introduction

The next release problem (NRP) refers to implementing the next release of software in the software industry. The problem arises from software companies' needs, which aim to develop and maintain the software systems that have been sold to customers. In previous studies, only single objective formulations were taken into account. However, in a multi-objective formulation, the software engineer aims to optimize at least two objectives that may be in conflict with each other. The reason for considering a multi-objective formulation is that requirements engineering typically involves dealing with complex and often conflicting demands, and the software engineer must therefore strive to achieve a reasonable balance between them. The problem is constrained by the software systems' total cost, whereby the objectives include maximizing total customer satisfaction and minimizing the total cost [1,2]. Meeting each requirement incurs a particular cost, and also fulfilling each requirement generates some benefit for the software development company. Companies are faced with the problem mentioned above when their customers request an extensive range of software requirements, some of which necessitate other requirements. Besides, depending on their ability to meet such requests, customers recognize companies at different levels of importance. The challenge is to choose the group of requirements that yields the greatest total benefit while keeping the required cost at a minimum.

The above-mentioned problem is also known as a cost-profit analysis problem [3], for which a Pareto optimal solution is an exciting approach. However, it would be hard for a decision-maker to find a suitable solution and determine how much cost would be acceptable for a corresponding increase in profit. Nowadays, firms developing and improving software structures and features must be identified and added as part of the next release. Hence, the companies would like to select these features to ensure the demands of their customer base are satisfied.

Since introducing the next released problem, only a few papers have studied the exact solution methods for optimization [4–6] and other studies related to interactive optimization and machine learning applied to the next release problem [7]. Other approaches, metaheuristics, have been used widely; for example, Chaves-González and Pérez-Toledano [8] used an adaptive multi-objective version of differential evolution for the multi-objective next release problem. do Nascimento Ferreira et al. [9] proposed a user-interactive model for the next release problem utilizing

* Corresponding author at: Faculty of Engineering & Information Technology, University of Technology Sydney, Australia.
E-mail address: gandomi@uts.edu.au (A.H. Gandomi).

https://doi.org/10.1016/j.asoc.2023.110472
1568-4946/

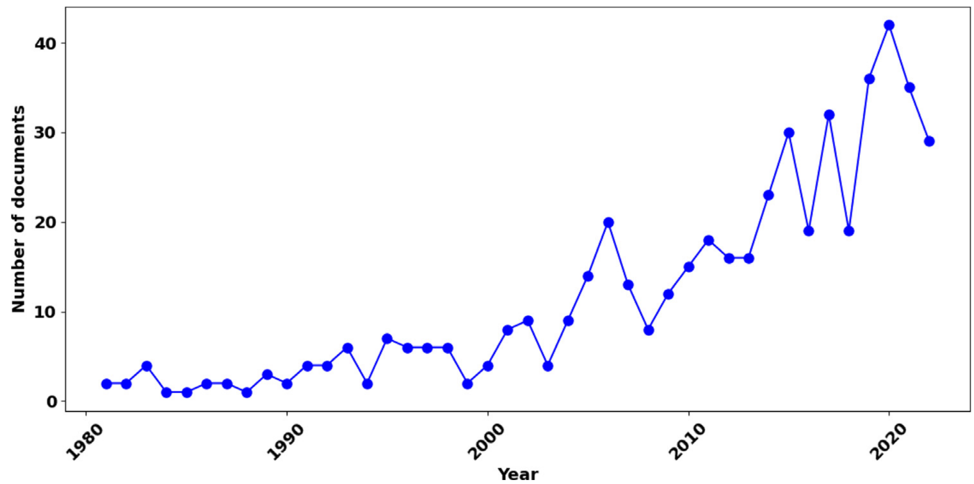


Fig. 1. Trend of published documents on the next release problem since 1981 (database: Scopus).

ant colony optimization. The proposed model allows users to specify which requirements they want to include or exclude in the upcoming release. Ghasemi et al. [10] developed multi-objective variants of the Grey Wolf Optimizer and the Whale Optimization Algorithm with the aim of solving the bi-objective version of NRP. They planned to incorporate the cost-to-score ratio measure and the roulette wheel mechanism to address the constraints present in the problem effectively. By doing so, the authors provided a more efficient and effective solution approach for the bi-objective NRP. Although classic algorithms can find the optimal solution for some special problems, as the number of customers grows so the problem will become complicated. Therefore, as an NP-hard problem, in this study, nineteen state-of-the-art evolutionary algorithms are used to find high-quality solutions as they are more common because of their pros, such as robustness and high flexibility in implementation.

Fig. 1 illustrates the trend of published documents since 1981 filtered using “Next release problem” and “Software” keywords. It can be seen that the focus on the next release problem optimization has increased significantly.

The remainder of this work is planned as follows. Section 2 defines the related works. Then, Section 3 illustrates the methodology. In Section 4 the results of the work are shown and Section 5 shows a summary of the findings and conclusions.

2. Related works

The following subsections provide an overview of the problem statements and evolutionary algorithms used to tackle the above-mentioned problem.

2.1. Multi-objective background

This section presents some definitions of multi-objective concepts which is related to this study. To achieve this aim, the concept of multi-objective optimization problems (MOOPs) will be defined. For this definition, without losing the generality, we assume maximizing all objectives is the goal of optimization.

Eqs. (1)–(3) present a simplified MOOP:

$$maximizing\ F(x) = (f_1(x), \ldots, f_t(x)) \tag{1}$$

$$\text{s.t. } h_i(x) \le 0 \text{ for i} \in \{1, \ldots, n\} \tag{2}$$

$$g_j(x) = 0 \text{ for j} \in \{1, \ldots, m\} \tag{3}$$

where $F(x)$ is the objective vector that consists of several objectives (t is the number of objective functions); n and m are numbers of inequality and equality constraints, and x is the decision variable subjected to lower bound and upper bound vectors. Rather than producing a single solution, these equations yield several optimal solutions instead, which are called Pareto optimal solutions. Without losing the generality, suppose that for t-objective problem in (1)–(3), there exist two feasible solutions namely, x_1 and x_2. We call x_2 is better than x_1 if the following conditions are being held:

$$\forall i : f_i(x_1) \le f_i(x_2) \text{ and } \exists\, j : f_j(x_1) < f_j(x_2) \tag{4}$$

In this case, we say that x_2 dominates x_1 or equivalently x_2 is better than x_1. If x_2 is not dominated by any other feasible solution, it is called Pareto-optimal solution of the t-objective problem in (1)–(3). The set of all Pareto-optimal solutions forms the trade-off surface in the objective space.

2.2. Problem statement: the next release problem (NRP)

The NRP is also considered a combinatorial optimization problem [1,11]. Zhang et al. [2] introduced a multi-objective next release problem (MONRP) and provided some benchmark data to analyze the proposed model. Because of the fact minimizing a given function, for example, (f), is the same as maximizing (-f), the proposed model by [2] could be written as follows:

$$Maximize\ \ f_1(x) = \sum_{i=1}^{n} score_i.x_i \tag{5}$$

$$Maximize\ \ f_2(x) = -\sum_{i=1}^{n} \cos t_i.x_i \tag{6}$$

$$x_i \in 0, 1 \tag{7}$$

An important assumption is that all requirements are independent. The decision vector $X = \{x_1, x_2, \ldots, x_n\}$ presents whether the requirements are satisfied in the next release of the software. The objective functions are as follows: (1) maximizing customer satisfaction and (2) minimizing the total required cost. Constraint (3) shows that the decision variables are binary. As it was mentioned above, the NRP aims to maximize customer profits along with minimizing required costs from a set of dependent requirements under budget constraints. Using the NRP, engineers can decide to balance between customer profits and company profits. Moreover, as a combinatorial optimization problem, it has been proved that NRP is NP-hard even if it is a basic problem and the customer requirements are independent [12,13].

This paper is the first comprehensive comparative study in the field of the next released problem. Although many studies addressed the problem and solved it with some evolutionary algorithms (EAs), this study is the first work that uses several new EAs to solve the next released problem with two classic and realistic data.

2.3. Solution approaches

In multi-objective optimization (MOO), there are two main ideas known as the Pareto dominance and the Pareto front. In this concept, there is no unique optimal solution for a problem, but Pareto front of solutions could be found [14,15], which optimize the objective functions along with the constraints. For these high-quality solutions, two properties should be satisfied; first, every two solutions should be non-dominated, and the second property is that any other solution found should be dominated by at least one solution in the set [14–16].

Since metaheuristics do not require concavity or convexity and also can produce several alternative solutions in a single run (i.e., evolutionary algorithms) [17], they are often used to tackle multi-objective combinatorial optimization problems [18–21]. Additionally, metaheuristics can integrate with specific decomposition algorithms [22] and, generally, many metaheuristics have been developed to deal with some MOOPs [20,21,23–26].

Zhang et al. [2] presented MONRP and provided some benchmark data to analyze the proposed model. Four solution techniques, namely Pareto GA, Single-objective GA, Random Search, and NSGA-II, were applied during their study.

Cai et al. [27] applied a multi-objective evolutionary algorithm (MOEA), NSGA-II, Strength Pareto Evolution Algorithm (SPEA2), random search, a multi-objective version of Invasive Weed Optimization (IWO/MO), and a proposed IWO/MO2. The authors utilized two types of datasets for the problem mentioned above: the first data sets include random data, and the second one is from Motorola [28]. In the aforementioned paper, MOEAs had better performance than random search. Amongst four other algorithms, IWO/MO outperformed other MOEAs on a large scale (for the random data).

Herein, a set of state-of-the-art evolutionary algorithms were elected to implement on MONRP. The comparison algorithms used in this study have been shown to be effective in many applications (e.g., NSGA-II has been cited in over 40,000 publications); Also, these algorithms have been around for a while, and their performance has been well-documented in the literature. This means that we can be confident that these algorithms are capable of finding good solutions to many multi-objective optimization problems.

According to the works of Li et al. [29], Hedayat et al. [30], Wagner and Zitzler [31], Li and Zhang [32], Behmanesh et al. [16] and Bader and Zitzler [33], the EAs are categorized into several groups as follows: (a) Indicator-based, (b) reference set-based, (c) Neighbor-based, (d) Pareto-based, (e) Decomposition-based, (f) diversity, and (g) Preference-based. In the study of [34], three main categories have been introduced: the first group is Indicator-based, which Diversity and Preference-based could potentially be included in the "indicator-based" category, as diversity and preference measures are often used as indicators in many-objective optimization. The second group is dominated-based, which Reference set-based, Pareto-based, and Neighbor-based could potentially fall under the umbrella of "dominated-based" algorithms, as they both involve some form of comparison or ranking of solutions based on dominant relationships. The third group is decomposition-based. These categories are shown in Table 1. The above-mentioned algorithms include NSGA-II, MOEA/D, SPEA2, and NNIA from a set of multi-objective evolutionary algorithms; MOMBI-II, KnEA, NSGA-III, tDEA, EFRRR, HypE, PICEAg, GrEA, ANSGA-III, SPEA2+SDE, BiGE, I-DBEA, SPEA/R, and RVEA from a set of many-objective evolutionary algorithms; Reference-point-based NSGA-II (rNSGA-II); and multi-objective particle swarm optimization algorithm (MOPSO).

The Non-dominated Sorting Genetic Algorithm-II (NSGA-II) is one of the most popular evolutionary algorithms and is known as a very efficient algorithm as it employs an elitist principle and a diversity-preserving mechanism [35,36].

Coello and Lechuga [41] proposed MOPSO, in which particles follow the concept of Pareto dominance to determine the flight direction. In this way, the particles maintain the global repository that other particles could use later to guide their own flight. MOPSO has been used widely in continuous and discrete optimization problems [52–55].

Baker et al. [28] proposed a many-objective evolutionary algorithm based on the NSGA-II framework known as NSGA-III, emphasizing non-dominated solutions to be close to a set of provided reference points. NSGA-III is based on the predefined multiple targeted search principle, such that a set of Pareto-optimal points could be found by using points corresponding to each reference point [39] proposed the EFR-RR algorithm that enhances two decomposition-based MOEAs, namely MOEA/D and EFR [56], and also maintains the desired diversity of solutions.

Cai et al. [27] suggested a Knee Point Driven Evolutionary Algorithm (KnEA) for many-objective optimization, which enhances the convergence performance. Recently, performance indicators have been introduced as a selection approach in multi-objective optimization. For instance, Hernández Gómez and Coello Coello [44] proposed an improved version of metaheuristics called MOMBI-II based on the R2 indicator, considering two important aspects, i.e., computational cost and Pareto compatibility. In the mentioned paper, MOMBI-II outperformed some evolutionary algorithms, specifically NSGA-III, on test problems known as DTLZ and WFG. Zitzler et al. [40] presented an improved version of the Strength Pareto evolutionary algorithm (SPEA2) and compared the results with those of other evolutionary algorithms, such as NSGA-II, on some classic test problems (DTZ and knapsack). It was concluded that SPEA2 has better performance over NSGA-II, specifically in higher dimensional objective spaces. Yuan et al. [39] introduced a new evolutionary multi-objective (EMO) algorithm, the Territory Defining Evolutionary Algorithm (tDEA), and tested its performance against well-known MOEAs in the literature. The results revealed that tDEA outperformed the other algorithms.

Said et al. [46] established a new dominance relation for interactive evolutionary multi-criteria decision-making (r-NSGA-II) and compared the proposed algorithm to other EMO algorithms. Gong et al. [45] suggested a novel non-dominated neighbor-based selection approach (NNIA) in which the proposed algorithm uses an immune-inspired operator, two heuristic search operators, and elitism. The algorithm introduced by Gong et al. [45] was compared with some evolutionary algorithms, including NSGA-II and SPEA2, to solve some benchmarks, such as DTLZ, and ZDT. The results showed that NNIA performs better when considering convergence metrics, coverage of two sets, and spacing as performance metrics.

The reference vectors could be used for two key applications in multi-objective optimization: (1) to decompose the original optimization problem and (2) to clarify user preferences of the whole front. Regarding the above-mentioned matter, Cheng et al. [34] proposed RVEA, a reference-vectors approach, to decompose the original MOOP. The RVEA was adopted to maintain a good balance between convergence and diversity and tested against some state-of-the-art algorithms, namely MOEA/DD, NSGA-III,

Table 1
Multi-objective evolutionary algorithms included in this study.

EA/Group	Indicator-based			Dominated-based			Decomposition-based
	a[a]	f	g	b	c	d	e
KnEA [37]		●				●	
NSGA-III [38]				●		●	
tDEA [39]				●		●	
SPEA2 [40]						●	
MOPSO [41]						●	
NSGA-II [35]						●	
EFR-RR [42]							●
MOEA/D [43]						●	●
RVEA [34]				●			●
MOMBI-II [44]	●						
NNIA [45]					●		
rNSGA-II [46]				●		●	
ANSGA-III [38]				●			●
BiGE [47]	●					●	
GrEA [48]						●	
I-DBEA [49]						●	●
SPEA2+SDE [50]						●	
HypE [33]	●					●	
SPEA/R [51]				●		●	

[a]a: indicator-based, b: reference-based, c: Neighbor-based, d: Pareto-based, e: Decomposition-based, f: Diversity-based, g: Preference-based.

MOEA/D-PBI, GrEA, and KnEA. The experimental results on various benchmark test problems, including DTLZ1-DTLZ4, SDTLZ1, SDTLZ3, and WFG1-WFG9, indicate that RVEA is effective and cost-efficient. Zhang and Li [43] proposed an MOEA based on decomposition (MOEA/D). The algorithm decomposes the multi-objective optimization problem into a number of scalar sub-problems and optimizes all sub-problems simultaneously, resulting in generally lower computational complexity. The author applied the algorithm mentioned above to a multi-objective 0-1 knapsack problem and showed that MOEA/D outperformed or performed similarly to NSGA-II.

3. Methodology

Although many-objective EAs are usually used for problems with more than three objective functions, some efforts have been addressed to employ these algorithms for single and multi-objective optimization problems [57–61]. The following subsections provide the performance evaluation metrics and data collection, which are parts of the methodology section.

3.1. Performance evaluation metrics

This part presents some of the main performance metrics used in this work. The aim of using several performance metrics is to compare the quality of the solutions set found by the suggested algorithms. Although several main metrics of MOOPs have been presented in the literature, including generational distance (GD), inverted generational distance (IGD), spread, hypervolume (HV), normalized HV (NHV), spacing measure (SM), diversity metric release [5,16,25], in this study, HV and spread, as two well-known metrics, were implemented. Three main aspects can be considered for metrics. These aspects are accuracy, diversity and cardinality [62]: accuracy refers to the convergence aspect, diversity refers to the distribution and the extent of the obtained solutions, and cardinality refers to the number of solutions that exist in an approximation set. HV and spread, as two well-known metrics, possess these three aspects. As such, the spread metric, which uses the euclidean distances between the extreme solutions and the boundary of the obtained solution, is a performance indicator that measures the distribution and the extent of the spread obtained among the solution in an approximation set. Therefore, the spread metric is an excellent indicator to consider the diversity of solutions in the approximation set.

- Hyper-volume (HV) has recently been addressed as an indicator by many researchers in the context of MOEA to evaluate the performance of search algorithms [63]. The bigger value of HV of the approximation indicates that its Pareto set completely dominates other approximations, which means that the HV indicator shows a set quality measure considered the dominated slice of the objective space [64,65].
- Spread: based on the Euclidean distances between the extreme solutions and the boundary solutions of the obtained non-dominated set [15]. The smaller value of the spread of the approximation, the better the distribution.

The framework of the methodology is displayed in Fig. 2.

3.2. Data collection

Two types of data were tested in this paper to evaluate the algorithms. The first group includes a classic set with 5 test instances (nrp1-nrp5) [12,13]. The five datasets, nrp1 to nrp5, correspond to individual problem instances for the bi-objective NRP. When considering the single-objective scenario, each dataset produces two identical instances except for the cost constraint limit. This limit is determined by multiplying the sum of the total costs with a coefficient of either 0.3, 0.5 or 0.7. The data is generated using a multi-level methodology, whereby each level comprises predetermined requirements. The cost of each requirement and the number of associated child requirements are uniformly selected from a range that is dependent on the level [1]. The second group includes a realistic dataset suggested by Xuan et al. [13], which was suggested to use the bug repositories for the Eclipse, Mozilla and Gnome open-source projects. In this way, bugs are viewed as requirements, while stakeholder requests are equated to users' comments on the bugs. A bug's severity determines requirement's cost, while the profit per stakeholder is randomly selected from a predetermined range. These data were gathered from Eclipse (Nrp-e1 to Nrp-e4), Gnome (Nrp-g1 to Nrp-g4), and Mozilla (Nrp-m1 to Nrp-m3). Table 2 presents the corresponding customers, requirements, cost, and profits for both types (classic and realistic) with a specific number of customers (m) and the number of requirements (n), including cost and profit for each set.

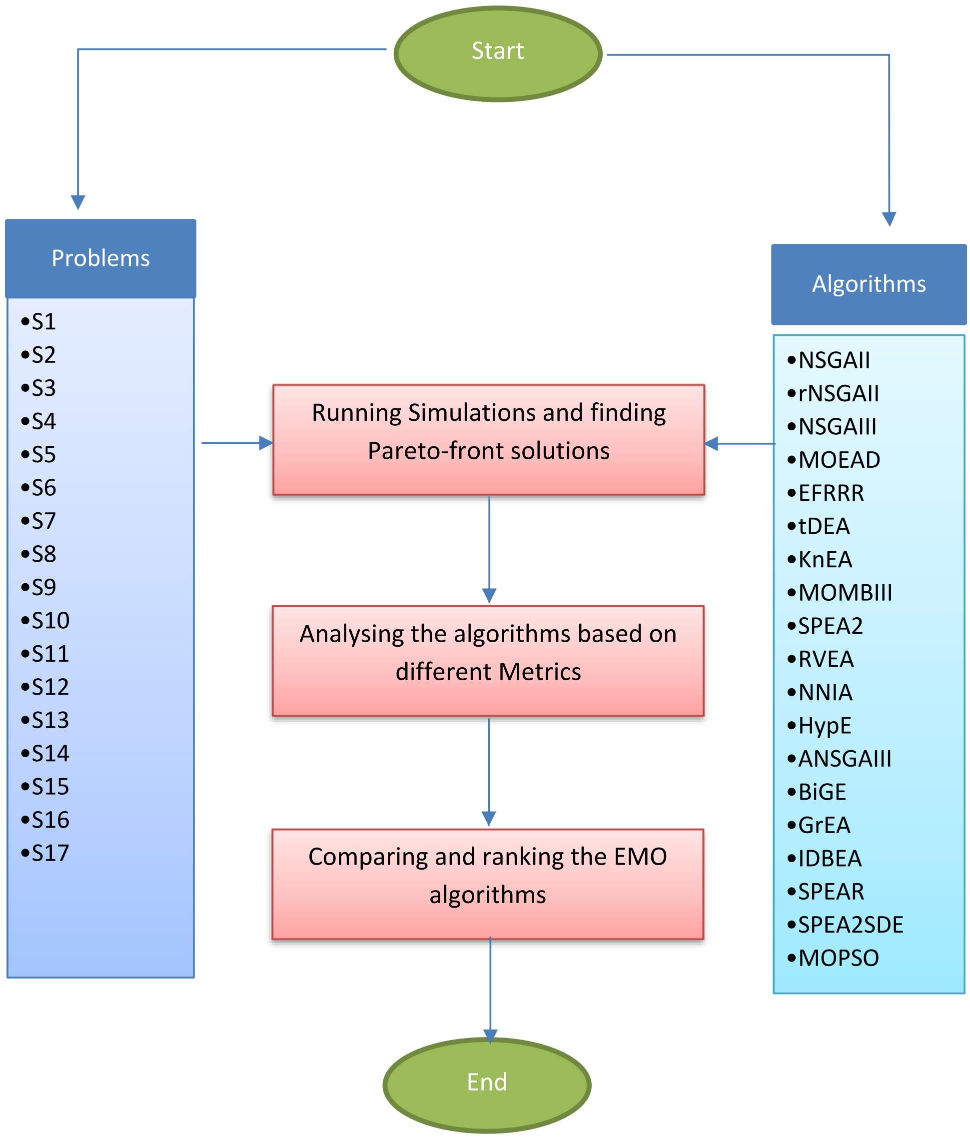


Fig. 2. The framework methodology.

Table 2
Test sets for classic and realistic data.

Test sets	Name	Customers (m)	Requirements (n)	Cost	Profit
S1	nrp1	100	140	5–10	10–50
S2	nrp2	500	620	5–15	10–50
S3	nrp3	500	1500	5–10	10–50
S4	nrp4	750	3250	5–15	10–50
S5	nrp5	1000	1500	3–5	10–50
S6	Nrp-e1	536	3502	1–7	10–50
S7	Nrp-e2	491	4254	1–7	10–50
S8	Nrp-e3	456	2844	1–7	10–50
S9	Nrp-e4	399	3186	1–7	10–50
S10	Nrp-g1	445	2690	1–7	10–50
S11	Nrp-g2	315	2650	1–7	10–50
S12	Nrp-g3	423	2512	1–7	10–50
S13	Nrp-g4	294	2246	1–7	10–50
S14	Nrp-m1	768	4060	1–7	10–50
S15	Nrp-m2	617	4368	1–7	10–50
S16	Nrp-m3	765	3566	1–7	10–50
S17	Nrp-m4	568	3643	1–7	10–50

4. Results

This work employed PlatEMO [66], to test and implement the different evolutionary algorithms for various problems. Several algorithms from a set of multi- and many-objective optimization have been selected. Although many-objective EAs are usually implemented for problems with more than three objective functions, some efforts have been addressed to address these algorithms for single- and multi-objective optimization problems since they can explore the solution space more thoroughly and efficiently. Also, in this study, different types of algorithms have been addressed, these include: Indicator-based, (b) reference set-based, (c) Neighbor-based, (d) Pareto-based, (e) Decomposition-based, (f) diversity, and (g) Preference-based. Two types of parameter settings have been used: general parameter settings, which be used for all algorithms (e.g. population size, number of evaluations) and some specific parameter tuning for certain algorithms. For general settings, the following parameters have been addressed:

Number of runs: 10 times for each algorithm on each test scale and take the average of the values,
Population size (N): 200,

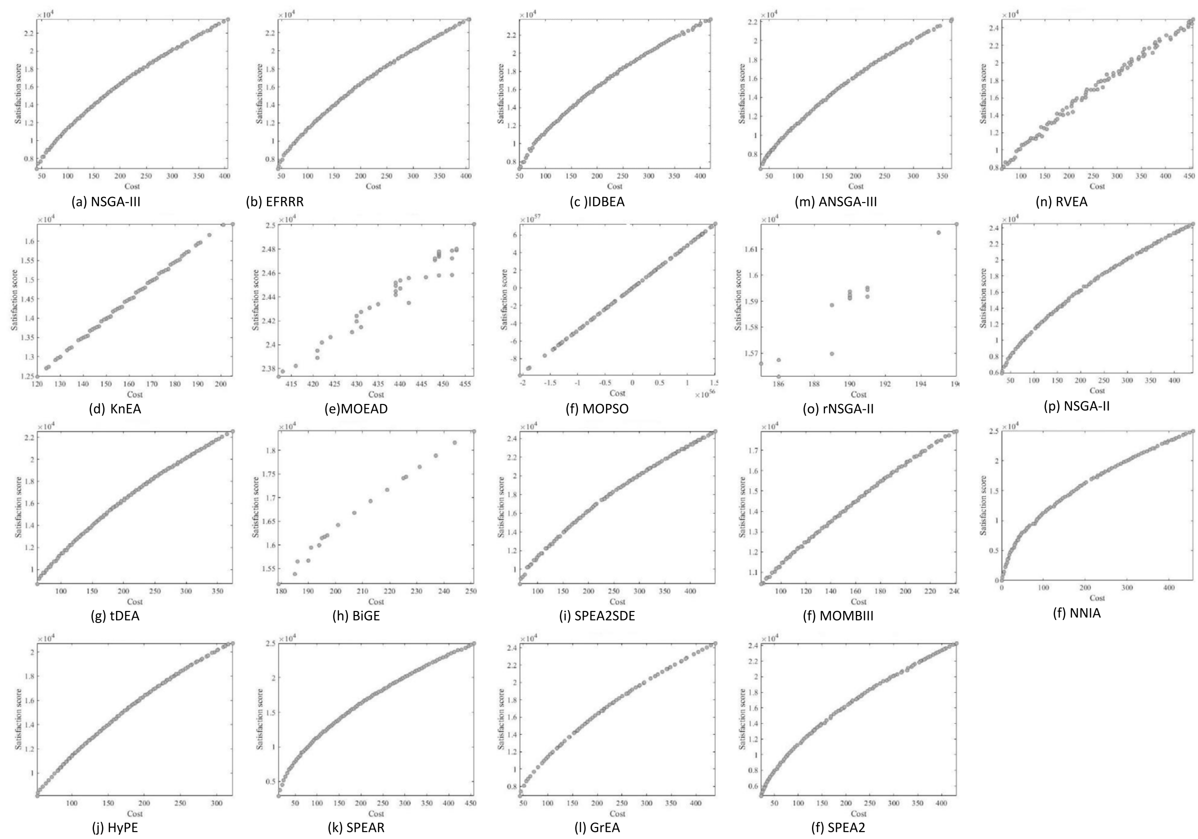


Fig. 3. Comparison of solutions found by different algorithms on MONRP.

Table 3
Parameters setting of some specific evolutionary algorithms.

Algorithm	Parameters setting
NNIA	nA = 20, nC = 100
RVEA	alpha = 2, fr = 0.1
MOPSO	Div = 10
MOMBIII	Alpha = 0.5, epsilon = 0.001, record = 5
MOEAD	Type = 1
HypE	nSample = 10,000
EFRRR	K = 2
KnEA	Rate = 0.5
rNSGA-II	Delta = 0.1
GrEA	Div = 10

Number of objectives: 2,
Number of evaluations (E): 100,000, and the extra parameter settings for the specific algorithms are provided in Table 3. Also, the other parameters have been set by the program defaults.

There are 19 algorithms that run 10 times for each test scale (17 test scales in general), resulting in 3230 experiments. It is noteworthy to mention that the experiments have been run on all test scales and because of the fact there are the same results, only S1 test scale results have been provided in this paper (Figs. 3 and 4). As mentioned, the aforementioned algorithms are classified into two major groups, namely, multi-objective and many-objective evolutionary algorithms. From a set of multi-objective evolutionary algorithms NSGA-II, MOEA/D, SPEA2, and NNIA are chosen and from a set of many-objective evolutionary algorithms MOMBI-II, KnEA, NSGA-III, tDEA, EFRRR, HypE, PICEAg, GrEA, ANSGA-III, SPEA2+SDE, BiGE, I-DBEA, SPEA/R, and RVEA are selected. In addition, two other algorithms rNSGA-II as a reference-point-based NSGA-II; and multi-objective particle swarm optimization algorithm (MOPSO) are also implemented in our experiments. Fig. 3 (a–s) shows the solutions found by the previously mentioned algorithms for the S1 test scale, revealing that many-objective evolutionary algorithms, such as EFRRR, NSGA-III, and IDBEA, could find a large number of Pareto solutions for the S1 test scale, while reference-point-based NSGA-II (rNSGA-II), many-objective evolutionary algorithmBiGE and multi-objective evolutionary algorithm MOEAD performed poorly. Some other algorithms, such as MOPSO and RVEA, were more robust than rNSGA-II and MOEAD but did not work well like the earlier mentioned algorithms.

Fig. 4 shows the changes in HV values against 100,000 evaluations for the 1st, 2nd, and 3rd best algorithms (See Appendix file A for other algorithms). As it can be seen from Fig. 4, NNIA, SPEAR, and SPEA2 possess the best HV values among the other algorithms, for which most HV values for NNIA are bigger than 0.708 and smaller than 1, while the HV values for SPEAR vary between 0.706 and 0.708, and HV values for SPEA2 vary between 0.663 and 0.703.

It is clear that the amount of HV values for NSGA-III, ANSGA-III, tDEA, HypE, SPEA2SDE, EFRRR, NNIA, MOMBI-II, and SPEAR gradually increased over the evaluation scales. In contrast, the values of HV for rNSGA-II sharply declined over the evaluation scales. Based on the results of deep analysis, it can be said that the values of HV for NSGA-II, RVEA, GrEA, MOEAD, and SPEA2 sharply increased, while MOPSO exhibited some fluctuations (Appendix). One of the key factors in the next generation population and achieving Pareto is selection. The Pareto solution consists of the non-dominated solutions that are obtained in the final iteration. In some cases, the Pareto archive contains a huge number of non-dominated solutions, which are out of computer memory limit. Some techniques could be invented to remove some of the non-dominated solutions and maintain the diversity of the solution

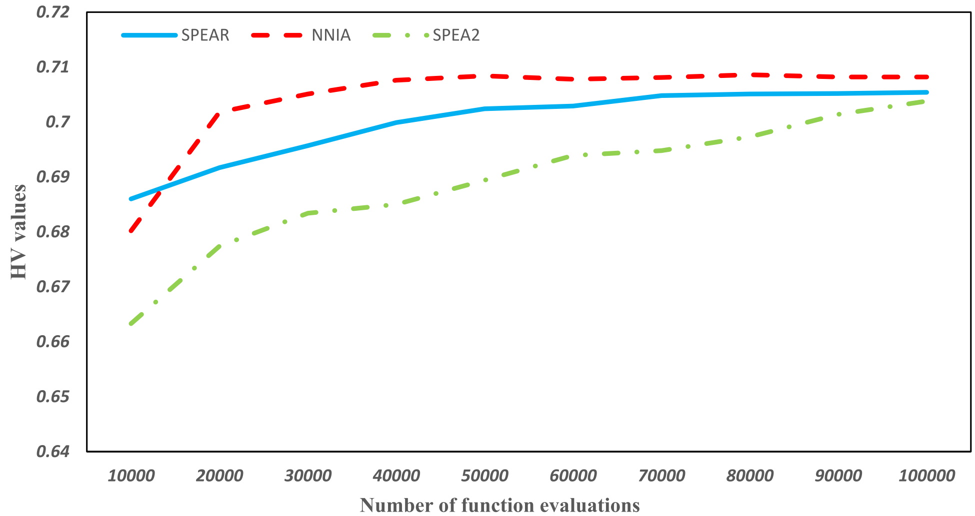


Fig. 4. HV values vs. number of function evaluations on MONRP for the 1st, 2nd, and 3rd best algorithms.

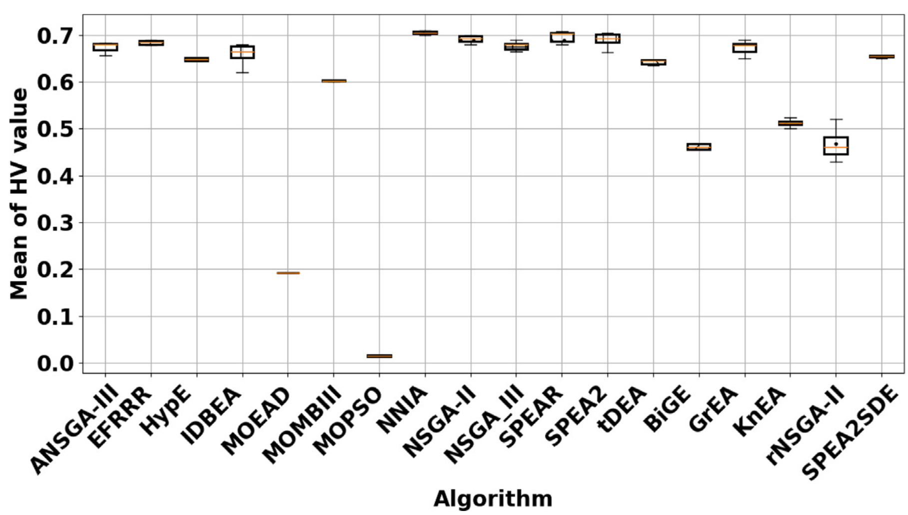


Fig. 5. Boxplot of the HV obtained by the algorithms in the test problems (S1).

as much as possible resulting in drop of the HV values in later iterations in some implementations such as KnEA, MOPSO, BiGE, SPEA2SDE, ANSGA-III, and rNSGA-II.

Figs. 5 and 6 present the mean of HV and spread values for the sets (S1). Using these metrics, three main aspects of performance, namely, accuracy, diversity, and cardinality are considered. In Figs. 5 and 6, each rectangle's size shows the Interquartile Range (IQR). The short line at the end of each rectangle presents the minimum and maximum values, and the short line inside the boxes represents each rectangle's median. Also, Fig. 5 shows that NNIA, from a set of multi-objective evolutionary algorithms, owns the best value. In Fig. 6, NNIA also presents the best value of the spread and possesses good diversity, while GrEA, from a set of many-objective evolutionary algorithms, illustrates the worst value of the spread. Tables 4 and 5 depict the mean and standard deviation of spread and HV indicators for all algorithms over all test scales (S1–S17), respectively[1]. According to the spread mean and standard deviation in Table 4, it is clear that NNIA possesses most of the best values (gray color), while GrEA owns the worst spread values for all datasets. Also, the light gray in Tables 4 and 5 show the 2nd best values among the algorithms. Again, it is apparent that NNIA owns the best value of HV for all test datasets, while MOPSO possesses the worst value.

Table 6 presents the 1st and 2nd best values of CPU run time for each dataset. It can be deduced that NSGA-II has the best performance for all datasets except S4, for which rNSGA-II ranks best among all algorithms. Also, rNSGA-II possesses the 2nd best performance for S1–S2, S5–S7, and S11–S17, while tDEA has the 2nd performance for S4, S8, and S10. For other datasets, S3 and S9, RVEA has the 2nd performance among all algorithms.

5. Summary of findings

This paper addresses 19 state-of-the-art evolutionary algorithms to can tackle the multi-objective next-released problem. The algorithms discussed earlier are classified into different categories: (a) Indicator-based, (b) reference set-based, (c) Neighbor-based, (d) Pareto-based, (e) Decomposition-based, (f) diversity,

[1] The signs '+' presents that the result is significantly better of the result in the control column while the sign '−' shows significantly worst, and the sign '=' illustrates that statistically similar to the result in the control column.

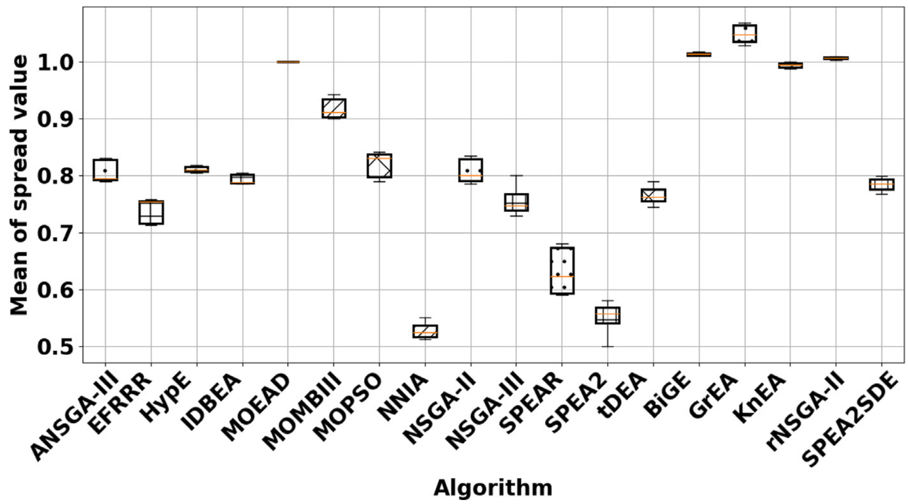


Fig. 6. Boxplot of the spread obtained by the algorithms in the test problems (S1).

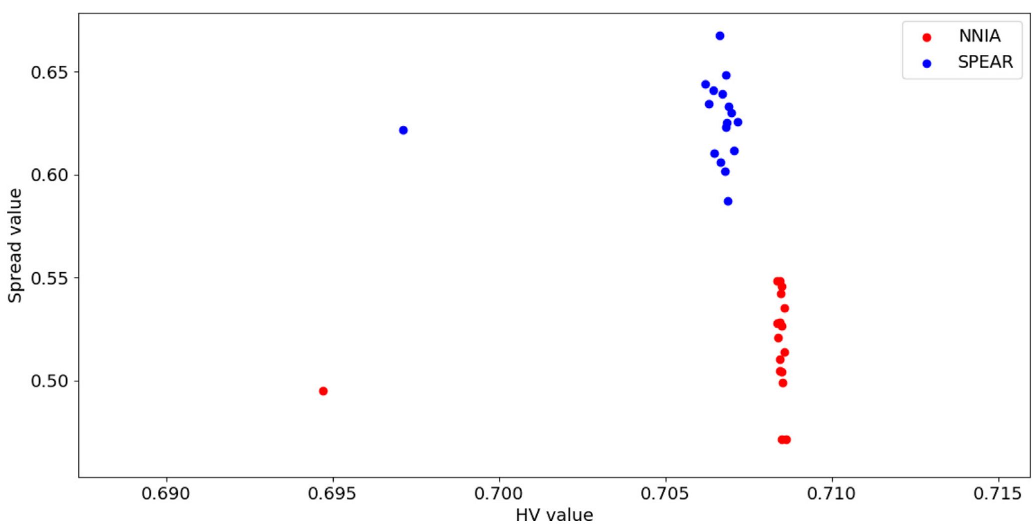


Fig. 7. HV and spread values of 1st and 2nd best algorithms.

and (g) Preference-based. NSGA-II, MOEA/D, SPEA2, and NNIA from a set of multi-objective evolutionary algorithms; MOMBI-II, KnEA, NSGA-III, tDEA, EFRRR, HypE, PICEAg, GrEA, ANSGA-III, SPEA2+SDE, BiGE, I-DBEA, SPEA/R, and RVEA from a set of many-objective evolutionary algorithms; Reference-point-based NSGA-II (rNSGA-II); and multi-objective particle swarm optimization algorithm (MOPSO) have been selected.

Table 7 presents the summary of indicators' performance, in which the 1st and 2nd best performance for each indicator is shown. In the literature analysis, SPEA2 outperforms NSGA-II in higher-dimensional spaces. NNIA has been reported to be better than NSGA-II and SPEA-2 in DTLZ and ZDT, as it is expected, NNIA is better than the other proposed above-mentioned algorithms in the paper. It is worth mentioning that NNIA only chooses minority isolated nondominated individuals in the current population and focuses more on the less-crowded areas of the current Pareto front [45].

Fig. 7 displays the HV and spread values of 1st and 2nd best algorithms (NNIA and SPEAR), for which most HV values for NNIA are bigger than 0.708 and smaller than 1, while the HV values for SPEAR vary between 0.706 and 0.708.

6. Discussion and conclusion

This study evaluated several evolutionary algorithms to solve the MONRP. This problem arises when a developed software system of urgent need, has been sold to customers, and a set of customer requirements must be met. Solving the MONRP problem involves two objectives: maximizing customer profits along with minimizing required costs from a set of dependent requirements under budget constraints. Using the MONRP, developers can decide to balance between customer profits and company profits. Also, MONRP, as a combinatorial optimization problem, has been proved to be known as an NP-hard even if it is a basic problem and the customer requirements are independent. Furthermore, Two types of datasets were examined to verify the simulation results. The first type includes classic data, and the second type involves realistic data, such as Mozilla, Genome, and Eclipse.

Therefore, 19 state-of-the-art EAs, namely NSGA-II, rNSGA-II, NSGA-III, MOEAD, EFRRR, tDEA, KnEA, MOMBIII, SPEA2, RVEA, NNIA, HypE, ANSGA-III, BiGE, GrEA, IDBEA, SPEAR, SPEA2SDE, and MOPSO, were selected and categorized into several groups

Table 4
Spread mean and standard deviation obtained by the algorithms in the test problems.

Test size	ANSGA-III	EFRRR	HypE	IDBEA	MOEAD	MOMBIII	MOPSO	NNIA
S1	0.825 ±0.039 =	0.73602 ±0.023=	0.815± 0.010 =	0.794 ±0.027 =	1.000± 0.000 =	0.917 ±0.026 =	0.831±0.071=	0.548 ±0.038 +
S2	0.821±0.032 =	0.78018 ±0.044=	0.804±0.009 =	0.783 ±0.027 =	1.000± 0.001 =	0.911 ±0.018=	0.778±0.076 =	**0.471± 0.072 +**
S3	0.867±0.067 =	0.74456 ±0.038 =	0.814± 0.014 =	0.811 ±0.041 =	1.000±0.000 =	0.911±0.024 =	0.802±0.066=	0.548±0.059 +
S4	0.814 ±0.015=	0.7573±0.021 =	0.816± 0.007=	0.807 ±0.019 =	1.000±0.001 =	0.915 ±0.020 =	0.777±0.064 =	**0.471±0.044+**
S5	0.814 ±0.027 =	0.76118±0.046 =	0.813± 0.006 =	0.774 ±0.041 =	1. 000±0.001=	0.915±0.013 =	0.804±0.010 =	0.526±0.075+
S6	0.822± 0.031=	0.76678±0.026 =	0.805± 0.013 =	0.757±0.035 =	1.000±0.001 =	0.919±0.014 =	0.804±0.071 =	**0.499± 0.046+**
S7	0.814±0.033 =	0.74051 ±0.036 =	0.813± 0.014 =	0.805±0.053 =	1.000±0.001 =	0.916±0.019=	0.795±0.010=	**0.510±0.058+**
S8	0.805 ±0.026=	0.75091 ±0.029=	0.821±0.0091=	0.774±0.030 =	1.000±0.001 =	0.914±0.020 =	0.756±0.061=	0.545±0.060+
S9	0.823±0.040=	0.73720 ±0.031 =	0.816±0.006 =	0.804±0.044=	1.000±0.000 =	0.915±0.019 -	0.779±0.094 =	**0.542±0.059+**
S10	0.817±0.028=	0.74956 ±0.048 =	0.818± 0.010 =	0.7920.058=	1.000±0.000=	0.910 ±0.015 =	0.784±0.065 =	**0.513 ±0.052 +**
S11	0.807± 0.024=	0.73563±0.021 =	0.812±0.008 =	0.788±0.049=	1.000±0.001 =	0.915±0.015 =	0.762±0.066=	**0.535±0.071+**
S12	0.822 ±0.028=	0.75562±0.043 =	0.811±0.011 =	0.787 ±0.044=	1.000±0.001 =	0.916±0.019 =	0.744±0.070 =	**0.504 ±0.030 +**
S13	0.827 ±0.033=	0.78479 ±0.029=	0.810± 0.011=	0.785 ±0.033 =	1.000±0.001=	0.911±0.020 -	0.794±0.052=	0.527 ±0.061+
S14	0.809± 0.017=	0.74988 ±0.037=	0.809± 0.012 =	0.792±0.044 =	1.000±0.000=	0.916±0.024=	0.821±0.011 =	**0.520± 0.024 +**
S15	0.831± 0.025=	0.73287 ±0.021=	0.822± 0.010 =	0.793 ±0.038=	1.000±0.001 =	0.924 ±0.017=	0.807±0.089 =	0.528± 0.081+
S16	0.805 ±0.037=	0.74194 ±0.044 =	0.806± 0.011 =	0.792 ±0.022 =	1.000±0.001 =	0.903±0.022 =	0.837±0.010 =	**0.504± 0.050+**
S17	0.806± 0.029 =	0.76040 ±0.034 =	0.811 ±0.011 =	0.798 ±0.054=	1.000±.000=	0.906±0.022=	0.796±0.090 =	**0.494± 0.063 +**

Test size	NSGA-II	NSGA-III	RVEA	SPEAR	SPEA2	tDEA
S1	0.817 ±0.093=	0.762± 0.036 =	0.831 ±0.026 =	0.648 ± 0.066 +	**0.508± 0.046 +**	0.768± 0.024 =
S2	0.779 ±0.052=	0.761 ± 0.028 =	0.822± 0.023 =	0.601± 0.039 +	0.530± 0.055 +	0.791± 0.034 =
S3	0.822 ±0.038 =	0.752 ± 0.025 =	0.824± 0.026 =	0.641± 0.086 +	**0.521± 0.038 +**	0.778± 0.034 =
S4	0.797± 0.043 =	0.782 ± 0.021 =	0.831± 0.044 =	0.623± 0.064 +	0.537± 0.040 +	0.780± 0.036 =
S5	0.760±0.042 =	0.772 ± 0.031 =	0.815± 0.022 =	0.630± 0.075 +	0.519 ±0.075 +	0.790± 0.043 =
S6	0.772±0.043 =	0.786 ±0.046 =	0.820± 0.029 =	0.611 ± 0.056 +	0.550± 0.067 +	0.771± 0.046 =
S7	0.759 ±0.034 =	0.766± 0.040 =	0.836± 0.046 =	0.634 ±0.038 +	0.534± 0.063 +	0.781± 0.030 =
S8	0.753± 0.045 =	0.745± 0.025 =	0.833± 0.023 =	0.606 ±0.031 +	**0.541 ± 0.056 +**	0.773± 0.022 =
S9	0.786 ±0.074 =	0.763± 0.036 =	0.817± 0.039 =	0.639± 0.047 +	0.562± 0.049 +	0.776± 0.035 =
S10	0.770± 0.055 =	0.744± 0.027 =	0.838 ± 0.021 =	0.587± 0.025 +	0.518± 0.046 +	0.790 ± 0.033 =
S11	0.761± 0.057 =	0.753± 0.038 =	0.841 ±0.025 =	0.625± 0.050 +	0.547± 0.028 +	0.756± 0.019 =
S12	0.772± 0.063 =	0.768± 0.026 =	0.829± 0.020 =	0.643± 0.070 +	0.537 ± 0.048 +	0.795 ± 0.031 =
S13	0.803 ±0.044 =	0.742± 0.029 =	0.842± 0.046 =	0.625± 0.066) +	**0.527± 0.030 +**	0.770 ±0.029 =
S14	0.784± 0.023 =	0.765± 0.035 =	0.827± 0.035 =	0.610 ±0.042 +	0.537 ± 0.041 +	0.797± 0.039 =
S15	0.794± 0.042 =	0.757± 0.040 =	0.826 ± 0.035 =	0.633± 0.034 +	**0.508± 0.044 +**	0.796± 0.035 =
S16	0.780± 0.035 =	0.765 ± 0.039 =	0.836 ± 0.035 =	0.667± 0.072 +	0.520± 0.052 +	0.784 ± 0.029 =
S17	0.776± 0.041 =	0.741 ± 0.039 =	0.818 ±0.031 =	0.621± 0.036 +	0.532± 0.051 +	0.788 ±0.047 =

Test size	BiGE	GrEA	KnEA	rNSGA-II	SPEA2SDE
S1	1.029± (0.019) =	1.048 ± (0.019) =	0.996± (0.008) =	1.006 ± (0.003) =	0.893 ± (0.028)
S2	1.024± (0.009) =	1.053± (0.022) =	0.997± (0.009) =	1.005 ± (0.004) =	0.904± (0.016)
S3	1.024± (0.009) =	1.048 ± (0.015) =	1.001± (0.013) =	1.002 ± (0.003) =	0.906± (0.020)
S4	1.030 ± (0.007) =	1.041 ± (0.015) =	1.000± (0.011) =	1.006± (0.005) =	0.906 ± (0.017)
S5	1.021± (0.009) =	1.041 ± (0.023) =	0.995 ± (0.008) =	1.008± (0.005) =	0.921 ± (0.019)
S6	1.026± (0.010) =	1.042 ± (0.008) =	1.000± (0.010) =	1.006 ± (0.006) =	0.911 ± (0.022)
S7	1.021± (0.006) =	1.043± (0.014) =	1.003± (0.009) =	1.004± (0.004) =	0.912 ± (0.021)
S8	1.026 ± (0.014) =	1.047 ± (0.017) =	0.995 ± (0.008) =	1.004± (0.003) =	0.886± (0.028)
S9	1.024 ± (0.007) =	1.034 ± (0.012) =	0.994 ± (0.013) =	1.006 ± (0.003) =	0.915 ± (0.011)
S10	1.027± (0.006) =	1.048 ± (0.011) =	0.997± (0.010) =	1.004± (0.003) =	0.894 ± (0.024)
S11	1.024± (0.01) =	1.039 ± (0.013) =	1.000± (0.012) =	1.005± (0.003) =	0.898 ± (0.018)
S12	1.022± (0.007) =	1.047 ± (0.014) =	0.952 ± (0.010) =	1.004± (0.003) =	0.895 ± (0.027)
S13	1.022± (0.009) =	1.051 ± (0.001) =	0.973± (0.007) =	1.002± (0.003) =	0.902 ± (0.021)
S14	1.018± (0.007) =	1.040± (0.012) =	0.948± (0.005) =	1.007 ± (0.004) =	0.903± (0.017)
S15	1.032± (0.015) =	1.031± (0.015) =	1.009± (0.010) =	1.006 ± (0.004) =	0.905 ± (0.025)
S16	1.022± (0.009) =	1.047 ± (0.010) =	0.978 ± (0.004) =	1.005± (0.006) =	0.897± (0.018)
S17	1.026± (0.013) =	1.046 ± (0.016) =	0.956± (0 .010) =	1.004± (0.003) =	0.906 ± (0.026)

of EMO algorithms are categorized as follows: Indicator-based, reference set-based, Neighbor-based, Pareto-based, Decomposition-based, diversity, and Preference-based. Moreover, two performance evaluation metrics, namely, HV and spread, were used. Using these metrics, three main aspects of performance, namely, accuracy, diversity, and cardinality were considered.

Moreover, while our study may not have used the most complex benchmark problems, we believe that it adds value to the field by providing a comprehensive and systematic comparison of several state-of-the-art evolutionary algorithms on the NRP problem. Our study also highlights the strengths and weaknesses of different algorithms and identifies areas where further research is needed to improve the performance of evolutionary algorithms on the NRP problem.

As a limitation of our study, in this paper, we have set the operators based on program default; it is interesting to check the performance according to various operators, such as different types of crossover.

The results have been obtained as follows:

- Amongst the proposed algorithms, (a) EFRRR, (d) MOMBIII, (f) NNIA, (g) NSGA-II, (h) NSGA-III, (k) SPEA2, (m) ANSGA-III, (p) IDBEA, (q) PICEAg, (r) SPEA2SDE, (s) SPEAR, (t) HypE, and (l) tDEA could find a large number of Pareto solutions for the S1 test scale, while (i) rNSGA-II, (n) BiGE, and (c) MOEAD showed weak performance. On the other hand, most of the many-objective evolutionary algorithms possess good performance for this problem.
- NNIA possesses the best value for the HV indicator, while MOPSO owns the worst value amongst all algorithms for all test scales.

Table 5
HV mean and standard deviation obtained by the algorithms in the test problems.

Test size	ANSGA-III	EFRRR	HypE	IDBEA	MOEAD	MOMBIII	MOPSO	NNIA
S1	0.679±(0.004) =	0.683 ± (0.005) =	0.647 ± (0.005) =	0.674 ± (0.005) =	0.182± (0.006) -	0.588± (0.011) -	**0.015± (0.001) -**	**0.708± (0.000)** +
S2	0.679± (0.005) =	0.684 ± (0.006) =	0.647± (0.006) =	0.673± (0.005) +	0.184± (0.006) -	0.588 ± (0.012) -	**0.015 ± (0.003) -**	**0.708± (0.000)** +
S3	0.668 ± (0.047) =	0.684 ± (0.004) =	0.646± (0.006) =	0.672± (0.005) =	0.182± (0.006) -	0.591± (0.012) -	**0.016± (0.005) -**	**0.708± (0.000)** +
S4	0.680 ± (0.004) =	0.683 ± (0.005) =	0.648± (0.006) =	0.673 ± (0.004) +	0.184± (0.006) -	0.587 ± (0.012) -	**0.014 ± (0.000) -**	**0.708± (0.000)** +
S5	0.681 ± (0.003) =	0.684± (0.005) =	0.647± (0.006) =	0.674± (0.11) =	0.186± (0.006) -	0.588 ± (0.011) =	**0.0153 ± (0.001) -**	**0.708± (0.000)** +
S6	0.680± (0.004) =	0.684± (0.006) =	0.648± (0.006) =	0.675± (0.004) =	0.181± (0.006) -	0.589 ± (0.013) =	**0.015 ± (0.001) -**	**0.708 ± (0.000)** +
S7	0.680 ± (0.005) =	0.685 ± (0.006) =	0.647± (0.006) =	0.672 ± (0.007) =	0.184± (0.006) -	0.591 ± (0.015) -	**0.015 ± (0.002) -**	**0.708± (0.000)** +
S8	0.680± (0.003) =	0.685 ± (0.004) =	0.645 ± (0.007) =	0.675 ± (0.003) =	0.182± (0.006) -	0.590 ± (0.014) =	**0.015± (0.001) -**	**0.708 ± (0.000)** +
S9	0.680± (0.004) =	0.682 ± (0.006) =	0.650± (0.006) =	0.673± (0.004) =	0.185± (0.006) -	0.590± (0.012) -	**0.015± (0.002) -**	**0.708 ± (0.000)** +
S10	0.680 ± (0.004) =	0.684± (0.004) =	0.646± (0.005) =	0.676± (0.005) =	0.184± (0.006) -	0.590± (0.013) =	**0.015 ± (0.002) -**	**0.708± (0.000)** +
S11	0.680 ± (0.003) =	0.683± (0.007) =	0.645± (0.005) =	0.673 ± (0.005) =	0.184 ± (0.006) -	0.588 ± (0.011) =	**0.014 ± (0.001) -**	**0.708 ± (0.000)** +
S12	0.679± (0.004) =	0.682± (0.006) =	0.648 ± (0.005) =	0.676± (0.005) =	0.185 ± (0.006) -	0.591 ± (0.013) -	**0.015± (0.002) -**	**0.708±(0.000)** +
S13	0.680± (0.004) =	0.683± (0.005) =	0.647± (0.005) =	0.675± (0.006) =	0.185 ± (0.006) -	0.590± (0.014) =	**0.015± (0.001) -**	**0.708 ± (0.000)** +
S14	0.681± (0.004) =	0.683± (0.004) =	0.648 ± (0.006) =	0.674 ± (0.004) =	0.185± (0.006) -	0.590 ± (0.013) =	**0.016 ± (0.003) -**	**0.708 ± (0.000)** +
S15	0.680± (0.004) =	0.685± (0.004) =	0.645 ± (0.006) =	0.674 ± (0.003) =	0.184± (0.006) -	0.588 ± (0.013) -	**0.015 ± (0.001) -**	**0.708 ± (0.000)** +
S16	0.679 ± (0.004) =	0.686 ± (0.004) =	0.648 ± (0.008) =	0.674± (0.006) =	0.184± (0.006) -	0.593 ± (0.010) -	**0.015 ± (0.003) -**	**0.708 ± (0.000)** +
S17	0.657 ± (0.018) =	0.657 ± (0.021) =	0.636 ± (0.013) =	0.640 ± (0.025) =	0.178± (0.004) -	0.590± (0.013) -	**0.015 ± (0.002) -**	**0.694 ± (0.011)** =

Test size	NSGA-II	NSGA-III	RVEA	SPEAR	SPEA2	tDEA
S1	0.681 ±(0.068) =	0.685± (0.005) =	0.642 ± (0.005) =	0.706 ± (0.001) +	0.702 ± (0.002) +	0.735± (0.007) =
S2	0.699 ± (0.000) =	0.687± (0.004) =	0.642± (0.007) =	0.706 ± (0.001) +	0.702± (0.002) +	0.725± (0.006) =
S3	0.699 ± (0.002) =	0.688± (0.005) =	0.642 ± (0.007) =	0.706 ± (0.001) +	0.702± (0.002) +	0.755 ± (0.005) =
S4	0.700 ± (0.002) =	0.686± (0.005) =	0.642 ± (0.009) =	0.706± (0.001) +	0.702 ± (0.002) +	0.745 ± (0.007) =
S5	0.700 ± (0.002) +	0.685 ± (0.004) =	0.642 ± (0.005) =	0.706 ± (0.001) +	0.702 ± (0.003) +	0.764 ± (0.006) =
S6	0.699± (0.002) =	0.684± (0.004) =	0.643 ± (0.006) =	0.707± (0.001) +	0.701 ± (0.003) +	0.742 ± (0.006) =
S7	0.700 ± (0.002) =	0.684± (0.005) =	0.642± (0.007) =	0.706± (0.001) +	0.701 ± (0.002) =	0.735 ± (0.007) =
S8	0.700± (0.002) +	0.686 ± (0.003) =	0.641± (0.008) =	0.706 ± (0.002) +	0.701 ± (0.002) +	0.743± (0.005) =
S9	0.700 ± (0.003) +	0.686 ± (0.005) =	0.641± (0.008) =	0.706 ± (0.001) +	0.700 ± (0.002) +	0.742± (0.005) =
S10	0.700 ± (0.002) +	0.687 ± (0.004) =	0.643± (0.005) =	0.706 ± (0.001) +	0.701 ± (0.003) +	0.756 ± (0.005) =
S11	0.700± (0.003) +	0.685 ± (0.004) =	0.635 ± (0.040) =	0.706 ± (0.001) +	0.701 ± (0.001) +	0.738 ± (0.005) =
S12	0.700± (0.002) +	0.687 ± (0.005) =	0.645 ± (0.007) =	0.706 ± (0.002) +	0.701 ± (0.003) +	0.730 ± (0.006) =
S13	0.699 ± (0.003) +	0.686 ± (0.005) =	0.640 ± (0.008) =	0.707 ± (0.001) +	0.702 ± (0.002) +	0.725± (0.0077) =
S14	0.700 ± (0.002) +	0.686 ± (0.006) =	0.641 ± (0.006) =	0.706± (0.001) +	0.701 ± (0.002) +	0.721 ± (0.007) =
S15	0.699± (0.002) =	0.686 ± (0.005) =	0.641± (0.007) =	0.706± (0.001) +	0.703± (0.002) +	0.748 ± (0.007) =
S16	0.700± (0.002) =	0.687 ± (0.005) =	0.640 ± (0.007) =	0.706± (0.001) +	0.701± (0.003) +	0.754 ± (0.006) =
S17	0.669± (0.023) =	0.654 ± (0.023) =	0.591 ± (0.039) -	**0.697± (0.007) =**	0.674± (0.021) =	0.459± (0.022) =

Test size	BiGE	GrEA	KnEA	rNSGA-II	SPEA2SDE
S1	0.461± (0.018) -	0.642 ± (0.007) =	0.513± (0.009) -	0.429 ± (0.004) -	0.671 ± (0.010)
S2	0.460± (0.013) -	0.642± (0.007) =	0.514 ± (0.006) -	0.430 ± (0.004) -	0.673 ± (0.006)
S3	0.454 ± (0.016) -	0.642± (0.006) =	0.512 ± (0.011) -	0.429 ± (0.003) -	0.675 ± (0.007)
S4	0.465 ± (0.012) -	0.644± (0.006) =	0.512± (0.005) -	0.430 ± (0.004) -	0.672 ± (0.009)
S5	0.461 ± (0.018) -	0.643 ± (0.006) =	0.515± (0.006) -	0.430± (0.005) -	0.671 ± (0.006)
S6	0.462 ± (0.016) -	0.642 ± (0.007) =	0.512± (0.010) -	0.430 ± (0.004) -	0.672± (0.007)
S7	0.453 ± (0.015) -	0.641± (0.008) =	0.514± (0.004) -	0.430 ± (0.003) -	0.674 ± (0.008)
S8	0.458 ± (0.014) -	0.645± (0.008) =	0.515± (0.007) -	0.429± (0.004) -	0.671 ± (0.008)
S9	0.465 ± (0.016) -	0.642± (0.007) =	0.512± (0.007) -	0.430± (0.003) -	0.671 ± (0.009)
S10	0.461± (0.017) -	0.643± (0.008) =	0.513 ± (0.006) -	0.430± (0.003) -	0.671 ± (0.010)
S11	0.466± (0.020) -	0.645± (0.006) =	0.514± (0.007) -	0.430 ± (0.003) -	0.669± (0.009)
S12	0.459 ± (0.011) -	0.643± (0.007) =	0.515± (0.007) -	0.430± (0.004) -	0.672± (0.006)
S13	0.457± (0.014) -	0.643± (0.006) =	0.512 ± (0.007) -	0.429± (0.004) -	0.672 ± (0.007)
S14	0.460 ± (0.015) -	0.643± (0.007) =	0.513 ± (0.009) -	0.429± (0.003) -	0.669 ± (0.007)
S15	0.463 ± (0.017) -	0.643 ± (0.007) =	0.516 ± (0.010) -	0.430 ± (0.005) -	0.672 ± (0.008)
S16	0.461 ± (0.015) -	0.642 ± (0.007) =	0.514 ± (0.009) -	0.430 ± (0.003) -	0.674± (0.008)
S17	0.497± (0.055) -	0.657± (0.013) =	0.525 ± (0.010) -	0.432 ± (0.005) -	0.661 ± (0.012)

Table 6
Average CPU time.

Algorithm	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11	S12	S13	S14	S15	S16	S17
NSGA-II	1	1	1		1	1	1	1	1	1	1	1	1	1	1	1	1
rNSGA-II	2	2		1	2	2	2				2	2	2	2	2	2	2
tDEA				2				2		2							
RVEA			2						2								

- For the spread indicator, NNIA and SPEA2 possess the best values, while GrEA owns the worst value amongst all algorithms for all test scales.
- Average CPU time shows that NSGA-II possesses the best performance of all algorithms for all test scales, except S4, for which rNSGA-II has the first ranking.

Table 7
Summary of indicators performance studied in the paper (1st and 2nd best performance).

Test scale	HV		Spread		Run time	
	1st best	2nd best	1st best	2nd best	1st best	2nd best
S1	NNIA	SPEAR	NNIA	SPEA2	NSGA-II	rNSGA-II
S2	NNIA	SPEAR	SPEA2	NNIA	NSGA-II	rNSGA-II
S3	NNIA	SPEAR	NNIA	SPEA2	NSGA-II	RVEA
S4	NNIA	SPEAR	SPEA2	NNIA	rNSGA-II	tDEA
S5	NNIA	SPEAR	NNIA	SPEA2	NSGA-II	rNSGA-II
S6	NNIA	SPEAR	SPEA2	NNIA	NSGA-II	rNSGA-II
S7	NNIA	SPEAR	SPEA2	NNIA	NSGA-II	rNSGA-II
S8	NNIA	SPEAR	NNIA	SPEA2	NSGA-II	tDEA
S9	NNIA	SPEAR	SPEA2	NNIA	NSGA-II	RVEA
S10	NNIA	SPEAR	NNIA	SPEA2	NSGA-II	tDEA
S11	NNIA	SPEAR	SPEA2	NNIA	NSGA-II	rNSGA-II
S12	NNIA	SPEAR	SPEA2	NNIA	NSGA-II	rNSGA-II
S13	NNIA	SPEAR	NNIA	SPEA2	NSGA-II	rNSGA-II
S14	NNIA	SPEAR	SPEA2	NNIA	NSGA-II	rNSGA-II
S15	NNIA	SPEAR	NNIA	SPEA2	NSGA-II	rNSGA-II
S16	NNIA	SPEAR	NNIA	SPEA2	NSGA-II	rNSGA-II
S17	SPEAR	SPEAR	NNIA	SPEA2	NSGA-II	rNSGA-II

- rNSGA-II, tDEA, and RVEA own the 2nd best performance in average CPU time.

The future study could be addressed as follows:

- Implementing other types of metaheuristics, such as swarm intelligence, is proposed.
- Additionally, considering that researchers have recently proposed newer formulations of the next-release problem, it would be valuable to apply the suggested algorithms to these problems.
- Furthermore, it is recommended to validate the algorithms discussed in this paper using other datasets available in the literature.
- Another potential research area involves evaluating the performance of the algorithms using different metrics found in the literature.

CRediT authorship contribution statement

Iman Rahimi: Conceptualization, Methodology, Formal analysis, Software, Writing – original draft, Data curation, Investigation. **Amir H. Gandomi:** Supervision, Methodology, Validation, Writing – review & editing. **Mohammad Reza Nikoo:** Revising, Review & editing, Visualization. **Fang Chen:** Revising, Review & editing, Visualization.

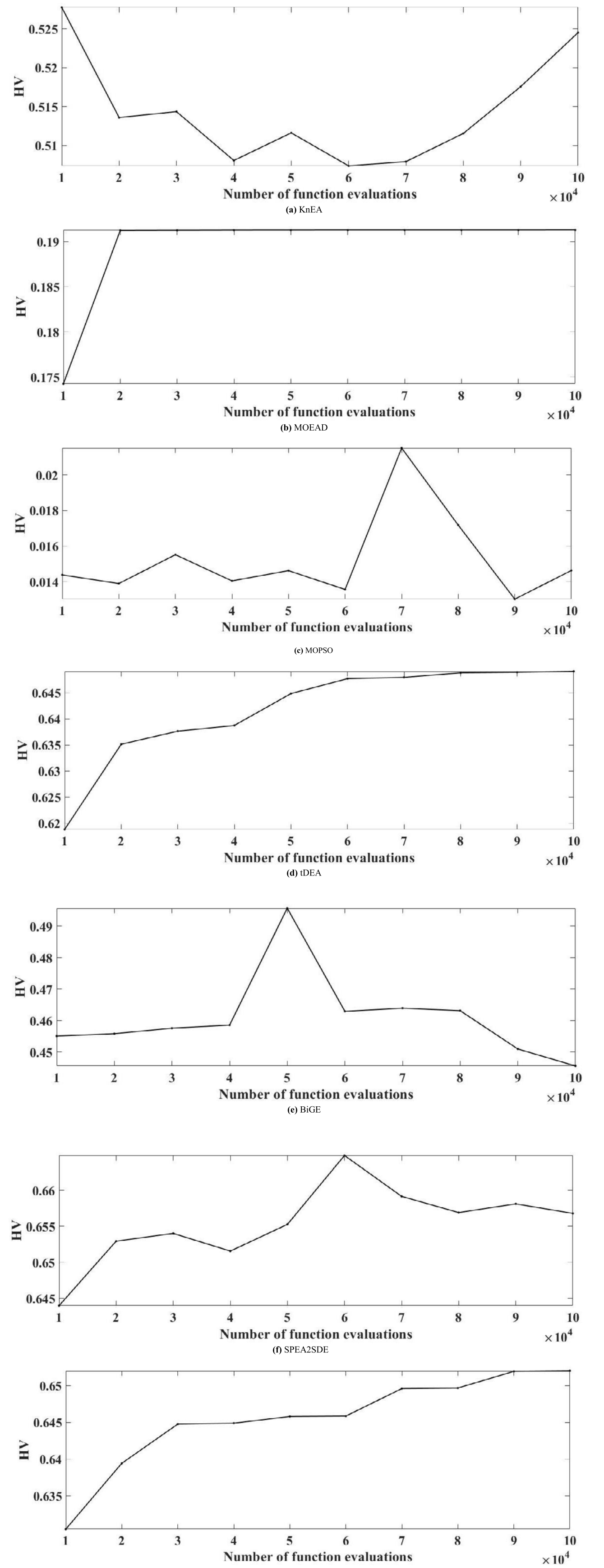
Declaration of competing interest

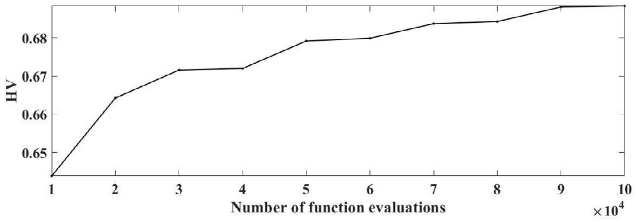
The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

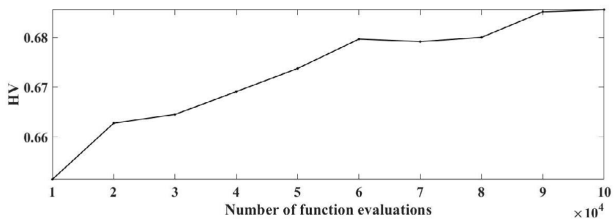
Data will be made available on request.

Appendix. Hypervolume (HV) index for different multi-objective optimization algorithms vs. number of function evaluations

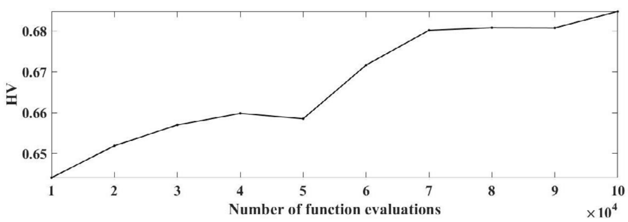




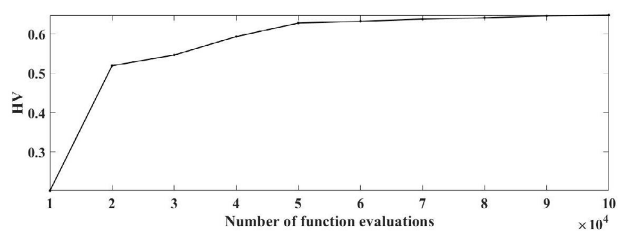
(h) NSGA-III



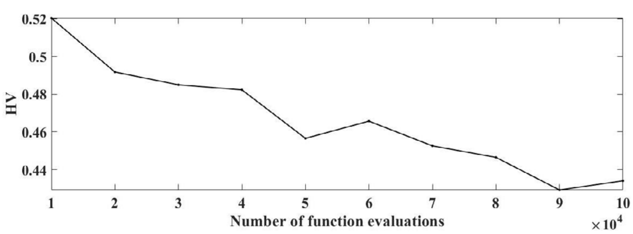
(i) GrEA



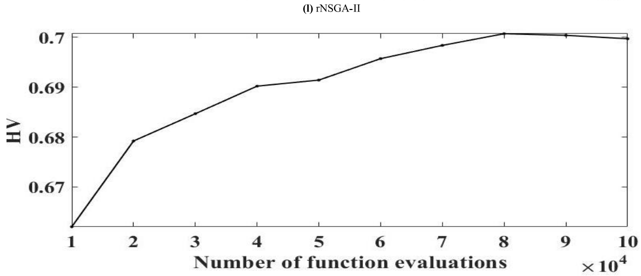
(j) ANSGA-III



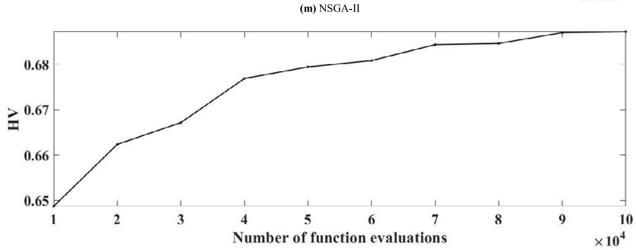
(k) RVEA



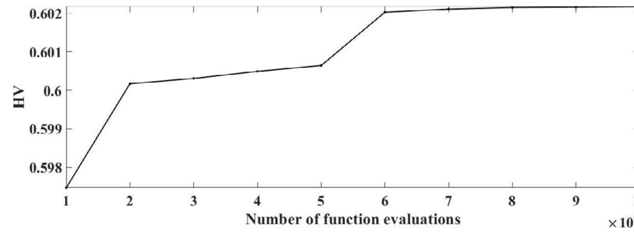
(l) rNSGA-II



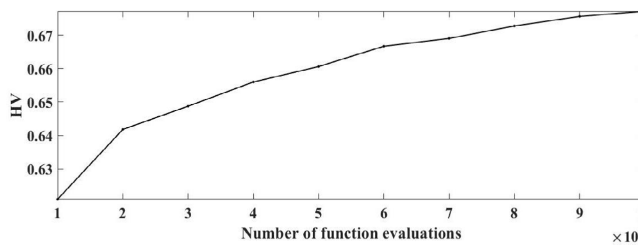
(m) NSGA-II



(n) EFRRR



(o) MOMBIII



(p) IDBEA

References

[1] N. Veerapen, G. Ochoa, M. Harman, E.K. Burke, An integer linear programming approach to the single and bi-objective next release problem, Inf. Softw. Technol. 65 (2015) 1–13.

[2] Y. Zhang, M. Harman, S.A. Mansouri, The multi-objective next release problem, in: Proceedings of the 9th Annual Conference on Genetic and Evolutionary Computation, 2007, pp. 1129–1137.

[3] J.J. Durillo, Y. Zhang, E. Alba, M. Harman, A.J. Nebro, A study of the bi-objective next release problem, Empir. Softw. Eng. 16 (1) (2011) 29–60.

[4] J.C. Almeida, F. de C. Pereira, M.V.A. Reis, B. Piva, The next release problem: Complexity, exact algorithms and computations, in: International Symposium on Combinatorial Optimization, 2018, pp. 26–38.

[5] S. Dong, Y. Xue, S. Brinkkemper, Y.-F. Li, Multi-objective integer programming approaches to next release problem—Enhancing exact methods for finding whole pareto front, Inf. Softw. Technol. (2022) 106825.

[6] F.G. Freitas, D.P. Coutinho, J.T. Souza, et al., Software next release planning approach through exact optimization, Int. J. Comput. Appl. 22 (8) (2011) 1–8.

[7] A.A. Araújo, M. Paixao, I. Yeltsin, A. Dantas, J. Souza, An architecture based on interactive optimization and machine learning applied to the next release problem, Autom. Softw. Eng. 24 (2017) 623–671.

[8] J.M. Chaves-González, M.A. Pérez-Toledano, Differential evolution with Pareto tournament for the multi-objective next release problem, Appl. Math. Comput. 252 (2015) 1–13.

[9] T. do Nascimento Ferreira, A.A. Araújo, A.D.B. Neto, J.T. de Souza, Incorporating user preferences in ant colony optimization for the next release problem, Appl. Soft Comput. 49 (2016) 1283–1296.

[10] M. Ghasemi, K. Bagherifard, H. Parvin, S. Nejatian, K.-H. Pho, Multi-objective whale optimization algorithm and multi-objective grey wolf optimizer for solving next release problem with developing fairness and uncertainty quality indicators, Appl. Intell. (2021) 1–30.

[11] L. Li, M. Harman, E. Letier, Y. Zhang, Robust next release problem: handling uncertainty during optimization, in: Proceedings of the 2014 Annual Conference on Genetic and Evolutionary Computation, 2014, pp. 1247–1254.

[12] A.J. Bagnall, V.J. Rayward-Smith, I.M. Whittley, The next release problem, Inf. Softw. Technol. 43 (14) (2001) 883–890.

[13] J. Xuan, H. Jiang, Z. Ren, Z. Luo, Solving the large scale next release problem with a backbone-based multilevel algorithm, IEEE Trans. Softw. Eng. 38 (5) (2012) 1195–1212.

[14] C.A. Coello Coello, Evolutionary multi-objective optimization: some current research trends and topics that remain to be explored, Front. Comput. Sci. China 3 (1) (2009) 18–30, http://dx.doi.org/10.1007/s11704-009-0005-7.

[15] K. Deb, Multi-objective optimization, in: Search Methodologies, Springer, 2014, pp. 403–449.

[16] R. Behmanesh, I. Rahimi, A.H. Gandomi, Evolutionary many-objective algorithms for combinatorial optimization problems: a comparative study, Arch. Comput. Methods Eng. 28 (2) (2021) 673–688.

[17] R. Sarker, T. Ray, An improved evolutionary algorithm for solving multi-objective crop planning models, Comput. Electron. Agric. 68 (2) (2009) 191–199.

[18] H.R. Cheshmehgaz, H. Haron, A. Sharifi, The review of multiple evolutionary searches and multi-objective evolutionary algorithms, Artif. Intell. Rev. 43 (3) (2015) 311–343.

[19] K.C. Tan, T.H. Lee, E.F. Khor, Evolutionary algorithms for multi-objective optimization: Performance assessments and comparisons, Artif. Intell. Rev. 17 (4) (2002) 251–290.

[20] I. Rahimi, A.H. Gandomi, F. Chen, E. Mezura-Montes, A review on constraint handling techniques for population-based algorithms: from single-objective to multi-objective optimization, Arch. Comput. Methods Eng. 30 (2023) 2181–2209.

[21] I. Rahimi, A.H. Gandomi, K. Deb, F. Chen, M.R. Nikoo, Scheduling by NSGA-II: review and bibliometric analysis, Processes 10 (1) (2022) 98.

[22] C.A. Poojari, J.E. Beasley, Improving benders decomposition using a genetic algorithm, European J. Oper. Res. 199 (1) (2009) 89–97.

[23] C. Peng, H.-L. Liu, F. Gu, An evolutionary algorithm with directed weights for constrained multi-objective optimization, Appl. Soft Comput. 60 (2017) 613–622.

[24] S. Yuan, G. Deng, Q. Feng, P. Zheng, T. Song, Multi-objective evolutionary algorithm based on decomposition for energy-aware scheduling in heterogeneous computing systems, J. Univ. Comput. Sci. 23 (7) (2017) 636–651.

[25] C.A.C. Coello, G.B. Lamont, D.A. van Veldhuizen, et al., Evolutionary Algorithms for Solving Multi-Objective Problems, Vol. 5, Springer, 2007.

[26] J.W. Herrmann, C.-Y. Lee, J. Hinchman, Global job shop scheduling with a genetic algorithm, Prod. Oper. Manage. 4 (1) (1995) 30–45.

[27] X. Cai, O. Wei, Z. Huang, Evolutionary approaches for multi-objective next release problem, Comput. Inform. 31 (4) (2012) 847–875.

[28] P. Baker, M. Harman, K. Steinhofel, A. Skaliotis, Search based approaches to component selection and prioritization for the next release problem, in: 2006 22nd IEEE International Conference on Software Maintenance, 2006, pp. 176–185.

[29] Hui Li, et al., Many-objective optimization: A survey of key concepts, methods and challenges, IEEE Trans. Evol. Comput. 19 (4) (2015) 501–524.

[30] A. Hedayat, et al., A review of evolutionary multi-objective optimization: algorithms and applications, Swarm Evol. Comput. 50 (2020) 100664.

[31] Markus Wagner, Eckart Zitzler, A reference vector guided evolutionary algorithm for many-objective optimization, in: Proceedings of the 2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence), IEEE, 2008.

[32] H. Li, Q. Zhang, Multiobjective optimization problems with complicated Pareto sets, MOEA/D and NSGA-II, IEEE Trans. Evol. Comput. 13 (2) (2008) 284–302.

[33] Jonathan Bader, Eckart Zitzler, Hype: An algorithm for fast hypervolume-based many-objective optimization, Evol. Comput. 19 (1) (2011) 45–76.

[34] R. Cheng, Y. Jin, M. Olhofer, B. Sendhoff, A reference vector guided evolutionary algorithm for many-objective optimization, IEEE Trans. Evol. Comput. 20 (5) (2016) 773–791.

[35] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multiobjective genetic algorithm: NSGA-II, IEEE Trans. Evol. Comput. 6 (2) (2002) 182–197.

[36] J.G. Hobbie, A.H. Gandomi, I. Rahimi, A comparison of constraint handling techniques on NSGA-II, Arch. Comput. Methods Eng. 28 (2021) 3475–3490.

[37] X. Zhang, Y. Tian, Y. Jin, A knee point-driven evolutionary algorithm for many-objective optimization, IEEE Trans. Evol. Comput. 19 (6) (2014) 761–776.

[38] K. Deb, H. Jain, An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part I: solving problems with box constraints, IEEE Trans. Evol. Comput. 18 (4) (2013) 577–601.

[39] Y. Yuan, H. Xu, B. Wang, X. Yao, A new dominance relation-based evolutionary algorithm for many-objective optimization, IEEE Trans. Evol. Comput. 20 (1) (2015) 16–37.

[40] E. Zitzler, M. Laumanns, L. Thiele, SPEA2: Improving the Strength Pareto Evolutionary Algorithm, TIK-Report, 103, 2001.

[41] C.A.C. Coello, M.S. Lechuga, MOPSO: A proposal for multiple objective particle swarm optimization, in: Proceedings of the 2002 Congress on Evolutionary Computation. CEC'02 (Cat. No. 02TH8600), Vol. 2, 2002, pp. 1051–1056.

[42] Y. Yuan, H. Xu, B. Wang, B. Zhang, X. Yao, Balancing convergence and diversity in decomposition-based many-objective optimizers, IEEE Trans. Evol. Comput. 20 (2) (2015) 180–198.

[43] Q. Zhang, H. Li, MOEA/D: a multiobjective evolutionary algorithm based on decomposition, IEEE Trans. Evol. Comput. 11 (6) (2007) 712–731.

[44] R. Hernández Gómez, C.A. Coello Coello, Improved metaheuristic based on the R2 indicator for many-objective optimization, in: Proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation, 2015, pp. 679–686.

[45] M. Gong, L. Jiao, H. Du, L. Bo, Multiobjective immune algorithm with nondominated neighbor-based selection, Evol. Comput. 16 (2) (2008) 225–255.

[46] L. ben Said, S. Bechikh, K. Ghédira, The r-dominance: a new dominance relation for interactive evolutionary multicriteria decision making, IEEE Trans. Evol. Comput. 14 (5) (2010) 801–818.

[47] M. Li, S. Yang, X. Liu, Bi-goal evolution for many-objective optimization problems, Artificial Intelligence 228 (2015) 45–65.

[48] S. Yang, M. Li, X. Liu, J. Zheng, A grid-based evolutionary algorithm for many-objective optimization, IEEE Trans. Evol. Comput. 17 (5) (2013) 721–736.

[49] M. Asafuddoula, T. Ray, R. Sarker, A decomposition-based evolutionary algorithm for many objective optimization, IEEE Trans. Evol. Comput. 19 (3) (2014) 445–460.

[50] M. Li, S. Yang, X. Liu, Shift-based density estimation for Pareto-based algorithms in many-objective optimization, IEEE Trans. Evol. Comput. 18 (3) (2013) 348–365.

[51] S. Jiang, S. Yang, A strength Pareto evolutionary algorithm based on reference direction for multiobjective and many-objective optimization, IEEE Trans. Evol. Comput. 21 (3) (2017) 329–346.

[52] W. Elloumi, A.M. Alimi, A more efficient MOPSO for optimization, in: ACS/IEEE International Conference on Computer Systems and Applications-AICCSA, Vol. 2010, 2010, pp. 1–7.

[53] A.M. Sharaf, A.A.A. El-Gammal, A novel discrete multi-objective particle swarm optimization (MOPSO) of optimal shunt power filter, in: 2009 IEEE/PES Power Systems Conference and Exposition, 2009, pp. 1–7.

[54] S. Lalwani, S. Singhal, R. Kumar, N. Gupta, A comprehensive survey: Applications of multi-objective particle swarm optimization (MOPSO) algorithm, Trans. Comb. 2 (1) (2013) 39–101.

[55] V. Mokarram, M.R. Banan, A new PSO-based algorithm for multi-objective optimization with continuous and discrete design variables, Struct. Multidiscip. Optim. 57 (2) (2018) 509–533.

[56] Y. Yuan, H. Xu, B. Wang, Evolutionary many-objective optimization using ensemble fitness ranking, in: Proceedings of the 2014 Annual Conference on Genetic and Evolutionary Computation, 2014, pp. 669–676.

[57] D. Anghinolfi, M. Paolucci, R. Ronco, A bi-objective heuristic approach for green identical parallel machine scheduling, European J. Oper. Res. 289 (2) (2021) 416–434.

[58] Y. Fu, M. Zhou, X. Guo, L. Qi, K. Sedraoui, Multiverse optimization algorithm for stochastic biobjective disassembly sequence planning subject to operation failures, IEEE Trans. Syst. Man Cybern. 52 (2) (2021) 1041–1051.

[59] V. Kayvanfar, S.M.M. Husseini, B. Karimi, M.S. Sajadieh, Bi-objective intelligent water drops algorithm to a practical multi-echelon supply chain optimization problem, J. Manuf. Syst. 44 (2017) 93–114.

[60] W. Peng, J. Zhang, L. Chen, et al., A bi-objective hierarchical program scheduling problem and its solution based on NSGA-III, Ann. Oper. Res. 308 (1) (2022) 389–414.

[61] H. Seada, K. Deb, A unified evolutionary optimization procedure for single, multiple, and many objectives, IEEE Trans. Evol. Comput. 20 (3) (2015) 358–369.

[62] N. Riquelme, C. von Lücken, B. Baran, Performance metrics in multi-objective optimization, in: 2015 Latin American Computing Conference, CLEI, 2015, pp. 1–11.

[63] A. Auger, J. Bader, D. Brockhoff, E. Zitzler, Theory of the hypervolume indicator: optimal μ-distributions and the choice of the reference point, in: Proceedings of the Tenth ACM SIGEVO Workshop on Foundations of Genetic Algorithms, 2009, pp. 87–102.

[64] J. Bader, K. Deb, E. Zitzler, Faster hypervolume-based search using Monte Carlo sampling, in: Multiple Criteria Decision Making for Sustainable Energy and Transportation Systems, Springer, 2010, pp. 313–326.

[65] E. Zitzler, S. Künzli, Indicator-based selection in multiobjective search, in: International Conference on Parallel Problem Solving from Nature, 2004, pp. 832–842.

[66] Y. Tian, R. Cheng, X. Zhang, Y. Jin, PlatEMO: A MATLAB platform for evolutionary multi-objective optimization [educational forum], IEEE Comput. Intell. Mag. 12 (4) (2017) 73–87.