A Comparative Study on Evolutionary Multi-objective Algorithms for Next Release Problem

Iman Rahimi
University of Technology Sydney
Amir H Gandomi (gandomi@uts.edu.au)
University of Technology Sydney
Mohammad Reza Nikoo
Sultan Qaboos University
Fang Chen
University of Technology Sydney

Research Article

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

Posted Date: August 9th, 2022

DOI: https://doi.org/10.21203/rs.3.rs-1929133/v1

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 the 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 shows that the HV and spread values of 1st and 2nd best algorithms (NNIA and SPEAR), for which most HV values for NNIA are bigger 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 the needs of software companies, which aims to develop and maintains the software systems that have been sold to customers. The problem is constrained by the software systems' total cost, whereby the objectives include maximizing total customer satisfaction and minimizing the total cost (Veerapen et al., 2015; Y. Zhang et al., 2007). 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, companies are recognized at different levels of importance by customers. Figure 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 above-mentioned problem is also known as a cost-profit analysis problem (Durillo et al., 2011), 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 the 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 (Almeida et al., 2018; Dong et al., 2022; Freitas et al., 2011). Although classic algorithm are able to find the optimal solution for some special problems, as the number of customers growths so the problem will become complicated. Therefore, as a NP- hard problem, in this study, nineteen state-of-the-art evolutionary algorithms are used to find the high quality solutions as they are more common because of their pros such as robustness and high flexibility in implementation.

This paper is the first comprehensive comparative study in the field of next released problem. Although many studies addressed the problem and solved them with some evolutionary algorithms (EAs), this study is the first work, wich uses several new EAs for solving the next released problem with two classic and realistic data. 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 on the problem statements and evolutionary algorithms that have been used to tackle the above-mentioned problem.

2.1 Problem statement: the next released problem

The next release problem is also considered a combinatorial optimization problem (L. Li et al., 2014; Veerapen et al., 2015) .(Y. Zhang et al., 2007) 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(Y. Zhang et al., 2007) could be written as follows:

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

$$Maximize \;\; f_2(x) = -\sum_{i=1}^{n} \cos t_i . x_i \tag{2}$$

$$x_i \in \{0,1\} \tag{3}$$

An important assumption is that all requirements are independent. The decision vector $X = \{x_1, x_2, ..., x_n\}$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.

2.2 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 a Pareto front of solutions could be found (Coello Coello, 2009) (Deb, 2014b), 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 solutions, and the second property is that any other solution found should be dominated by at least one solution in the set (Behmanesh et al., 2021; Coello Coello, 2009; Deb, 2014b) .

Since metaheuristics do not require concavity or convexity and also can produce several alternative solutions in a single run (i.e., evolutionary algorithms) (Sarker & Ray, 2009), they are often used to tackle multi-objective combinatorial optimization problems (Cheshmehgaz et al., 2015; Tan et al., 2002). Additionally, metaheuristics can integrate with specific decomposition algorithms(Poojari & Beasley, 2009) and, generally, many metaheuristics have been developed to deal with some multi-objective optimization problems (MOOPs)(C. Peng et al., 2017) (S. Yuan et al., 2017) (Coello et al., 2007) (Herrmann et al., 1995) .

(Y. Zhang et al., 2007) 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., 2012) 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 (Baker et al., 2006). In the aforementioned paper, MOEAs had better performance than the 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. These algorithms are categorized into several groups: (a) Indicator-based, (b) reference set-based, (c) Neighbor-based, (d) Pareto-based, (e) Decomposition-based,

(f) diversity, and (g) Preference-based, as 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 (Deb et al., 2002) .

Table 1
Multi -objective evolutionary algorithms included in this study

EA/Group	a	b	c	d	e	f g
KnEA (X. Zhang et al., 2014)				●		●
NSGA-III (Deb & Jain, 2013)		●		●		
tDEA (Y. Yuan, Xu, Wang, & Yao, 2015)		●		●		
SPEA2 (Zitzler et al., 2001)				●		
MOPSO (Coello & Lechuga, 2002)				●		
NSGA-II (Deb et al., 2002)				●		
EFR-RR (Y. Yuan, Xu, Wang, Zhang, et al., 2015)					●	
MOEA/D (Q. Zhang & Li, 2007)				●	●	
RVEA (Cheng et al., 2016)		●			●	
MOMBI-II (Hernández Gómez & Coello Coello, 2015)	●					
NNIA (Gong et al., 2008)			●			
rNSGA-II (Said et al., 2010)		●		●		
ANSGA-III (Jain & Deb, 2013)		●			●	
BiGE (M. Li et al., 2015)	●			●		
GrEA (Yang et al., 2013)				●		
I-DBEA (Asafuddoula et al., 2014)				●	●	
SPEA2 + SDE (M. Li et al., 2013)				●		
HypE (Bader & Zitzler, 2011)	●			●		
SPEA/R (Jiang & Yang, 2017)		●		●		

(Coello & Lechuga, 2002) 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 have been used widely in continuous and discrete optimization problems (Elloumi & Alimi, 2010; Sharaf & El-Gammal, 2009)(Lalwani et al., 2013; Mokarram & Banan, 2018).

(Baker et al., 2006) 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 principle of the predefined multiple targeted search, such that a set of Pareto-optimal points could be found by using points corresponding to each reference point.(Y. Yuan, Xu, Wang, & Yao, 2015) proposed the EFR-RR algorithm that enhances two decomposition-based MOEAs, namely MOEA/D and EFR (Y. Yuan et al., 2014), and also maintains the desired diversity of solutions.

(Cai et al., 2012) 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 & Coello Coello, 2015) 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., 2001) 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.(Y. Yuan, Xu, Wang, & Yao, 2015) 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., 2010) established a new dominance relation for interactive evolutionary multicriteria decision making (r-NSGA-II) and compared the proposed algorithm to other EMO algorithms.(Gong et al., 2008) 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., 2008) 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 has better performance 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., 2016) 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, 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.(Q. Zhang & Li, 2007) proposed a 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 used usually for the problems with more than three objective functions, some efforts have been addressed to employ these algorithms for single and multi-objective optimization problems (Anghinolfi et al., 2021; Fu et al., 2021; Kayvanfar et al., 2017; W. Peng et al., 2022; Seada & Deb, 2015). The following sub-sections provide the perforemance evaluation metrics and data collection, which are parts of the methodology section. The framework of the methodology is displayed in Fig. 2.

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 GD, IGD, spread, HV, normalized HV (NHV), SM, diversity metricreleae(Dong et al., 2022) (Behmanesh et al., 2021) (Behmanesh et al., 2021) (Coello et al., 2007) ,in this study, HV and spread, as two welknown metrics, were implemented as accuracy, diversity and cardinality are considered with these metrics (Riquelme et al., 2015):

- 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 (Auger et al., 2009). 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(Bader et al., 2010; Zitzler & Künzli, 2004).

- Spread: based on the Euclidean distances between the extreme solutions and the boundary solutions of the obtained non-dominated set (Deb, 2014a). The smaller value of the spread of the approximation, the better the distribution.

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 (Bagnall et al., 2001; Xuan et al., 2012). For the classic dataset, each dataset corresponds to a single problem. Table 2 provides the classic datasets, nrp1 to nrp5, with a specific number of customers (m) and the number of requirements (n), including cost and profit for each set. The second group includes a realistic dataset suggest by (Xuan et al., 2012). 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.

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 (Tian et al., 2017), to test and implement the different evolutionary algorithms for various problems. The parameters for implementing all algorithms are as follows:

Number of runs: 10 times for each algorithm on each test scale and take the average of the values,

Population size (N): 200,

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.

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

There are 19 algorithms that running 10 times for each test scale (17 test scales in general), resulting in 3230 experiments. It is nothworthy to mention that the expriments have been run on all test scales and because of the fact there are same results, only S1 test scale results have been provided in this paper (Figs. 3 and 4). Figure 3 (a-s) show the solutions found by different algorithms for the S1 test scale, revealing that algorithms such as EFRRR, NSGA-III, and IDBEA could find a large number of Pareto solutions for the S1 test scale, while rNSGA-II, BiGE and 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. Figure 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). 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 file A). One of the key factors in the next generation population and achieving Pareto is selection. The Pareto solution consist of the non-dominated solutions that are obtained in the final iteration. In some cases, the Pareto archive contains 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 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.

Figures 5 and 6 present the mean of HV and spread values of for the sets (S1), where 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 boxes represents each rectangle’s median. Also, Fig. 5 shows that NNIA owns the best value. In Fig. 6, NNIA also presents the best value of the spread, while GrEA illustrates the worst value of the spread. Tables 4 and 5 depict the mean and the standard deviation of spread and HV indicators for all algorithms over all test scales (S1-S17), respectively. According to the spread mean and standard deviation in Table 4, it is clear that NNIA possesses most of the best values (grey 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 amongst 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 owns the best performance for all datasets except S4, for which rNSGA-II has the best ranking 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.

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 +**

Table 4
Spread mean and standard deviation obtained by the algorithms in the test problems (continue)

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 =

Table 4 *Spread mean and standard deviation obtained by the algorithms in the test problems (continue)*

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)

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) =**

Table 5
HV mean and standard deviation obtained by the algorithms in the test problems (Continue)

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) =

Table 5 *HV mean and standard deviation obtained by the algorithms in the test problems (continue)*

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								

5. Summary Of Findings

This paper addresses 19 state-of-the-art evolutionary 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, to can tackle the multi-objective next released problem. 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 better NSGA-II with 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 worthy to mention 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 (Gong et al., 2008).

Figure 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 0.708 and smaller than 1, while the HV values for SPEAR vary between 0.706 and 0.708.

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

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 satisfaction and minimizing the total cost regarding the requirements for developing a software system. 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 are categorized into several groups of EMO algorithms: (a) Indicator-based, (b) reference set-based, (c) Neighbor-based, (d) Pareto-based, (e) Decomposition-based, (f) diversity, and (g) Preference-based.

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. In general, 17 test scales were addressed. From the experiments, te following results are summarized:

- 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.

- NNIA possesses the best value for the HV indicator, while MOPSO owns the worst value amongst all algorithms for all test scales.
- 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.
- rNSGA-II, tDEA, and RVEA own the 2nd best performance in average CPU time.

Applying other types of evolutionary algorithms is proposed as a direction for future study. Moreover, considering that researchers have recently proposed newer formulations of the next release problem, it would be valuable to implement the proposed algorithms for these problems. Furthermore, it is also suggested to verify the algorithms addressed in this paper with other datasets available in the literature. Moreover, 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. Another research area could be evaluating the performance of the algorithms with other metrics available in the literature.

Declarations

Compliance with Ethical Standards:

Sources of Funding, conflict of interest, and human/animal participants: No

References

1. Almeida, J. C., Pereira, F. de C., Reis, M. V. A., & Piva, B. (2018). The next release problem: Complexity, exact algorithms and computations. *International Symposium on Combinatorial Optimization*, 26–38.
2. Anghinolfi, D., Paolucci, M., & Ronco, R. (2021). A bi-objective heuristic approach for green identical parallel machine scheduling. *European Journal of Operational Research*, *289*(2), 416–434.
3. Asafuddoula, M., Ray, T., & Sarker, R. (2014). A decomposition-based evolutionary algorithm for many objective optimization. *IEEE Transactions on Evolutionary Computation*, *19*(3), 445–460.
4. Auger, A., Bader, J., Brockhoff, D., & Zitzler, E. (2009). Theory of the hypervolume indicator: optimal $μ$-distributions and the choice of the reference point. *Proceedings of the Tenth ACM SIGEVO Workshop on Foundations of Genetic Algorithms*, 87–102.
5. Bader, J., Deb, K., & Zitzler, E. (2010). Faster hypervolume-based search using Monte Carlo sampling. In *Multiple criteria decision making for sustainable energy and transportation systems* (pp. 313–326). Springer.
6. Bader, J., & Zitzler, E. (2011). HypE: An algorithm for fast hypervolume-based many-objective optimization. *Evolutionary Computation*, *19*(1), 45–76.
7. Bagnall, A. J., Rayward-Smith, V. J., & Whittley, I. M. (2001). The next release problem. *Information and Software Technology*, *43*(14), 883–890.
8. Baker, P., Harman, M., Steinhofel, K., & Skaliotis, A. (2006). Search based approaches to component selection and prioritization for the next release problem. *2006 22nd IEEE International Conference on Software Maintenance*, 176–185.
9. Behmanesh, R., Rahimi, I., & Gandomi, A. H. (2021). Evolutionary many-objective algorithms for combinatorial optimization problems: a comparative study. *Archives of Computational Methods in Engineering*, *28*(2), 673–688.
10. Cai, X., Wei, O., & Huang, Z. (2012). Evolutionary approaches for multi-objective next release problem. *Computing and Informatics*, *31*(4), 847–875.
11. Cheng, R., Jin, Y., Olhofer, M., & Sendhoff, B. (2016). A reference vector guided evolutionary algorithm for many-objective optimization. *IEEE Transactions on Evolutionary Computation*, *20*(5), 773–791.

12. Cheshmehgaz, H. R., Haron, H., & Sharifi, A. (2015). The review of multiple evolutionary searches and multi-objective evolutionary algorithms. *Artificial Intelligence Review*, *43*(3), 311–343.
13. Coello, C. A. C., Lamont, G. B., van Veldhuizen, D. A., & others. (2007). *Evolutionary algorithms for solving multi-objective problems* (Vol. 5). Springer.
14. Coello, C. A. C., & Lechuga, M. S. (2002). MOPSO: A proposal for multiple objective particle swarm optimization. *Proceedings of the 2002 Congress on Evolutionary Computation. CEC'02 (Cat. No. 02TH8600)*, *2*, 1051–1056.
15. Coello Coello, C. A. (2009). Evolutionary multi-objective optimization: some current research trends and topics that remain to be explored. *FRONTIERS OF COMPUTER SCIENCE IN CHINA*, *3*(1), 18–30. https://doi.org/10.1007/s11704-009-0005-7
16. Deb, K. (2014a). Multi-objective optimization. In *Search methodologies* (pp. 403–449). Springer.
17. Deb, K. (2014b). Multi-objective optimization. Search methodologies. *Search Methodologies. New York: Springer*.
18. 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.
19. 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.
20. Dong, S., Xue, Y., Brinkkemper, S., & Li, Y.-F. (2022). Multi-objective integer programming approaches to Next Release Problem—Enhancing exact methods for finding whole pareto front. *Information and Software Technology*, 106825.
21. Durillo, J. J., Zhang, Y., Alba, E., Harman, M., & Nebro, A. J. (2011). A study of the bi-objective next release problem. *Empirical Software Engineering*, *16*(1), 29–60.
22. Elloumi, W., & Alimi, A. M. (2010). A more efficient MOPSO for optimization. *ACS/IEEE International Conference on Computer Systems and Applications-AICCSA 2010*, 1–7.
23. Freitas, F. G., Coutinho, D. P., Souza, J. T., & others. (2011). Software next release planning approach through exact optimization. *Int. J. Comput. Appl*, *22*(8), 1–8.
24. Fu, Y., Zhou, M., Guo, X., Qi, L., & Sedraoui, K. (2021). Multiverse optimization algorithm for stochastic biobjective disassembly sequence planning subject to operation failures. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, *52*(2), 1041–1051.
25. Gong, M., Jiao, L., Du, H., & Bo, L. (2008). Multiobjective immune algorithm with nondominated neighbor-based selection. *Evolutionary Computation*, *16*(2), 225–255.
26. Hernández Gómez, R., & Coello Coello, C. A. (2015). Improved metaheuristic based on the R2 indicator for many-objective optimization. *Proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation*, 679–686.
27. Herrmann, J. W., LEE, C.-Y., & Hinchman, J. (1995). Global job shop scheduling with a genetic algorithm. *Production and Operations Management*, *4*(1), 30–45.
28. Jain, H., & Deb, K. (2013). An evolutionary many-objective optimization algorithm using reference-point based nondominated sorting approach, part II: Handling constraints and extending to an adaptive approach. *IEEE Transactions on Evolutionary Computation*, *18*(4), 602–622.
29. Jiang, S., & Yang, S. (2017). A strength Pareto evolutionary algorithm based on reference direction for multiobjective and many-objective optimization. *IEEE Transactions on Evolutionary Computation*, *21*(3), 329–346.
30. Kayvanfar, V., Husseini, S. M. M., Karimi, B., & Sajadieh, M. S. (2017). Bi-objective intelligent water drops algorithm to a practical multi-echelon supply chain optimization problem. *Journal of Manufacturing Systems*, *44*, 93–114.
31. Lalwani, S., Singhal, S., Kumar, R., & Gupta, N. (2013). A comprehensive survey: Applications of multi-objective particle swarm optimization (MOPSO) algorithm. *Transactions on Combinatorics*, *2*(1), 39–101.
32. Li, L., Harman, M., Letier, E., & Zhang, Y. (2014). Robust next release problem: handling uncertainty during optimization. *Proceedings of the 2014 Annual Conference on Genetic and Evolutionary Computation*, 1247–1254.
33. Li, M., Yang, S., & Liu, X. (2013). Shift-based density estimation for Pareto-based algorithms in many-objective optimization. *IEEE Transactions on Evolutionary Computation*, *18*(3), 348–365.

34. Li, M., Yang, S., & Liu, X. (2015). Bi-goal evolution for many-objective optimization problems. *Artificial Intelligence*, *228*, 45–65.
35. Mokarram, V., & Banan, M. R. (2018). A new PSO-based algorithm for multi-objective optimization with continuous and discrete design variables. *Structural and Multidisciplinary Optimization*, *57*(2), 509–533.
36. Peng, C., Liu, H.-L., & Gu, F. (2017). An evolutionary algorithm with directed weights for constrained multi-objective optimization. *Applied Soft Computing*, *60*, 613–622.
37. Peng, W., Zhang, J., Chen, L., & others. (2022). A bi-objective hierarchical program scheduling problem and its solution based on NSGA-III. *Annals of Operations Research*, *308*(1), 389–414.
38. Poojari, C. A., & Beasley, J. E. (2009). Improving Benders decomposition using a genetic algorithm. *European Journal of Operational Research*, *199*(1), 89–97.
39. Riquelme, N., von Lücken, C., & Baran, B. (2015). Performance metrics in multi-objective optimization. *2015 Latin American Computing Conference (CLEI)*, 1–11.
40. Said, L. ben, Bechikh, S., & Ghédira, K. (2010). The r-dominance: a new dominance relation for interactive evolutionary multicriteria decision making. *IEEE Transactions on Evolutionary Computation*, *14*(5), 801–818.
41. Sarker, R., & Ray, T. (2009). An improved evolutionary algorithm for solving multi-objective crop planning models. *Computers and Electronics in Agriculture*, *68*(2), 191–199.
42. Seada, H., & Deb, K. (2015). A unified evolutionary optimization procedure for single, multiple, and many objectives. *IEEE Transactions on Evolutionary Computation*, *20*(3), 358–369.
43. Sharaf, A. M., & El-Gammal, A. A. A. (2009). A novel discrete multi-objective Particle Swarm Optimization (MOPSO) of optimal shunt power filter. *2009 IEEE/PES Power Systems Conference and Exposition*, 1–7.
44. Tan, K. C., Lee, T. H., & Khor, E. F. (2002). Evolutionary algorithms for multi-objective optimization: Performance assessments and comparisons. *Artificial Intelligence Review*, *17*(4), 251–290.
45. Tian, Y., Cheng, R., Zhang, X., & Jin, Y. (2017). PlatEMO: A MATLAB platform for evolutionary multi-objective optimization [educational forum]. *IEEE Computational Intelligence Magazine*, *12*(4), 73–87.
46. Veerapen, N., Ochoa, G., Harman, M., & Burke, E. K. (2015). An integer linear programming approach to the single and bi-objective next release problem. *Information and Software Technology*, *65*, 1–13.
47. Xuan, J., Jiang, H., Ren, Z., & Luo, Z. (2012). Solving the large scale next release problem with a backbone-based multilevel algorithm. *IEEE Transactions on Software Engineering*, *38*(5), 1195–1212.
48. 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.
49. Yuan, S., Deng, G., Feng, Q., Zheng, P., & Song, T. (2017). Multi-Objective Evolutionary Algorithm Based on Decomposition for Energy-aware Scheduling in Heterogeneous Computing Systems. *J. Univers. Comput. Sci.*, *23*(7), 636–651.
50. Yuan, Y., Xu, H., & Wang, B. (2014). Evolutionary many-objective optimization using ensemble fitness ranking. *Proceedings of the 2014 Annual Conference on Genetic and Evolutionary Computation*, 669–676.
51. 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.
52. Yuan, Y., Xu, H., Wang, B., Zhang, B., & Yao, X. (2015). Balancing convergence and diversity in decomposition-based many-objective optimizers. *IEEE Transactions on Evolutionary Computation*, *20*(2), 180–198.
53. Zhang, Q., & Li, H. (2007). MOEA/D: A multiobjective evolutionary algorithm based on decomposition. *IEEE Transactions on Evolutionary Computation*, *11*(6), 712–731.
54. Zhang, X., Tian, Y., & Jin, Y. (2014). A knee point-driven evolutionary algorithm for many-objective optimization. *IEEE Transactions on Evolutionary Computation*, *19*(6), 761–776.
55. Zhang, Y., Harman, M., & Mansouri, S. A. (2007). The multi-objective next release problem. *Proceedings of the 9th Annual Conference on Genetic and Evolutionary Computation*, 1129–1137.

56. Zitzler, E., & Künzli, S. (2004). Indicator-based selection in multiobjective search. *International Conference on Parallel Problem Solving from Nature*, 832–842.
57. Zitzler, E., Laumanns, M., & Thiele, L. (2001). SPEA2: Improving the strength Pareto evolutionary algorithm. *TIK-Report*, *103*.

Figures

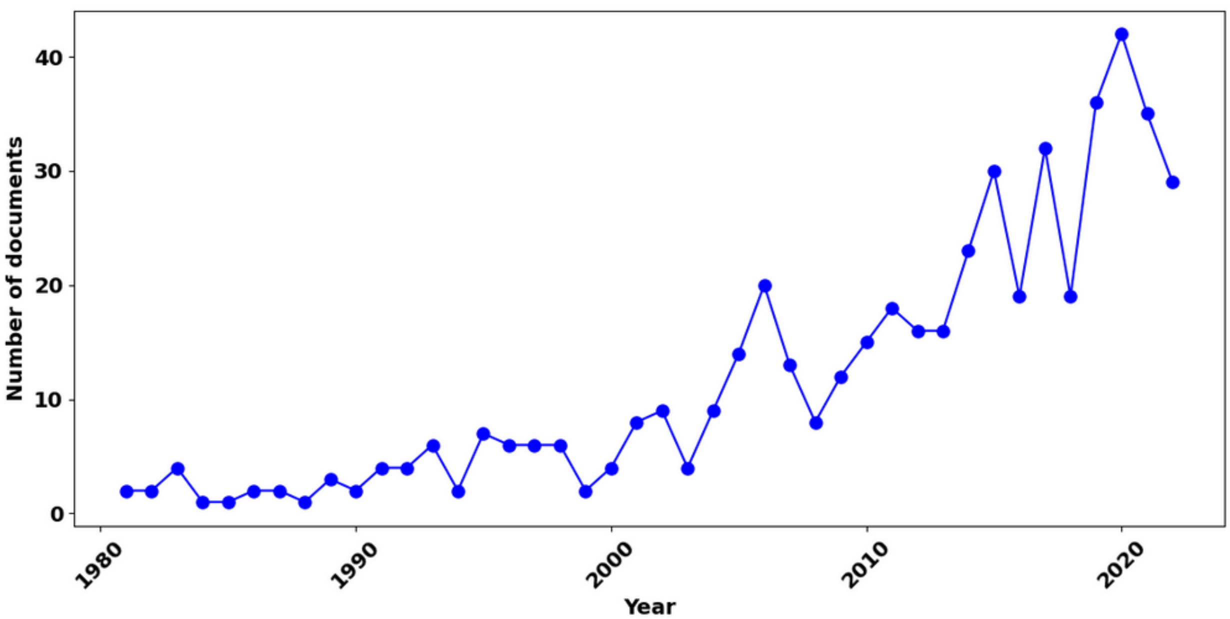

Figure 1

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

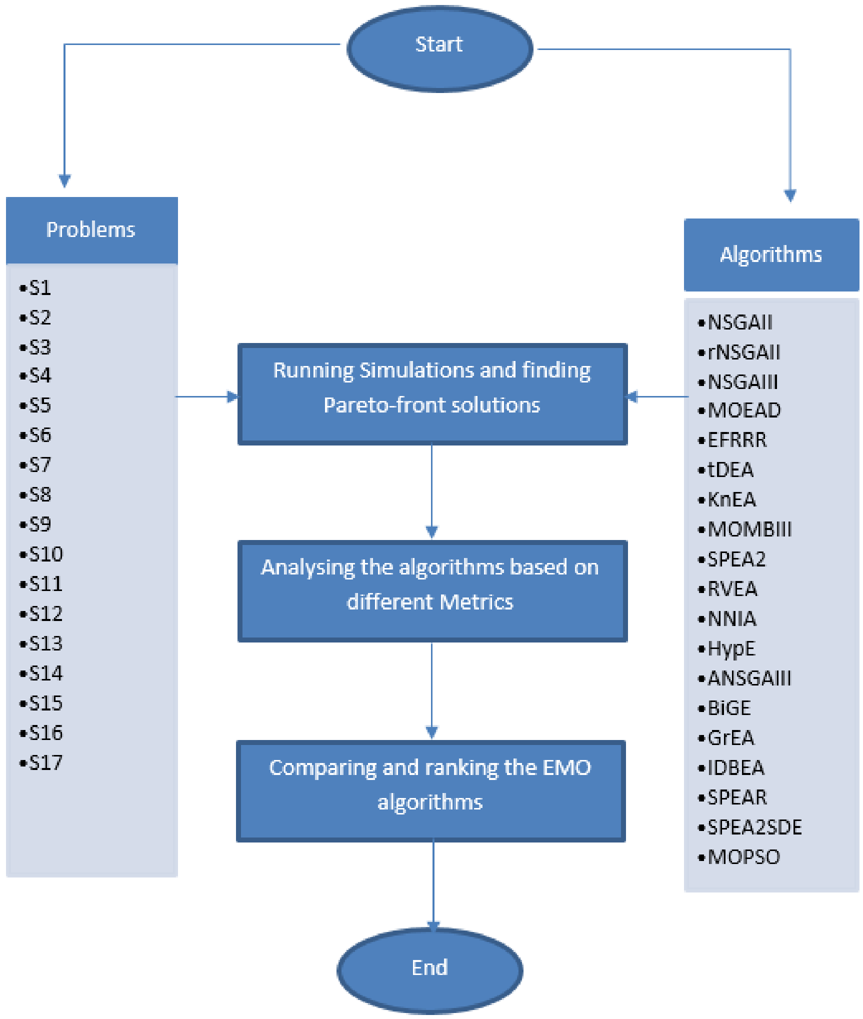

Figure 2

The framework methodology

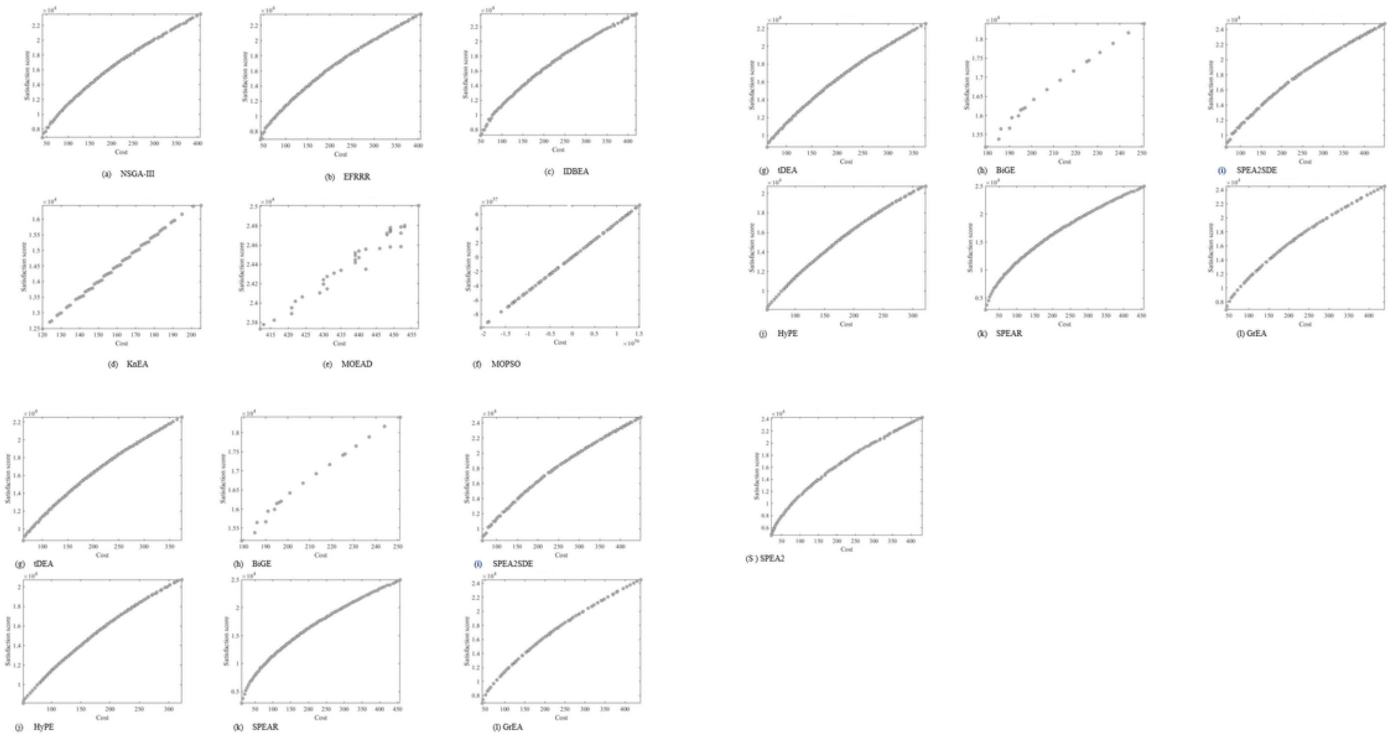

Figure 3

Comparison of solutions found by different algorithms on MONRP

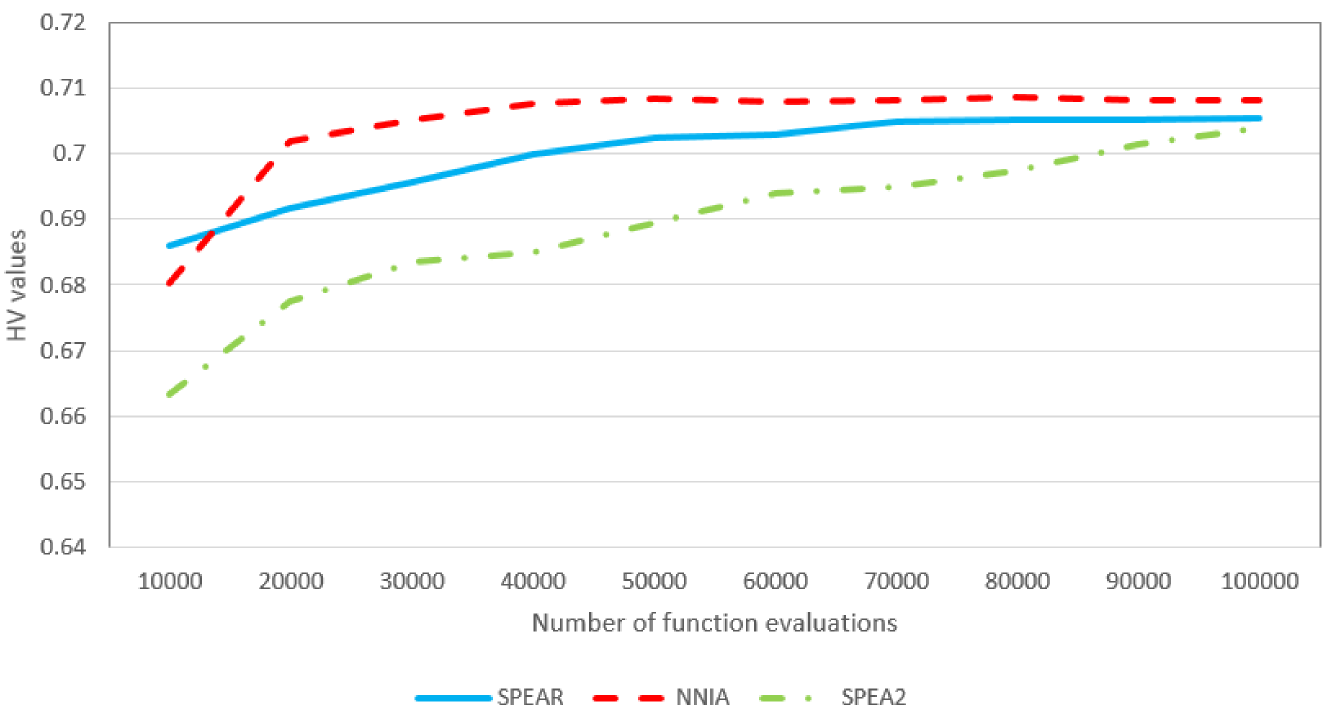

Figure 4

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

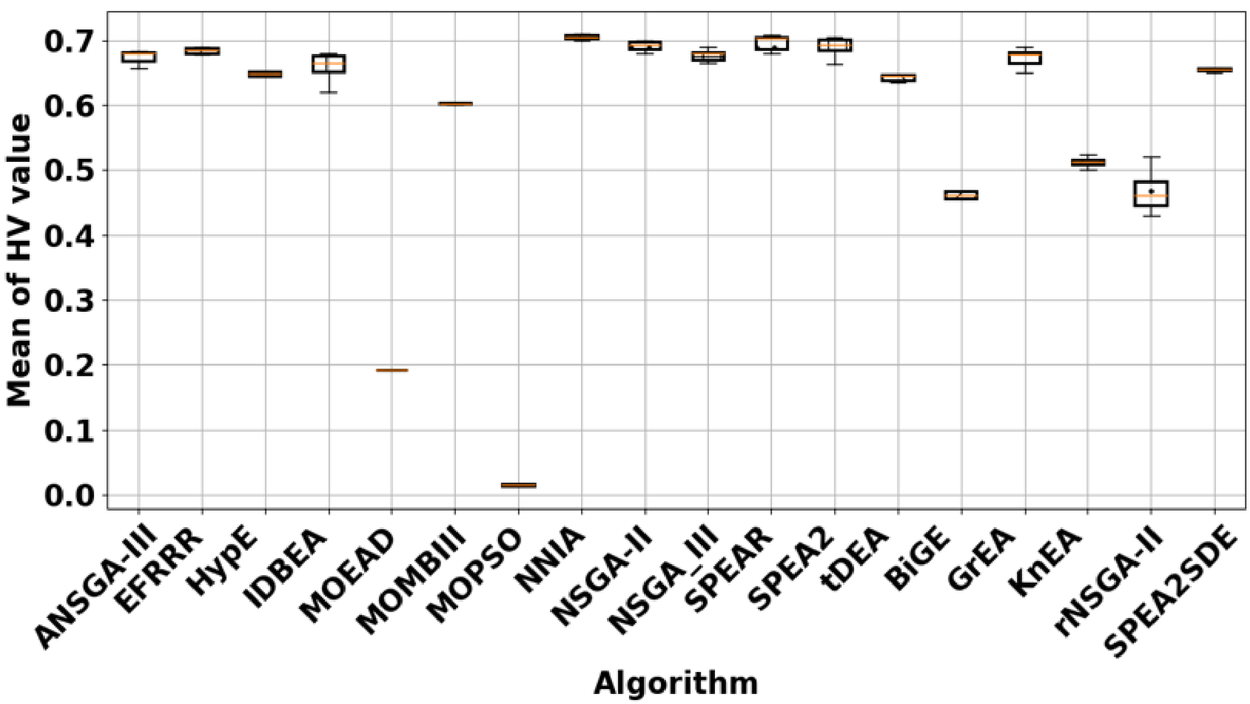

Figure 5

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

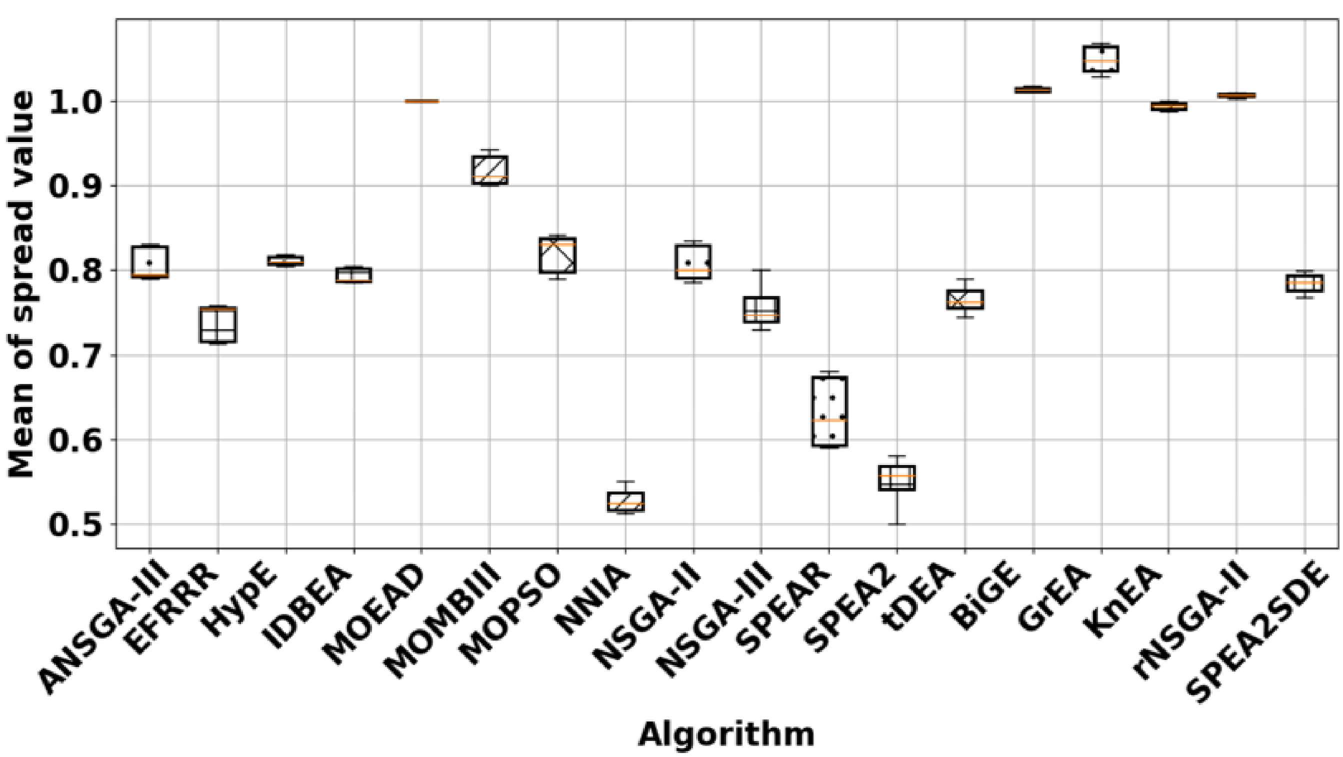

Figure 6

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

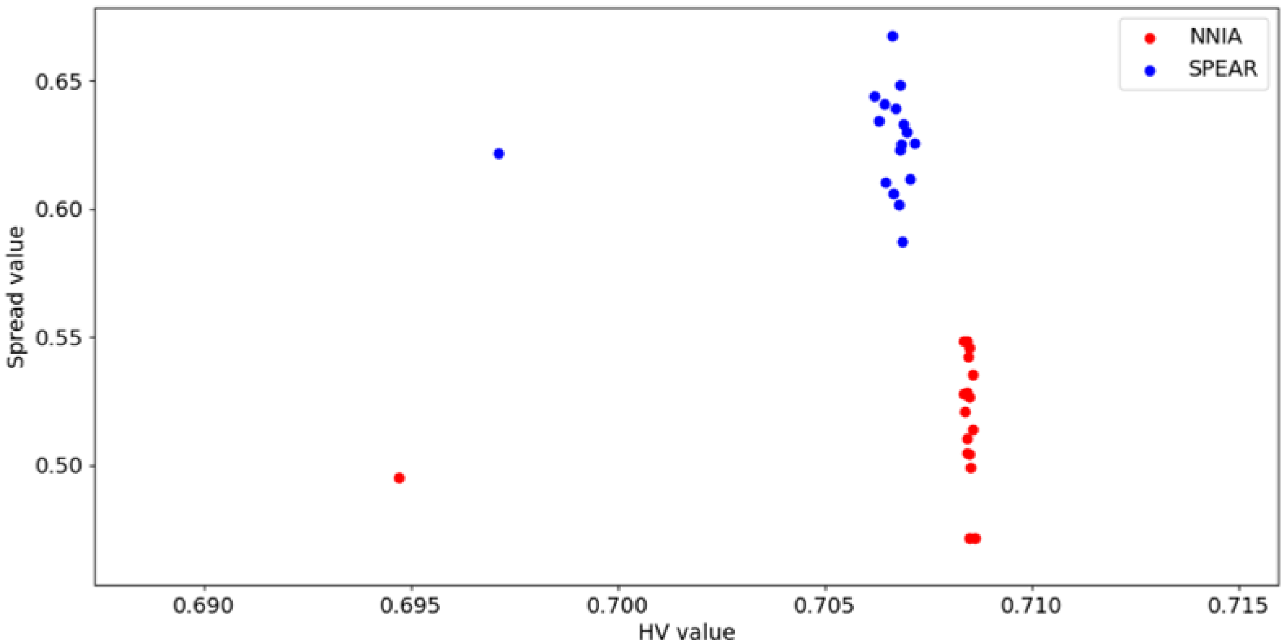

Figure 7

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

Supplementary Files

This is a list of supplementary files associated with this preprint. Click to download.

- AppendixA.docx