Complex & Intelligent Systems (2020) 6:189–197
https://doi.org/10.1007/s40747-019-00126-2

ORIGINAL ARTICLE

A repository of real-world datasets for data-driven evolutionary multiobjective optimization

Cheng He[1] · Ye Tian[2] · Handing Wang[3] · Yaochu Jin[4]

Received: 19 March 2019 / Accepted: 15 October 2019 / Published online: 11 November 2019

Abstract

Many real-world optimization applications have more than one objective, which are modeled as multiobjective optimization problems. Generally, those complex objective functions are approximated by expensive simulations rather than cheap analytic functions, which have been formulated as data-driven multiobjective optimization problems. The high computational costs of those problems pose great challenges to existing evolutionary multiobjective optimization algorithms. Unfortunately, there have not been any benchmark problems reflecting those challenges yet. Therefore, we carefully select seven benchmark multiobjective optimization problems from real-world applications, aiming to promote the research on data-driven evolutionary multiobjective optimization by suggesting a set of benchmark problems extracted from various real-world optimization applications.

Keywords Benchmark test suite · Data-driven optimization · Multiobjective optimization

Introduction

Evolutionary multiobjective optimization (EMO) has been flourishing for two decades in academia. However, the industry applications of EMO to real-world optimization problems are infrequent, due to the strong assumption that objective function evaluations are easily accessed. In fact, such objective functions may not exist, instead computationally expensive numerical simulations or costly physical experiments must be performed for evaluations. Such problems driven by data collected in simulations or experiments are formulated as data-driven optimization problems [1], which pose challenges to conventional EMO algorithms. Firstly, obtaining the minimum data for conventional EMO algorithms to converge requires a high computational or resource cost [2]. Secondly, although surrogate models that approximate objective functions can be used to replace the real function evaluations [3], the search accuracy cannot be guaranteed because of the approximation errors of surrogate models. Thirdly, since only a small amount of online data are allowed to be sampled during the optimization process, the management of online data significantly affects the performance of algorithms [4,5]. The research on data-driven evolutionary optimization is highly in demand for handling various real-world applications. One main reason is the lack of benchmark problems that can closely reflect real-world

Part of the work was supported in part by an EPSRC Grant (No. EP/M017869/1), in part by the Ministry of Science and Technology of China Grant (No. 2017YFC0804003), in part by the Anhui Provincial Natural Science Foundation Grant (No. 1908085QF271), in part by the Science and Technology Innovation Committee Foundation of Shenzhen Grant (No. ZDSYS201703031748284), and in part by the Young Scientists Fund Grant (No. 61903178).

✉ Handing Wang
hdwang@xidian.edu.cn

Cheng He
chenghehust@gmail.com

Ye Tian
field910921@gmail.com

Yaochu Jin
yaochu.jin@surrey.ac.uk

[1] Department of Computer Science and Engineering, Southern University of Science and Technology, Shenzhen 518055, China

[2] Institutes of Physical Science and Information Technology, Anhui University, Hefei 230601, China

[3] School of Artificial Intelligence, Xidian University, Xi’an 710071, Shaanxi, China

[4] Department of Computer Science, University of Surrey, Guildford, Surrey Gu2 7XH, UK

challenges, leading to a big gap between academia and industries. In real-world applications, there are a large amount of difficulties which are totally different from the existing benchmark test problems. For example, there may be no exact objective functions to reflect the mappings between the decision variables and the objectives in practice [6], or some noise factors are involved during the fitness evaluation [7], or the computation time of the algorithm is limited due to the hardware limitation/demands [8], or a number of constraints are involved [9], or even the "curse of dimensionality" can result in the failure of optimization algorithm [10].

Despite those mentioned difficulties in real-world applications, many benchmark test suites, which try to mimic the properties of real-world problems, have been used to examine the performance of data-driven EMO algorithms. For instance, the KNO and OKA problems was used in [11]; the Zitzler–Deb–Thiele test suite (ZDT) [12] was used in [13–16]; the Deb–Thiele–Laumanns–Zitzler test suite (DTLZ) [17] was used in [18,19]; and the MF test suite was used in [20]. It is highlighted that these benchmark test suites promote the development of data-driven evolutionary multi-objective optimization, but the abilities of these data-driven EMO algorithms in solving real-world expensive MOPs are not validated. On the other hand, a suite of computationally expensive shape optimization problems using computational fluid dynamics was proposed in [21]. This suite has somehow filled the aforementioned gaps, nevertheless these problems could be relatively too expensive and specific for designing a new algorithm.

Online data-driven evolutionary multiobjective optimization

Online data-driven EMO algorithms are based on conventional EMO algorithms but involve surrogate assists. Therefore, a very general process of online data-driven EMO algorithms consists of surrogate model building, multi-objective optimization, and model management. One or multiple surrogate models are trained to replace the expensive fitness evaluations to guide the search. In the search, new candidate solutions are generated using different variation operators such as crossover and mutation, but they are selected according to the predicted fitness using surrogate model rather than expensive fitness evaluations. During the optimization process, a small number of online data can be selectively sampled via model management strategies to enhance the quality of the surrogate models. To further discuss the methodology of online data-driven algorithms, we briefly introduce four representative algorithms (ParEGO [11], MOEA/D-EGO [16], K-RVEA [19], and CSEA [18]).

Efficient global optimization (EGO) [22] is a very classic online data-driven single-objective optimization algorithm, while it uses a Kriging model as surrogate model and selects new training data based on a infill sampling criterion (e.g., expected improvement). ParEGO [11] extends EGO to multi-objective optimization problems. It employs aggregation functions to decompose one multi-objective optimization problem into a set of single-objective optimization problems. Thus, ParEGO repeatedly uses EGO to solve one random single-objective optimization problem from those aggregation functions, where an evolutionary algorithm is adopted to maximize expected improvement for choosing new online data.

Different from the sequential search for each aggregation function, MOEA/D-EGO [16] simultaneously solve those single-objective optimization problems due to the parallelism of MOEA/D [23]. In MOEA/D-EGO, a Kriging model is built for each objective, then the prediction of aggregation functions and their expected improvement can be calculated. K-RVEA [19] also builds one Kriging model for each objective, but its problem decomposition follows the angle-penalized distance (APD) in RVEA [24].

In fact, classifiers can be used as surrogate models to help evolutionary algorithms distinguish promising candidate solutions for the next generation. CSEA [18] is a representative classification-based EMO algorithm, where a feedforward neural network is adopted to determine whether a solution can be selected or not.

Therefore, designing a new online data-driven EMO algorithm needs to consider the following key points.

– Choice of EMO algorithm: The chosen EMO algorithm is a foundation of an online data-driven EMO algorithm, which significantly affects its performance.
– Choice of surrogate model: The quality of the chosen surrogate model determines whether the evolutionary search can be corrected guided. To improve the robustness of surrogate models, multiple models can be used as an ensemble. Furthermore, surrogate models can approximate the objectives, aggregation functions, performance indicators, and selection for multiobjective optimization problems.
– Choice of online data: The chosen online data can efficiently and economically improve the surrogate models and benefit the following optimization search. Different online data sampling strategies would result in different performance of online data-driven EMO algorithms.

Test problems

We carefully select seven benchmark multiobjective optimization problems from real-world applications, including design of car cab [25], optimization of vehicle frontal structure [26], filter design [27], optimization of power systems [28], portfolio optimization [29], and optimization of neural networks [30]. The objective functions of those problems cannot be calculated analytically, but can be calculated by calling an executable program to provide true black-box evaluations for both offline and online data sampling. A set of initial data is generated offline using Latin hypercube sampling, and a predefined fixed number of online data samples is set as the stopping criterion.

- DDMOP1: This problem is a vehicle performance optimization problem, termed car cab design, which has 11 decision variables and 9 objectives. The decision variables include the dimensions of the car body and bounds on nature frequencies, e.g., thickness of B-Pillar inner, thickness of floor side inner, thickness of door beam, and barrier height. Meanwhile, the nine objectives characterize the performance of the car cab, e.g., weight of the car, fuel economy, acceleration time, road noise at different speed, and roominess of the car.
- DDMOP2: This problem aims at structural optimization of the frontal structure of vehicles for crashworthiness, which involves 5 decision variables and 3 objectives. The decision variables include the thickness of five reinforced members around the frontal structure. Meanwhile, the mass of vehicle, deceleration during the full-frontal crash (which is proportional to biomechanical injuries caused to the occupants), and toe board intrusion in the offset-frontal crash (which accounts for the structural integrity of the vehicle) are taken as objectives, which are to be minimized.
- DDMOP3: This problem is an LTLCL switching ripple suppressor with two resonant branches, which includes 6 decision variables and 3 objectives. This switching ripple suppressor is able to achieve zero impedance at two different frequencies. The decision variables are the design parameters of the electronic components, e.g., capacitors, inductors, and resistors. Meanwhile, the objectives of this problem involve the total cost of the inductors (which is proportional to the consume of the copper and economic cost) and the harmonics attenuations at two different resonant frequencies (which are related to the performance of the designed switching ripple suppressor).
- DDMOP4: This problem is also an LTLCL switching ripple suppressor but with nine resonant branches, including 13 decision variables and 10 objectives. This switching ripple suppressor is able to achieve zero impedance at nine different frequencies. The decision variables are the design parameters of the electronic components, e.g., capacitors, inductors, and resistors. Meanwhile, the objectives of this problem involve the total cost of the inductors and the harmonics attenuations at nine different resonant frequencies.
- DDMOP5: This problem is a reactive power optimization problem with 14 buses, which involves 11 decision variables and 3 objectives. The decision variables include the dimensions of the system conditions, e.g., active power of the generators, initial values of the voltage, and per-unit values of the parallel capacitor and susceptance. Meanwhile, the five objectives characterize the performance of the power system, e.g., active power loss, voltage deviation, reciprocal of the voltage stability margin, generation cost, and emission of the power system.
- DDMOP6: This problem is a portfolio optimization problem, which has 10 decision variables and 2 objectives. The data consist of 10 assets with the closing prices in 100 min. Each decision variable indicates the investment proportion on an asset. The first objective denotes the overall return, and the second objective denotes the financial risk according to the modern portfolio theory.
- DDMOP7: This problem is a neural network training problem, which has 17 decision variables and 2 objectives. The training data consist of 690 samples with 14 features and 2 classes. Each decision variable indicates a weight of the neural network with a size of $14 \times 1 \times 1$. The first objective denotes the complexity of the network (i.e., the ratio of nonzero weights), and the second objective denotes the classification error rate of the neural network.

Specifically, this repository includes six different types of real-world MOPs with different properties, e.g., irregular Pareto fronts/sets, different number of decision variables/objectives, or different problem complexities. For instance, DDMOP1 and DDMOP2 involve complex Pareto fronts/sets; DDMOP3 and DDMOP4 involve complex numbers; DDMOP5 involves multiple local optima; DDMOP6 involves time-series property; DDMOP7 involves noisy during the training. We do not aim to propose a benchmark test suite with specific properties in each test instance. Instead, we aim at evaluating the average performance of data-driven algorithms on different types of problems to support engineers in selecting the candidate optimizer.

General shape of the approximate Pareto front

To generally characterize the Pareto optimal fronts (POFs) of our test problems, we have conducted a long-term simulation on six problems (CSEA [18], NSGA-II [31], K-RVEA [19], ParEGO [11], SPEA2 [32], and NSGA-III [33] with a budget of 1000 real function evaluations are used to optimize each problem), and the non-dominated solutions of the obtained

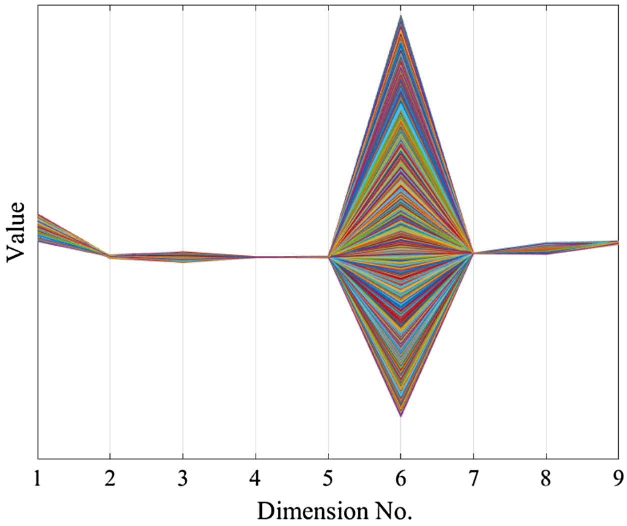

Fig. 1 The approximate POF of DDMOP1

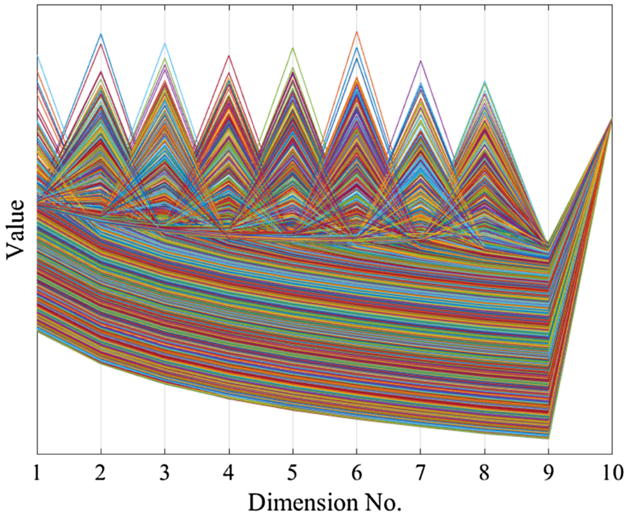

Fig. 2 The approximate POF of DDMOP4

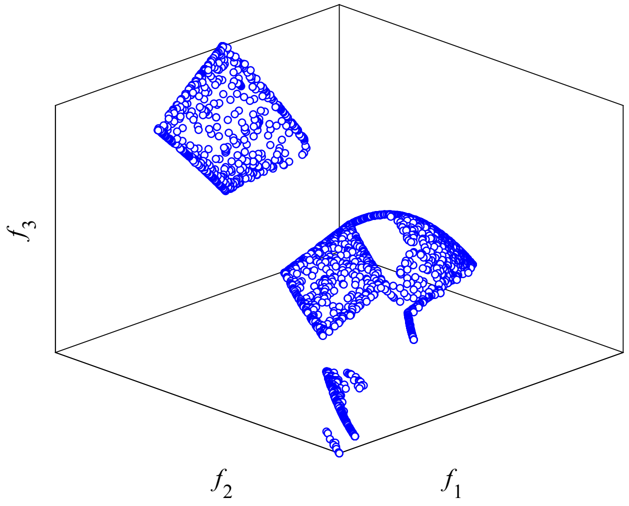

Fig. 3 The approximate POF of DDMOP2

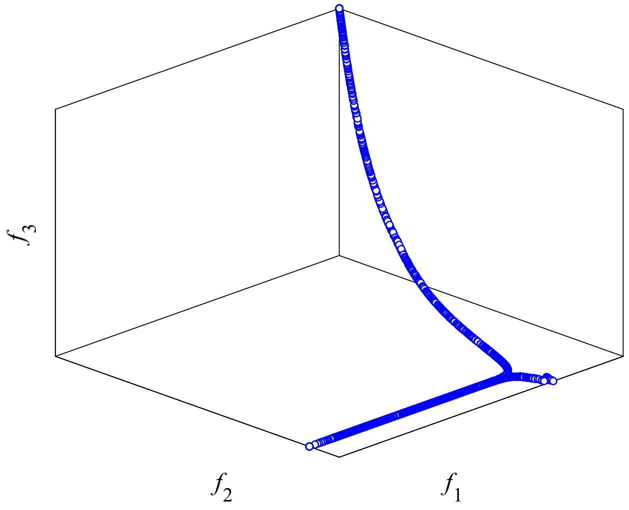

Fig. 4 The approximate POF of DDMOP3

solutions are used to approximate the POFs.[1] Note that we do not give the objective values of the obtained solutions, since we cannot ensure the obtained solutions are exactly on the POFs due to the computationally expensive cost of the real function evaluations.

For DDMOP1 in Fig. 1 and DDMOP4 in Fig. 2, their numbers of objectives are more than eight. It can be observed from these two plots that the variations of the function values are different for all the objectives. Nevertheless, the shape of the approximate Pareto fronts is relatively regular. Hence, the main difficulty is to ensure the convergence of the obtained solution set.

DDMOP2, DDMOP3, and DDMOP5 are problems with three objectives. In Fig. 3, the approximated Pareto front of DDMOP2 is discontinuous and there is a hole on the second part of the approximate Pareto front. Meanwhile, the approximate Pareto front of DDMOP3 degenerates into an irregular curve as shown in Fig. 4. It is difficult to obtain a set of representative solutions evenly distributed around the entire POF for DDMOP2 and DDMOP3. In contrast, the approximate Pareto front of DDMOP5 in Fig. 5 is relatively simple in comparison with the above two problems. Hence, MOEAs should pay more attention to convergence enhancement in solving DDMOP5.

Finally, for DDMOP6 in Fig. 6, the obtained approximate Pareto front is simple, and it can be used to reflect the general performance of MOEAs on solving online data-driven multiobjective optimization problems.

[1] We do not give the result of DDMOP7 as the number of the obtained non-dominated solutions is too small, and it is meaningless to display the result.

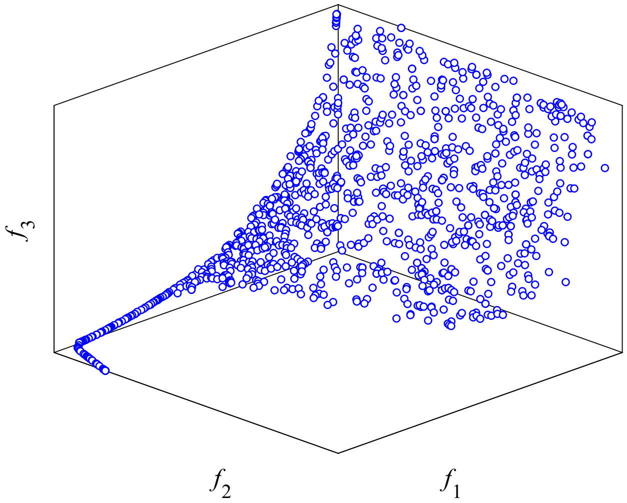

Fig. 5 The approximate POF of DDMOP5

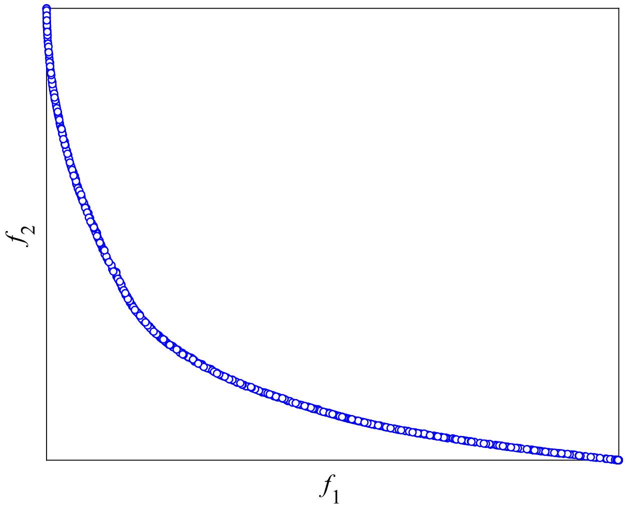

Fig. 6 The approximate POF of DDMOP6

In contrary to most existing benchmark problems with regular formulations, the proposed benchmark test problems are extracted from real-world applications, and the irregularity in the shape of the Pareto fronts encourages us to develop efficient MOEAs with strong ability of diversity maintenance. In all these test problems, the approximate POFs are irregular despite DDMOP6, where the objectives have different scale degrees in DDMOP1 and DDMOP4, the approximate POF of DDMOP3 is a combination of several degenerated curves, the approximate POF of DDMOP2 is discontinuous, the approximate POF of DDMOP5 is a combination of curve and surface, the approximate POF of DDMOP6 is concave, and the objective functions of DDMOP7 are complex due to the existence of neural network.

Software platform information

The proposed test suite has been implemented in MATLAB code.[2] We suggest conducting experiments on the proposed test suite via PlatEMO [34], which is an open source MATLAB-based platform for EMO. PlatEMO currently includes more than 90 representative multiobjective evolutionary algorithms and over 120 benchmark problems, along with a variety of widely used performance indicators. Moreover, PlatEMO provides a simple interface and a friendly graphical user interface, which enable users to efficiently conduct experiments on the proposed test suite with a low learning cost, and users can also investigate the performance of their algorithms on the proposed test suite in comparison to state-of-the-art algorithms.

To test an algorithm on the proposed test suite, users should embed the algorithm in PlatEMO with the specified interface and form, then use the following command: *main* (‘-$algorithm$’,@Alg,‘-$problem$’,@$DDMOP1$,‘-N’,256, ‘-$evaluation$’,400), where @Alg denotes the function handle of the algorithm to be tested, @$DDMOP1$ denotes the function handle of one of the proposed benchmark problems, ‘-N’,256 defines the population size, and ‘-$evaluation$’,400 defines the number of function evaluations (i.e., number of online data samples).

Comparative study

To further examine the performance of existing data-driven optimization algorithms on these problems, four popular EMO algorithms are compared.

Compared algorithms

In this work, we compared three representative data-driven evolutionary algorithms, i.e., CSEA [18], K-RVEA [19], ParEGO [11], and the model-free NSGA-II [31]. NSGA-II is used as the baseline to indicate the superiority of data-driven EMO algorithms in solving computationally expensive multiobjective optimization problems. It is worth noting that K-RVEA and ParEGO both adopt Kriging models, but their approximation targets are different (one Kriging model is adopted to surrogate an objective function in K-RVEA while it is used to surrogate the aggregation function in ParEGO); on the contrary, a feedforward neural network is adopted in CSEA to surrogate a classification criterion.

[2] https://github.com/HandingWang/DDMOP/tree/master/DDMOP_Exp.

Experimental settings

To obtain a set of acceptable solutions from each problem within a bearable time consumption. We recommend the following settings, including the population size of the algorithm and the predefined fixed number of online data samples.

The number of population size is set to 100 for problems with two objectives, i.e., DDMOP6 and DDMOP7. It is set to 105 for problems with three objectives, i.e., DDMOP2, DDMOP3, and DDMOP5. As for problems with ten objectives, i.e., DDMOP1 and DDMOP4, the population size is set to 256. The setting of population size enables the decomposition-based MOEAs to generate a set of uniformly distributed weight vectors/points.

The terminal criterion for the algorithms that will be tested on these problems is the predefined fixed number of online data samples. We set the predefined fixed number of online data samples according to the number of decision variables of the test problems. Hence, it is set to 400, 300, 400, 600, 800, 300, and 600 for DDMOP1 to DDMOP7, respectively. Note that these settings are based on the experimental analysis over a long period function evaluations, and conventional algorithms can achieve an acceptable result with this setting. We do not want to spend too much computational/ economical cost for gaining a relatively small improvement.

Meanwhile, we recommend that each test problem is tested for more than ten independent runs, so we can obtain the statistical results, e.g., mean, variance, and worst/best case result, to analyze the performance of the algorithm.

We recommend that the prefixed number of generations before updating the surrogate model(s) should be less than 30 to create a fair environment for comparison. Meanwhile, we have given the initial population for the compared algorithms to avoid the disturbance caused by the initialization procedure. To conduct fair comparisons, we have used the recommended settings of specific parameters in each adopted algorithm. To be more specific, the number of weight vectors is set to 15, and the maximum number of surrogate-assisted fitness approximation before the surrogate update is set to 200,000 as recommended in ParEGO [11]. For K-RVEA, parameter δ is set to $0.05N$ with N being the population size, and the number of generations $w_{\max}$ before updating the Kriging models is set to 20 as recommended in [19]. Regarding the settings of CSEA, the number of surrogate-assisted prediction before updating the models is equal to that in K-RVEA, the maximum epochs for training the FNN T is set to 500 and the training is terminated once the change of the weights is smaller than 0.001, the number of hidden neurons H is set to $2 \times D$ with D being the number of decision variables, and the number of reference solutions is set to 6 for all the problems [18]. Besides, there is no specific parameter involved in NSGA-II. In this part, we use the MATLAB toolbox DACE [35] to construct the Kriging models for both ParEGO and K-RVEA, where the regression model is set to a constant function, the correlation model is set to the Gaussian process, and other parameters are set the same as the default settings.

Performance indicators

Since the true Pareto fronts of the proposed benchmark problems are unknown, the widely used performance indicator hypervolume (HV) [36] is suggested to quantitatively assess the population obtained in each run. The HV value of a population P with respect to a reference point set R in the objective space is defined as

$$HV(P, R) = \lambda(H(P, R)), \tag{1}$$

where

$$H(P, R) = \{z \in Z | \exists x \in P, \exists r \in R : f(x) \leq z \leq r\},$$

and λ is the Lebesgue measure with

$$\lambda(H(P, R)) = \int_{\mathbb{R}^n} 1_{H(P,R)}(z)\mathrm{d}z, \tag{2}$$

where $1_{H(P,R)}$ is the characteristic function of $H(P, R)$. In short, the HV value of P is the area covered by P with respect to R, and a higher HV value indicates a better convergence as well as a diversity of the points.

To calculate the HV value of a population obtained on each benchmark problem, the reference point set R is set to a single point $(1, 1, \ldots, 1)$. Moreover, we collect a set of non-dominated solutions by conducting a long-term simulation on each problem, which can be used to approximately normalize the population before calculating HV. To be specific, all the objective values of P are normalized according to z^* and $1.1 \times z^{\text{nad}}$, where z^* is the ideal point that consists of the minimum values of all the objectives of the obtained non-dominated solution set, and z^{nad} is the nadir point that consists of the maximum values of all the objectives of the obtained non-dominated solution set. In addition, since the calculation of HV is ineffective for populations with many objectives, the Monte Carlo estimation method with 1,000,000 sampling points is suggested for populations with more than four objectives for higher computational efficiency.

Results

Each problem is tested for 20 independent runs, and the experimental results of the four compared algorithms are given in Table 1. It can be observed that ParEGO has

Table 1 The HV results achieved by the compared algorithms on DDMOP1 to DDMOP7

Problem	CSEA	K-RVEA	NSGA-II	ParEGO
DDMOP1	8.23E+07(4.52E+06)	6.96E+07(2.91E+06)	5.81E+07(2.00E+06)	**1.35E+08(8.67E+05)**
DDMOP2	6.14E+02(1.16E+01)	6.08E+02(6.81E+00)	5.68E+02(9.38E+00)	**6.58E+02(6.45E−01)**
DDMOP3	3.66E+02(2.10E+00)	3.52E+02(6.52E−01)	3.64E+02(1.63E+00)	**3.70E+02(1.45E+00)**
DDMOP4	4.33E+21(4.29E+19)	4.18E+21(3.12E+19)	4.01E+21(5.90E+19)	**4.48E+21(1.37E+19)**
DDMOP5	**2.00E−02(3.66E−18)**	0.00E+00(0.00E+00)	2.00E-02(3.66E−18)	0.00E+00(0.00E+00)
DDMOP6	0.00E+00(0.00E+00)	0.00E+00(0.00E+00)	0.00E+00(0.00E+00)	0.00E+00(0.00E+00)
DDMOP7	**3.00E−01(2.21E−02)**	2.70E−01(2.26E−02)	2.70E−01(1.49E−02)	0.00E+00(0.00E+00)

The best result in each row is highlighted in bold

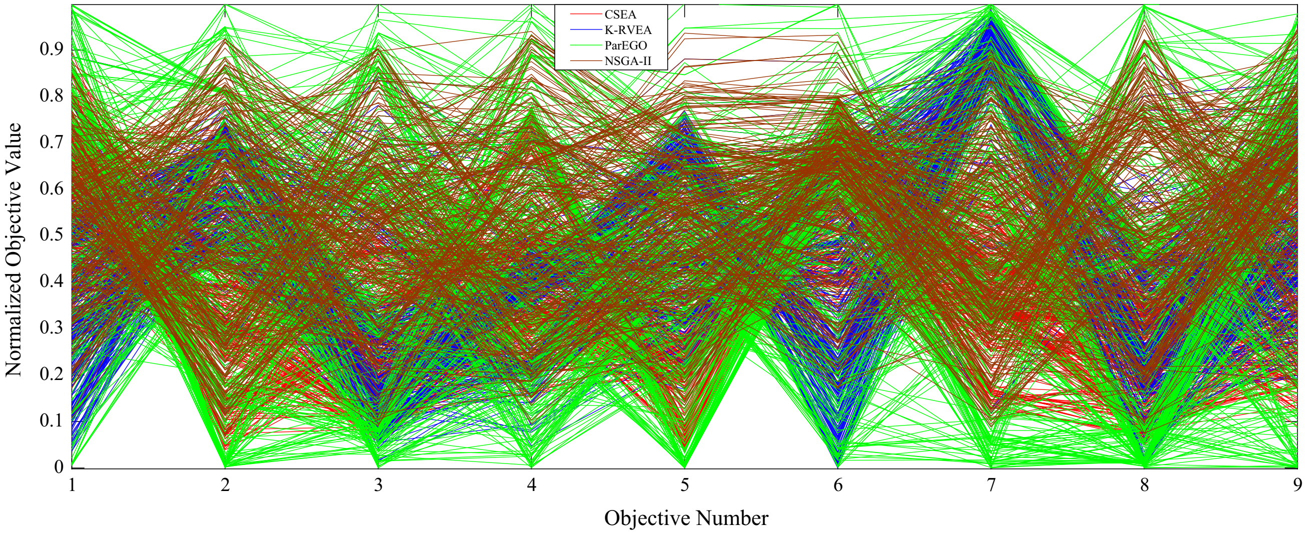

Fig. 7 The non-dominated solutions obtained by each compared algorithm on DDMOP1 in the run associated with the medium HV value

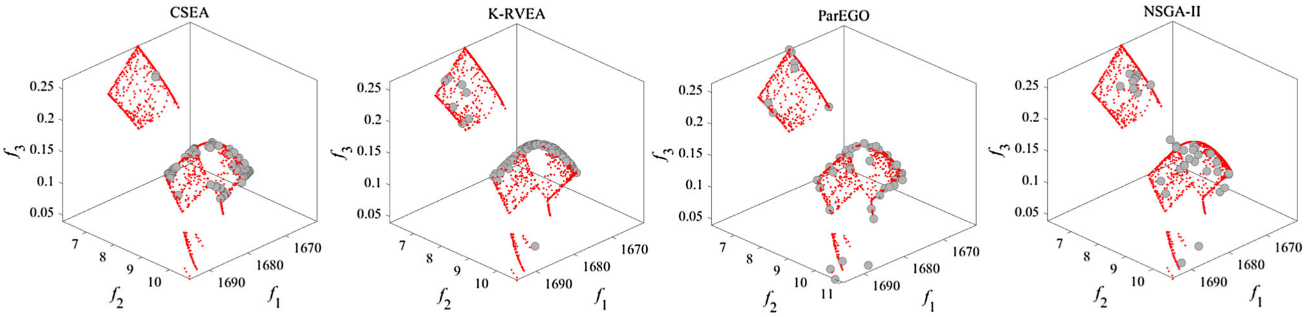

Fig. 8 The non-dominated solutions obtained by each compared algorithm on DDMOP2 in the run associated with the medium HV value

achieved four best results while CSEA has achieved two best results. Besides, the non-dominated solutions obtained by each algorithm on DDMOP1 and DDMOP2 are given in Figs. 7 and 8, respectively, where each solution set is selected from the run in association with the medium HV value. It can be observed from these two figures that CSEA and K-RVEA perform well on DDMOP1 with nine objectives, while ParEGO performs the best on DDMOP2 with two objectives; by contrast, NSGA-II has failed to obtain a set of well-converged solutions. Moreover, the promising results achieved by ParEGO may be attributed to the fact that ParEGO is suitable for this repository. To be more specific, a random weight vector is adopted to transfer the original MOP into a single-objective optimization problem and optimize it independently and, thus, it can obtain a well-converged solution in association with each weight vector greedily. Thus, the bias on convergence over diversity has resulted in better HV results. By contrast, CSEA and K-RVEA tried to strike a

Table 2 The average computation time of all the compared algorithms on each test problem

Problem	CSEA	K-RVEA	NSGA-II	ParEGO
DDMOP1	6.41E+03	6.48E+03	6.40E+03	6.75E+03
DDMOP2	5.73E+03	5.72E+03	5.70E+03	6.19E+03
DDMOP3	5.63E+03	5.62E+03	5.60E+03	5.92E+03
DDMOP4	5.19E+03	5.28E+03	5.10E+03	5.10E+03
DDMOP5	5.99E+03	6.04E+03	5.92E+03	6.78E+03
DDMOP6	4.39E+03	4.38E+03	4.35E+03	4.58E+03
DDMOP7	4.05E+03	3.88E+03	3.74E+03	4.91E+03

balance between the convergence enhancement and diversity maintenance, and thus wasted real-objective evaluations on problems with complex PFs, e.g., DDMOP1 to DDMOP5. Overall, the three data-driven algorithms have outperformed NSGA-II, indicating their effectiveness in handling computationally expensive optimization problems.

Computation time

The average computation time of each algorithm on DDMOP1 to DDMOP7 over ten independent runs is displayed in Table 2. It can be observed that the computation time of all the compared algorithms on each test problem is similar, which is attributed to the computationally expensive properties of the proposed problems. To be more specific, NSGA-II has achieved the shortest computation time since it is a model-free algorithm, followed by CSEA, K-RVEA, and ParEGO. Meanwhile, CSEA has achieved the similar results with K-RVEA; in contrast, ParEGO has spent the most computation time on each problem, which may be attributed to the increasing scale of the training set. Note that in ParEGO, all the newly evaluated solutions are merged to the dataset for training the Kriging model; by contrast, K-RVEA maintains a constant number of samples for training the model.

Conclusion

In this work, we have proposed a repository of real-world datasets for data-driven EMO. We first give the prosperities of these real-world problems and their approximate Pareto optimal fronts. Then, the performance of four popular algorithms, including three data-driven EMO algorithm and a model-free EMO algorithm, is analyzed. From the perspective of problem properties, the proposed repository of real-world datasets has covered different problems with different irregular/regular Pareto optimal fronts. Besides, the problem complexities of the problems are different, which can be observed from Table 1.

This repository has been used as the benchmark test problems for IEEE Congress on Evolutionary Computation 2019 "Online Data-Driven Multi-Objective Optimization Competition". The motivation of proposing this repository is to promote the research in data-driven multiobjective optimization, in terms of both algorithm design and application of these algorithms to real-world problems. Furthermore, this repository could provide a new benchmark test suite for examining the performance of existing data-driven EMO algorithms on real-world problems.

References

1. Jin Y, Wang H, Chugh T, Guo D, Miettinen K (2018) Data-driven evolutionary optimization: an overview and case studies. IEEE Trans Evol Comput 23(3):442–458
2. Tinkle Chugh, Karthik Sindhya, Jussi Hakanen, Kaisa Miettinen (2017) A survey on handling computationally expensive multiobjective optimization problems with evolutionary algorithms. Soft Comput 2017:1–30
3. Jin Y (2011) Surrogate-assisted evolutionary computation: recent advances and future challenges. Swarm Evol Comput 1(2):61–70
4. Handing W, Yaochu J, Jansen Jan O (2016) Data-driven surrogate-assisted multiobjective evolutionary optimization of a trauma system. IEEE Trans Evol Comput 20(6):939–952
5. Cheng R, He C, Jin Y, Yao X (2018) Model-based evolutionary algorithms: a short survey. Complex Intell Syst 4(4):283–292
6. Zingg David W, Marian N, Pulliam Thomas H (2008) A comparative evaluation of genetic and gradient-based algorithms applied to aerodynamic optimization. Eur J Comput Mech 17(1–2):103–126
7. Wang H, Zhang Q, Jiao L, Yao X (2015) Regularity model for noisy multiobjective optimization. IEEE Trans Cybern 46(9):1997–2009
8. Mavrovouniotis M, Li C, Yang S (2017) A survey of swarm intelligence for dynamic optimization: algorithms and applications. Swarm Evol Comput 33:1–17
9. Zhou Aimin Q, Bo-Yang LH, Shi-Zheng Z, Nagaratnam SP, Qingfu Z (2011) Multiobjective evolutionary algorithms: a survey of the state of the art. Swarm Evol Comput 1(1):32–49
10. Forrester A, Sobester A, Keane A (2008) Engineering design via surrogate modelling: a practical guide. Wiley, New York
11. Knowles J (2006) ParEGO: a hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems. IEEE Trans Evol Comput 10:50–66
12. Zitzler E, Deb K, Thiele L (2000) Comparison of multiobjective evolutionary algorithms: empirical results. Evol Comput 8(2):173–195
13. Zhan D, Cheng Y, Liu J (2017) Expected improvement matrix-based infill criteria for expensive multiobjective optimization. IEEE Trans Evol Comput 21(6):956–975
14. Loshchilov I, Schoenauer M, Sebag M (2010) A mono surrogate for multiobjective optimization. In: Proceedings of the 12th annual conference on genetic and evolutionary computation. ACM, pp 471–478

15. Hussein R, Deb K (2016) A generative Kriging surrogate model for constrained and unconstrained multi-objective optimization. In: Proceedings of the genetic and evolutionary computation conference 2016. ACM, pp 573–580
16. Zhang Q, Liu W, Tsang E, Virginas B (2010) Expensive multiobjective optimization by MOEA/D with gaussian process model. IEEE Trans Evol Comput 14(3):456–474
17. Deb K, Thiele L, Laumanns M, Zitzler E (2005) Scalable test problems for evolutionary multiobjective optimization. Advanced information and knowledge processing. Springer, London
18. Pan L, He C, Tian Y, Wang H, Zhang X, Jin Y (2019) A classification-based surrogate-assisted evolutionary algorithm for expensive many-objective optimization. IEEE Trans Evol Comput 23(1):74–88
19. Chugh T, Jin Y, Miettinen K, Hakanen J, Sindhya K (2018) A surrogate-assisted reference vector guided evolutionary algorithm for computationally expensive many-objective optimization. IEEE Trans Evol Comput 22(1):129–142
20. Dudy L, Yaochu J, Soon OY, Bernhard S (2010) Generalizing surrogate-assisted evolutionary computation. IEEE Trans Evol Comput 14(3):329–355
21. Daniels SJ, Rahat AAM, Everson RM, Tabor GR, Fieldsend JE (2018) A suite of computationally expensive shape optimisation problems using computational fluid dynamics. In: International conference on parallel problem solving from nature. Springer, pp 296–307
22. Jones Donald R, Matthias S, Welch William J (1998) Efficient global optimization of expensive black-box functions. J Glob Optim 13(4):455–492
23. Zhang Q, Li H (2007) MOEA/D: a multiobjective evolutionary algorithm based on decomposition. IEEE Trans Evol Comput 11(6):712–731
24. Cheng R, Jin Y, Olhofer M, Sendhoff B (2016) A reference vector guided evolutionary algorithm for many-objective optimization. IEEE Trans Evol Comput 20(5):773–791
25. Deb K, Gupta S, Daum D, Branke J, Mall AK, Padmanabhan D (2009) Reliability-based optimization using evolutionary algorithms. IEEE Trans Evol Comput 13(5):1054–1074
26. Deb K, Jain H (2014) 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):577–601
27. Zhang Z, He C, Ye J, Jinbang X, Pan L (2019) Switching ripple suppressor design of the grid-connected inverters: a perspective of many-objective optimization with constraints handling. Swarm Evol Comput 44:293–303
28. Kavasseri R, Srinivasan SK (2011) Joint placement of phasor and power flow measurements for observability of power systems. IEEE Trans Power Syst 26(4):1929–1936
29. Paolini R, Marson P, Vicarioto M, Ongaro G, Viero M, Girolami A (2012) Portfolio optimization using a multi-objective, risk-adjusted evolutionary algorithm. In: Computational intelligence for financial engineering and economics, pp 303–311
30. Jin Y, Sendhoff B (2008) Pareto-based multiobjective machine learning: an overview and case studies. IEEE Trans Syst Man Cybern Part C 38(3):397–415
31. Deb K, Pratap A, Agarwal S, Meyarivan T (2002) A fast and elitist multi-objective genetic algorithm: NSGA-II. IEEE Trans Evol Comput 6:182–197
32. Zitzler E, Laumanns M, Thiele L et al (2001) SPEA2: improving the strength Pareto evolutionary algorithm for multiobjective optimization. Eurogen 3242:95–100
33. Deb K, Jain H (2014) An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part I: solving problems with box constraints. IEEE Trans Evol Comput 18:577–601
34. Tian Y, Cheng R, Zhang X, Jin Y (2017) PlatEMO: a MATLAB platform for evolutionary multi-objective optimization. IEEE Comput Intell Mag 12(4):73–87
35. Lophaven Søren Nymand, Nielsen Hans Bruun, Søndergaard Jacob (2002) DACE-a matlab kriging toolbox, version 2.0
36. Coello Carlos A, Coello VV, David A, Lamont Gary B (2002) Evolutionary algorithms for solving multi-objective problems. Springer, Berlin

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.