Postprint of High-Dimensional Many-Objective Evolutionary Algorithm Based on New Fitness Function and Multi-Search Strategies

Authors: Dai Cai, Shi Xiaoqi

Date: 2018-11-29T00:00:00+00:00

Abstract

Balancing convergence and diversity in high-dimensional many-objective evolutionary algorithms constitutes a difficult and challenging endeavor. To enhance the performance of high-dimensional many-objective evolutionary algorithms, this paper proposes a high-dimensional many-objective evolutionary algorithm based on a novel fitness function and multi-search strategy. The algorithm introduces a new fitness function to balance diversity and convergence, and designs a multi-search strategy to assist the crossover operator in generating superior offspring, thereby improving convergence. The fitness function first selects individuals with better convergence from the current population and newly generated offspring, and then calculates the sparsity degree of these individuals; the multi-search strategy selects sparse and convergent solutions to execute global and local search. Numerical experiments were conducted on 15 test problems from the CEC2018 many-objective competition, with the number of objectives for each test problem being 5, 10, and 15, respectively. Experimental results demonstrate that the algorithm can obtain a solution set with better diversity and convergence than four representative algorithms (e.g., NSGAIII, MOEA/DD, KnEA, RVEA).

Full Text

Preamble

Vol. 37 No. 1
Application Research of Computers
ChinaXiv Partner Journal
Accepted Paper

Many-Objective Evolutionary Algorithm Based on a New Fitness Function and Multi-Search Strategy

Dai Cai, Shi Xiaoqi
(College of Computer Science, Shaanxi Normal University, Xi'an 710119, China)

Abstract: Balancing convergence and diversity in many-objective evolutionary algorithms is a difficult and challenging task. To improve the performance of many-objective evolutionary algorithms, this paper proposes a many-objective evolutionary algorithm based on a new fitness function and multi-search strategy. The algorithm introduces a novel fitness function to balance diversity and convergence, and designs a multi-search strategy to help crossover operators generate excellent offspring, thereby improving convergence. The fitness function first selects individuals with good convergence from the current population and newly generated offspring, then calculates the sparsity of these individuals. The multi-search strategy selects sparse and convergent solutions to perform global and local search. Numerical experiments test 15 benchmark problems from the CEC2018 competition on many-objective optimization, with each problem having 5, 10, and 15 objectives. Experimental results demonstrate that the proposed algorithm can find solution sets with better diversity and convergence than four representative algorithms (NSGAIII, MOEA/DD, KnEA, and RVEA).

Keywords: many-objective optimization; fitness function; multi-search strategy; evolutionary algorithm
Classification: TP391.9
doi: 10.19734/j.issn.1001-3695.2018.06.0485

---

0 Introduction

Over the past decades, numerous evolutionary algorithms [?] have been proposed to solve multi-objective optimization problems. Multi-objective evolutionary algorithms use population evolution strategies to obtain a set of optimal solutions and represent an effective approach for solving multi-objective optimization problems. In the real world, many multi-objective optimization problems [?] involve more than three optimization objectives, which are generally referred to as many-objective optimization problems [?]. Evolutionary algorithms are population-based stochastic search methods that have been widely applied to solve complex nonlinear problems, as they only require objective function values. However, when addressing many-objective optimization problems, most existing multi-objective evolutionary algorithms exhibit poor convergence performance.

When using evolutionary algorithms to solve many-objective optimization problems, two major challenges arise:
a) As the number of objectives increases, the number of solutions required to approximate the Pareto front grows exponentially. If the population size is large, the computational cost becomes substantial; if the population size is small, the proportion of non-dominated individuals increases rapidly with the number of objectives, causing the selection pressure of Pareto dominance to diminish and leading to convergence loss in Pareto dominance-based multi-objective evolu-

tionary algorithms.
b) As the number of objectives increases, the objective space grows exponentially. Maintaining population diversity reduces the correlation among individuals, which decreases the convergence speed of multi-objective evolutionary algorithms. Conversely, strengthening the correlation among individuals in the population makes it impossible to maintain population diversity.

---

1 Related Research

Given the characteristics of many-objective optimization problems, four main categories of methods currently exist for solving them:

a) **Dimensionality reduction methods.** The core idea of these approaches is to enhance algorithmic convergence pressure by reducing the number of objectives. Luo et al. [?] constructed a conflict information matrix by measuring the conflict among objectives on approximate solution sets, performed eigenvalue analysis on this matrix to determine the importance of objectives, and thereby achieved objective reduction. Zou et al. [?] determined which objectives to eliminate based on changes in the proportion of non-dominated solutions after removing different objective functions. However, even after reduction, the remaining number of objectives may still exceed three, meaning these methods can improve the convergence speed of many-objective evolutionary algorithms but cannot fundamentally solve many-objective optimization problems.

b) **Indicator-based methods.** These approaches guide algorithmic search by evaluating solution set quality (e.g., diversity and convergence). The most representative algorithm in this category is the indicator-based evolutionary algorithm framework proposed by Friedrich et al. [?], which uses indicator functions to rank individuals and requires these indicators to have "dominance relationship preservation." Since the true Pareto front of most problems may be unknown, the only metrics that can accurately evaluate solution set quality are the hypervolume indicator and the R2 indicator [?]. Falcon-Cardona et al. [?] applied the R2 indicator to ant colony algorithms for solving many-objective optimization problems. Bader et al. [?] used Monte Carlo simulation to approximate hypervolume values efficiently and applied it to indicator-based evolutionary algorithms for many-objective optimization. The main drawback of using hypervolume as an indicator is that its computational complexity grows exponentially with the number of objectives [?]. Even when approximating hypervolume values, the time complexity remains high. The R2 indicator's limitation is that finding an appropriate set of utility functions is difficult when facing problems with complex Pareto fronts, which directly affects the performance of R2-based multi-objective evolutionary algorithms. Although computing R2 is less complex than computing hypervolume, its complexity remains higher than other types of algorithms, limiting the application of these methods to many-objective optimization problems.

c) **Ranking methods.** The core idea of these approaches is to increase selection pressure by ranking individuals. For example, Fleming et al. [?] applied preference information to Pareto-based ranking methods, gradually determining preferences based on evolving results, continuously narrowing the region of interest to decision-makers, and finally isolating the decision-maker's preferred optimal solutions. Wang et al. [?] proposed a ranking method based on corner solutions. Dai et al. [?] balanced diversity and convergence in many-objective evolutionary algorithms by improving the α-dominance relation. Lin et al. [?] applied the α-dominance relation to evolutionary algorithms for solving many-objective optimization problems. Modified Pareto dominance methods can improve algorithm convergence speed, but additional methods are still needed to maintain solution set diversity.

d) **Decomposition-based methods.** These approaches use decomposition techniques to construct reference directions or reference points that guide population solutions toward them, thereby enhancing convergence while maintaining good solution set diversity (when reference points or directions are diverse, population diversity is generally also good). For example, Deb et al. [?] used a set of reference points as preferences to obtain the optimal solution set for each single-objective optimization subproblem, then guided non-dominated solutions to approach these reference solutions. Yuan et al. [?] classified the current population according to a set of reference points and used aggregation functions to rank solutions within each class, making population solutions approach reference points to achieve improved convergence and diversity. Several improved decomposition-based multi-objective evolutionary algorithms [?, ?, ?] have also been proposed for many-objective optimization problems.

We believe that both diversity and convergence of solution sets can be improved by generating excellent offspring. To achieve this goal, we first designed a multi-search strategy to help crossover operators perform global and local search. This search strategy selects sparse and well-converged individuals as parent individuals to enhance search efficiency. Additionally, we designed a new fitness function to balance population convergence and diversity. This fitness function first selects better-converged individuals from the current population and newly generated offspring, which helps improve population convergence. It then removes crowded individuals based on the product of the Euclidean norm and infinity norm between objective vectors, which helps maintain population diversity.

2 Concepts in Many-Objective Optimization

This section introduces the main concepts in many-objective optimization. A many-objective optimization problem [?] can be formulated as:

$$\begin{aligned} \min \quad & F(x) = (f_1(x), f_2(x), \dots, f_m(x)) \\ \text{s.t.} \quad & g_i(x) \le 0, \quad i = 1, 2, \dots, q \\ & h_j(x) = 0, \quad j = 1, 2, \dots, p \end{aligned}$$

where $x = (x_1, \dots, x_n) \in \mathbb{R}^n$ is an n-dimensional decision variable vector; $f_i(x)$ are objective functions, with m being the number of objectives (greater than 3); $g_i(x)$ are q inequality constraints; and $h_j(x)$ are p equality constraints. The set of all feasible solutions satisfying these constraints is defined as Ω.

Four important concepts in many-objective optimization are introduced below:
a) For two given solutions $x, z \in \Omega$, if $f_i(x) \le f_i(z)$ for all $i = 1, \dots, m$, then x is said to dominate z (or $x \preceq z$).
b) If a solution x is not dominated by any other solution, then x is a Pareto optimal solution.
c) The set of all Pareto optimal solutions is called the Pareto optimal solution set (PS).
d) The set of objective vectors corresponding to all Pareto optimal solutions is called the Pareto optimal front (PF).

---

3 Many-Objective Evolutionary Algorithm Based on a New Fitness Function and Multi-Search Strategy

To better balance convergence and diversity in many-objective evolutionary algorithms, a new fitness function is designed to maintain population diversity and increase convergence pressure, while a multi-search strategy is employed to help crossover operators generate excellent offspring, thereby improving population diversity and convergence. This section introduces the new fitness function and multi-search strategy in detail.

3.1 New Fitness Function

The primary goal of the new fitness function is to help the algorithm maintain population diversity and increase convergence pressure. To compute this fitness function, a set of uniformly distributed weight vectors $W = \{\lambda^1, \lambda^2, \dots, \lambda^N\}$ is required, where $\lambda^i = (\lambda^i_1, \dots, \lambda^i_m)$ for $i = 1, \dots, N$ are weight vectors. For each weight vector λ^i, the best individual is first identified using Equation (2), which is a Chebyshev function. The optimal solution of Equation (2) is a Pareto optimal solution of Equation (1).

$$\Delta(x, \lambda^i) = \max_{1 \le j \le m} \left\{ \lambda^i_j |f_j(x) - Z^*_j| \right\}$$
$$\text{where } Z^* = (Z_1, \ldots, Z_m) \text{ and } Z_j = \min\{f_j(x) | x \in \Omega\}$$

Second, a threshold DP for individual convergence is set, and the crowding degree of each individual is calculated using Equation (3).

$$DP = \{\|F(x) - Z^*\|_2 \mid x \in POP\}$$

where $\|F(x)-Z^*\|_2$ represents the Euclidean distance between two solutions, and $\|F(x) - Z^*\|_\infty$ represents the maximum difference between solutions. Finally, the fitness value of each individual is computed using Equation (4).

$$Fit(x) = \begin{cases} \max_{y \in POP}\{d(y)\} + \|F(x) - Z^*\|_2 - DP, & \text{if } \|F(x) - Z^*\|_2 \le DP \\ d(x), & \text{otherwise} \end{cases}$$

where $d(x) = \min_{y \in POP, y \neq x}\{\|F(x) - F(y)\|_2 \cdot \|F(x) - F(y)\|_\infty\}$. The larger the fitness value of an individual, the better the individual. The condition $\|F(x)-Z^*\|_2 \le DP$ can enhance algorithmic selection pressure, thereby improving convergence.

The fundamental principle of this fitness function is: using given weight vectors and the Chebyshev function, the many-objective optimization problem is transformed into a set of single-objective optimization problems. In each generation of evolution, the current optimal solution for each single-objective subproblem can be identified. Then DP is used to judge the convergence of other solutions, and Equation (3) measures the sparsity of each solution. This fitness function can measure both solution convergence and sparsity.

3.2 Multi-Search Strategy and Crossover Operator

Search strategies play a crucial role in many-objective evolutionary algorithms. An appropriate search strategy can help crossover operators perform global and local search. This paper employs a multi-search strategy to balance global and local search. Before introducing this strategy, the Euclidean distance between any two weight vectors is computed, and the T nearest weight vectors to each weight vector are identified. Let $B(i) = \{i_1, \ldots, i_T\}$ record the T nearest neighbor weight vectors of weight vector λ^i, where $\lambda^{i_1}, \ldots, \lambda^{i_T}$ are the T nearest weight vectors to λ^i.

The first search strategy helps the crossover operator perform local search. The crossover operator used here (differential evolution [?]) is shown in Equation (5).

$$x_j^{\text{new}} = \begin{cases} x_j^{r1} + L \cdot (x_j^{r2} - x_j^{r3}), & \text{if rand}(0,1) < CR \\ x_j^{r1}, & \text{otherwise} \end{cases}$$

where $x = (x_1, \dots, x_n)$, $j = 1, \dots, n$, $r1, r2, r3$ are two random numbers in $[0, 1]$, $L \in [0, 2]$ is a scaling factor controlling search step size, and CR is the crossover probability. This search strategy selects parent individuals from the convergence-promising set $P_1 = \{x^i \mid \Delta(x^i, \lambda^i) = \min_{x \in POP} \Delta(x, \lambda^i), i = 1, \dots, N\}$, and these parent individuals are neighbors of each other, which helps perform local search and accelerate convergence speed.

The second search strategy helps the crossover operator in Equation (5) perform global search. First, the fitness value of each individual in the current population is calculated using Equation (4). Then, several better individuals are selected using roulette wheel selection. In Equation (5), x^{r1} is one of these better individuals, while x^{r2} and x^{r3} are any two different solutions from the current population. Since x^{r2} and x^{r3} are selected arbitrarily, the value of $\|x^{r2} - x^{r3}\|$ will be relatively large, which facilitates global search. In this work, each search strategy produces half of the offspring.

3.3 Algorithm Steps

Based on the above components, a many-objective evolutionary algorithm based on a new fitness function and multi-search strategy (MaOEA-FM) is proposed. The pseudocode of this algorithm is as follows:

Algorithm: Pseudocode of MaOEA-FM

Input: MaOP (1); stopping condition; N: number of weight vectors and population size; T: number of neighbors for each weight vector, $0 < T < N$; $W = \{\lambda^1, \dots, \lambda^N\}$: N uniformly distributed weight vectors.

Output: Approximate Pareto optimal front.

Initialization: Randomly generate an initial population $POP = \{x^1, \dots, x^N\}$; determine $Z^* = (z_1, \dots, z_m)$; determine reference points $B(i) = \{i_1, \dots, i_T\}$ for $i = 1, \dots, N$, where $B(i)$ contains the T nearest neighbor weight vectors of λ^i.

while stopping condition is not satisfied **do**

1. For each weight vector λ^i, compute the current optimal solution for its subproblem using Equation (2). Generate N new solutions using the crossover operator (Equation (5)) and the first search strategy, and add these new solutions to set O.
2. Compute the fitness value of each individual in POP using Equation (4). Select N better individuals using roulette wheel selection. For each selected individual, generate a new solution using the second search strategy and the crossover operator (Equation (5)), and add these new solutions to set O.

3. For $j = 1, \dots, m$, update $z_j = \min\{z_j, f_j(x) \mid x \in O\}$.
4. Update population: $POP = POP \cup O$; compute the fitness function value of each individual in POP; let $POP = \emptyset$; select the N individuals with the largest fitness values and add them to POP.

end while

---

4 Experiments

4.1 Test Problems and Parameter Settings

This paper uses test problems from the CEC2018 [?] competition on many-objective optimization to verify the effectiveness of the proposed algorithm. The CEC2018 competition includes 15 test problems with diverse characteristics that can effectively test the strengths and weaknesses of many-objective evolutionary algorithms. Four state-of-the-art many-objective evolutionary algorithms (NSGAIII [?], MOEA/DD [?], KnEA [?], and RVEA [?]) are used for comparison. The source code for these four algorithms is obtained from PlatEMO [?], and their parameter settings use the default values from PlatEMO. In MaOEA-FM, $CR = 0.5$ and $L = 0.5$. Other experimental parameter settings (such as population size, maximum number of function evaluations, number of independent runs, problem dimensionality, and variable ranges) are the same as those used in the CEC2018 competition; details can be found in [?].

4.2 Performance Metrics

In this experiment, the inverted generational distance (IGD) [?] is used to quantitatively measure algorithm performance. IGD comprehensively measures both convergence and diversity of a population. A smaller IGD value indicates better diversity and convergence of the measured population. For each test problem, 10,000 points are uniformly sampled from the Pareto optimal front to calculate the IGD value. Additionally, the Wilcoxon rank-sum test [?] is used to measure statistical significance, with the significance level set to 0.05. The symbols ("+"), ("="), and ("-") indicate that MaOEA-FM performs better than, equal to, or worse than the compared algorithm, respectively.

4.3 Numerical Experimental Results

shows the IGD metric values obtained by MaOEA-FM and the four comparison algorithms on the CEC2018 competition problems, where F1-k denotes a many-objective problem with k objectives for F1. The best performance results are marked in bold.

From Table 1, it can be observed that across the 45 test problems, MaOEA-FM achieves significantly better IGD values than NSGAIII, MOEA/DD, KnEA, and RVEA on 38, 40, 36, and 39 problems, respectively. This indicates that on most problems, the quality of populations obtained by MaOEA-FM is superior to those obtained by NSGAIII, MOEA/DD, KnEA, and RVEA.

For the eight problems whose Pareto optimal fronts do not fully cover the unit hyperplane (F1, F2, F4, F7, F8, F9, and F15), the mean IGD values obtained by MaOEA-FM are smaller than those of NSGAIII, MOEA/DD, KnEA, and RVEA on 22, 21, 21, and 21 problems, respectively. For the six problems whose Pareto optimal fronts completely cover the unit hyperplane (F3, F10-F14), the mean IGD values obtained by MaOEA-FM are smaller than those of NSGAIII, MOEA/DD, KnEA, and RVEA on 15, 16, 14, and 15 problems, respectively. These comparison results demonstrate that the proposed fitness function can effectively maintain population diversity and convergence for problems with different types of Pareto optimal fronts.

For the three problems with many local Pareto optimal fronts (F3, F4, and F7), MaOEA-FM shows better convergence than NSGAIII, MOEA/DD, KnEA, and RVEA on most problems, indirectly indicating that the multi-search strategy helps improve search efficiency.

---

5 Conclusion

To address the challenges faced by existing multi-objective evolutionary algorithms when solving many-objective optimization problems, this paper proposes an evolutionary algorithm based on a new fitness function and multi-search strategy. The algorithm designs a novel fitness function to maintain population diversity and improve convergence, and employs a multi-search strategy to balance global and local search. Comparison with recent and effective many-objective evolutionary algorithms on CEC2018 competition problems demonstrates the effectiveness of the proposed algorithm. Future work will investigate the performance of this algorithm on real-world many-objective optimization problems.

References

[1] Cheng Ran, Rodemann F T M, Olhofer M, et al. Evolutionary many-objective optimization of hybrid electric vehicle control: from general optimization to preference articulation [J]. IEEE Trans on Emerging Topics in Computational Intelligence, 2017, 1(2): 97-111.

[2] Zuo Liyun, Shu Lei, Dong Shoubin, et al. A multi-objective optimization scheduling method based on the ant colony algorithm in cloud computing [J]. IEEE Access, 2014, 3: 2687-2699.

[3] Zhang Yong, Gong Dunwei, Cheng Jian. Multi-objective particle swarm optimization approach for cost-based feature selection in classification [J].

IEEE/ACM Trans on Computational Biology and Bioinformatics, 2017, 14(1): 64-75.

[4] Narukawa K, Rodemann T. Examining the performance of evolutionary many-objective optimization algorithms on a real-world application [C]//Proc of the 6th International Conference on Genetic and Evolutionary Computing. 2012: 316-319.

[5] Lygoe R J, Cary M, Fleming P J. A real-world application of a many-objective optimisation complexity reduction process [C]//Proc of the 7th International Conference on Evolutionary Multi-Criterion Optimization. 2013: 641-655.

[6] Liu Hai-Lin, Chen Lei, Zhang Qingfu, et al. Adaptively allocating search effort in challenging many-objective optimization problems [J]. IEEE Trans on Evolutionary Computation, 2018, 22(3): 433-448.

[7] Hughes E J. Evolutionary many-objective optimisation: many once or one many? [C]//Proc of IEEE Congress on Evolutionary Computation. 2005: 222-227.

[8] Luo Nai-li, Li Xia, Wang Na. Objective reduction using conflict information in many-objective optimization evolutionary algorithm [J]. Journal of Signal Processing, 2017, 33(9): 1169-1178.

[9] Zou Juan, Zheng Jinhua, Shen Ruiming, et al. A novel metric based on changes in Pareto domination ratio for objective reduction of many-objective optimization problems [J]. Journal of Experimental & Theoretical Artificial Intelligence, 2017, 29(5): 983-994.

[10] Friedrich T, Horoba C, Neumann F. Multiplicative approximations and the hypervolume indicator [C]//Proc of Genetic and Evolutionary Computation Conference. 2009: 571-578.

[11] Falcon-Cardona J G, Coello C A. A new indicator-based many-objective ant colony optimizer for continuous search spaces [J]. Swarm Intelligence, 2017, 11(1): 71-100.

[12] Bader J, Zitzler E. HypE: an algorithm for fast hypervolume-based many-objective optimization [J]. Evolutionary Computation, 2011, 19: 45-76.

[13] Beume N, Fonseca C, López-Ibáñez M, et al. On the complexity of computing the hypervolume indicator [J]. IEEE Trans on Evolutionary Computation, 2009, 13(5): 1075-1082.

[14] Fleming P J, Purshouse R C, Lygoe R J. Many-objective optimization: an engineering design perspective. Evolutionary multi-criterion optimization [M]. Berlin: Springer-Verlag, 2005: 14-32.

[15] Wang Handing, Yao Xin. Corner sort for Pareto-based many-objective optimization [J]. IEEE Trans on Cybernetics, 2014, 44(1): 92-102.

[16] Dai Cai, Wang Yuping, Hu Lijuan. An improved alpha-dominance strategy for many-objective optimization problems [J]. Soft Computing, 2016, 20(3): 899-910.

[17] Lin Mengman, Zhou Huan, Wang Liping. Research of many-objective evolutionary algorithm based on Alpha dominance [J]. Computer Science, 2017, 44(1): 264-270.

[18] Deb K, Jain H. An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, Part I: solving problems with box constraints [J]. IEEE Trans on Evolutionary Computation, 2014, 18(4): 577-601.

[19] Yuan Yuan, Xu Hua, Wang Bo, et al. A new dominance relation-based evolutionary algorithm for many-objective optimization [J]. IEEE Trans on Evolutionary Computation, 2016, 20(1): 16-37.

[20] Zhang Qingfu, Li Hui. MOEA/D: a multiobjective evolutionary algorithm based on decomposition [J]. IEEE Trans on Evolutionary Computation, 2007, 11(6): 712-731.

[21] Zheng Wei, Tan Yanyan, Fang Xiaonan, et al. An improved MOEA/D with optimal DE schemes for many-objective optimization problems [J]. Algorithms, 2017, 10(3): 86.

[22] Ishibuchi H, Doi K, Nojima Y. On the effect of normalization in MOEA/D for multi-objective and many-objective optimization [J]. Complex & Intelligent Systems, 2017, 3(4): 279-294.

[23] Price K, Storn R M, Lampinen J A. Differential evolution: a practical approach to global optimization, ser. natural computing [M]. Secaucus, NJ, USA: Springer-Verlag, 2005.

[24] Cheng Ren. Benchmark functions for CEC'2018 competition on many-objective optimization [R]. Birmingham: School of Computer Science, University of Birmingham Edgbaston, 2017.

[25] Li Ke, Deb K, Zhang Qingfu, et al. An evolutionary many-objective optimization algorithm based on dominance and decomposition [J]. IEEE Trans on Evolutionary Computation, 2015, 19(5): 694-716.

[26] Zhang Xingyi, Tian Ye, Jin Yaochu. A knee point driven evolutionary algorithm for many-objective optimization [J]. IEEE Trans on Evolutionary Computation, 2015, 19(6): 761-776.

[27] Cheng Ren, Jin Yaochu, Olhofer M, et al. A reference vector guided evolutionary algorithm for many-objective optimization [J]. IEEE Trans on Evolutionary Computation, 2016, 20(5): 773-791.

[28] Tian Ye, Cheng Ren, Zhang Xingyi, et al. PlatEMO: a MATLAB platform for evolutionary multi-objective optimization [J]. IEEE Comput. Intell. Mag., 2017, 12(4): 73-87.

[29] Zitzler E, Thiele L, Laumanns M, et al. Performance assessment of multi-objective optimizers: an analysis and review [J]. IEEE Trans on Evolutionary Computation, 2003, 7(2): 117-132.

[30] Robert S, Torrie J, Dickey D. Principles and procedures of statistics: a biometrical approach [M]. New York: McGraw-Hill, 1997.

Table 1 Mean and standard deviation values of IGD obtained by MaOEA-FM, NSGAIII, MOEA/DD, KnEA, and RVEA

Note: "+" indicates MaOEA-FM performs better than the compared algorithm; "=" indicates MaOEA-FM performs equally to the compared algorithm; "-" indicates MaOEA-FM performs worse than the compared algorithm.

Note: Figure translations are in progress. See original paper for figures.