Postprint: An Efficient Three-Objective Differential Evolution Method Using Pareto Non-Dominance

Authors: Xu Yulong, Pan Xu, Wang Zhongyi, Sheng Mengyuan, Wang Linjing

Date: 2018-05-18T00:00:00+00:00

Abstract

In multi-objective evolutionary algorithms, excessive time complexity represents a ubiquitous challenge, particularly when addressing problems with more than three objective functions, where solution rank assignment consumes considerable computational resources. Focusing specifically on three-objective problems and leveraging Pareto dominance relationships, this study investigates solution rank assignment and identifies inherent redundancies in classical ranking and assignment methods: such approaches necessitate sorting all solutions prior to rank assignment and subsequent generation selection, thereby incurring unnecessary computational overhead. To mitigate this redundancy, we propose an efficient three-objective evolutionary algorithm that integrates Pareto non-dominance relationships with differential evolution. The algorithm computes exclusively the highest-rank individuals in each iteration, concurrently performing rank assignment and offspring selection, and terminates computation upon offspring population generation, thus reducing the number of evaluated individuals and overall computational cost. Furthermore, we present comprehensive theoretical analysis and proofs for the proposed method. Subsequently, comparative experiments are conducted across a series of three-objective optimization problems, benchmarking our approach against the renowned NSGA-II ranking method and the recently prominent ENS method. Simulation results demonstrate that the proposed method surpasses classical approaches in both time complexity and convergence speed, while exhibiting marginally inferior performance relative to ENS. On standard test functions DTLZ1-DTLZ6, the proposed method achieves performance comparable to ENS and superior to NSGA-II, thereby validating its effectiveness and correctness.

Full Text

An Efficient Three-Objective Differential Evolution Method Using Pareto Non-Dominance Relationships

Xu Yulong[1],[2],[3], Pan Xu[1], Wang Zhongyi[1]†, Sheng Mengyuan[1], Wang Lingjing[1]

[1]Institute of Information Technology, Henan University of Chinese Medicine, Zhengzhou 450046, China

[2]Institute of Information Engineering, Zhengzhou University, Zhengzhou 450052, China

[3]Department of Computer Science & Engineering, Hong Kong University of Science and Technology, Hong Kong, China

Abstract

In multi-objective evolutionary algorithms, excessive time complexity is a pervasive issue, particularly when dealing with three or more objective functions, where solution ranking assignment consumes disproportionate computational resources. Addressing tri-objective problems, this paper investigates solution ranking assignment using Pareto dominance relationships and identifies redundant operations in classical ranking and allocation methods. Conventional approaches require sorting all solutions before rank assignment and selection of the next generation, incurring unnecessary computations. To eliminate this redundancy, we propose an efficient three-objective evolutionary algorithm that integrates Pareto non-dominance relationships with differential evolution. The algorithm processes only the highest-ranking individuals in each iteration, simultaneously performing rank assignment and offspring selection, terminating computation once the offspring population is generated. This strategy reduces the number of evaluated individuals and computational overhead. We provide theoretical analysis and proofs supporting the method. Comparative experiments against the renowned NSGA-II sorting method and the recently proposed ENS method on a series of tri-objective optimization problems demonstrate that our approach outperforms classical methods in time complexity and convergence speed, while remaining competitive with ENS. Performance on standard test functions DTLZ1-DTLZ6 shows results comparable to ENS and superior to NSGA-II, thereby validating the effectiveness and correctness of the proposed method.

Keywords: tri-objective optimization; Pareto; non-dominated sorting; convergence speed

0 Introduction

Multi-objective optimization problems (MOPs) generally involve more than two objective functions, with solutions forming a set known as the non-dominated set or Pareto-optimal set. Various methods exist for solving general optimiza-

tion problems, including linear programming, gradient descent, and evolutionary algorithms, though existing research indicates that multi-objective evolutionary algorithms (MOEAs) are particularly suitable for MOPs. MOEAs have garnered extensive international attention, yielding numerous outstanding algorithms, most notably NSGA-II and SPEA2. These algorithms evaluate solutions through spatial density and Pareto dominance, constructing Pareto candidate solution sets to obtain optimal solutions. Domestic scholars including Jiao Licheng, Chen Huowang, Cui Xunxue, Zheng Jianguo, and Cai Zixing have conducted in-depth research on the key technologies, theoretical analysis, and developmental history of multi-objective evolution, identifying research directions and challenges. Among these, improving computational efficiency represents a particularly significant problem.

To enhance algorithmic efficiency, researchers have pursued various approaches. NSGA-II offers substantial computational speed advantages over its predecessor and other contemporary algorithms, yet its global time complexity remains $O(mn^2)$. Tang et al. improved upon this using Pareto dominance relationships and the arena principle, randomly selecting a solution as the "champion" and comparing it with remaining solutions. If any solution dominates the champion, it replaces the original, achieving time and space complexities of $O(mn^2)$ and $O(n)$ respectively, demonstrating higher efficiency than NSGA-II in experiments. Clymont et al. proposed the deductive sort method, which records comparison results between all solutions to infer dominance relationships and avoid redundant comparisons, with time and space complexities below $O(mn^2)$ and $O(n)$. Recently, Zhang Xingyi et al. introduced the Efficient Non-dominated Sort (ENS) method, which determines a candidate solution' s rank when compared against already-ranked solutions, avoiding substantial redundant operations. ENS includes two variants: ENS-SS and ENS-BS. ENS-SS checks ranks sequentially from the first level until the candidate is assigned, while ENS-BS begins from the middle rank and halves the search range after each comparison. Experimental results show ENS significantly outperforms comparative algorithms in computational efficiency.

Differential evolution (DE) has emerged as an excellent algorithm in recent years, attracting considerable attention for its efficiency and simplicity. Abbass and Sarker first combined DE with non-dominance relationships for multi-objective optimization, achieving promising results. Robic and Filipic proposed DEMO, the most well-known multi-objective DE algorithm. Yang Ping et al. employed election rules to construct Pareto non-dominated solution sets with theoretical efficiency proofs, though their work focused only on comparing non-dominated sorting counts. Ren Changan et al. proposed an improved multi-objective evolutionary algorithm based on objective space partitioning using interval index values, but tested it only on bi-objective problems. Recent work has integrated clustering and data analysis methods to improve multi-objective DE, while other studies have proposed grid-index-guided multi-objective DE algorithms, all achieving favorable results. However, these multi-objective DE methods require rank assignment for all individuals during non-dominated sorting, creating

redundant operations. Our previous work addressed this for bi-objective DE, introducing a fast non-dominated sorting method that achieved good results, but remained limited to bi-objective problems.

Building upon this foundation, this paper presents an efficient three-objective DE method using Pareto non-dominated relationships. For tri-objective optimization problems, the method enables simultaneous non-dominated sorting and selection of individuals for the next generation. Comparative experiments against classical and state-of-the-art algorithms demonstrate through theoretical analysis and experimental results that the proposed method is both correct and highly efficient.

1 Related Theory

1.1 Differential Evolution Algorithm

Since its inception, differential evolution has become one of the most renowned algorithms in evolutionary computation due to its speed, ease of implementation, and minimal parameter requirements. DE is a heuristic algorithm comprising basic operations of mutation, crossover, selection, and update. The general procedure is summarized below.

First, an initial population is randomly generated, followed by mutation. Several mutation strategies exist, with the most common being:

a) **DE/rand/1**: $v_{i,g} = x_{r1,g} + F(x_{r2,g} - x_{r3,g})$

b) **DE/best/1**: $v_{i,g} = x_{best,g} + F(x_{r1,g} - x_{r2,g})$

c) **DE/current-to-best/1**: $v_{i,g} = x_{i,g} + F(x_{best,g} - x_{i,g}) + F(x_{r1,g} - x_{r2,g})$

where x represents the target vector, v the mutant vector, subscript i denotes the i-th individual ($i = 1, 2, 3, \ldots, NP$) with NP being the population size, and g indicates the g-th generation. In equation (3), $x_{best,g}$ is the best individual in the population, $x_{i,g}$ the current individual, $v_{i,g}$ the mutated individual, and $x_{r1,g}, x_{r2,g}, x_{r3,g}$ are three mutually distinct randomly selected individuals. The term $(x_{r1,g} - x_{r2,g})$ represents the differential vector, scaled by factor F to control search step size.

Next, crossover combines the mutant vector $v_{i,g}$ with the target vector $x_{i,g}$ to produce the trial vector $u_{i,g}$. DE employs either binomial or exponential crossover, with binomial being more commonly used:

$$u_{j,i,g} = \begin{cases} v_{j,i,g} & \text{if } rand_{i,j} \leq Cr \text{ or } j = j_{rand} \\ x_{j,i,g} & \text{otherwise} \end{cases}$$

where $j = 1, \ldots, D$ indexes the gene dimension, $Cr \in [0, 1]$ is the crossover probability controlling convergence speed, and j_{rand} is a random integer in $[1, D]$ ensuring at least one gene originates from the mutant vector.

Finally, selection employs a highly greedy strategy, comparing target and trial vectors individually, with the superior advancing to the next generation. For minimization problems, selection follows:

$$x_{i,g+1} = \begin{cases} u_{i,g} & \text{if } f(u_{i,g}) \leq f(x_{i,g}) \\ x_{i,g} & \text{otherwise} \end{cases}$$

This summarizes the DE procedure; detailed descriptions are available in the literature.

1.2 Multi-Objective Optimization

In multi-objective optimization, objective functions are mutually conflicting and non-commensurable, with optimal solutions typically forming a set rather than a single point. This fundamentally distinguishes multi-objective from single-objective optimization where a unique optimal solution exists. The dominance relationship between two solutions is defined as Pareto dominance. Since MOPs lack a global optimum, multiple Pareto-optimal solutions exist, constituting the problem' s optimal solution set. All solutions within this set are mutually non-dominated, allowing decision-makers to select one or more based on preferences. Multi-objective evolutionary algorithms aim to obtain numerous, uniformly distributed Pareto-optimal solutions approaching the theoretical Pareto front. Relevant concepts and mathematical formulations are detailed in the literature.

2 Efficient Three-Objective Differential Evolution Using Pareto Non-Dominance

2.1 Efficient Non-Dominated Sorting Method

NSGA-II is among the most famous multi-objective optimization algorithms, with many improved algorithms drawing upon its principles. However, NSGA-II exhibits computational redundancy: its non-dominated sorting process must assign ranks to all individuals before selection can occur. The recently proposed ENS algorithm also requires rank assignment for all individuals, though it avoids unnecessary dominance comparisons to improve speed. Both approaches determine ranks for the entire population before selecting individuals for the next generation in descending rank order from level 1 to level n.

In multi-objective evolution, assuming a population size of N, exactly N chromosomes must be selected for the next generation. Classical algorithms process all $2N$ chromosomes, requiring complete sorting, rank assignment, and comparison before selecting N individuals. Consider whether we could identify N suitable chromosomes by processing only a subset of the $2N$ individuals. Reducing the number of evaluated individuals would improve algorithmic efficiency. Therefore, based on inherent patterns in tri-objective non-dominated sorting, we propose an efficient tri-objective non-dominated sorting method that avoids

comparing all $2N$ individuals, thereby reducing time complexity. The detailed algorithm is presented below.

Algorithm 1: Efficient Tri-Objective Non-Dominated Sorting Method

Input: Combined parent and offspring population U with $2N$ individuals
Output: N chromosomes selected for the next generation

1. For rank = 1
2. For each individual $i = 1 : 2N$
3. Identify 4 special individuals belonging to rank level, store in set S_{rank}
4. These are individuals with minimum distances to the f_1 axis, f_2 axis, f_3 axis, and origin
5. End for
6. Assign rank to individuals in set S_{rank}
7. Compute set difference $U' = U \setminus S_{rank}$
8. For each individual $j = 1 : |U'|$
9. Compare individual j with all individuals in S_{rank}
10. Obtain all individuals of rank level, store in set S_{rank}
11. End for
12. End for individual i
13. Obtain complete set of individuals at rank level, store in S_{rank}
14. All individuals in S_{rank} enter offspring population
15. If total offspring individuals $> N$
16. Perform crowding distance calculation and select partial individuals
17. End if
18. If total offspring individuals $= N$
19. Terminate algorithm

20. End if

21. End for individual i

22. Compute set difference $Q = U \setminus S_{rank}$

23. Update $U = Q$

24. rank++, proceed to next rank level

25. End rank loop

Algorithm 1 assumes population size N and input set U of $2N$ chromosomes. The proposed sorting method selects next-generation individuals without exhaustive computation of all $2N$ individuals, reducing comparison counts and eliminating computational redundancy. Key features include: First, identify four special individuals in the current population belonging to the highest rank (assuming non-coincidence for the general case) and store them in S_{rank}. Then compare remaining individuals against those in S_{rank}. When an individual i is dominated by any member of S_{rank}, comparison terminates early, saving operations. When individual i is non-dominated with all members of S_{rank}, it is added to S_{rank}. Drawing from ENS methodology, comparisons begin with the last individual in S_{rank}, ensuring the set remains sorted by the first objective function value in ascending order, making later individuals more likely to dominate i. Repeating this process yields all highest-rank individuals (rank = 1) in S_{rank}. If admitting all S_{rank} individuals into the next generation exceeds N, crowding distance calculation selects a subset; otherwise, all individuals enter the next generation and rank increments to process the next level. If exactly N individuals are selected, the algorithm terminates without processing remaining individuals, improving efficiency. Typically, during mid-evolution, only the first few rank levels suffice to fill the next generation, eliminating computation for lower ranks and substantially saving time.

2.2 Theoretical Foundation of the Efficient Tri-Objective Non-Dominated Sorting Algorithm

The crucial aspect of the proposed algorithm is identifying four special individuals guaranteed to belong to the current population' s highest rank. For tri-objective optimization, at most four such special individuals exist (though they may coincide; we assume the general non-coincident case). Proving these individuals occupy the highest rank is fundamental.

Inference 1: If an individual $a(x, y, z)$ has the minimum x-coordinate value among all individuals in the current population, then a belongs to the highest rank.

Inference 2: If an individual $b(x, y, z)$ has the minimum y-coordinate value

among all individuals in the current population, then b belongs to the highest rank.

Inference 3: If an individual $c(x, y, z)$ has the minimum z-coordinate value among all individuals in the current population, then c belongs to the highest rank.

Inference 4: If an individual $d(x, y, z)$ has the minimum distance to the origin among all individuals in the current population, then d belongs to the highest rank.

Proof by contradiction for Inference 1: Given the current population, let x_{min} be the minimum x-coordinate value. For individual $a(x_{min}, a_y, a_z)$, no individual l exists with $l_x < x_{min}$. Assume there exists an individual l that dominates $a(x_{min}, a_y, a_z)$. By Pareto principles, this requires $l_x < x_{min}$, $l_y < a_y$, and $l_z < a_z$, which contradicts the known condition. Therefore, no such l exists, proving a belongs to the highest rank. Similar proofs apply to Inferences 2-4. Thus, the four special individuals identified by our method are guaranteed to be highest-rank individuals.

2.3 Algorithm Framework and Time Complexity Analysis

Based on the efficient tri-objective non-dominated sorting method, we propose a Fast Multi-objective Differential Evolution algorithm (FMODE) by integrating it with differential evolution. The overall framework is presented in Algorithm 2.

Algorithm 2: Fast Multi-Objective Differential Evolution

Input: Algorithm parameters, population size N, maximum function evaluations FES
Output: Final evolved solution set

1. Initialize population $X(0) : \{x_1(0), \dots, x_n(0)\}$
2. Load algorithm parameters: G, N, Cr, F
3. While evolution not terminated
4. Perform mutation using equation (1) to obtain M
5. Perform crossover using equation (4) to obtain Q
6. Combine parent and offspring populations: $U = X \cup Q$
7. Set $K = 0$
8. While $K < N$

9. Call Algorithm 1 for efficient non-dominated sorting
10. Set K individuals as selected for next generation
11. End while
12. If offspring population size equals N
13. Terminate algorithm
14. Else
15. Select appropriate individuals for next generation
16. End if
17. End while

FMODE processes only the highest-rank individuals during non-dominated sorting while monitoring whether sufficient next-generation individuals have been selected, terminating early when possible and substantially reducing computation time. Classical non-dominated sorting algorithms must assign ranks to all individuals before selection, incurring greater overhead. The original NSGA algorithm had $O(mn^3)$ time complexity, while NSGA-II improved this to $O(mn^2)$ but still required complete rank assignment before selection, containing redundant operations.

The renowned ENS algorithm comprises two main steps: (1) sorting the population by the first objective ($O(n \log n)$), and (2) assigning ranks to each solution. ENS-SS has worst-case $O(mn^2)$ complexity when all individuals share one rank, and best-case $O(mn)$ when individuals are distributed across approximately $\sqrt{n}$ ranks. ENS-BS has identical worst-case complexity but best-case $O(mn \log n)$ when all ranks are distinct.

For the proposed algorithm, the best case occurs when the first rank contains exactly N individuals, which are directly selected for the next generation, requiring $O(mn)$ time. Sorting by objective values costs $O(n \log n)$, and computing distances to the origin costs $O(n)$, yielding total complexity $O(mn) + 2O(n \log n) + O(n)$. The worst case matches NSGA-II, requiring rank assignment for all individuals at $O(mn^2)$, giving total complexity $O(mn^2) + 2O(n \log n) + O(n)$. While early generations process multiple ranks due to random population distribution, later generations typically require only the first few ranks, improving efficiency.

3 Simulation Experiments

3.1 Experimental Environment and Parameter Settings

To validate FMODE' s effectiveness, we employ international standard test problems DTLZ1-DTLZ6, whose Pareto fronts are representative. Function descriptions are shown in Table 1. FMODE parameters are set as: population size $N = 200$, mutation factor $F = 0.5$, crossover probability $Cr = 0.2$. Different maximum evolution generations are configured per function as shown in Table 1. The mutation strategy is "DE/Rand/1/bin". Comparison algorithms are the renowned NSGA-II and ENS, with parameters set according to their original literature. Experimental environment: Windows 10, Intel(R) CoreTM i5-4590 CPU @ 3.3GHz, MATLAB R2015b.

We adopt the Inverted Generational Distance (IGD) metric, which encompasses both diversity and convergence properties. IGD is calculated as:

$$IGD(P^*, P) = \frac{\sum_{v \in P^*} d(v, P)}{|P^*|}$$

where P^* represents the theoretical Pareto-optimal set, P the obtained solution set, and $d(v, P)$ the minimum Euclidean distance between v and P.

3.2 Runtime Comparison

To verify the sorting method' s efficiency, we compare its runtime and comparison counts against NSGA-II, ENS-SS, and ENS-BS non-dominated sorting methods. Results are shown in Tables 2 and 3, with time in seconds and comparison counts in iterations. Tests use populations with specific rank structures (approximately $\sqrt{n}$ ranks, where ENS-SS performs optimally) and randomly generated populations.

The results show that for small populations, all four algorithms exhibit similar sorting times. For larger populations, our algorithm' s sorting time is slightly longer than ENS but significantly shorter than NSGA-II. In summary, FMODE' s computational efficiency is marginally inferior to ENS but substantially superior to NSGA-II. Since ENS is a recently proposed prominent algorithm, these comparative results demonstrate FMODE' s competitiveness and computational efficiency.

3.3 Overall Performance Comparison

This section presents comprehensive performance comparisons between FMODE and classical algorithms. Results using the IGD metric are shown in Tables 4 and 5, with the best results highlighted in bold, representing means over five independent runs. Evolutionary curves and computational times for all algorithms across six test functions are shown in Figures 2-7.

The results indicate similar evolutionary curves across the four algorithms for all six functions, with ENS-BS and ENS-SS showing slight advantages due to minimized redundant computations. For DTLZ1 and DTLZ3, FMODE' s evolution time is slightly longer than ENS but far shorter than NSGA-II. On DTLZ4, FMODE outperforms NSGA-II in performance while using slightly more time. For DTLZ2 and DTLZ6, FMODE's performance matches comparison algorithms but with marginally higher time consumption. Notably, on DTLZ6, FMODE requires significantly more time because most individuals cluster in the first rank, preventing the algorithm from exploiting its advantage of simultaneous selection during sorting. Comparisons between obtained fronts and theoretical fronts are shown in Figure 8.

Overall, FMODE' s IGD performance is comparable to ENS and NSGA-II, with nearly identical performance characteristics. In terms of computational time, it is slightly slower than ENS but markedly faster than NSGA-II, thereby validating the algorithm' s effectiveness and correctness.

3.4 Parameter Settings for FMODE

This section investigates FMODE' s parameter sensitivity. With mutation strategy "DE/Rand/1/bin" , $F = 0.5$, $NP = 200$, and $FES = 1000$, we test various crossover probability Cr values. IGD results are shown in Table 7. The best performance occurs at $Cr = 0.2$ or 0.3, establishing $F = 0.5$ and $Cr = 0.2$ as empirical optimal values. Other mutation strategies with different parameter combinations perform worse than "DE/Rand/1/bin" , with detailed data omitted for brevity. The recommended parameter settings for the proposed algorithm are: "DE/Rand/1/bin" mutation, scaling factor $F = 0.5$, and crossover probability $Cr = 0.2$.

4 Conclusion

This paper introduces an efficient tri-objective non-dominated sorting method that reduces the number of individuals processed during sorting while enabling simultaneous selection of next-generation individuals, significantly improving computational efficiency over conventional rank assignment. Integrating this method with differential evolution yields the Fast Multi-objective Differential Evolution (FMODE) algorithm. Validation experiments against NSGA-II and ENS on standard test functions DTLZ1-DTLZ6 demonstrate that FMODE surpasses classical NSGA-II in both time complexity and performance, while remaining competitive with or slightly inferior to ENS. Future work will analyze FMODE' s limitations and consider integrating other evolutionary frameworks or adaptive parameter control to further enhance performance.

References

[1] Coello C A, Van Veldhuizen D A, Lamont G B. Evolutionary algorithms for solving multi-objective problems. Berlin: Springer, 2007: 1–140.

[2] Deb K, Pratap A, Agarwal S, et al. A fast and elitist multiobjective genetic algorithm NSGA-II. IEEE Trans on Evolutionary Computation, 2002, 6(2): 182-197.

[3] Zitzler E, Laumanns M, Thiele L. SPEA2: improving the strength Pareto evolutionary algorithm. Proc of Evolutionary Methods for Design, Optimization and Control with Applications to Industrial Problem, 2002: 95-100.

[4] Gong Maoguo, Wang Shanfeng, Liu Wenfeng, et al. Evolutionary computation in China: A Literature Survey. CAAI Trans on Intelligence Technology, 2016, 1(4): 334-354.

[5] Xie Tao, Chen Huowang, Kang Lishan. Multi-objective optimization evolutionary algorithms. Chinese Journal of Computers, 2003, 26(8): 997-1003.

[6] Cui Xunxue, Li Qin, Tao Qing. Genetic algorithm for Pareto optimum-based route selection. Journal of Systems Engineering and Electronics, 2007, 2(18): 360-368.

[7] Wang Yong, Cai Zixing, Zhou Yuren, et al. Constrained evolutionary algorithms. Journal of Software, 2009, 20(1): 11-29.

[8] Zheng Jianguo, Wang Xiang, Liu Ronghui. -DE algorithm for constrained optimization problems. Journal of Software, 2012, 23(9): 2374-2387.

[9] Tang Suqin, Cai Zixing, Zheng Jinhua. A fast method of constructing the non-dominated set: arena' s principle. Proc of the 4th International Conference on Natural Computation, 2008: 391-395.

[10] McClymont K, Keedwell E. Deductive sort and climbing sort: new methods for non-dominated sorting. Evolutionary Computation, 2012, 20(1): 1-19.

[11] Zhang Xingyi, Tian Ye, Cheng Ran, et al. An efficient approach to non-dominated sorting for evolutionary multi-objective optimization. IEEE Trans on Evolutionary Computation, 2015, 19(2): 201-213.

[12] Xu Yulong, Fang Jian' an, Zhu Wu, et al. Differential evolution using a superior-inferior crossover scheme. Computational Optimization and Applications, 2015, 61(4): 243-274.

[13] Das S, Mullick S S, Suganthan P N. Recent advances in differential evolution: an updated survey. Swarm and Evolutionary Computation, 2016, 4(27): 1-30.

[14] Abbass H, Sarker R. The Pareto differential evolution algorithm. International Journal on Artificial Intelligence Tools, 2002, 11(4): 531-552.

[15] Robic T, Filipic B. DEMO: differential evolution for multi-objective optimization. Proc of the 3nd Int Conf Evolutionary Multi-criterion Computation, 2005: 520-533.

[16] Yang Ping, Zheng Jinhua, Li Miqing, et al. A fast method for constructing multi-objective Pareto non-dominated sets: election rule. Computer Application Research, 2009, 26(2): 488-491.

[17] Ren Changan, Li Zhiyong, Chen Youwen. An improved multi-objective evolutionary algorithm based on objective space partitioning. Computer Application Research, 2010, 27(4): 1311-1314.

[18] Wu Jiaying, Li Ping, Zheng Jinhua. A multi-objective optimization algorithm based on multi-parent genetic mechanism. Computer Applications and Software, 2008, 25(2): 52-53, 79.

[19] Tan Yang, Tan Yuewu, Tang Zhaoyi. Multi-objective evolutionary algorithm based on Hamming distance evaluation. Computer Engineering, 2014, (2): 212-218.

[20] Bao Peiming, Zhu Qingbao. Normalized ranking non-dominated set construction method for multi-objective evolution. Acta Electronica Sinica, 2009, 37(9): 2010-2015.

[21] Huang Ying, Li Yang, Gao Ciwei. Multi-objective power network planning based on non-dominated sorting differential evolution algorithm. Power System Technology, 2011, 35(3): 85-89.

[22] Li Bin, Zhong Runtian, Xiao Jinchao, et al. A multi-objective optimization algorithm based on edge distribution estimation. Journal of Electronics & Information Technology, 2007, 29(11): 2683-2687.

[23] Wang Xianping, Tang Lixin. An adaptive multi-population differential evolution algorithm for continuous multi-objective optimization. Information sciences, 2016, 348: 124-141.

[24] Cheng Jixiang, Gary G. Yen, Zhang Gexiang. A grid-based adaptive multi-objective differential evolution algorithm. Information sciences, 2016, 367: 890-908.

[25] Li Wen, Li Hecheng. Multi-objective evolutionary algorithm based on spatial grid partitioning. Computer Engineering and Applications, 2014, (8): 53-56, 117.

[26] Xu Yulong, Fang Jian'an, Zhang Han, et al. Fast multi-objective differential evolution algorithm based on non-dominated sorting. Computer Applications, 2014, 34(9): 2547-2551.

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