Hindawi
Mathematical Problems in Engineering
Volume 2020, Article ID 3094941, 12 pages
https://doi.org/10.1155/2020/3094941

Research Article

Fast Nondominated Sorting Genetic Algorithm II with *Lévy* Distribution for Network Topology Optimization

Maoqing Zhang,[1] **Lei Wang**,[1] **Zhihua Cui**,[2] **Jiangshan Liu**,[3] **Dong Du**,[4] **and Weian Guo**[5]

[1]*School of Electronics and Information, Tongji University, 201804 Shanghai, China*
[2]*Complex System and Computational Intelligence Laboratory, Taiyuan University of Science and Technology, Taiyuan, Shanxi 030024, China*
[3]*School of Mechanical Engineering, Tongji University, 201804 Shanghai, China*
[4]*Shanghai Jiangcon Technology Company Limited, 200000 Shanghai, China*
[5]*Sino-German College of Applied Sciences, Tongji University, 201804 Shanghai, China*

Correspondence should be addressed to Dong Du; duzhengyu115@163.com

Received 11 October 2019; Revised 18 December 2019; Accepted 31 December 2019; Published 20 January 2020

Academic Editor: Samuel N. Jator

Fast nondominated sorting genetic algorithm II (NSGA-II) is a classical method for multiobjective optimization problems and has exhibited outstanding performance in many practical engineering problems. However, the tournament selection strategy used for the reproduction in NSGA-II may generate a large amount of repetitive individuals, resulting in the decrease of population diversity. To alleviate this issue, *Lévy* distribution, which is famous for excellent search ability in the cuckoo search algorithm, is incorporated into NSGA-II. To verify the proposed algorithm, this paper employs three different test sets, including ZDT, DTLZ, and MaF test suits. Experimental results demonstrate that the proposed algorithm is more promising compared with the state-of-the-art algorithms. Parameter sensitivity analysis further confirms the robustness of the proposed algorithm. In addition, a two-objective network topology optimization model is then used to further verify the proposed algorithm. The practical comparison results demonstrate that the proposed algorithm is more effective in dealing with practical engineering optimization problems.

1. Introduction

In the real world, there exist many engineering optimization problems, such as water resource optimization problem [1], malicious code detection [2,3], and big data optimization problem [4–6]. Over the past years, many efficient methods [7] have been proposed to tackle them. Multiobjective optimization problems [8] (MOPs) generally refer to problems having two or three objectives. In general, these objectives are contradictory with each other because the improvement of one objective may result in the deterioration of another objective. Over the past few decades, many excellent methods have been proposed for MOPs, such as NSGA-II [9], SPEA2 [10], and PSEA-II [11]. These methods can be roughly divided into the four categories in terms of core ideas for improving algorithm performance.

The first category is to introduce new strategies. This idea has been proved to be useful in single optimization problems, and many researchers have tried it in MOPs. For example, Zitler et al. [12] investigated various problem features and concluded that elitism is proved to be a vital factor in keeping the efficiency of the multiobjective evolutionary algorithm. Furthermore, Zitzler et al. [13] introduced an improved fitness assignment strategy to SPEA, resulting in the improvement of overall performance. Wang et al. [7] systemically investigated the control parameter of the firefly algorithm and proposed a firefly algorithm with adaptive control parameters.

Reducing the computational complexity is another aspect. The most classical exemplar is NSGA-II [9], which incorporates a fast nondominated sorting strategy to NSGA, effectively reducing the complexity of NSGA. MOEA/D [14] is also a famous method for its decomposition-based strategy, which decomposes a multiobjective problem into many single-objective optimization problems and then optimizes the subproblems simultaneously.

Optimizing search manner corresponds to the third category. Exploration and exploitation are two important factors, based on which many methods have been proposed. For example, Li et al. [15] proposed a general learning paradigm based on jumping genes to enhance the exploration ability of multiobjective evolutionary algorithms. In contrast to that, to enhance the overall exploitation of a memetic algorithm for multiobjective optimization, Chong [16] employed a multiobjective evolutionary gradient search as a form of local search.

Research on application of various algorithms is also a vital part. To estimate the demand of water resources, Wang et al. [1] specially designed three estimation models to evaluate the historical water use and local economic structure and then incorporated the three models into the firefly algorithm, achieving high prediction accuracy. Abd and Lu [17] proposed a many-objective optimization problem with seven objectives and then employed the knee-driven evolutionary algorithm to optimize the model. To optimize the network topology structure, Li and Chen [18] specially designed a modified NSGA-II, in which asexual crossover and double swap mutation operation are proposed.

The tournament selection strategy, which is used to select the individuals for reproduction in NSGA-II [9], may repeatedly select the same individual with better quality, resulting in a large amount of repetitive individuals and decreasing the population diversity. To alleviate this issue, this paper attempts to introduce the *Lévy* distribution to fast nondominated sorting genetic algorithm II and proposes the fast nondominated sorting genetic algorithm II with the *Lévy* distribution (LDNSGA-II). Note that the *Lévy* distribution is originally introduced to cuckoo search algorithm and has been proved to be effective in improving search ability. In addition, parameter sensitivity analysis is further conducted to demonstrate the robustness of the *Lévy* distribution. After that, this paper further applies LDNSGA-II to a practical network topology optimization problem. The contributions of this paper can be summarized as follows:

(1) Firstly, this paper analyzes the phenomenon of repetitive individuals caused by the tournament selection strategy. Then, to alleviate that problem, this paper proposes to employ the *Lévy* distribution, which has shown excellent performance in helping individuals to conduct effective local search and escape from the local optima.

(2) To verify the performance of the proposed method, this paper utilizes three common test sets including ZDT, DTLZ, and MaF test suits. Experimental results demonstrate that LDNSGA-II is more promising in dealing with common multiobjective optimization problems compared with some state-of-the-art algorithms.

(3) Parameter sensitivity analysis is conducted to test the impact of *delta* in the *Lévy* distribution. To have a direct understanding of parameter *delta*, different settings of *delta* are tested. According to the experimental results, the performance of LDNSGA-II is less sensitive to the change of *delta*.

(4) This paper further applies LDNSGA-II to a network topology optimization problem, which is a discrete multiobjective optimization problem. The results illustrate that the proposed method is effective in dealing with practical optimization problems.

This paper is organized as follows: Section 2 introduces the basic concepts of MOPs and related work. Section 3 presents the basic framework of NSGA-II and our motivation for this paper and proposes the nondominated sorting genetic algorithm with *Lévy* distribution (LDNSGA-II). Section 4 verifies LDNSGA-II using various test problems and applies the proposed method to a network topology optimization problem. The conclusion is drawn in Section 5.

2. Definition and Related Work

2.1. Definition. A typical multiobjective problem [8] can be defined as follows:

$$\begin{aligned} \min f(x) &= \min\left[f_1(x), f_2(x), \ldots, f_M(x)\right], \\ x &= \left(x_1, x_2, \ldots, x_D\right) \in R^D, \\ &\begin{cases} g_i(x) \geq 0, i = 1, 2, \ldots, K, \\ h_j(x) = 0, j = 1, 2, \ldots, P, \end{cases} \end{aligned} \tag{1}$$

where $f_M(x)$ is the M-th subobjective function and x is a vector of the solution, which should satisfy the above constraints. R^D is the decision variable space, $g_i(x) \geq 0$ is an inequality constraint, and $h_j(x) = 0$ is an equality constraint.

There are two kinds of relationships between any two solutions: dominated relationship and nondominated relationship. Let $u = (u_1, u_2, \ldots, u_M)$ and $v = (v_1, v_2, \ldots, v_M)$ be two vectors consisting of M objectives. u is said to dominate v if and only if $u_i \leq v_i$ for each corresponding component in both u and v, and there exists at least one index j which makes $u_j < v_j$. Otherwise, u and v have the nondominated relationship.

2.2. Related Work. The *Lévy* distribution is firstly incorporated into the cuckoo search (CS) by Yang and Deb [19] to deal with single-objective optimization problems. In CS, the *Lévy* distribution is adopted to satisfy the heavy-tailed probability distribution. As a result, CS boosts a more effective random search than the genetic algorithm and other swarm intelligent algorithms [20–22]. Besides, CS gets wide recognition for its strong global search ability. The section mainly focuses on the review of the *Lévy* distribution and its variants.

The *Lévy* distribution is a particular class of random walk, in which the step lengths during the walk are described as a heavy-tailed probability distribution. As Barthelemy et al. [23] pointed out that the *Lévy* distribution can be applicable to a diverse range of fields, describing animal foraging patterns and the distribution of human travel. The popularity of CS has confirmed the efficiency of the *Lévy* distribution. In 2009, Yang [24] formulated a new meta-heuristic algorithm by combining the *Lévy* distribution with the firefly algorithm, in which the *Lévy* distribution provides a random step for a random sign or direction. In 2010, Yang and Deb [25] further proposed an eagle strategy (ES) to combine the *Lévy* distribution with the firefly algorithm. In ES, the *Lévy* distribution replaces simple randomization, resulting in more search space. In 2013, Xie et al. [26] incorporated the *Lévy* distribution into the bat algorithm to ensure the diversity of the population against premature convergence and to make the algorithm effectively jump out of local minima. In 2018, Xiong et al. [27] adapted the continuous CS to a discrete multiobjective CS with *Lévy* distribution being selection probability. Different from the methods above, in 2019, Cui et al. [28,29] attempted to apply the *Lévy* distribution to many-objective optimization problems and further replaced the global best individual with the upper and lower boundaries of variables.

From the discussion above, it can be known that the *Lévy* distribution is commonly used in swarm-based intelligent optimizers, and it is considered to be effective in avoiding premature convergence. In addition, the fact can be seen that few papers apply *Lévy* distribution to NSGA-II. Therefore, combining the *Lévy* distribution with NSGA-II is the main contribution of this paper.

3. Proposed Method

3.1. Basic Framework of NSGA-II. NSGA-II [9] has two vital components, including the fast nondominated sorting strategy and the crowding distance. The fast nondominated sorting strategy is used to control evolutionary pressure, which can divide the population into multiple Pareto fronts, while the crowding distance is useful in improving the population diversity. The content below mainly introduces the core ideas of them.

Let P_i and Q_i correspond to the parent and offspring populations with N individuals. The first step is to select N better individuals from the combined population $R_i = P_i \cup Q_i$. To achieve that goal, the fast nondominated sorting strategy is used to generate various Pareto fronts. Generally speaking, the number of the first Pareto front is less than N. Then, the second and even the third Pareto fronts may be included until the i-th front, of which individuals will be selected partially because the number of the i-th front plus previous Pareto fronts will exceed the total number N. To distinguish the individuals in the i-th Pareto front, crowding distance is used. Individuals with better diversity will be selected as the candidate individuals to do crossover operator. The crossover and mutation operators are common operators, and they can be conducted as usual. Readers are encouraged to refer to the original paper [5] (Algorithm 1).

3.2. Our Motivation. In NSGA-II [9], the tournament selection strategy is employed for the selection of better individuals. The whole process can be described as shown in Figure 1. As illustrated in Figure 1, all individuals are firstly compared with each other. After that, better individuals are randomly selected as the parents for the reproduction. Note that, due to the characteristics of multiobjective optimization problems, there does not exist a global best individual but a set of trade-off solutions. Therefore, if individual I is a solution in the first Pareto front, then, it is likely to be selected in the next round, thus resulting the same individual as the parents and reducing the diversity of the next population.

To visually exhibit the analysis above, this paper conducts an experiment on ZDT1 and records the number of repetitive individuals with 100 individuals being population size and 100 iterations. Detailed settings are presented in later experiments. Figure 2 presents the statistics of repetitive individuals per generation, and Figure 3 exhibits the statistics of repetitive individuals at the 80$^{\text{th}}$ generation. From Figure 2, it can be seen that there are many repeatedly selected individuals from generation 7 to 60. As can be seen from Figure 3, the first individual is selected six times, and some other individuals are also picked up several times.

To avoid the phenomenon above, this paper intends to incorporate the *Lévy* distribution into the crossover operator and alleviates the phenomenon without destroying the guiding information. *Lévy* distribution is an excellent method, which can not only help efficiently search local potential individuals but also help escape from the local optima, resulting in improving the diversity of population. Detailed illustration of *Lévy* distribution can be found in the next section.

3.3. Proposed Method. For clarity, this section firstly introduces the *Lévy* distribution and then incorporates it into NSGA-II. *Lévy* distribution can be formally expressed as follows:

$$\text{Lévy}(\delta) \sim \mathrm{u} = \mathrm{t}^{-1-\delta}, \quad 0 < \delta < 2. \tag{2}$$

For simplicity, it can be represented as follows:

$$Lévy(\delta) \sim \phi \times \frac{\mathrm{u}}{|\mathrm{v}|^{1/\delta}}, \tag{3}$$

where u and v are the two parameters based on the Gaussian distribution and δ (delta) is originally set to 1.5 [14], and in later experiments, it will be systematically examined, and ϕ is defined as follows:

$$\phi = \left(\frac{\Gamma(1+\delta) \times \sin(\pi \times \delta/2)}{\Gamma(1+\delta/2) \times \delta \times 2^{(\delta-1)/2}} \right)^{1/\delta}. \tag{4}$$

Figure 4 plots five hundred points generated with *Lévy* distribution. From Figure 4, it can be seen that most points are centered towards (0, 0), while several points are far away from the center. Statistically, the points in the rectangle account for more than 96% of all the points. That is to say, the *Lévy* distribution is able to conduct sufficient local search

```
Input: population size, crossover probability, mutation probability, and maximum evaluation time
Output: the best solution set
Begin
    For each individual, initialize the positions
    Evaluate the positions
    Perform selection operator (using nondominated sorting and crowding distance)
    Perform crossover operator
    Perform mutation operator
    Combine parent and offspring populations
    While (stop criterion is met)
      Evaluate the positions
      Update the population (using nondominated sorting and crowding distance)
      Perform selection operator (using nondominated sorting and crowding distance)
      Perform crossover operator
      Perform mutation operator
      Combine parent and offspring population
End
Output the optimal solutions of the population
End
```

ALGORITHM 1: The pseudocode of NSGA-II.

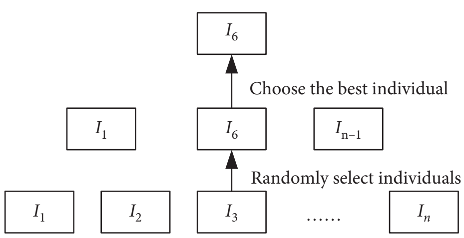

FIGURE 1: Illustration of the tournament selection strategy.

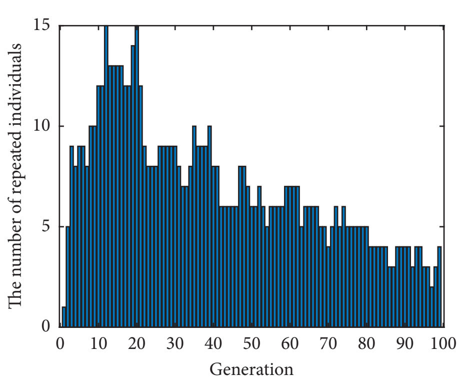

FIGURE 2: Statistics of the repetitive individuals per generation.

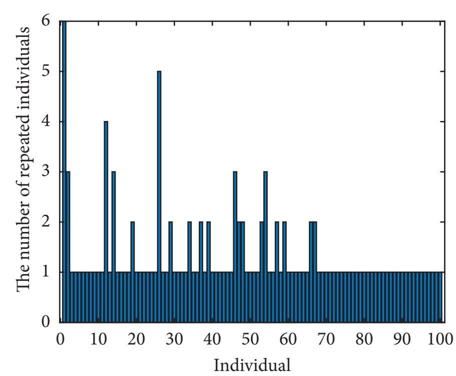

FIGURE 3: Statistics of the repetitive individuals at 80th generation.

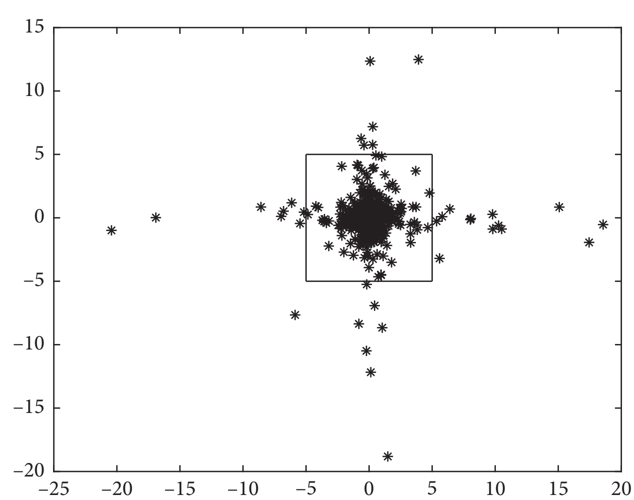

FIGURE 4: Illustration of the *Lévy* distribution.

to enhance the convergence, as well as the global search to avoid premature convergence and enhance diversity.

The crossover operator has been discussed and analyzed widely. Here, this paper does not intend to describe it in detail but presents how to combine it with *Lévy* distribution. Formally, the crossover operator can be expressed as follows:

$$Q_1 = \frac{(P_1 + P_2)}{2} + beta \times \frac{(P_1 - P_2)}{2}, \quad (5)$$

where P_1 and P_2 are the two selected parents and *beta* [5] is a parameter related to the crossover probability. Readers can refer to the original paper [5] for more details. Q_1 is the generated offspring individual. Then, the combination of the crossover operator and *Lévy* distribution can be formulated as follows:

$$Q_1 = \frac{(P_1 + P_2)}{2} + beta \times Lévy \times \frac{(P_1 - P_2)}{2}, \quad (6)$$

where *Lévy* is a parameter sampled with *Lévy* distribution. The *Lévy* distribution is used not only to alleviate the phenomenon of repetitive individuals without destroying the leading information but also to help enhance the overall performance of the crossover operator (Algorithm 2).

4. Experimental Results and Analysis

This section is divided into four subsections. Section 4.1 introduces the test problems, comparison algorithms [9–11], and the indicator used to evaluate the performance of the proposed methods. After that, comparisons of LDNSGA-II with other outstanding methods are conduced and corresponding analyses are also presented in Section 4.2. Following that, the sensitivity analysis of parameter *delta* in *Lévy* distribution is presented in Section 4.3. In the end, LDNSGA-II is applied to a network topology optimization problem to verify LDNSGA-II in practical optimization problem in Section 4.4.

4.1. Test Problems and Experimental Settings. Parameter settings of comparison algorithms are listed in Table 1. pc is the crossover probability, and pm indicates the mutation probability. ZDT [30], DTLZ [31], and MaF [32] test sets are the widely used test problems. In the experiments, ZDT1, ZDT2, ZDT3, DTLZ4, DTLZ5, DTLZ6, MaF11, and MaF12 are employed, and corresponding parameters are listed in Table 2, where D is the dimension of decision space and M is the dimension of objective space. Note that, in Table 2, due to the extreme complexity of MaF11 and MaF12, they are not presented completely. More specific details of them are ignored considering the limited paper length. Readers can refer to paper [32] for more detailed presentation. The indicator used in the experiments is IGD [33], which can measure the convergence and diversity of one algorithm. IGD is formulated as follows:

$$\mathrm{IGD}(P, P^*) = \frac{\sum_{v \in P^*} \mathrm{dist}(v, P)}{|P^*|}, \quad (7)$$

where P^* is a set of uniformly distributed points on the true Pareto front of objective space. P is the solution set obtained with one algorithm. $\mathrm{dist}(v, P)$ is the minimum Euclidean distances between v and point in P. For all test problems, the number (the closest integer to 500 among the possible values) of reference points is used for the IGD calculation.

For each test instance, each algorithm is run 20 independent times on the same machine with Intel Core i5-2400 3.10 GHz CPU, 6.00 GB memory, and Windows 7 operating system with Matlab 7.9. In this paper, the maximum number of evaluations is used as the stopping criterion. For all the test problems, the population is set to 100. For the ZDT test suit, the maximum evaluation number is set to 10000. For DTLZ4, DTLZ5, and DTLZ6, the maximum evaluation is 25000. The maximum evaluation is set to 40000 for MaF11 and MaF12. For detailed parameter settings of test problems, refer to paper [34].

4.2. Comparison with State-of-the-Art Algorithms. Table 3 lists the average IGD values of different algorithms over 20 independent runs. Note that the best results are highlighted in bold. The values in parentheses are the variances of the IGD values. From Table 3, it can be seen that the LDNSGA-II performs better on ZDT1, ZDT2, ZDT3, DTLZ4, DTLZ5, MaF11, and MaF12. On DTLZ6, it is evident that LDNSGA-II has superiority over NSGA-II but is still worse than SPEA2. From the comparison results, it can be concluded that the overall performance of LDNSGA-II has been significantly improved.

To have a direct comparison, Figure 5 further presents the Pareto fronts of LDNSGA-II and NSGA-II. From Figure 5, it can be seen that both LDNSGA-II and NSGA-II have similar Pareto fronts on ZDT1, ZDT2, DTLZ4, DTLZ5, DTLZ6, MaF11, and MaF12. Note that although MaF11 and ZDT3 have similar shapes, they are essentially different problems, which can be seen from the formulations presented in Table 2. Furthermore, on ZDT3, LDNSGA-II performs better than NSGA-II because the bottom right region has no solutions found by NSGA-II.

4.3. Parameter Sensitivity Analysis. To test the effect of parameter *delta* on *Lévy* distribution, Figure 6 presents the comparison curves with different *delta* settings on ZDT2 and DTZL5. The horizontal axis indicates the generation, while the vertical axis is the IGD value. From the comparison curves, it can be seen that different settings of *delta* result in similar curves. The performance of LDNSGA-II is less sensitive to the change of *delta*.

4.4. Application of LDNSGA-II to Network Typology Optimization. The fourth industrial revolution marks the coming of deep integration of information and communication technology. A stable and reliable network topology is greatly essential for users to achieve the data acquisition in Internet of Things. The switched Ethernet (as illustrated in Figure 7 [18], there are various network devices with different functions. Smart camera, Robots, PLC, and CNC are common devices in Internet of Things) has been widely applied to industrial control field due to its acceptable fault tolerance, high communication speed, and broadcast storm restrict.

$$A = \begin{bmatrix} 0 & \dots & a_{1i} & \dots & a_{1n} \\ \dots & \dots & \dots & \dots & \dots \\ a_{i1} & \dots & 0 & \dots & a_{in} \\ \dots & \dots & \dots & \dots & \dots \\ a_{n1} & \dots & a_{ni} & \dots & 0 \end{bmatrix}, \quad (8)$$

However, in practical operations, designer trends to add new devices to those devices with higher connectivity under the influence of the Matthew effect [18], reducing the stability and reliability of horizontal device interconnection and vertical networking integration. To this end, Li and Chen [18] proposed a multiobjective switched industrial Ethernet topology structure, in which each network device is

```
Input: population size, crossover probability, mutation probability, and maximum evaluation time
Output: the best solution set
Begin
    For each individual, initialize the positions
    Evaluate the positions
    Perform selection operator (using nondominated sorting and crowding distance)
    Perform crossover operator
    Perform mutation operator
    Combine parent and offspring populations
    While (stop criterion is met)
        Evaluate the positions
        Update the population (using nondominated sorting and crowding distance)
        Perform selection operator (using nondominated sorting and crowding distance)
        Perform crossover operator using equation (6)
        Perform mutation operator
        Combine parent and offspring population
End
Output the optimal solutions of the population
End
```

ALGORITHM 2: The pseudocode of LDNSGA-II.

TABLE 1: Parameter settings.

Algorithms	Parameter settings
NSGA-II	$pc = 1$, $pm = 1/\text{dimension}$
SPEA2	$pc = 1$, $pm = 1/\text{dimension}$
PEASII	$pc = 1$, $pm = 1/\text{dimension}$, div = *10*
LDNSGA-II	*delta* = 1.5, $pc = 1$, $pm = 1/\text{dimension}$

presented with a symbol. Supposing there are n leaf nodes (such as PC, industrial personal computer, and radio frequency identification sensor) which are added to the specified switch Ethernet, the communication between each device is expressed with the following matrix:where a_{ij} is the communication quantity from node i to node j, and it can be expressed with the unit kbps. Note that the communication quantity from node i to itself is considered to be zero.

To exhibit the connectivity of the specified switch Ethernet, the adjacency mapping matrix is further employed. Supposing there are n leaf devices and N switches as the backbone of the switch Ethernet, if node k is connected to node l, then the corresponding element can be expressed as $X_{kl} = 1$, otherwise $X_{kl} = 0$. Due to the control requirement of industrial Ethernet, the n leaf nodes in the switch Ethernet can be connected with each other. Therefore, the entire mapping adjacent matrix can be presented as follows:

$$X = \begin{bmatrix} 0 & \dots & X_{1N} & X_{1(N+1)} & \dots & X_{1(N+n)} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ X_{N1} & \dots & X_{(N-1),N} & X_{N(N+1)} & \dots & X_{N(N+n)} \\ X_{(N+1)1} & \dots & X_{(N+1)N} & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ X_{(N+n)1} & \dots & X_{(N+n)N} & 0 & \dots & 0 \end{bmatrix}, \quad (9)$$

Where $X_{(N-1),N}$ indicates that the N-th switch is connected to the (N–1)-th switch and $X_{N(N+1)}$ means that the N-th switch is connected to the first leaf node. Note that the leaf nods are not allowed to connected with each other, so $X_{i,j} = 0$, where $i > N$, $j > N$.

In industrial Ethernet topology, there are two special requirements, information transmission load between subnetworks and transmission load difference between switches, both of which should be minimized to achieve the load balance. The first goal is defined as follows:

$$f_1 = \sum_{m=2}^{D-1} \sum_{i=1,j=1}^{n} (m-1) \times B_{ij}^{m+1} \times a_{ij}, \quad (10)$$

where $D = \{\max(d) \mid \exists X_{kl}^{D} = 1,\ N+1 \le K < L \le N+n\}$, $1 \le m \le D-1$, and a_{ij} is the corresponding element of the communication matrix A. If $X_{(N+i)(N+j)}^{m+1} = 1$, then $B_{i,j}^{(m+1)} = 1$; otherwise, $B_{i,j}^{(m+1)} = 0$.

The second goal is to minimize the transmission load difference between switches, and it can be defined as follows:

$$f_2 = \sum_{s=2}^{n} |w(s) - w(s-1)|, \quad (11)$$

where $w(s)$ the transmission load of each switch, and it can be defined as follows:

$$w(s) = \sum_{i=1,j=1}^{n} \left(X_{s(N+i)} \times a_{ij}\right), \quad (12)$$

where $s = \{1, 2, \dots, N\}$.

Different from the loosely connected Internet, there are relatively strict requirements in the industrial Ethernet network:

(1) Strong constraint: isolated lead node is not allowed.
(2) Weak constraint: there are two constraints. The first is that each switch should be connected to one leaf node. The second is that maximum connection degree of each switch is limited. In this paper, the

Table 2: Illustrations of test problems.

Problems	Dimension	Variable bounds	Objective functions
ZDT1	30	[0, 1]	$f_1(x) = x_1$ $f_2(x) = g(x)[1 - \sqrt{x_1/g(x)}]$ $g(x) = 1 + 9(\sum_{i=2}^{n} x_i)/(n-1)$
ZDT2	30	[0, 1]	$f_1(x) = x_1$ $f_2(x) = g(x)[1 - (x_1/g(x))^2]$ $g(x) = 1 + 9(\sum_{i=2}^{n} x_i)/(n-1)$
ZDT3	30	[0, 1]	$f_1(x) = x_1$ $f_2(x) = g(x)[1 - \sqrt{x_1/g(x)} - x_1/g(x)\sin(10\pi x_1)]$ $g(x) = 1 + 9(\sum_{i=2}^{n} x_i)/(n-1)$
ZDT4	10	$x_1 \in [0, 1]$ $x_i \in [-5, 5]$ $i = 2, \ldots, n$	$f_1(x) = x_1$ $f_2(x) = g(x)[1 - \sqrt{x_1/g(x)}]$ $g(x) = 1 + 10(n-1) + \sum_{i=2}^{n}[x_i^2 - 10\cos(4\pi x_i)]$
DTLZ4	11	[0, 1]	$f_1(x) = (1 + g(x_M))\prod_{t=1}^{M-1}\cos(x_i^{\alpha}\pi/2)$ $f_{m=2:\,M-1}(x) = (1 + g(x_M))\sin(x_{M-m+1}\pi/2)\prod_{t=1}^{M-1}\cos(x_i^{\alpha}\pi/2)$ $f_M(x) = (1 + g(x_M))\sin(x_i^{\alpha}\pi/2)$ $s.t.\quad g(x_M) = \sum_{x_i \in x_M}(x_i^{\alpha} - 0.5)^2,\ i = 1, 2, \ldots, n,\ \alpha = 100$
DTLZ5	11	[0, 1]	$f_1(x) = (1 + g(x_M))\prod_{t=1}^{M-1}\cos(((1 + 2g(x_M)x_i)/(2 + (1 + g(x_M)))\pi)/2)$ $f_{m=2:\,M-1}(x) = (1 + g(x_M))\sin(x_{M-m+1}\pi/2)\prod_{t=1}^{M-1}\cos(((1 + 2g(x_M)x_i)/(2 + (1 + g(x_M)))\pi)/2)$ $f_M(x) = (1 + g(x_M))\sin(((1 + 2g(x_M)x_i)/(2 + (1 + g(x_M)))\pi)/2)$ $s.t.\quad g(x_M) = \sum_{x_i \in x_M}((1 + 2g(x_M)x_i)/(2 + (1 + g(x_M))) - 0.5)^2,\ i = 1, 2, \ldots, n$
DTLZ6	11	[0, 1]	$f_1(x) = (1 + g(x_M))\prod_{t=1}^{M-1}\cos(((1 + 2g(x_M)x_i)/(2 + (1 + g(x_M)))\pi)/2)$ $f_{m=2:\,M-1}(x) = (1 + g(x_M))\sin(x_{M-m+1}\pi/2)\prod_{t=1}^{M-1}\cos(((1 + 2g(x_M)x_i)/(2 + (1 + g(x_M)))\pi)/2)$ $f_M(x) = (1 + g(x_M))\sin(((1 + 2g(x_M)x_i)/(2 + (1 + g(x_M)))\pi)/2)$ $s.t.\quad g(x_M) = \sum_{x_i \in x_M} x^{0.1},\ i = 1, 2, \ldots, n$
MaF11	11	$x \in \prod_{i=1}^{D}[0, 2i]$	$f_1(x) = y_M + 2(1 - \cos((\pi/2)y_1))\ldots(1 - \cos((\pi/2)y_{M-2}))(1 - \cos((\pi/2)y_{M-1}))$ $f_2(x) = y_M + 4(1 - \cos((\pi/2)y_1))\ldots(1 - \cos((\pi/2)y_{M-2}))(1 - \sin((\pi/2)y_{M-1}))$ ... $f_{M-1}(x) = y_M + 2(M-1)(1 - \cos((\pi/2)y_1))(1 - \sin((\pi/2)y_2))$ $f_M(x) = y_M + 2M(1 - y_1\cos^2(5\pi y_1))$
MaF12	11	$x \in \prod_{i=1}^{D}[0, 2i]$	$f_1(x) = y_M + 2\sin((\pi/2)y_1)\ldots\sin((\pi/2)y_{M-2})\sin((\pi/2)y_{M-1})$ $f_2(x) = y_M + 4\sin((\pi/2)y_1)\ldots.\sin((\pi/2)y_{M-2})\sin((\pi/2)y_{M-1})$ ... $f_{M-1}(x) = y_M + 2(M-1)\sin((\pi/2)y_1)\cos((\pi/2)y_2)$ $f_M(x) = y_M + 2M\cos((\pi/2)y_1)$

Table 3: Comparison of different algorithms.

Problem	LDNSGA-II	NSGA-II	SPEA2	PESAII
ZDT1	**1.0477e − 1 (3.83e − 2)**	2.2805e − 1 (8.00e − 2)	1.3602e − 1 (7.55e − 2)	3.5496e − 1 (1.07e − 1)
ZDT2	**2.3905e − 1 (1.69e − 1)**	5.7633e − 1 (9.83e − 2)	5.5212e − 1 (1.89e − 1)	5.1306e − 1 (1.15e − 1)
ZDT3	1.0448e − 1 (1.86e − 2)	**1.0396e − 1 (4.38e − 2)**	1.3252e − 1 (6.10e − 2)	3.0291e − 1 (6.38e − 2)
DTLZ4	**5.2499e − 3 (2.65e − 1)**	1.5241e − 1 (3.11e − 1)	2.2555e − 1 (3.56e − 1)	1.5651e − 1 (3.09e − 1)
DTLZ5	**5.0841e − 3 (2.41e − 3)**	5.2258e − 3 (1.85e − 4)	5.1050e − 3 (4.35e − 5)	1.0215e − 2 (1.54e − 3)
DTLZ6	5.0579e − 3 (1.22e − 4)	5.5626e − 3 (2.70e − 4)	**4.0756e − 3 (2.33e − 5)**	1.4282e − 2 (2.76e − 3)
MaF11	**1.3151e − 2 (8.90e − 4)**	2.0795e − 2 (7.85e − 4)	2.3795e − 2 (8.80e − 4)	1.3309e − 2 (6.73e − 4)
MaF12	**2.7729e − 2 (2.90e − 3)**	2.9268e − 2 (2.61e − 3)	4.0268e − 2 (3.97e − 3)	2.9456e − 2 (2.48e − 3)

maximum connection degree is set to 16. Additional illustrations can be found in original paper [18].

To achieve a balanced network topology, LDNSGA-II is used to optimize the model above in combination with modified NSGA-II method in paper [18] (MNSGA-II for short in this experiment). To exhibit the optimization performance, this randomly generated network topology is also presented as a baseline. In this experiment, 4 switches and 40 leaf nodes are used, and the communication matrix between leaf nodes in [35] is applied. The maximum generation is set to 1000, and 100 individuals constitute the

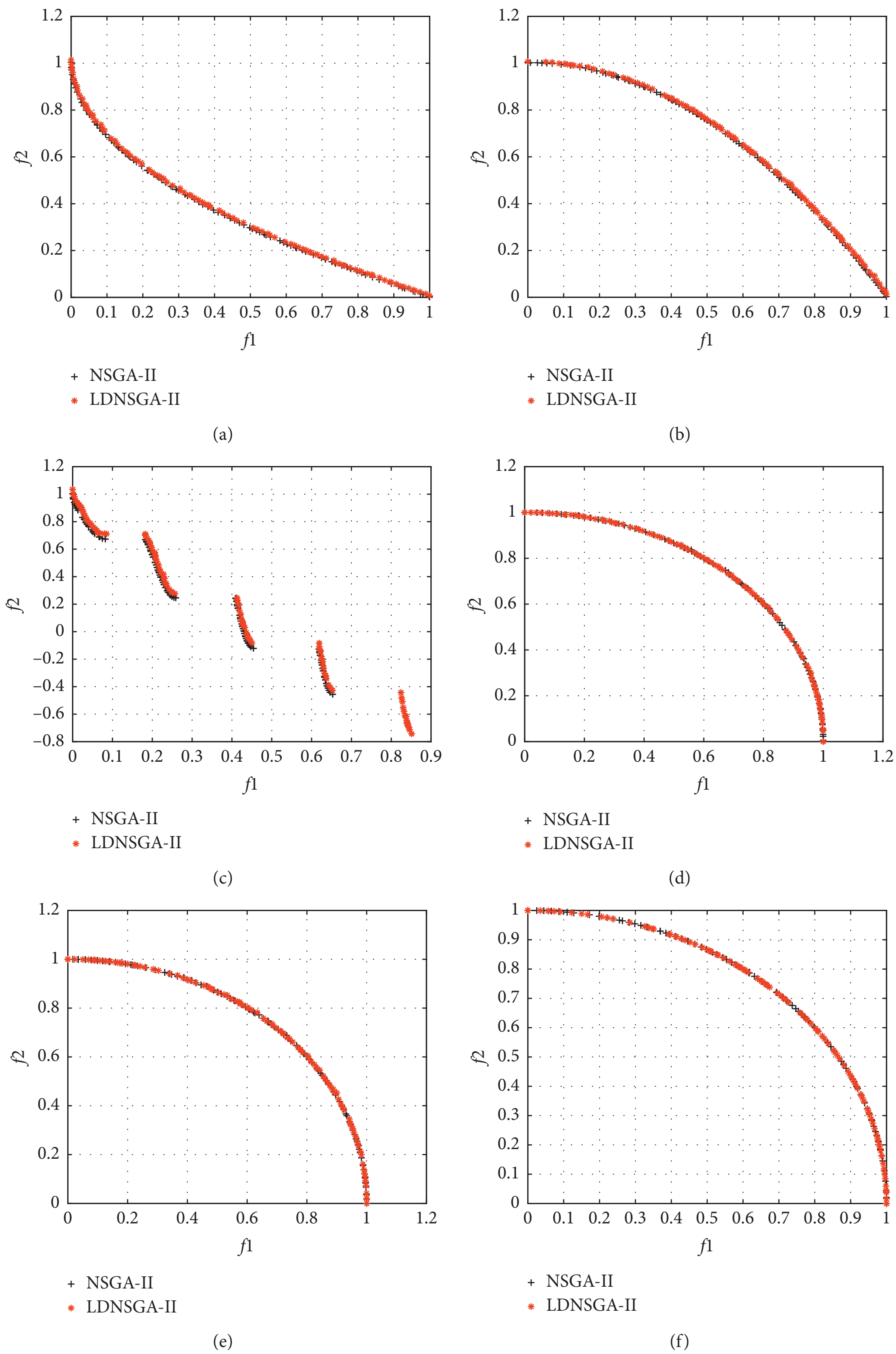

Figure 5: Continued.

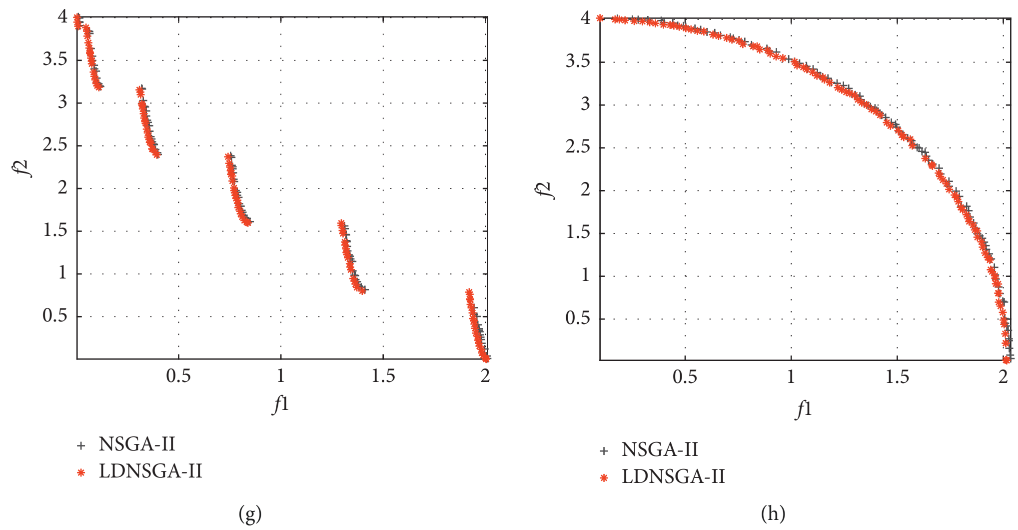

Figure 5: Comparison of Pareto fronts of LDNSGA-II and NSGA-II. (a) ZDT1. (b) ZDT2. (c) ZDT3. (d) DTLZ4. (e) DTLZ5. (f) DTLZ6. (g) MaF11. (h) MaF12.

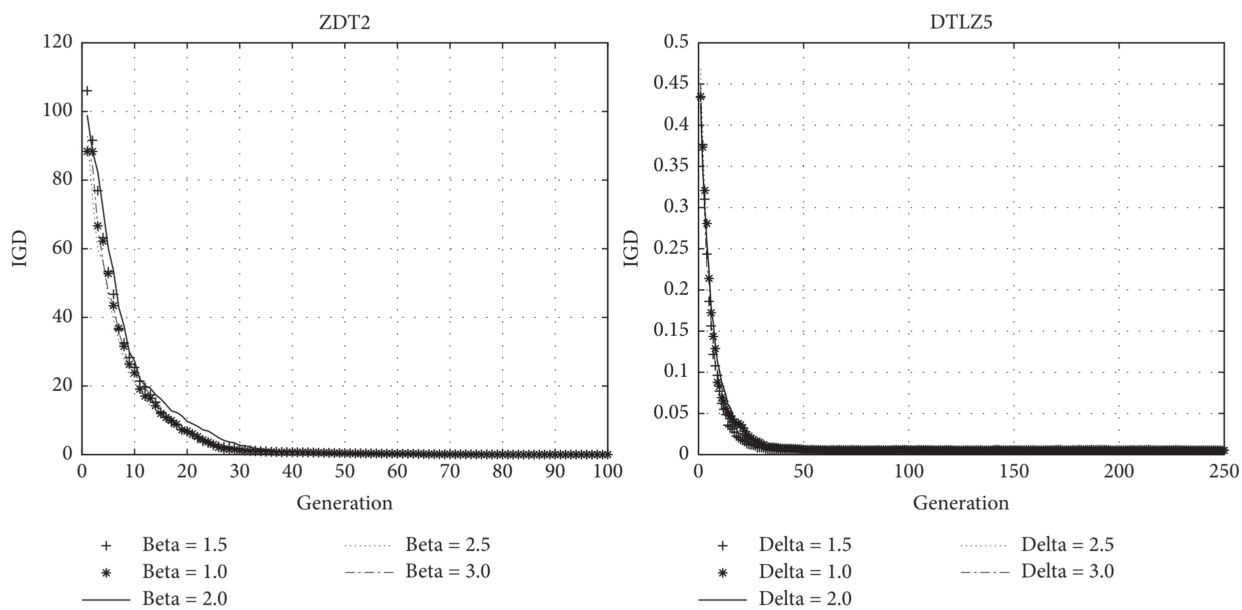

Figure 6: Comparison curves with different *delta* settings. (a) ZDT2. (b) DTLZ5.

population. The crossover probability is 0.9, while the mutation probability is set to $1/D$, where D is the number of variables. Due to the characteristics of discrete optimization problem, the *Lévy distribution* cannot be directly applied to the model above. Instead, this paper converts the *Lévy distribution* into a probability model. As illustrated in Section 3.2, the *Lévy distribution* can help individual escape from local optima. In this experiment, to achieve the same purpose, each node is randomly replaced with other leaf node with probability 0.1 under strict requirements mentioned above. Note that the probability 0.1 is set according to paper [18], where each node can be randomly replaced with probability 0.1 to increase the diversity of network topology. Figure 8 presents the obtained results. Note that the symbol "◊" indicates the solutions which are generated randomly and they can be regarded as the common case without optimization.

From Figure 8(a), it can be seen that the network topology without optimization may increase the information transmission load between switches. Meanwhile, the transmission load difference between switches is also obvious, which is harmful for the entire network. On the contrary, for both MNSGA-II and LDNSGA-II, the optimized network topology can greatly reduce the transmission load and achieve a balanced transmission load. In addition, compared to MNSGA-II proposed in original paper [18], LDNSGA-II is more efficient in optimizing the network topology according to the obtained Pareto fronts. From Figure 8(b), it can be seen that the Pareto front obtained by LDNSGA-II is more outstanding than other methods. Also, the fact can be

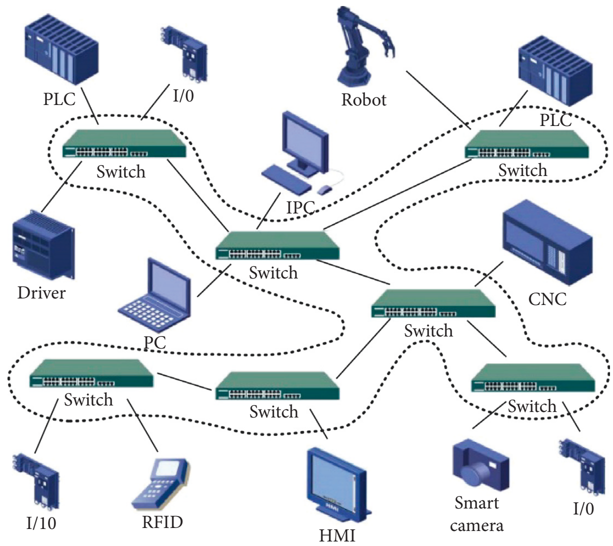

Figure 7: Illustration of network topology of switched industrial Ethernet.

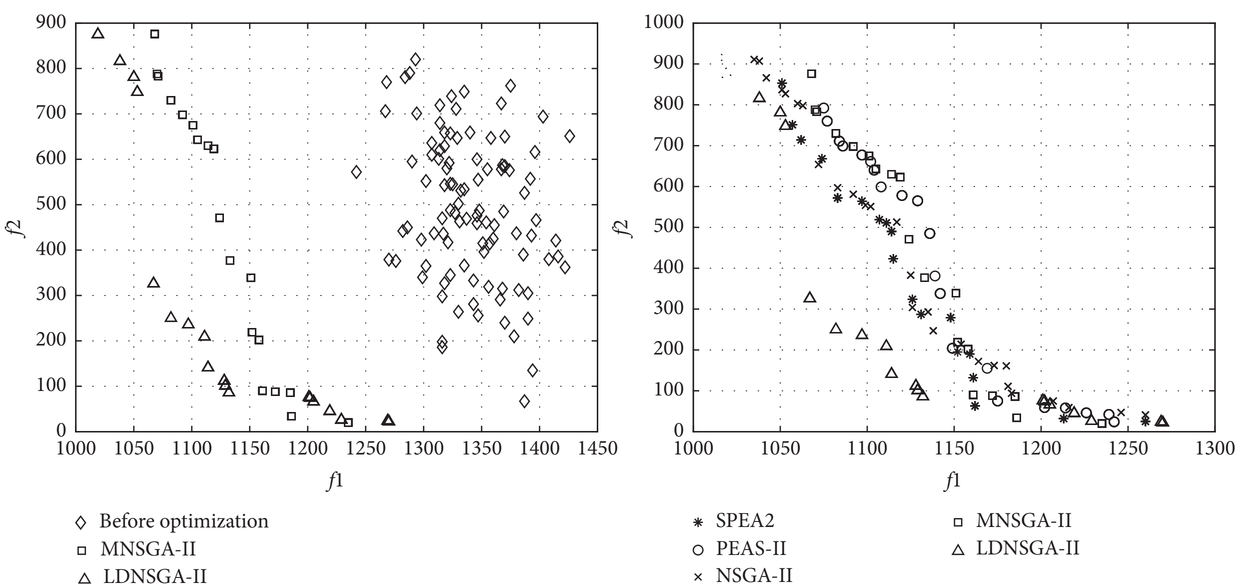

Figure 8: Comparison of solutions obtained with different methods.

observed that SPEA2 and NSGA-II are slightly better than PEASII. For users, if the time delay is a primary goal, then the corresponding network topology of the upper left points should be considered, while if good extendibility is preferred, then the lower right points can be selected.

5. Conclusion and Further Work

Lévy distribution is widely researched in cuckoo search because of its outstanding search ability. To incorporate *Lévy* distribution into NSGA-II, this paper proposes a fast nondominated sorting genetic algorithm II with *Lévy* distribution (LDNSGA-II). To verify the proposed algorithm, this paper employs three test suits and IGD indicator in our experiments. In addition, this paper systematically investigates the effect of parameter *delta*. The experimental results demonstrate that LDNSGA-II is comparable with the state-of-the-art algorithms and the parameter *delta* has less influence on LDNSGA-II. Further application of LDNSGA-II to the network topology optimization verifies its efficiency and effectiveness. Further work will focus on the improvement of NSGA-II and its applications.

Data Availability

The data sets used in this paper are standard test data sets which are all available online. The experimental data in Section 4.4 are obtained from the authors of paper [18].

Conflicts of Interest

There are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (nos. 71771176 and 61503287), Natural Science Foundation of Shanghai (no. 19ZR1479000), and Science and Technology Winter Olympic Project (no. 2018YFF0300505).

References

[1] H. Wang, W. Wang, Z. Cui, X. Zhou, J. Zhao, and Y. Li, "A new dynamic firefly algorithm for demand estimation of water resources," *Information Sciences*, vol. 438, pp. 95–106, 2018.

[2] Z. Cui, L. Du, P. Wang, X. Cai, and W. Zhang, "Malicious code detection based on CNNs and multi-objective algorithm," *Journal of Parallel and Distributed Computing*, vol. 129, pp. 50–58, 2019.

[3] X. Cai, Y. Niu, S. Geng et al., "An under-sampled software defect prediction method based on hybrid multi-objective cuckoo search," *Concurrency and Computation: Practice and Experience*, vol. 10, 2019.

[4] H. Wang, W. Wang, L. Cui et al., "A hybrid multi-objective firefly algorithm for big data optimization," *Applied Soft Computing*, vol. 69, pp. 806–815, 2018.

[5] X. Cai, S. Geng, D. Wu, L. Wang, and Q. Wu, "A unified heuristic bat algorithm to optimize the LEACH protocol," *Concurrency and Computation: Practice and Experience*, vol. 10, 2019.

[6] X. Cai, P. Wang, L. Du, Z. Cui, W. Zhang, and J. Chen, "Multi-objective 3-dimensional DV-hop localization algorithm with NSGA-II," *IEEE Sensors Journal*, vol. 19, no. 21, pp. 10003–10015, 2019.

[7] H. Wang, X. Zhou, H. Sun et al., "Firefly algorithm with adaptive control parameters," *Soft Computing*, vol. 21, no. 17, pp. 5091–5102, 2017.

[8] M. Zhang, H. Wang, Z. Cui, and J. Chen, "Hybrid multi-objective cuckoo search with dynamical local search," *Memetic Computing*, vol. 10, no. 2, pp. 199–208, 2018.

[9] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.

[10] E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2: Improving the strength Pareto evolutionary algorithm," in *Proceedings of the Fifth Conference on Evolutionary Methods for Design, Optimization and Control with Applications to Industrial Problems*, pp. 95–100, Athens, Greece, September 2001.

[11] D. W. Corne, N. R. Jerram, J. D. Knowles, and M. J. Oates, "PESA-II: Region-based selection in evolutionary multi-objective optimization," in *Proceedings of the 3rd Annual Conference on Genetic and Evolutionary Computation*, pp. 283–290, San Francisco, CA, USA, July 2001.

[12] E. Zitler, K. Deb, and L. Thiele, "Comparison of multiobjective evolutionary algorithms: empirical results," *Evolutionary Computation*, vol. 8, no. 2, pp. 173–195, 2000.

[13] E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2: Improving the strength Pareto evolutionary algorithm for multiobjective optimization," *International Center for Numerical Methods in Engineering (CIMNE)*, vol. 10, pp. 95–100, 2001.

[14] Q. Zhang and H. Li, "MOEA/D: a multiobjective evolutionary algorithm based on decomposition," *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 6, pp. 712–731, 2007.

[15] K. Li, S. Kwong, R. Wang, K.-S. Tang, and K.-F. Man, "Learning paradigm based on jumping genes: a general framework for enhancing exploration in evolutionary multiobjective optimization," *Information Sciences*, vol. 226, no. 3, pp. 1–22, 2013.

[16] J. K. Chong, "A novel multi-objective memetic algorithm based on opposition-based self-adaptive differential evolution," *Memetic Computing*, vol. 8, no. 2, pp. 147–165, 2016.

[17] E. M. Abd and S. Lu, "Many-objectives multilevel thresholding image segmentation using Knee Evolutionary Algorithm," *Expert System with Applications*, vol. 10, 2019.

[18] J. Li and M. Chen, "Multiobjective topology optimization based on mapping matrix and NSGA-II for switched industrial internet of things," *IEEE Internet of Things Journal*, vol. 3, no. 6, pp. 1235–1245, 2017.

[19] X. Yang and S. Deb, "Cuckoo search via Lévy distributions," in *Proceedings of World Congress on Nature and Biologically Inspired Computing*, pp. 210–214, Kitakyushu, Japan, December 2010.

[20] Y. Da and G. Xiurun, "An improved PSO-based ANN with simulated annealing technique," *Neurocomputing*, vol. 63, pp. 527–533, 2005.

[21] Z. Cui, F. Li, and W. Zhang, "Bat algorithm with principal component analysis," *International Journal of Machine Learning and Cybernetics*, vol. 10, no. 3, pp. 603–622, 2019.

[22] Z. Cui, F. Xue, X. Cai, Y. Cao, G.-G. Wang, and J. Chen, "Detection of malicious code variants based on deep learning," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 7, pp. 3187–3196, 2018.

[23] P. Barthelemy, J. Bertolotti, and D. S. Wiersma, "A Lévy flight for light," *Nature*, vol. 453, no. 7194, pp. 495–498, 2008.

[24] X. Yang, "Firefly algorithm, *Lévy distribution*s and global optimization," in *Research and Development in Intelligent Systems XXVI*, M. Bramer, R. Ellis, and M. Petridis, Eds., Springer, London, UK, 2010.

[25] X.-S. Yang and S. Deb, "Eagle strategy using Lévy walk and firefly algorithms for stochastic optimization," *Nature Inspired Cooperative Strategies for Optimization (NICSO 2010)*, vol. 284, pp. 101–111, 2010.

[26] J. Xie, Y. Zhou, and H. Chen, "A novel bat algorithm based on differential operator and *Lévy distribution*s trajectory," *Computational Intelligence and Neuroscience*, vol. 10, 2013.

[27] W. Xiong, B. Guo, Y. Shen, and W. Zhang, "A discrete multi-objective optimization method for hardware/software partitioning problem based on cuckoo search and elite strategy," *Neuro Quantology*, vol. 16, no. 5, pp. 749–756, 2018.

[28] Z. Cui, M. Zhang, H. Wang et al., "A hybrid many-objective cuckoo search algorithm," *Soft Computing*, vol. 10, 2019.

[29] Z. Cui, Y. Chang, J. Zhang, X. Cai, and W. Zhang, "Improved NSGA-III with selection-and-elimination operator," *Swarm and Evolutionary Computation*, vol. 49, pp. 23–33, 2019.

[30] E. Zitzler, K. Deb, and L. Thiele, "Comparison of multi-objective evolutionary algorithms: empirical results," *Evolutionary Computation*, vol. 8, no. 2, pp. 173–195, 2000.

[31] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, "Scalable multiobjective optimization test problems," *Proceedings of the IEEE Congress on Evolutionary Computation*, vol. 1, pp. 825–830, 2002.

[32] R. Cheng, M. Li, Y. Tian et al., "A benchmark test suite for evolutionary many-objective optimization," *Complex & Intelligent Systems*, vol. 3, no. 1, pp. 67–81, 2017.

[33] C. A. C. Coello and N. C. Cortés, "Solving multiobjective optimization problems using an artificial immune system," *Genetic Programming and Evolvable Machines*, vol. 6, no. 2, pp. 163–190, 2005.

[34] X. Zhang, Y. Tian, and Y. Jin, "A knee point-driven evolutionary algorithm for many-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 19, no. 6, pp. 761–776, 2015.

[35] Q. Zhang and W. Zhang, "Network partition for switched industrial Ethernet using genetic algorithm," *Engineering Applications of Artificial Intelligence*, vol. 20, no. 1, pp. 79–88, 2007.