Automatika

Journal for Control, Measurement, Electronics, Computing and Communications

ISSN: 0005-1144 (Print) 1848-3380 (Online) Journal homepage: https://www.tandfonline.com/loi/taut20

New intelligent optimization framework

As Xie

To cite this article: As Xie (2018) New intelligent optimization framework, Automatika, 59:2, 231-253, DOI: 10.1080/00051144.2018.1465680

To link to this article: https://doi.org/10.1080/00051144.2018.1465680

Published online: 28 Sep 2018.

AUTOMATIKA
2018, VOL. 59, NO. 2, 231–253
https://doi.org/10.1080/00051144.2018.1465680

REGULAR PAPER

OPEN ACCESS   Check for updates

New intelligent optimization framework

As Xie

China Institute for SMEs, Zhejiang University of Technology, Hangzhou, People's Republic of China

ABSTRACT

Generally, the "intelligence" of the intelligent optimization algorithms is mainly dependent on the probability and operational rules. Thus there are always some probability equations or mathematical formulations that need to be updated. This paper proposes an algorithm model that needs no probability tuning. The algorithm designed according to the guiding principles and specific methods of benchmarking proposed in this paper is able to achieve the synergistic coexistence and automatic balance of exploration and exploitation, thus the population diversity will be kept during the running. The algorithm model proposed here is between engineering technology and cognitive philosophy, it is not just a specific algorithm, but a kind of general methodology and/or a mode of thinking. The successful application of some realistic issues, like distributed power generation optimization configuration, verified its applicability.

ARTICLE HISTORY
Received 5 July 2017
Accepted 12 April 2018

KEYWORDS
Intelligent optimization; optimization algorithm; meta-heuristic method; encoding scheme; benchmarking philosophy

Summary/highlights

Compare with other intelligent optimization algorithms, this method has the following advantages:

1. The "intelligence" of this method does not depend on the probability rules of the operators, but their organizing tactics;
2. There are no probability equations or mathematical formulations that need to be updated;
3. Synergistic coexistence and automatic balance of the exploration and the exploitation are achieved;
4. The population diversity has been kept during the running;
5. Most useless and ineffective repetitious operations are avoided, thus the needed consumption of storage space and running time are lessened largely.

1. Introduction

Many issues in science, engineering and management can be transformed into optimization problems. For example, the problem of predicting the stability of the protein in Bioinformatics can be converted into the optimization problem of searching the lowest potential energy value of protein structure. The classification of tumour microarray gene expression data in Molecular Biology can be transformed into the optimization problem of solving the optimal tumour feature gene extraction and classification model. The problem of structural determination of complex networks in big data can be converted into the optimization problem of searching the optimal network structure. The real-time scheduling problem of modern intelligent industrial mobile robots can be turned into the optimization problem of the mobile path of robots. For these optimization problems, some of which can be solved by using traditional operations research methods, but the vast majority is impossible. For example, the field of photoelectric conversion. In the design phase of solar cells, to find the key factors of the photoelectric conversion efficiency of solar cells with different materials, it needs to analyse and extract the parameters of the equivalent circuit model. At this point, we can use the traditional optimization algorithms (TOAs), although the actual effect is not as good as the intelligent method. In the practical application of solar cells, the photovoltaic array is a real-time source of electricity in a daily environment. Its output characteristic will vary with ambient temperature, sunshine intensity and other factors, and it is a complex nonlinear form. Therefore, the maximum power points cannot be obtained simply by using an equivalent model. Also, it is hard to express in mathematical models. That is to say, we cannot use the TOAs to get the maximum power points. Therefore, to improve the energy efficiency of photovoltaic power generation system and obtain the maximum power output, it is very important to keep tracking of the maximum power points. At this point, the traditional operations research methods are helpless, and it can only be helped by using IOAs.

As the above, many of the issues in science, engineering and management can be transformed into optimization problems. The various methods applied to these problems are themselves, various models.

CONTACT As Xie shermanxas@163.com

Designing different methods is in itself designing different models. The theme of this paper is to model the benchmarking philosophy as a general method framework for solving different optimization problems. So this paper put forward a general model for intelligent optimization/search based on benchmarking philosophy, and whose "intelligence" is mainly dependent on the organizing tactics rather than the probability rules of its operators. According to the guiding principles and the specific methods of benchmarking, it is easy to design efficient optimization/search algorithms for various complex problems in science, engineering and management. This paper mainly includes four parts. The first part briefly introduced the literature about original IOAs and made a brief comment. The second part elaborated the content of this general model, including the theoretical basis and the algorithm model. The third part made some technical analyses of this algorithm model by simulation experiments. The fourth part proved the applicability of this algorithm model through the real problem of distributed power generation optimization configuration.

2. Literature about original IOAs

2.1. Rough classification of the original IOAs

Since genetic algorithm was put forward, many scholars have done much pioneering work. In recent years, there are tens of thousands of literature, and the original IOAs have more than a hundred by far. Some of which obtained a high degree of recognition. According to the principle of optimization and its overall characteristic, they can be broadly divided into four categories.

The first type is the evolutionary computation. There are several kinds of evolutionary algorithms and various improved versions. Although the specific search techniques are different, the same optimization idea is to imitate the evolutionary process of biology and generate candidate solutions by selecting, crossover and mutation. *The second type* is the swarm intelligence, which is mainly inspired by the behaviour of social insects or animals. There are a large number of living species in the biosphere, so this kind of algorithms develops very quickly. Entering the twenty-first century, almost one or two new algorithms emerged every year. Swarm intelligence is a type of distributed algorithm, and the so-called swarm intelligence refers to that the non-intelligent individuals expressed the characteristics of intelligent behaviour through the cooperation between each other. In the swarm intelligence system, the unit individual has simple intelligence and behaviour, but this group of individuals can cooperate to solve the complex problems. So the group has a kind of whole intelligence. In swarm intelligence algorithms, there is no control centre, and the interaction between individuals is also extremely simple. The most well-known of swarm intelligence is particle swarm optimization and ant colony optimization. *The third type* is the phenomenon algorithm, which is mainly inspired by the internal rules behind the various natural phenomena. The mechanism behind phenomenon algorithm is different from the first two types. Each random collection of the search agents is realized by moving or communication of the search space according to the physical rules. The search behaviour is performed by analogical gravitation, weight, light, projection, inertial force, electromagnetic force, etc. Because of the complexity of physical phenomena, the mechanisms and the rules of which are also varied, it can be predicted that this kind of IOAs will usher in a wave of a high tide of prosperity with the rapid development of artificial intelligence, data mining, knowledge discovery, mobile networks and other fields. *The final type* cannot be put into the three previous categories. For example, the simulated annealing algorithm, although derived from the solid annealing concept, has only a single candidate solution for each iteration. The microcanonical annealing algorithm, although imitating the annealing mechanism, the state transfer in the process of annealing is in a deterministic way instead of using the metropolis criterion. Taboo search, variable neighborhood algorithm and predatory search, these three algorithms should not be categorized as the optimization algorithm. Which, in essence, should be considered as three unique search strategies.

2.2. Brief comment on the original IOAs

The above division of four types is rough and not strict. In fact, there are some algorithms, whose own designs ideas have various features. For example, memetic algorithm, which included the concept of evolutionary computing, but also reflected the characteristic of swarm intelligence. Classifying the original IOAs into four categories, there is no special purpose. Just because it would be confusing to put so many original algorithms together. Moreover, the above classification is purely based on the intrinsic characteristic of the algorithm itself, but not according to the characteristics of the problems to be solved. Thus, there is no classification from the perspective of single goal and multi-objective. They were just classified by intuition, so the classification is rough. In the same way as TOAs based on strict mathematical logic, in essence, the IOAs are also the kind of search algorithms based on the hill-climbing model. The TOAs are the type of deterministic hill-climbing. Because the direction and step size are determined by the derivatives of the objective function. This is the advantage of TOAs but also their limitation. Because the prerequisite for their application is that the problem can be built into an accurate mathematical model, and the objective function must be continuous and differentiable. However, the IOAs almost have no such restrictions. Although the search behaviour of the single individual is non-deterministic, the search behaviour at the population level has a strong tendency towards the "peak(s)". Usually, The IOAs are inspired by the laws of nature or behaviours of biological groups. The design idea is that the random sample points in the problem domain are regarded as moving individuals, and the individuals have some properties as fitness, speed and position. Under the guidance of some probability rules, these individuals can get the possible optimal solution by continuous updating of speed and position. This search process has a certain degree of "intelligence". Comparing with the TOAs, although these methods have been applied satisfactorily in practice, there is no rigorous and complete mathematical basis for their application. Therefore, the IOAs need further mathematical analysis of the aspects of distribution density, optimization probability, convergence, complexity and accuracy of the algorithm. The method proposed in this paper also has this kind of deficiency. Due to the complexity of the constraints and the variable ranges, the solution space structure is often irregular. Coupled with the influence of other factors, the feasible solution and infeasible solution are often cross-distribution. Some design on operators would lead to too much repetitive and ineffective search. Also, the use of elitism strategy might result in a decrease in the population diversity and ultimately lead to a decline in search performance. To sum up, there are several significant shortcomings. First, the global search and local optimization are not easy to balance. Second, it was difficult to maintain population diversity in the search process. Third, there are too many useless and ineffective operations.

Regarding the designing of IOAs, the existing work can be divided into three levels. The primary level is to improve the existing algorithms. For example, according to the characteristics of the problem domain, to extract certain specific rule, and design certain new operator, and so forth. So many original IOAs have some improved versions. The intermediate level is to put forward new search ideas. For example, like genetic algorithm (GA), the original IOAs shown in Table 1 all have their own unique search patterns. The high level is to break through the traditional idea of intelligent computing. Using a variety of encoding schemes in IOAs is essentially equivalent to mapping the problem to be solved from the current space-time to another. Using the population and individuals to search the optimal solution is essentially equivalent to using the enumeration method to get the answer. But the use of probability rules makes the emergence of enumeration answers have a certain tendency, that is, the so-called intelligence of search behaviour. Therefore, it can be said that the GA firstly created the idea of using the coded individuals to get the optimal solution via "intelligent enumeration". Comparing with the TOAs based on strict mathematical logic, this is a kind of brand-new intelligent computation thought. At present, almost all the existing IOAs are all failed to escape from this mode of thinking.

3. Algorithm model based on benchmarking philosophy

3.1. The fundamental philosophy of algorithm model

Benchmarking,[1] in short, it is to find the gap with the best case and quickly narrow that gap via studying and emulating and eventually go beyond the rival. Therefore, finding the best solution based on benchmarking philosophy is starting from the current existing solution, and through recurring setting benchmark, studying and emulating, resetting benchmark, gradually from the worse option to the bad one, from the sub-optimal option to the optimal one, and ultimately getting the best solution. This is a process of iteration step by step, and this optimization process can be performed in parallel with the help of the modern computation and communication technology. It can be seen that from the point of view of optimization/search, there are a series of optimization/search rules implied in the benchmarking philosophy, which can provide references for the designing of new IOAs.

3.1.1. The principle of setting the benchmark

Which one was set as the benchmark, it depends on the rule of setting the benchmark. As a consequence, determine the suitable rule of setting benchmark according to the characteristics of current problems to be solved is the first step. In reality, most of the complex problems in the field of science, engineering and management, ultimately may be transformed into optimization problems. That is, how to construct the model, including conceptual model, mathematical model, or another type, and how to solve this model. In the modelling process, firstly, it is necessary to establish the causal relationship between the decision variables and the evaluation objectives. Secondly, it is necessary to define the constraint conditions. These two parts constitute the model, that is, it is also the evaluation criteria for the alternatives. Take building the mathematical model as an example, the evaluation function and constraints constitute the evaluation criteria for decision-making programmes. Implementing benchmarking, the first step is to determine the rules of setting the benchmark. That is, setting the proper benchmark according to evaluation criteria for alternative options.

3.1.2. The principle of studying and emulating

After selecting the benchmark, the next step is how to study, emulate and improve the object. In the real world, all kinds of problems, have the characteristics

Table 1. Brief classification of original IOAs with high recognition degree[a].

Category	Name (author, year)
Evolutionary computation	Evolutionary Programming (Fogel, 1966), Evolutionary Strategy (Ingo, 1973), Genetic Algorithm (Holland, 1975), Co-evolving Algorithm (Hillis, 1990), Genetic Programming (Koza, 1992), Cultural Algorithm (Reynolds, 1994), Estimation of Distribution Algorithm (Mühlenbein & Paass, 1996), Differential Evolution (Storn & Price, 1997)
Swarm intelligence	Memetic Algorithm (Pablo, 1989), Ant Colony Optimization (Dorigo et al., 1991), Continuous Particle Swarm Optimization (Eberhart & Kennedy, 1995; Shi & Eberhart, 1998), Binary Particle Swarm Optimization (Kennedy & Eberhart, 1997), Artificial Immune System (Castro & José, 1999), Clonal Selection Algorithm (Castro & Zuben, 2000), Shuffled Frog Leaping Algorithm (Eusuff & Lansey, 2001), Self-Organizing Migrating Algorithm (Zelinka & Lampinen, 2000), Bacteria Foraging Optimization Algorithm (Passino, 2002), Artificial Fish-swarm Algorithm (LI, 2002), Queen-bee Evolution (Jung, 2003), Bee Dance Algorithm (Gordon et al., 2003), Bees Swarm Optimization Heuristic Algorithm (Lučić & Teodorović, 2003), Bee Algorithm (Nakrani & Tovey, 2004), Bee-Hive Algorithm (Wedde et al., 2004), Honey Bee Swarm Optimization Algorithm (Karaboga, 2005), Glowworm Algorithm (Krishnanand & Ghose, 2005), Virtual Bee Algorithms (X.S. Yang, 2005), Monkey Search (Mucherino & Seref, 2007), Cat Search (Chu & Tsai, 2007), Roach Infestation Optimization (Havens, 2008), Biogeography-based Optimization (Simon, 2008), Firefly Algorithm (X.S. Yang, 2009), Cuckoo Search (X.S. Yang & Deb, 2009), Bat-inspired Algorithm (X.S. Yang, 2010), Fruit Fly Optimization Algorithm (Pan, 2011), Teaching–learning-based Optimization (Rao et al., 2012), Krill herd Algorithm (Gandomi & Alavi, 2012), Grey Wolf Optimizer (Mirjalili et al., 2014), Chicken Swarm Optimization (X. B. Meng et al., 2014), Bird Swarm Algorithm (X. B. Meng et al., 2015)
Phenomena algorithm	Photosynthetic Algorithm (Murase, 2000), Harmony Search (Zong et al., 2001), Gravitation Search Optimization Algorithm (Webster et al., 2003), Big bang–big crunch based Optimization (Erol & Eksin, 2006), Central Force Optimization (Formato, 2007), Gravitational Search Algorithm (Rashedi et al, 2009), Charged System Search (Kaveh & Talatahari, 2010), Galaxy-based Search Algorithm (Shah-Hosseini & Hamed, 2011), Curved Space Optimization (Moghaddam et al., 2012), Ray Optimization (Kaveh & Khayatazad, 2012), Black Hole Optimization Algorithm (Hatamlou, 2013)
Other type	Simulated Annealing Algorithm (Kirkpatrick et al., 1983), Micro-canonical Annealing Algorithm (Creutz, 1983), Taboo Search (Glover, 1986), Variable Neighborhood Algorithm (Mladenovic, 1995), Predatory Search (Linhares, 1998), Slime Mold Algorithm (Tero et al., 2010)

[a] The filter rule is that if an algorithm had been cited more than 100 times by others based on Google Scholar, it can be said that it has obtained a high degree of recognition. Here only lists the original author and date of these algorithms, whose improved versions in later are not listed here. The original innovation is strongly emphasized. Of course, due to the limited horizon, there may be some other excellent original algorithms that are not listed in the table. Readers can easily retrieve the relevant literature according to the above information. Limited to paper length, they will not be included in the references of this paper.

of their own domain and professionalism. Therefore, in the process of studying and emulating, benchmarking must be carried out based on their own domain features and their own evaluation criteria of the alternative schemes. As mentioned above, the main idea of IOAs is to use the encoded individuals to get the optimal solution via "intelligent enumeration". Therefore, when an encoding scheme is adopted, then it will have its own appropriate principle for studying and emulating. Through studying and emulating, constantly exceeding, constantly updating the alternative programmes to make the result continue to be improved and optimized.

3.1.3. The principle of resetting benchmark

After setting the benchmark, followed by studying and emulating, there may be two results. The first is that the desired effect has not been achieved, and another is that the desired effect has been achieved, or even exceeded. In either case, a new benchmark is needed to be set. Many of the decisions in the real world, for a variety of reasons, some did not achieve the positive effects, and some had achieved the desired results. Regardless of the effect of decision-making, the common goal of management decisions in all sectors is ultimate to solve the problem that how to realize sustainable development. Similarly, for the IOA with excellent performance, after the end of an iteration, regardless of the current results, the search results of next iteration are expected to be better, at least not worse than the last. To achieve this goal, it is necessary to re-establish a new benchmark according to the actual situation of search space.

3.1.4. The principle of teamwork

Once the role model was set up, it became the object of learning and emulating. Especially in the modern enterprise, when benchmarking is put into practice, at least a team, such as a business sector in the company, is the basic unit. This implies achieving common progress through group cooperation in the process of benchmarking. For the IOAs, each individual is relatively simple, and its search capability is relatively limited, however, the group's intelligence is relatively advanced, and the search performance of the group is improved significantly. Therefore, the IOAs, the vast majority of which is based on the group, rather than individual, to carry out searching and optimization.

3.1.5. The principle of diversity

When benchmarking is put into implementation, even if the specific industries or areas is explicit, even the specific business or sector is definite, it still requires a certain degree of diversity in alternative schemes and benchmarking objects. Because there are some knowable and controllable factors, especially some unknowable and uncontrollable factors, such as the diversity of decision variables and other reasons. For many problems in the real world, from the standpoint of systematics, maintaining a certain degree of redundancy is conducive to the overall system stability and overall interests. And therefore, it is good that alternative schemes and benchmarking objects have a certain diversity. For the IOAs, if one can maintain the population diversity during the process while running, this just showed its superior performance. On account of their own optimization idea and search strategy, nowadays, most of the IOAs often encountered premature phenomena and faced the challenge of keeping the diversity of population during the search process.

3.1.6. The principle of efficiency

As with other management methods, benchmarking should also pay attention to efficiency issues, that is, input–output ratio. Here assumed that the output effect was fixed, only talk about the cost of inputs. In the real enterprise management, it needs to invest a lot of people, financial, material and other resources in the various stages of benchmarking, and naturally, it is important to try every means to reduce the cost of inputs. For the IOAs, searching for the good candidate solutions, it also needs to invest in costs. For example, the consumption of storage space, runtime and other computing resources, of course, the less the better.

3.2. The basic model of benchmarking algorithm (BA)

3.2.1. The basic procedure of algorithm model

The general framework of BA is like this: within the whole ecosystem (i.e. the solution space), some niche populations (i.e. initial solutions) are generated (randomly, or by some rules). These niche populations are equivalent to all kinds of enterprises in the global market. Those individuals within the niche populations are equal to the business sectors or employees. First, set the benchmark according to the purpose of optimization based on the evaluation function. The best individual in each niche population (i.e. the local best individual, i.e. the local optimal solution), and the best individual in the whole ecosystem (i.e. the global best individual, i.e. the global optimal solution) is determined respectively. This is equivalent to establishing the local benchmark and the global benchmark. The previous result is recorded. Second, every individual within each niche population emulates their own benchmark in accordance with the guiding principles of benchmarking, which will be elaborated later. Each strives for close to, even beyond its object through learning and emulating. After this round of benchmarking, reevaluate the situation and reset the local benchmark and the global benchmark, and update the record. And then, start a new round of benchmarking, so back and forth, until iteration to meet the stop condition. At this time, the global best individual is just the global optimal solution after decoding. And the local best individuals are just the local optimal solutions after decoding. It can be seen that this idea of learning includes both competition learning mechanism and learning competition mechanism. The basic procedure of algorithm model is shown as follow.

- Step1: Input initial condition. Initialize all kinds of parameters, like the populations, individuals, stop condition, etc.
- Step2: Setting benchmark according to the guiding principles of benchmarking.
- Step3: Studying and emulating according to the specific methods of benchmarking.
- Step4: Output the result.

The basic algorithm flowchart is shown in Figure 1.

The intelligent optimization method based on benchmarking philosophy, whose general idea is studying and emulating a good example. However, its core idea is the guiding principles of benchmarking and the specific methods of studying and emulating. The former is the guiding ideology of designing specific algorithms for specific problems.

3.2.2. The guiding principles of benchmarking

3.2.2.1. Benchmark record is updated in real time. While the algorithm is running, both the best individual in each niche population and the best individual in the whole ecosystem generated in each round iteration are recorded with a record-keeping. That is, the local benchmark and the global benchmark generated in each round iteration are registered and constantly updated in real time.

3.2.2.2. Studying and emulating chosen on demand. Benchmarking includes three behaviours, which consists of emulating the global benchmark, emulating the local benchmark, and self-learning. Based on different practical problems and different selection criteria, we can design different optimization algorithms according to whether to implement certain emulating behaviour. For example, the three emulating behaviours do not need to be executed in sequence. After emulating the global benchmark, if the individual did not get improvement, then it will emulate the local benchmark. Otherwise, it will quit from the current round of iteration.

3.2.2.3. Learning object is flexibly changed. After each iteration, the best individual in each niche population, i.e. the local best individual, will be set as the benchmark in its own niche population, i.e. the local benchmark. Moreover, the best individual in the whole ecosystem will be established as the global benchmark. Theoretically, the local benchmarks in each round of iteration will continue to change. The global benchmark is selected from those local benchmarks, so it is bound to change as well. Unless the global optimal

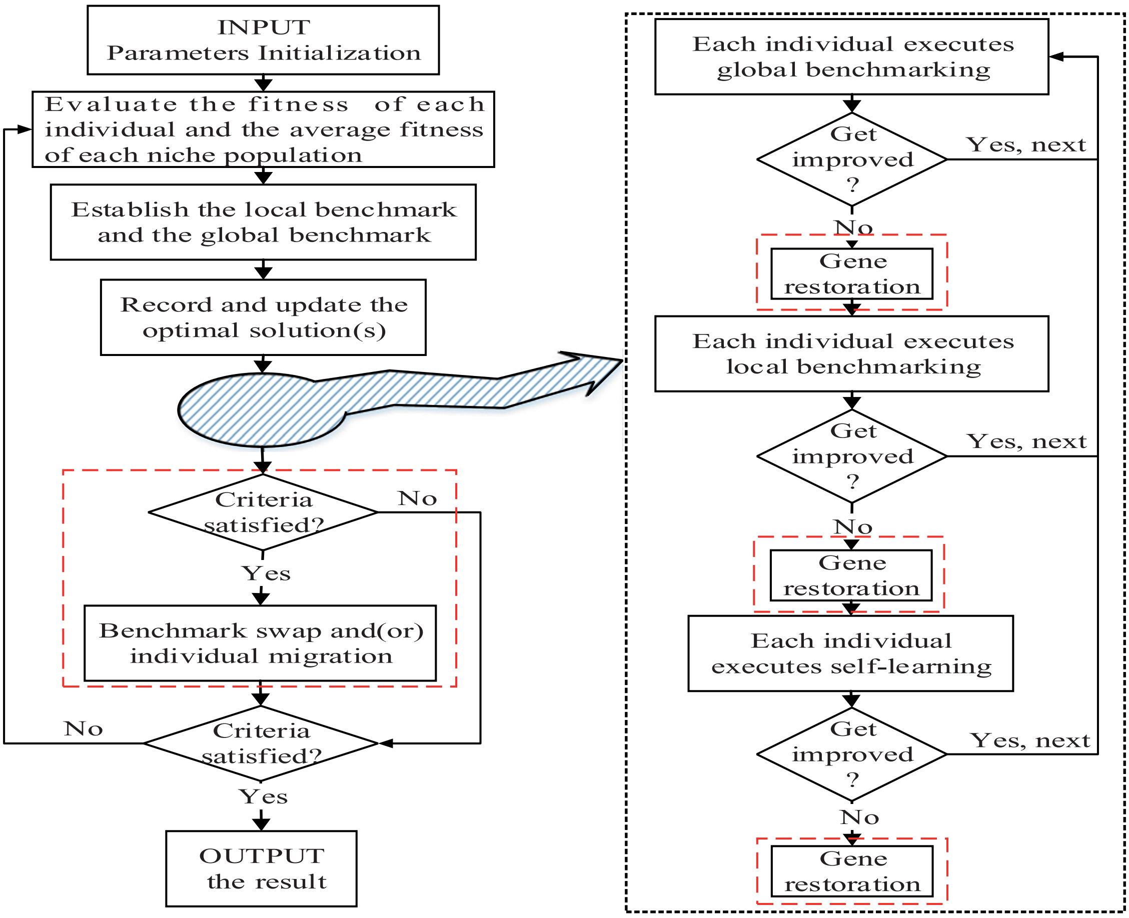
INPUT
Parameters Initialization
Evaluate the fitness of each individual and the average fitness of each niche population
Establish the local benchmark and the global benchmark
Record and update the optimal solution(s)
Criteria satisfied?
No
Yes
Benchmark swap and(or) individual migration
No
Criteria satisfied?
Yes
OUTPUT the result
Each individual executes global benchmarking
Get improved ?
Yes, next
No
Gene restoration
Each individual executes local benchmarking
Get improved ?
Yes, next
No
Gene restoration
Each individual executes self-learning
Get improved ?
Yes, next
No
Gene restoration

Figure 1. The basic flowchart of BA.

solution has been found, then the global benchmark will no longer change. Alternatively, a local extreme value is figured out by some niche population, then the local benchmark within it will remain unchanged. The most particular case is that in the implementation of distributed search, each niche population has found out a local extreme point after certain round of iteration. Moreover, after that, all the local benchmarks, of course, including the global benchmark, will remain unchanged. However, the global extreme point within the solution space has still not been searched out this moment. This is the most peculiar situation. In theory, the occurrence probability of this extreme situation is negligible, but to be absolutely safe, the additional rule can be adopted to avoid this extreme situation. For example, we can stipulate that if the global benchmark has not been improved after several rounds of iteration, it is mandatory to exchange the local benchmark from each niche population in a random way. This is like the rotation system for officials in enterprise management. In this way, theoretically, the learning objects could continue to change in the process of benchmarking. According to the different characteristics and rules of the problem domain, we can design some different search algorithms, and ensure that the global extreme point(s) in the ecosystem can be found out.

3.2.2.4. Efficient use of Matthew effect. IOAs are based on probabilistic search, and whose intelligence is mainly reflected in their own probability rules. The benchmarking philosophy based optimization idea attaches great importance to studying and emulating, but also contains the competition model of strong will be stronger. So the Matthew effect will be rationally used in the implementation of benchmark learning. Each individual in the ecosystem has different learning desire and learning intensity in the process of studying and emulating. The desire and the intensity, i.e. the probability, is determined by the degree of differences and (or) similarity between itself and the benchmark object. The benchmarking philosophy based optimization idea contains such a rule that if the degree of difference is smaller, and (or) the degree of similarity is larger, the greater the learning desire, followed by the greater learning intensity. This is the Matthew effect on the individual level. For example, when the algorithm used binary coding, the smaller the Hamming distance between the individual and the benchmark, the greater the desire for the individual to perform the benchmark learning, followed by the greater the strength of the benchmark learning. At this point, the individual will further reduce the Hamming distance with the benchmark, and get closer to, even beyond the object by studying and emulating. As a result, the opportunity that the individual got improved is greatly increased.

3.2.2.5. Rational use of cluster effect. In the process of studying and emulating, if certain niche population found out a better global solution, that is, it set a new global benchmark, other niche populations within the ecosystem would assign individuals to the niche population to assist in intensive searches. However, if this niche population did not find out a better global solution during the benchmarking process, all its individuals would gradually be assigned to other niche populations, and this niche population would eventually die out. This is the cluster effect on the population level. Similarly, if a certain individual found out a better global solution, that is, it itself became the new global benchmark by studying and emulating, it would attract many individuals from other niche populations to its neighbourhood, thus forming a new niche population, to assist in the intensive search. This is the cluster effect on the individual level.

3.2.3. The specific methods of benchmarking

After giving the guiding principles of benchmarking, it is now necessary to elaborate on the specific ways of benchmarking. It is not just the core idea of algorithm theory based on benchmarking philosophy, but also this is the guiding principles for designing BAs. According to the core idea of benchmarking, the general principle is to emulate those better than oneself. Specifically, the implementation of benchmarking is to reduce the differences between one of itself and the benchmark object, and (or) to increase the similarity between one of itself and the benchmark object. In general, this initiative search idea based on studying and emulating is applicable, as to how to implement this specific behaviour of studying and emulating, it depends on the encoding scheme the algorithm itself used. For different subjects, different professions and various fields, due to the particularity of the problem to be optimized, and also the particularity of the scene in which the problem is to be optimized, it needs to use different modelling methods, construct different mathematical models and use different encoding schemes. The same studying and emulating, if the encoding schemes are different, it will have different data structures, and the implementation methods are various. Here are some common encoding schemes and their corresponding emulating methods.

3.2.3.1. Emulating method based on 0/1 encoding scheme. There are three common 0/1 encoding modes, binary code, grey code and one-hot code. They have different advantages and disadvantages for various problems. There is no need to elaborate on the differences and relationship between these three encoding modes. However, for the IOAs, if binary encoding scheme was used, the encoding operation and decoding operation of the individual (i.e. the candidate solution) is relatively straightforward, and the computation time consumed is relatively less. However, the transcoding operation among individuals, i.e. the search operation of a candidate solution, is relatively complex and the storage space consumed is relatively more. If grey or one-hot encoding scheme was used, the encoding operation and decoding operation are fairly complex and the computation time consumed is relatively more, but the translating operation among individuals is fairly simple and the storage space consumed is relatively less. This situation is just different from the situation in the digital control system. It will use fewer flip-flops and consume more combinatorial logic when the binary or grey code encoding scheme is used, and it is the opposite when one-hot encoding scheme is used.

If 0/1 encoding modes were adopted, the degree of difference between two individuals is measured by the Hamming distance. The studying and emulating method is just to reduce the Hamming distance between itself and the benchmark object. The adjustment of search direction and the step size is mainly reflected in the variation of its gene expression. The strength of the learning desire is primarily reflected in size of the learning rate. The intensity of emulating behaviour is mainly reflected in the number of changed genes.

3.2.3.2. Emulating method based on float encoding scheme. Float encoding is one of the most common encoding modes in the IOAs. If the float encoding scheme was adopted, the degree of difference between two individuals could be characterized by the concept of distance, and the degree of similarity between two individuals can be represented by the correlation coefficient. For the concept of distance, the Minkowski distance [1] is excellent. In practical applications, based on the data structure and the features of the problem to be solved, it is better to convert Minkowski distance to Manhattan distance [2], or Euclidean distance, or Chebyshev distance [3], by selecting appropriate parameters. For the correlation coefficient, based on the data structure and actual demand, it is also possible to choose appropriate calculation methods. For example, Pearson correlation coefficient [4], or cosine similarity, or Tanimoto coefficient [5], or Spearman correlation coefficient [4], or Kendall correlation coefficient [6].

When float encoding mode is used in the IOAs, the studying and emulating method is to reduce the Minkowski distance between one of itself and the benchmark object, or (and) increase the correlation coefficient between one of itself and the benchmark object.

In addition, *for float encoding scheme, it is better to pay particular attention to the following two issues,* which are the in-depth experience of actual programming practice.

Firstly, on the reasonable choice of method measuring the degree of difference and similarity. From the point of view of management, the degree of difference and the degree of similarity is just two similar concepts, and it seems that you can convert each other based on the arithmetic. However, from data science and computer technology, they are two completely different concepts with different precise definitions. Their connotation and denotation are completely different, and they should not and must not be converted to each other. Based on various analysis models and data structures, the degree of difference characterized by distance, and the degree of similarity measured by the correlation coefficient, the difference between the two is huge. The degree of difference is more to reflect the differences between two individuals in the numerical value, but the degree of similarity is more to take into account the differences between two individuals in the direction. In the process of calculation and searching, there may appear a kind of situation, in which the degree of difference between two individuals has changed, but the degree of similarity between them has not yet changed. Alternatively, conversely, the degree of similarity between two individuals has changed, but the degree of difference between them has not yet changed. Therefore, when to design specific optimization algorithm aiming at a specific problem, it is necessary to select the reasonable method to measure the degree of difference and similarity based on specific analysis model and data structure.

Secondly, on the transcoding, calculation and accuracy of a floating-point number in algorithm designing and testing. At present, the majority of CPU/GPU, which is in line with the IEEE Standard 754 and support the SSE2 instruction set, are the hardware foundation of algorithm designing and testing. However, according to the floating-point storage format specified by IEEE Standard 754, there are some decimal numbers with decimal places, which cannot be accurately represented using the 0/1 binary code currently used on the CPU/GPU. Therefore, if there is no reasonable way to deal with this situation, the accuracy misalignment will become serious in transcoding and calculation. In short, the precision of floating-point numbers and their transcoding and calculation is largely determined by the exponential range. Take the 32-bit single precision floating-point number as an example, the minimum of exponent part should not be lower than $2^{-7}-1$, and the maximum of exponent part should not be higher than $2^{8}-1$ (which has excluded the particular circumstances that the exponent part all is 0 or 1). If beyond the exponential range, it will have to abandon the end of the decimal and result in up overflow or down overflow. Sometimes, for a very small value, CPU/GPU only has to deal with it as 0, because it cannot be represented by the normalized value. If the low-precision floating-point number is discarded many times, then the accuracy misalignment will become worse, even the divisor may become as 0, and cause an exception. If the floating-point number was treated in a non-normalized way, although the accuracy misalignment can be remedied, the efficiency of transcoding and computing would exponentially decline, because it has a higher request for CPU/GPU and compiler. Therefore, the designing and testing of IOAs not only need to consider the characteristics of the problem domain but also need to consider the hardware CPU/GPU and the software compiler in the calculation process. Finally, when the calculation results return, it also needs to consider the decoding problem in the algorithm itself, that is, the transcoding issue of between the binary machine code and the floating-point number. Yes, this involves algorithmic code optimization.

3.2.3.3. Emulating method based on character(s) or Boolean encoding scheme. Many of the problems in science, engineering and management, especially in the field of Data Science, the model must use symbolic functions to build. For example, the usual commodity evaluation and recommendation problems in E-commerce, which involved human–computer interaction. In this case, it often requires character or string encoding scheme, or Boolean encoding scheme.

When character(s) or Boolean coding mode is used in the IOAs, the degree of difference and similarity between two individuals can be represented by Levenshtein distance [7], or Jaro-Winkler distance [8], or Jaccard coefficient [9], etc. The studying, emulating method is just to reduce the degree of difference between one of itself and the benchmark object, and (or) to increase the degree of similarity between one of itself and the benchmark object by gene editing.

3.2.4. The operators in BA

Here, it is important to note that, in general, the operators and their probability rules are essential for an intelligent algorithm, because the performance is entirely determined by the probability rules of the operators when the algorithm is running. However, for the BA, its performance is not determined by the probability rules of the operators but depends entirely on the organizing tactics of the operators. The following four operators are just a kind of concrete implementation of benchmarking philosophy aiming at the common encoding schemes. Compared with the existing IOAs, There is nothing special about these operators. Write them out here, just for the completeness of algorithm description.

3.2.4.1. Global benchmarking. Let X_E^{best} be the best individual within the whole ecosystem (whose fitness value takes the maximum or minimum according to the optimization purpose, the same below), i.e. the global optimal individual, i.e. the global benchmark, whose gene expression is G_E^{best}; X_K^i is the ith individual in population P_K, whose gene expression is G_K^i. For the following two different encoding schemes commonly used, we give the corresponding global benchmarking programme according to the specific methods of benchmarking shown above.

If 0/1 encoding mode is used, the individual X_K^i executes global benchmarking, which means that the different gene bit in G_K^i from G_E^{best} is replaced by the corresponding one in the latter by some probability. I.e. the individual X_K^i actively shrinks the Hamming distance from the global benchmark.

If float encoding mode is used, the individual X_K^i implements global benchmarking, which means that its gene expression G_K^i is updated with a certain probability by the equation of $G_K^i = G_K^i + \text{rand} * (G_E^{\text{best}} - G_K^i)$. I.e. the individual X_K^i actively shrinks the Euclidean distance from the global benchmark.

For the probability rules of global benchmarking, it can be designed according to the guiding principles of benchmarking. That is, efficiently use the Matthew effect and rationally use the cluster effect. Then the probability rules of global benchmarking can be given by the following equation:

$$\begin{cases} \text{if } \max f(x) : \text{Grate}_K^i = \text{Grate}' + f_K^i/\tilde{f}_K - 1, \\ \text{if } \min f(x) : \text{Grate}_K^i = \text{Grate}' + \tilde{f}_K/f_K^i - 1, \end{cases}$$

wherein Grate' stands for the initial rate of the global benchmarking, f_K^i stands for the fitness of the individual X_K^i, $\tilde{f}_K$ stands for the average fitness of the niche population P_K.

Under the algorithm model proposed in this paper, the probability of global benchmarking is to follow the above rules or completely take random value, the results of a large number of simulation experiments made by the author showed that there is not much difference between the two. Therefore, the algorithm model proposed here does not need any so-called adaptive and intelligent probability rules. The operators in the algorithm model just need to be organized according to the guiding principles mentioned above, the algorithm will be sufficiently "intelligent". Therefore, the probabilities of other operators just need to take random values. The algorithm model proposed here needs no probability tuning. It sounds incredible at first, but it is true. The readers can follow this idea to design their own algorithm or just download the procedure code to test.

3.2.4.2. Local benchmarking. Let X_K^{best} be the best individual in the niche population P_K, i.e. the local optimal individual, i.e. the local benchmark, whose gene expression is G_K^{best}; X_K^i is the ith individual in population P_K, whose gene expression is G_K^i. Similar to the global benchmarking, we give the local benchmarking programme for the following two different encoding schemes commonly used.

If 0/1 encoding mode is used, the individual X_K^i executes local benchmarking, which means that the different gene bit in G_K^i from G_K^{best} is replaced by the corresponding one in the latter by some probability. I.e. the individual X_K^i actively shrinks the Hamming distance from the local benchmark.

If float encoding mode is used, the individual X_K^i implements local benchmarking, which means that its gene expression G_K^i is updated with a certain probability by the equation $G_K^i = G_K^i + r\text{and} * (G_K^{\text{best}} - G_K^i)$. I.e. the individual X_K^i actively shrinks the Euclidean distance from the local benchmark.

Please note that both the effect of global benchmarking and local benchmarking is to reduce the Hamming/Euclidean distance, which appears to be the same, but in fact, there is a significant difference between the two. Moreover, this is the essence of the core ideas of BA. The behaviour of actively shrinking the Hamming/Euclidean distance from the global/local benchmark is beneficial for cluster search of the population. It is helpful to form the cluster effect, and quickly find out the optimal global/local solution. At the same time, it is also the best way to maintain the population diversity, because the learning objective of each individual is constantly dynamic changed, so the level of population clusters within the ecosystem is also dynamic changed.

3.2.4.3. Self-learning. Let G_K^i be the gene expression of individual X_K^i, we give the self-learning programme for the following two different encoding schemes commonly used.

If 0/1 encoding mode is used, the individual X_K^i executes self-learning, which means the each gene bit in its gene expression G_K^i performs dual mapping by some probability, i.e. 0↔1. For example, the original gene expression is like this: 1 1 1 0 0 0 1 1 1, then the dual gene expression is: 0 0 0 1 1 1 0 0 0.

If float encoding mode is used, the individual X_K^i executes self-learning, which means its gene expression G_K^i is updated with a certain probability by the equation $G_K^i = a + \text{rand} * (b - a)$, where $G_K^i = [x_1, x_2, \ldots, x_n]$, $x_i \in [a_i, b_i]$.

3.2.4.4. Tentative learning. Tentative learning refers to that the tentative strategy is used in the process of benchmarking. That is, after benchmarking, the individual will automatically return to the state before benchmarking if its fitness value has not been improved accordingly. This can be achieved by gene resetting.

3.2.5. Five factors in algorithm design

In the design of a specific algorithm, the following factors have significant impacts, which need to focus on consideration.

First, probability rules of the operators. Traditionally, the "intelligence" of IOAs is mainly reflected in the probability rules, which determined the probability of the operators in IOAs. Therefore, there are some probability equations or mathematical formulations in IOAs. However, the "intelligence" of algorithm model proposed here depends on the organizing tactics rather than the probability rules. Since the probabilities of operators, including global benchmarking, local benchmarking, self-learning and tentative learning, have no need to set initial values and are all set as random values during the running process. That is, the algorithm model proposed here, does not need any so-called self-adaptive or intelligent probability rules. The operators in the algorithm model just need to be organized according to the guiding principles mentioned in later, which has shown in the above algorithm flowchart, the algorithm will be sufficiently "intelligent". Of course, if the probability rules were set according to the guiding principles of benchmarking, the search effect is not bad as well. But a large number of numerical simulations made by the author showed that there is not much difference between the two.

Second, the criteria for setting a benchmark. The best individual within the whole ecosystem (whose fitness value takes the maximum or minimum according to the optimization purpose) will be set as the global benchmark. That is, the global optimal individual will be fixed as the global benchmark. However, for the global optimal individual, there are two different cases. The first is that the individual is the best individual of the ecosystem which evolved to the present generation, i.e. the so-called the global optimal individual to the current generation. The second is that the individual is the best individual of the ecosystem in the current generation, i.e. the so-called the global optimal individual in the current generation. Similarly, the best individual in each niche population will be set as the local benchmark. There are two different cases as well. The first is that the local benchmark is served by the local optimal individual to the current generation. The second is that the local benchmark is served by the local optimal individual in the current generation.

Third, the position of tentative learning. Tentative learning refers to that the tentative strategy is used in the process of benchmarking. That is, after benchmarking, the individual will automatically return to the state before benchmarking if its fitness value has not been improved accordingly. This can be achieved by gene resetting. The tentative learning strategy plays the role in protecting healthy genes from damage, which should be embedded in the three operators of global benchmarking, local benchmarking and self-learning. In the actual designing, whether the tentative learning is necessary or not, and where it is placed, depends on the specific circumstances.

Fourth, the condition for each niche population to exchange their local benchmarks. This condition can be set in a variety of ways. It can be configured according to the information about the algorithm itself, or according to the heuristic information related to the problem to be optimized. For example, if the global benchmark within the ecosystem does not change for three consecutive generations, it is required that the niche populations randomly exchange their local benchmarks. Exchanging the local benchmarks, in theory, it is good for algorithmic performance. However, it is not as obvious as expected in the numerical simulation experiments. Because the local benchmarks in each generation are always changed in the running process.

Fifth, the step size of benchmarking under the float encoding scheme. In the after-mentioned basic operators, for the sake of simplicity, the factor of step size was set to a random value between 0 and 1. In addition, it can be adjusted adaptively according to the fitness of the individual. Theoretically, the search result will be better.

3.2.6. The simple benchmarking algorithm (SBA)

The SBA mainly refers to such a set of configurations. The global benchmark is served by the global optimal individual to the current generation. The local benchmark is served by the local optimal individual in the current generation. The tentative learning strategy is nested in the global benchmarking and the local benchmarking. The probabilities of global benchmarking, local benchmarking and self-learning are completely set to random values between 0 and 1. The pseudo code of SBA is shown as follows.

Let $E = [P_1, P_2, \ldots, P_{np}]$ be the whole ecological system consisting of *np* niche populations, N_i be the number of individuals in niche population P_i, P_i^j be the *i*th individual in niche population P_i, P_i^{best} be the best individual in niche population P_i. Let f_i^j be the evaluation function value of individual P_i^j, $\tilde{f}_i$ be the average value of niche population P_i at current generation, $\tilde{f}_E$ be the average value of the ecological system E at current generation, P_{best} be the best individual in the ecological system E. Let $\max\_gen$ be the maximum iteration times. Then the pseudo code of SBA can be shown as below.

```
♦ Initialization. $np$, $N$ and other parameters if necessary.
♦ For $gen = 1$: $\max\_gen$, do
    (a) For $i = 1{:}np$, do
          ✓ Evaluate $f_i^j$ and $\tilde{f}_i$,
          ✓ Find and record $P_i^{best}$, i.e. set the local benchmark
    (b) Find out and record $P_{best}$, i.e. set the global benchmark
    (c) Find out, record and update the best individual in the ecological system.
    (d) Evaluate $\tilde{f}_E = \left(\sum^{\tilde{f}_i}\right)/np$
    (e) For $i = 1{:}np$, do
        ✓ $\begin{cases} maxf(x): N_i = N_i + \tilde{f}_i/\tilde{f}_E - 1 \\ minf(x): N_i = N_i - \tilde{f}_i/\tilde{f}_E + 1 \end{cases}$
        ✓ If $N_i = 0$, then the niche population $P_i$ die out
    (f) For $i = 1{:}np$, do
        ✓ $P_i^j$ conduct global benchmarking
        ✓ If $f_i^j$ does not be improved, then, $P_i^j$ will conduct local benchmarking
        ✓ If $f_i^j$ does not be improved yet, then, $P_i^j$ will carry out self-learning
    (g) If $\tilde{f}_E$ does not be improved or the best individual in the ecological system does not be replaced, then each niche population exchange their local benchmarks
♦ Output the optimal solution(s)
```

3.2.7. Other versions of BA

As mentioned above, the global benchmark and the local benchmark, respectively, have two different ways for the establishment. Some other factors, for example, whether the tentative learning is adopted or not, and where it is placed; the condition for exchanging local benchmarks; the step size of benchmarking; the probability rules of the core operator, etc., there are a variety of ways. In addition to the SBA, any combination of the above factors can constitute a different version of BA.

4. Some technical analyses of this algorithm model

4.1. The core merit of this algorithm model

Any versions of BA designed according to the guiding principles and specific methods of benchmarking mentioned above will be able to achieve the synergistic coexistence and automatic balance of exploration and exploitation, thus the population diversity will be kept during the running. This assertion was confirmed by theoretical analysis and simulation experiments.

4.1.1. The "intelligence" mainly depends on the organizing tactics rather than the probability rules

As we all know, traditionally, the IOAs based on probabilistic search, whose "intelligence" is mainly reflected in their probability rules. Moreover, these probability rules mainly refer to the rules that the operators of IOAs are implemented according to a certain probability. Therefore, almost all of the IOAs need to set a variety of probability rules for their operators and the corresponding initial values. For example, the GA and its various improved versions need to set the probability equations for their three operators of selection, crossover, and mutation, and the corresponding suitable initial values. The particle swarm optimizations (PSOs) and its various improved versions need to set the probability equations for their inertia weight factors and acceleration coefficients and the corresponding suitable initial values. The ant colony algorithm (ACO) and its various improved versions need to set the probability equations of the pheromone and heuristic factors and their suitable initial values. These probability rules and the corresponding initial values are set appropriately or not, which fundamentally determine whether the algorithm can play its proper role or not.

However, how to configure the probability parameters of an algorithm to maximize its performance, which in itself is a combinatorial optimization problem. This issue, in theory, has not yet been well resolved. Therefore, almost all of the existing IOAs how to configure the probability parameters mainly rely on statistical experience. It is currently the joint shortcoming of all the existing IOAs.

There is no such problem in BA proposed this paper. As can be seen from the preceding, it is not necessary to set the probability equations for the operators. The probability of the operator is completely set as a random value between 0 and 1. Of course, when designing other versions of BA, we can set different probability rules for each operator and the corresponding initial value. However, it increased the complexity of BA, and many numerical experiments made by the author showed that which did not make the performance of BA have significantly improved. In terms of traditional thinking, the probability of the operators should be the core parameter, which should have a substantial effect on the performance of the algorithm. However, the search performance of BA does not depend on the probability rules of the operators but the idea of benchmarking. Specifically, it depends on the guiding principles and the specific methods of benchmarking. Based on specific encoding scheme and data structure, the core operators in BA are just a form of implementation of the idea of benchmarking and its guiding principles. In fact, as long as following the ideas and principles of benchmarking, any versions of BA based on any encoding scheme and data structure will naturally get excellent performance.

In summary, this institutional arrangement without probability rules completely avoided the adverse impact of probability rules on the performance of BA. The "intelligence" of BA is not reflected in the probability rules of its operators but on the organizing tactics represented by the guiding principles of benchmarking. Moreover, this is one of the most significant differences from the existing IOAs, which was confirmed in the following simulation experiments.

4.1.2. Synergistic coexistence and automatic balance of exploration and exploitation

The exploration of an algorithm reflects its ability to perform the global search and develop new spaces, while the exploitation reflects its ability to perform local search and refining the improvement. In general, the stronger the exploration, the weaker the exploitation and vice versa. So traditionally, the exploration and the exploitation are contradictory and conflicting. If the exploration is too strong and the exploitation is too weak, the search result is easy to diverge and easy to miss the extreme points. It is ultimately not conducive to find out the global optimal solution. On the contrary, if the exploration is too weak and the exploitation is too strong, the search result is easy to gather and easy to fall into local extreme points. It is also ultimately not conducive to find out the global optimal solution. For estimating the performance of an IOA, the most important point is to see what strategy it used to balance its exploration and exploitation. It determined the advantages and disadvantages of an algorithm to a large extent.

In order to balance the exploration and exploitation, the common strategy used by almost all of the existing IOAs is attaching the corresponding probability parameters for different operators. For how to set the values of those probability parameters, there are two ways. The first is to set fixed values in advance based on statistical experience. The second is to adjust the values dynamically based on certain principles in the search process. For example, the GA and its various improved versions are often using the three parameters of selection rate, crossover rate and mutation rate to balance the exploration and exploitation. The PSO and its various improved versions are often using one inertia weight factor and two acceleration coefficients to balance the exploration and exploitation. The ACO and its various improved versions are often using the pheromone strength factor to balance the exploration and exploitation. Similar cases are numerous. For how to set the values of those probability parameters, although each algorithm has its own skills, all of them are not perfect. Since those probability parameters, as control variables, how to set their values in itself is an optimization problem as well, and has not yet been satisfactorily resolved. In fact, regardless of what kind of skills is used to dispose of those probability parameters, their values and the corresponding balance effect may not be optimal. Therefore, in general, for the exploration and the exploitation, i.e. the global search and local optimization, how to balance the two of them, the existing IOAs have no perfect solutions.

The exploration and the exploitation in itself are antagonistic and incompatible, but in algorithm design, it is possible to achieve synergistic coexistence and automatic balance. Therefore, the "hidden rule" that you need to rely on probability rules to balance the exploration and the exploitation, which can be completely broken or avoided. The general frame of BA abandoned the traditional strategy that using probability parameters to balance the exploration and the exploitation. As a matter of factor, as long as based on the guiding principles of benchmarking, i.e. relying on reasonable institutional design, rather than control parameters, any version of BAs, whose exploration and the exploitation, will achieve synergistic coexistence and automatic balance. In the general frame of BA, the global benchmarking is equal to exploring the probability distribution of the unknown extreme point in the solution space. This kind of global search plays the role in opening up new space and embodied the exploration of BA. The local benchmarking is equivalent to that using the known actions to get more feedback to make the next "revenue maximization" based on the current observations. This kind of local optimization plays the role of refining the improvement and incarnated the exploitation of BA. The second part of the guiding principles of benchmarking, i.e. the principle of studying and emulating chosen on demand, which ensures that the global benchmarking and the local benchmarking must be implemented in the same round of iterative. This is the same round but unsynchronized system arrangement cleverly neutralized the contradictions that the global search and local optimization cannot coexist in parallel, and thus fundamentally ensured the balance of exploration and exploitation. It was confirmed in the following simulation experiments.

4.1.3. Simulation experiments on the exploration and exploitation of this algorithm model

Now let's take the function Peaks built-in Matlab as an example to show how to achieve synergistic coexistence and automatic balance of the exploration and the exploitation of BA. Use the function Peaks as an example, because its 3-d map is relatively simple, it is easy to observe the individual's dynamic search process.

The SBA was used in this simulation. There are five niche populations in the initial stage of the ecosystem, and each containing three individuals. That is, there is a total of 15 individuals in the search space. The binary encoding scheme was adopted, and the length of each individual's gene expression is 30. Maximum iterations were set as 100. The optimization goal is to find out the maximum of function Peaks.

There are two different cases. One is that the initial population is randomly distributed in the search space. The second is that the initial population is placed at the same point (3, 3) in the search space. The following showed the dynamic search process of each individual in the two cases. It would be very confusing if the dynamic search processes of 15 individuals were shown in the animation at the same time. So the dynamic search process of each individual was displayed separately to show how to achieve synergistic coexistence and automatic balance of its exploration and exploitation.

Please note that the dynamic search process of each individual was shown in a GIF image. There are

30 GIF images. Subject to the file format, it cannot show the animations here. So all the 30 GIF images are placed in a zip file called "Searching Process.rar" (http://www.mediafire.com/folder/7m1bfpflz4byl/Relevant_Data). From the dynamic search process of all the 15 individuals with 2 different initial distributions, we can see 3 observable phenomena.

First, each individual in the ecosystem carried out the "hill-climbing" for many times in the process of 100 iterations. In the search process, it is evident that when the individual was in the hillside, it was attracted by the local benchmark, and thus generated some behaviours of "hill-climbing", and eventually climbed to the top of the hill, and found out the maximum. The behaviour of "hill-climbing" is equivalent to that using the known actions to get more feedback to make the next "revenue maximization" based on the current observations. It plays the role of refining the improvement and incarnated the exploitation of BA.

Second, the movement trajectory of each individual in the ecosystem has covered almost the entire search space. The maximum iterations were set to 100, i.e. each individual only evolved for 100 generations. However, it can be seen that the individual's search path almost covered the entire solution space in this 100 iterations. In theory, as long as the number of iterations is big enough, individuals in the ecosystem will traverse the entire search space. This large-scale and global search capability reflected the ability to open up a new area and embodied the exploration of BA.

Third, almost every individual in the ecosystem had found out the global maximum for multiple times. For example, from the gif format images, we can see that under the two conditions with different initial population distribution, the first individual found out the global maximum of 15 times and 17 times, respectively. This is benefited from the search strategy, and which ensures the BA has relatively high search efficiency.

Finally, the need for special note is that the data used in this simulation were produced while SBA was run casually once, and they were not specially picked. In other words, they are not deliberately chosen to show such good results. Readers can use the programme code to execute the algorithm to generate data to verify.

4.2. The second merit of this algorithm model

4.2.1. Population diversity has been maintained during the process of running

How to maintain population diversity in the process of running is one of the most critical challenges faced by all the existing IOAs. In particular, for dynamic optimization problems, the external environment of the population will change with the constraints, environmental variables and other factors. For example, if the number of control variables changed, the search space was deformed, and eventually, the optimal solution of the problem will change. To deal with this kind of dynamic optimization problems, it is necessary to exquisitely detect the changes of the external environment, and then make the corresponding rapid response to track the drift or jump trajectory of the optimal solution in the search space. Therefore, for the IOAs, whether to maintain the population diversity in the search process is one of the core indicators of determining whether the performance is excellent.

Traditionally, in order to keep the population diversity in the process, it is taking the same measures, which plays the role of balancing the exploration and the exploitation. That is, using the corresponding probability parameters to adjust the operators that have an impact on the population diversity. For example, the GA and its various improved versions, are often expanding the population diversity by increasing the crossover rate or mutation rate; the PSO and its various improved versions, are often using the inertia weight factor and the acceleration coefficients to maintain the population diversity. For the other of IOAs, they have the similar approaches, that is, by adjusting the corresponding probability of some operators to control the population diversity in the search process. However, how to configure the probability parameters, mainly rest upon experimental design, and there is no complete theoretical support at present.

In the framework of BA, as mentioned above, the global benchmarking or local benchmarking, both the effect aims to reduce the differences and/or to increase the similarity between one of itself and the benchmark object. In other words, the implementation of benchmarking will form a certain clustering effect, and the clustering effect will lead to a certain reduction of population diversity in the search process. However, due to the artful system design, the benchmarking defined under the framework of BA (please refer to the section "The guiding principles and specific methods of benchmarking"), will not cause the loss of population diversity but speed up the process of searching for the optimal solution and promote the conservation of population diversity, because there will produce many new benchmarks after each round of competitive learning. (For this, it's hard to believe that it's true if you didn't write the algorithm codes and run it yourself. Even the author himself, it's also because this feature always emerged in the algorithm testing, after repeated testing and confirmation, this feature has been put forward separately as the second merit of this algorithm model. Readers can download code to verify themselves.) Even if there was an extreme case emerged in the process of benchmarking (in fact, so far, after thousands of runs and tests, there has been no such extreme situation. I guess it is because there would produce many new benchmarks after each round of competitive learning), for example, the benchmarks in the solution space have not changed for successive rounds of iterations. Then according to the third guiding principle of benchmarking, i.e. the learning object is flexibly switched, we will force the niche populations to exchange their local benchmarks. In this way, the learning objective of an individual will continue to change, and the individual cluster level within the search space will change dynamically in the search process. Thus, it is both conducive to forming cluster effect to help for searching the optimal solution, but also cleverly solved the problem of maintaining the population diversity in the search process. This is exactly of that the "intelligence" of BA is embodied in the organizing tactics of its operator. The organizing tactics based on this institutional design, which makes BA can naturally maintain the population diversity in the search process. It was confirmed in the following simulation experiments as well.

4.2.2. Simulation experiments on the population diversity in this algorithm model

Now let's still take the function Peaks built-in MATLAB as an example to show how BA to keep the population diversity in the search process. The SBA was still used in this simulation. The parameters are the same as in the previous section.

In general, for IOAs, the population diversity in the search process will be greatly affected by the two factors of the encoding scheme and the initial population distribution. In this test, the encoding scheme has two different cases, i.e. binary encoding and float encoding. The initial population distribution also has two different cases, i.e. random distribution and designated distribution. The latter refers to that the initial population is placed at the same point (3, 3) in the search space. Now, in the two cases of finding out the maximum and the minimum of the function Peaks, let's test the BA's ability to keep the population diversity in the process of running. Therefore, there are eight different initial conditions shown in Table 2 and correspondingly eight various distribution.

Please note that the population distribution state at each generation will be photographed on a static picture, and the population distribution state of 100 generations constitute a dynamic picture, so there are 8 GIF images. Subject to the file format, it cannot show the animations here, so all the eight GIF images are placed in a zip file called "Population Distribution.rar" (http://www.mediafire.com/folder/7m1bfpflz4byl/Relevant_Data). From the distribution states in 8 different cases, we can see three observable phenomena as well.

Table 2. Eight different initial conditions.

Case number	Optimization goal	Encoding scheme	Initial population distribution
1	Maximum	Binary	Random
2	Maximum	Binary	Designated
3	Maximum	Float	Random
4	Maximum	Float	Designated
5	Minimum	Binary	Random
6	Minimum	Binary	Designated
7	Minimum	Float	Random
8	Minimum	Float	Designated

First, BA's ability to maintain the population diversity is not affected by the two factors of the encoding scheme and the initial population distribution. Regardless of whether the encoding scheme is binary or float, regardless of whether the initial population is random or designated at the same point (3, 3), BA maintained good population diversity during the run. At the simulation in the previous section, we found the movement trajectory of each individual in the ecosystem has covered almost the entire search space. The results of this experiment indirectly confirmed the experimental results of the former.

Second, the local clustering search is synchronized with the decentralized global search. The simulation in the previous section showed the search process of each individual at the micro level. The experiments in this section showed the search process of the population from the whole level. It can be observed that, regardless of the conditions, the entire population sometimes gathered together and sometimes decentralized, which is consistent with the previous theoretical analysis and prediction, and also confirmed at the population level that BA relied on the system design to achieve the balance between exploration and exploitation.

Third, almost all the individuals have found out the global extremes. In order to facilitate the observation, specifically assigned a numerical order for each individual in the search space. We can see that, regardless of the encoding method and the initial population distribution, the vast majority of individuals have searched the global maximum or minimum. Which also indirectly confirmed the results of the simulation in the previous section.

Similarly, the data used in this experiment was generated while SBA was run casually once, and they were not specially picked. That is to say, they are not deliberately chosen to show such good results. Readers can use the procedure code to run the algorithm to generate data to verify.

4.3. The third merit of this algorithm model

4.3.1. Lots of useless and ineffective repetitive operation are avoided

"Useless operations" and "ineffective operations" here refer to the two types of search actions carried out by the operators in the algorithm. The candidate solutions generated by the former are in accordance with the constraints and are legal, but they are often executed repeatedly and wasted much running time and storage space. The candidate solutions generated by the latter are nonconforming and illegal, so they need to be repeatedly corrected and also wasted much running time and storage space. The common feature of these two types of operations is that they are repeatedly executed and are not conducive to solving the optimization problem. For example, GA and its various improved versions often generate a large number of "useless operations". When a feasible candidate solution is generated, the individual may repeatedly execute the crossover operation with other individuals without returning the corresponding fitness values. Thus the great genes obtained in the previous crossover operation may be damaged by subsequent repetitive crossover operations. The simulated annealing algorithm will also generate a large number of repetitive exploratory operations during the search process, and these operations are "useless operations". The PSO and its various improved versions, there are no "useless operations", but if the inertia weight and acceleration coefficients were set too big, they would produce a large number of "ineffective operations".

For the IOAs, under the same hardware and software conditions, if the "useless operations" and "ineffective operations" are less, the less the running time and storage space, the higher the search efficiency. In the framework of BA, as mentioned above, the three operators of global benchmarking, local benchmarking and self-learning are not sequentially executed but on demand. In the process of benchmarking, only its fitness value is not improved, an individual will execute the next operator. So it will not generate a lot of "useless operations", and has the opportunity to modify the genetic structure and improve the fitness value before the end of the current iteration. Therefore, the search efficiency is very high, and the convergence speed is rather quick. "Ineffective operations" in IOAs are mainly generated in the float encoding scheme. For example, because the inertia weight factor and acceleration coefficient were set too large, in the PSO and its various improved versions, the legitimate candidate solutions tend to fly out of the bound boundary and become illegal. So a series of operations has emerged as the "ineffective operations". For this problem of "ineffective operations", there is some literature on theoretical analysis [10] and experimental demonstration [11]. The typical approach at present is that if some of the vectors overflowed the boundary of the decision variable, they would be replaced by the boundary value so that the current illegal candidate solution will be washed out as a legitimate one. There are some other methods, for example, if some of the vectors overflowed the boundary of the decision variable, then implement modulo operation (Mod) or remaining computing (Rem) between the current illegal candidate solution and the boundary value, or perform the operators again to regenerate the legitimate candidate solution. Although these methods solved the legitimacy problem of candidate solutions, they destroyed the original "intelligence" of the algorithm.

BA has no such problem of "ineffective operations" in the search process. In the case of float encoding scheme, the search behaviour of an individual is benchmark-oriented to reduce the Euclidean distance for the purpose. Because the benchmarks are all legally valid, and the step size is a random value between 0 and 1, so in the implementation of benchmarking, the individual will not fly out of the boundary into an illegal solution, and naturally will not produce the problem of "ineffective operations" as PSO. Of course, this is mainly in terms of weak constraints. For very complex and strict constraints, it is not necessarily valid. The obvious differences in the social information sharing mechanism between BA and PSO will be no longer repeated here. In a word, the convergence speed of BA is rather quick, and the search efficiency is very high. It was confirmed in the following simulation experiments.

4.3.2. Simulation experiments on the convergence speed of this algorithm model

Now let's take the dynamic function optimization problem as an example to show the convergence speed of the algorithm model proposed here. Why do we design this dynamic function optimization issues? There are three reasons. The first is because many of the matters in science, engineering and management can be translated into the optimization problems. The second is that most of these optimization problems can be transformed into the problems of solving the corresponding mathematical model. The final reason is that these static mathematical models are the special case of them in the real world, and they are all the ideal state of their actual state. If an algorithm can solve the dynamic model of some problem well, then it is natural to solve its static model in the ideal state well.

With regard to the construction of dynamic functions, many methods have been proposed. Here, we use two simple methods to construct a dynamic function from a static function. The first is to change the constraints of the function to generate deformation of the search space. For example, the number of the variables and its range of values in the static function, changed according to some specific rule to construct dynamic function optimization problem. The second is to modify the individual's gene expression to produce deformation of the gene sequence. For example, when the 0/1 coding scheme was used, the gene code of all individuals periodically performed exclusive-or operation with a particular template to construct dynamic function optimization problem.

Subject to file size limit, more simulation results on benchmarking functions were not included in this file. Now take one of the CEC2017 benchmark functions, $f(x) = \sum_{i=1}^{D} x_i \sin\sqrt{|x_i|}, x_i \in [-500, 500]$, as an example. It is a very special multimodal function, there are numerous local extreme points in its solution space, but only one global minimum point. It is extremely difficult to find out the global optimal solution. It is a very good test function, and its optimization problem is a typical deception problem. Using this function, we can construct two dynamic function optimization problems based on the above two methods. Which are the dynamic function optimization problem based on space variation and the dynamic function optimization problem based on gene variation.

The SBA was used in these simulation experiments. To compare with SBA, some newest optimization methods, including the top four algorithms in the real-parameter single-objective optimization competition in CEC2017, i.e. jSO [12], LSHADE-cnEpSin [13], LSHADE-SPACMA [14] and improved differential evolution strategy (DES) [15], have been put to use in the experiments. Although all the controls parameters involved in these methods are various, to ensure a fair comparison as possible, the hardware and software computing environments of the tested algorithms were set to the same. In addition to the settings that are closely related to simulation experiments, like encoding schemes and maximum iterations, the other critical control parameters of these algorithms were set respectively with the optimal configuration in accordance with their respective references.

4.3.2.1. Dynamic function optimization based on spatial variation. The dynamic function based on spatial variation is defined as follows: the function Schwefel has three variables if the domain of definition is (−500, 500); it has two variables if the domain of definition is (−1000, 1000), so that the global minimum of the function will drift accordingly. The dynamic function Schwefel is showed as follows.

$$\min f(x) = \begin{cases} \sum\limits_{i=1}^{3} x_i \sin\sqrt{|x_i|} & x_i \in [-500, 500], \\ \sum\limits_{i=1}^{2} x_i \sin\sqrt{|x_i|} & x_i \in [-1000, 1000]. \end{cases}$$

The intensity of the environmental change is measured by the degree of change of the definitional domain, which can be used to test the dynamic tracking capability of an algorithm for the global minimum in different solution spaces.

In order to ensure the fairness, in the experiment, the float encoding scheme was used in the five algorithms. The maximum iterations of the five algorithms were all set as 1000. The definitional domain will switch over automatically once every 100 generations, i.e. the environment underwent a total of 10 cycles. The number of individuals (or particles) of the five algorithms was all set as 100, and they were all designated at the same point (3, 3). The results take the average value of 100 times.

From the experimental results showed in Figure 2, we can observe several distinct phenomena. First, the dynamic tracking ability and the robustness of SBA are much stronger than the other four algorithms. Regardless of how the search space changed, SBA always can find out the global optimal solution within the new solution space. Second, SBA's convergence is very fast. In the face of the new environment, SBA can quickly converge to the global optimal solution. This is the benefit of organic integration of learning-based competition and competitive learning. Moreover, this is related to the number of the individuals in the ecosystem, the greater the number, the faster the global convergence. Third, for these four algorithms in addition to SBA, the searching performance occurred in varying degrees of oscillation when the search space took place in various levels of deformation. The dynamic optimization performance of these four algorithms did not seem to have much difference, but in fact, they are very different from each other. As can be seen from the above figure, the optimal solution locus of each algorithm is smooth as a whole, because they are the average value of 100 times. The search performance of these four algorithms is extremely unstable. As we can see, when the search space is 3d, the search results of DES are always better than the LSHADE-SPACMA, but when the search space is 2d, whose search results are always worse than the LSHADE-SPACMA. LSHADE-CNEPSIN has opposite performance, we can see that the search performance of LSHADE-CNEPSIN was improved while the search space changed from 3d to 2d, and it apparently declined while the search space changed from 2d to 3d. The cause is unknown.

4.3.2.2. Dynamic function optimization based on gene variation. The dynamic function based on gene variation is defined as this way. In the case of 0/1 encoding scheme, in the process of searching for the global minimum of function Schwefel, $f(x) = \sum_{i=1}^{2} x_i \sin\sqrt{|x_i|}, x_i \in [-500, 500]$, the gene code of all individuals periodically performed exclusive or operation with a particular template whose length is the same with the gene code of the individuals. If a gene bit in the template is 1, then the corresponding gene bit in the individual's gene expression will change, and if a gene bit in the template is 0, the corresponding gene bit value in the individual's gene expression will remain unchanged. So the proportion that the gene bits whose value is 1 in the template can be used to characterize the intensity of environmental changes (IECs). And which can be used to test the dynamic tracking capability of an algorithm for the global minimum in different environments.

In this experiment, parameter configurations of the involved algorithms were exactly the same as in the previous experiments. In order to ensure the fairness, in the experiment, the maximum iterations of the involved algorithms in the test were all set to 1000. The domain of definition will switch over automatically once every

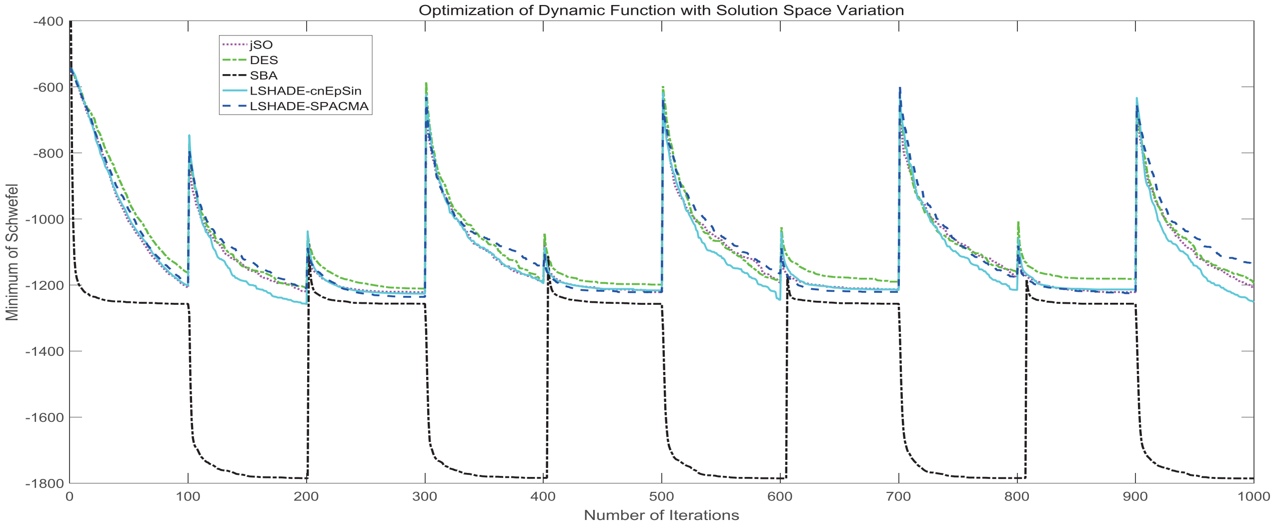

Figure 2. The mean value of 100 runs for dynamic function optimization based on space variation.

100 generations, i.e. the environment underwent a total of 10 cycles. The number of individuals (or particles) in a population of the involved algorithms is all set as 100, and they are all designated at the same point (3, 3). The involved algorithms all use a binary encoding scheme, and all run 100 times. The results take the average of 100 times. When the IEC was 0.1, 0.5 and 0.9, the dynamic tracing trajectories of the involved algorithms were shown respectively in the following figures.

From the experimental results respectively shown in Figures 3–5, it can be observed several distinct phenomena. First, SBA is robust, and its convergence ability is powerful, which is far superior to the other four algorithms. Second, the IEC has a significant impact on the algorithms. We can see that the tracing trajectories of the involved algorithms all had been strong vibrated while the IEC increased from 0.1 to 0.5. When the IEC rose from 0.5 to 0.9, the oscillation of tracing trajectories of LSHADE-SPACMA and jSO became more intense, but that of SBA was reduced. This may be related to that SBA has the strong ability to maintain good population diversity in the search process. On the eve of the oscillation of environment, i.e. before the exclusive-or operation with the template, some third-class individuals were far away from the global benchmark originally. While the IEC increased to 0.9, i.e. 90% of the gene bits in the template were 1, the genes of those third-class individuals got improved on the contrary after executing the exclusive-or operation with the template. Therefore, as shown in the above, compared with that when the IEC is 0.5, the oscillation in the tracking trajectory of SBA became moderate. Finally, it is similar to dynamic function optimization based on space variation, for these four algorithms in addition to SBA, the searching performance occurred in varying degrees of oscillation when the gene sequence happened in varying degrees of deformation. Sometimes better and sometimes relatively poor. The specific details will no longer be repeated

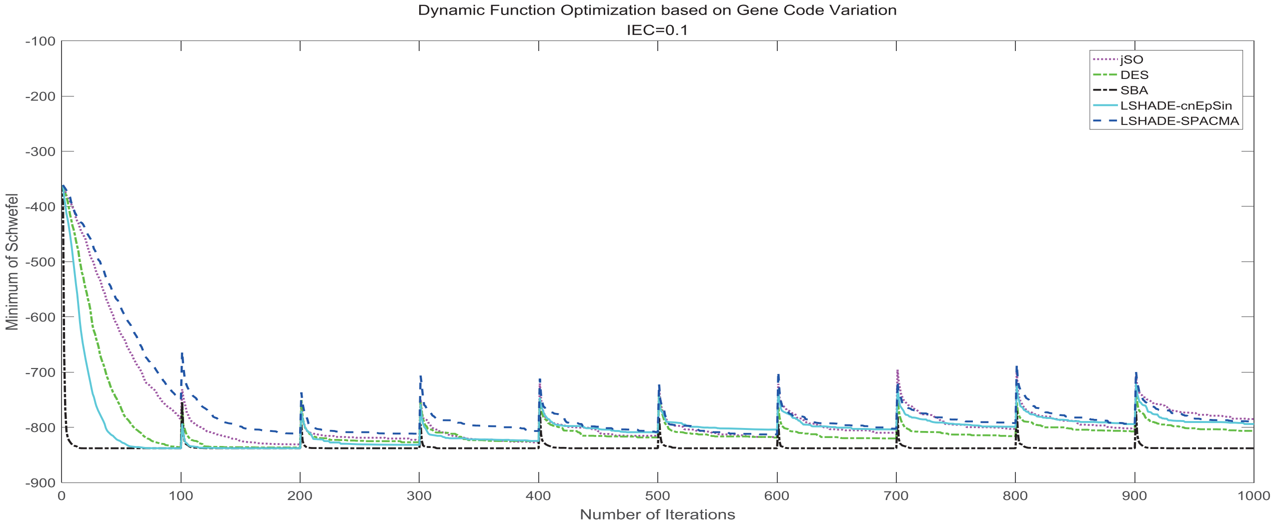

Figure 3. The tracking trajectory when the IECs is 0.1.

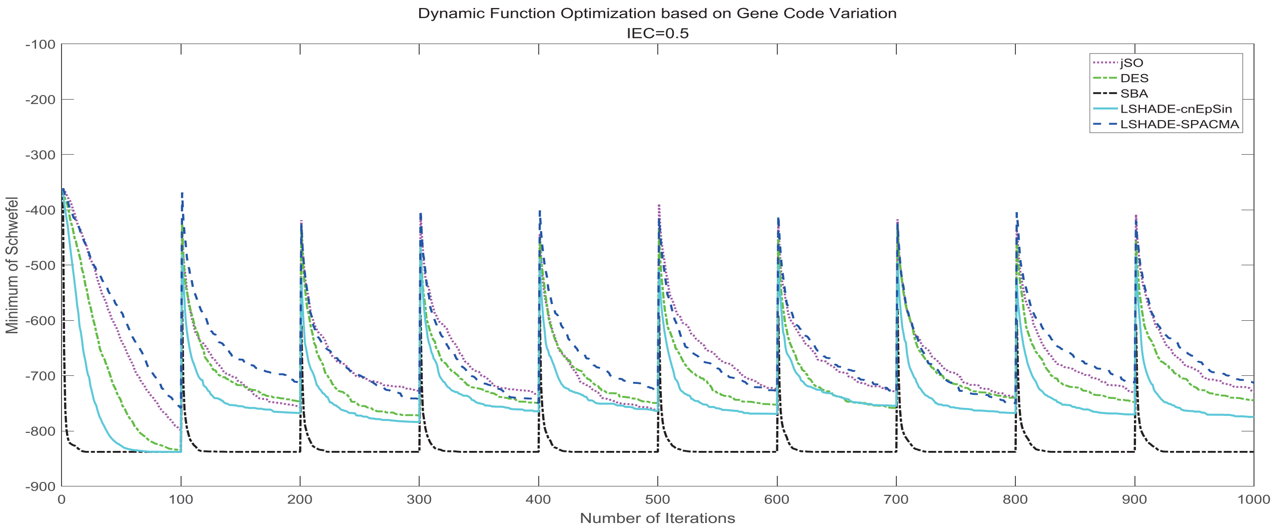

Figure 4. The tracking trajectory when the IECs is 0.5.

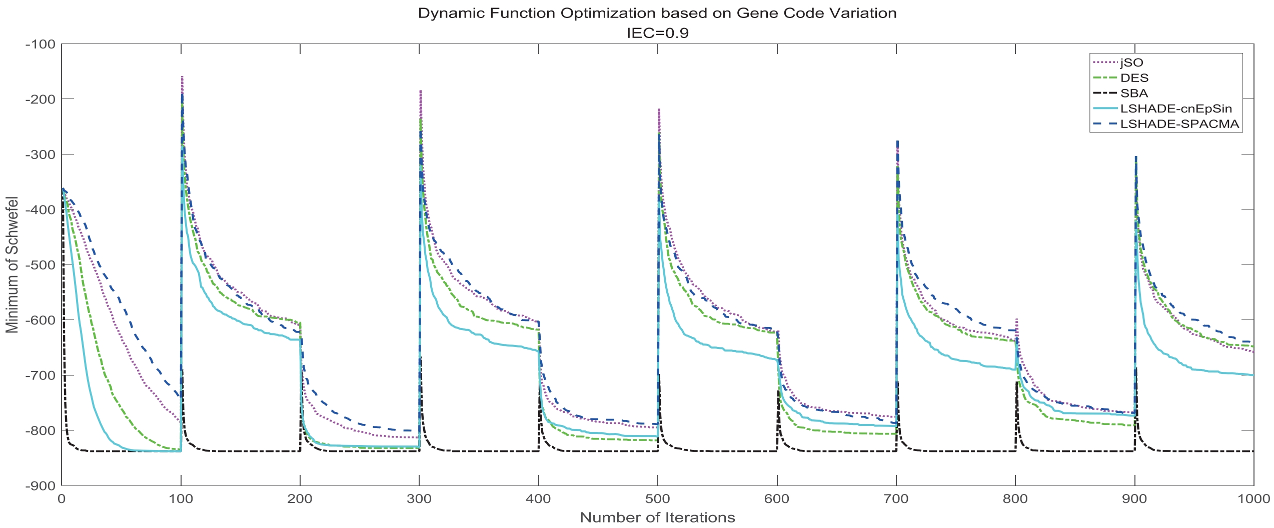

Figure 5. The tracking trajectory when the IECs is 0.9.

here. All in all, SBA has a relatively higher search efficiency.

4.4. Dynamic niche population technology

According to the guiding principles of benchmarking, in the search process, some niche populations cannot adapt because of environmental changes, in which the number of individuals gradually reduced owing to individual migration, and then they died out due to Matthew effect. At the same time, some new niche populations would be generated in virtue of cluster effect. Therefore, the total number of individuals in the whole ecosystem would not change. That is, the total number of individuals in the ecosystem is constant. This can be demonstrated as below.

Assume that the initial number of individuals in the whole ecosystem is N_{sum}, i.e. $\sum_{i=1}^{np} N_i^0 = N_{\text{sum}}$. When the algorithm run to the tth generation, the total number of individuals in the whole ecosystem is N'_{sum}, i.e. $\sum_{i=1}^{np} N_i^t = N'{}_{\text{sum}}$. In the search process, the number of individuals in each niche population would change according to the following equation.

$$\begin{cases} \max f(x) : N_i = N_i + \tilde{f}_i/\tilde{f}_E - 1, \\ \min f(x) : N_i = N_i - \tilde{f}_i/\tilde{f}_E + 1. \end{cases} \tag{1}$$

Then, take the $\min f(x)$ as an example, when the algorithm run to the $(t+1)$th generation, the total number of individuals in the whole ecosystem can be given as the following equation.

$$\sum_{i=1}^{np} N_i^{t+1} = \sum_{i=1}^{np} (N_i^t - \tilde{f}_i/\tilde{f}_E + 1) = \sum_{i=1}^{np} N_i^t - \sum_{i=1}^{np} (\tilde{f}_i/\tilde{f}_E) + \sum_{i=1}^{np} 1. \tag{2}$$

So

$$\sum_{i=1}^{np} N_i^{t+1} = N'_{\text{sum}} - \frac{1}{\tilde{f}_E} \sum_{i=1}^{np} (\tilde{f}_i) + np. \tag{3}$$

Since $\tilde{f}_E = \left(\sum_{i=1}^{np}(\tilde{f}_i)\right)/np$ substituted to Equation (3), we get

$$\sum_{i=1}^{np} N_i^{t+1} = N'_{\text{sum}} - np + np = N'_{\text{sum}} = \sum_{i=1}^{np} N_i^t. \quad (4)$$

Therefore, when $t = 0$, we get

$$N_{\text{sum}} = \sum_{i=1}^{np} N_i^0 = \sum_{i=1}^{np} N_i^1 = \sum_{i=1}^{np} N_i^2 = \cdots$$
$$= \sum_{i=1}^{np} N_i^t = N'_{\text{sum}}. \quad (5)$$

At the initial stage, all the individuals within the ecosystem were classified into different niche populations. But during the search process, these individuals migrated dynamically among the different niche populations. Thus, some niche populations died out, another one(s) would be newly generated. This reflected the competition and collaboration relations of individuals and populations. It is conducive to the formation of cluster effect and conducive to maintaining the population diversity.

4.5. It is suitable for largely distributed parallel computation

From the basic framework of BA, we can see that the overall framework is similar to the ideal social organization structure. Take the enterprise as an example; there are many different departments within a company, each of which consists of a department manager and some employees. All the department managers are the best performers in the department. Every employee in the department learns and competes with each other to become the manager. Moreover, the best one of all the department managers automatically becomes the general director of the enterprise. The candidates of department managers are not fixed but adjusted dynamically according to the performance of each employee's work. The initial setting of departments and positions is hetero-organizational, but the dynamic adjustment is self-organizational. This framework included both the learning competition and the competitive learning. On the one hand, it avoided the defects that PSO has only one population and is easy to fall into premature convergence. On the other hand, it avoided the problem that the members of the multi-agent system have no global information, and the communication between the members is limited in the same section. This kind of social organization structure with the synergistic coexistence of hetero-organization and self-organization, on the one hand, is conducive to balance the exploration and the exploitation; on the other hand, is very suitable for the organization and implementation of large-scale distributed parallel computing. It is the inevitable requirement and unavoidable trend in the upcoming quantum computer age.

5. Distributed power generation optimization configuration

For any new method, besides the theoretical analysis, it is also very important to see whether it can effectively deal with real problems. Therefore, the SBA proposed in this paper will be applied to the optimal configuration of distributed power supply to see how its optimization performance.

As an important component of the smart power grid, distributed generation (DG) has received increasing attention in recent years. Distributed power supply access to the distribution network will lead to the change of the power flow and the direction of each branch. Consequently, the system loss is not only related to the load size, but also with the siting and sizing of DG in the distribution network. Therefore, it is of great significance to study the rational planning of DG.

As with the above simulation experiments, the top four algorithms in the real-parameter single-objective optimization competition in CEC2017, i.e. jSO [12], LSHADE-cnEpSin [13], LSHADE-SPACMA [14] and DES [15], were put to use in this practical issue.

Since this paper is a new intelligent optimization method, so the background information related to DG will not be introduced in detail. We directly present the mathematical model of distributed power planning here. The objective function established here not only takes into account the interests of the distribution network operators but also takes into account the cost of electricity purchase and the system voltage stability. Consequently, this creates a multi-objective programming problem. We use the super efficiency data envelopment analysis (DEA) method to transform this multi-objective programming problem into a single-objective programming problem, and then the above-mentioned various IOAs are compared and experimented on this realistic optimization issues.

5.1. The mathematical model of distributed power planning

1. Sub-goal planning model

(1) The electricity purchasing cost of customer

$$\min f_i^1 = (P_w - P_{\sum \text{DG}} - \Delta P_{\text{loss}})T_{\max}C_e$$
$$= \left[P_w - \sum_{i=1}^{n} n_i S_{\text{DG}i} - (P_{\text{loss}} - P'_{\text{loss}})\right]$$
$$\times T_{\max}C_e, \quad (6)$$

wherein $T_{\max}$ represents the maximum load utilization time; P_w represents the total network capacity; $P_{\sum \text{DG}}$ represents the total active power output of the installed DGs; P_{loss} represents the network loss before optimization; P'_{loss} represents the wattful loss after access to distributed power supply;

(2) Investment cost of distributed power supply

$$\min f_i^2 = \sum_{i=1}^{N_d} x_i \left[\left(C_{r,i} + C_{D,i}\frac{r(r+1)^2}{(r+1)^n - 1}\right)\right] \times P_{\mathrm{DG}i}, \tag{7}$$

wherein n is the planning period; r is the fixed annual rate; $C_{D,i}$ represents the installation and purchase cost of DG for the ith node; $C_{r,i}$ represents the operating expense of DG; $P_{\mathrm{DG}i}$ represents the installed capacity for the ith node; x_i indicates whether the ith node is equipped with a distributed power supply and N_d represents the number of nodes with DG.

(3) Cost of network power loss
Combined with the node injection power of DG, the calculation formula of system loss is given as below based on Newton–Raphson power flow equations.

$$\min f_i^3 = \sum_{i}^{N_b}\sum_{j}^{N_b}[(a_{i,j}(P_iP_j + Q_iQ_j) + b_{i,j}(Q_iP_j - P_iQ_j))], \tag{8}$$

$$a_{i,j} = \frac{R_{ij}}{V_iV_j}\cos(\partial_i - \partial_j), b_{i,j} = \frac{X_{ij}}{V_iV_j} sin(\partial_i - \partial_j), \tag{9}$$

wherein N_b represents the total number of nodes in the system; $Z_{i,j} = R_{i,j} + jX_{i,j}$ is the impedance of branch i–j; $V_i\angle\partial_i$ is the voltage at node i; P_i and Q_i are respectively the input active and reactive power of node i;
After the DG was connected to the power distribution network, the system loss is related to the corresponding active and reactive power capacity of the DG installation node, that is

$$P_i = P_{\mathrm{DG}i} - P_{Di}, \tag{10}$$

$$Q_i = Q_{\mathrm{DG}i} - Q_{Di}, \tag{11}$$

wherein $P_{\mathrm{DG}i}$ and P_{Di} are respectively the input power and load power of node i. Substitute Equations (10) and (11) into Equation (8), we get the system active power loss after access to distributed power supply, which is shown as below:

$$\Delta P_{\mathrm{loss}} = \sum_{i}^{N_b}\sum_{j}^{N_b}[(a_{i,j}(P_i(P_{\mathrm{DG}i} - P_{Di}) + Q_i(Q_{\mathrm{DG}i} - Q_{Di})) + b_{i,j}(Q_i(P_{\mathrm{DG}i} - P_{Di}) - P_i(Q_{\mathrm{DG}i} - Q_{Di})))]. \tag{12}$$

So the cost of network power loss can be given as below:

$$f_i^3 = \Delta P_{\mathrm{loss}} T_{\max} C_{pu}, \tag{13}$$

wherein $T_{\max}$ and C_{pu} are respectively the maximum load utilization time and the unit electricity price.

(4) Voltage stability margin

The greater the line voltage stability margin, the better the stability of the system. The voltage stability margin of branch i–j can be given as below:

$$\max f_{ij}^4 = 4[(P_jX_{i,j} - Q_jR_{i,j})^2 + V_i^2(P_jR_{i,j} + Q_jX_{i,j})]/V_i^4. \tag{14}$$

2. Constraint conditions

(1) Node power balance constraint

$$\begin{cases} P_i - \sum\limits_{j=1}^{N} e_i(G_{ij}e_j - B_{ij}f_j) + f_i(G_{ij}f_j - B_{ij}e_j) = 0, \\ Q_i - \sum\limits_{j=1}^{N} f_i(G_{ij}e_j - B_{ij}f_j) - e_i(G_{ij}f_j + B_{ij}e_j) = 0, \end{cases} \tag{15}$$

wherein N is the number of nodes in the DG system; P_i and Q_i are respectively the input active and reactive power of node i; e_i and f_i are respectively the real part and imaginary part of the voltage of node i; G_{ij} and B_{ij} are respectively the electric conductance and electrical susceptance between node i and node j.

(2) nodes voltage constraint

$$U_{i\min} \le U_i \le U_{i\max}, \tag{16}$$

wherein $U_{i\min}$ and $U_{i\max}$ are respectively the upper limit and lower limit of the voltage of node i.

(3) The transmission power limitation of the transmission line

$$P_{ij} < P_{ij}^{\max} \tag{17}$$

Wherein P_{ij} is the transmission power of node i to node j.

(4) The total capacity limitation of DG installation

$$\sum_{i=1}^{N} P_{\mathrm{DG}i} < \frac{P}{4}, \tag{18}$$

wherein P is the total capacity of the system load.

3. Total-goal planning model

Now we use the super efficiency DEA method to transform the multi-objective planning model into single-objective planning model. Let the weight coefficient vector $[\alpha_i, \beta_i]$ of the objective function be the decision units. α_i is the weight coefficient vector of the input, β_i is the weight coefficient vector of the output. The minimized target value and the maximized target value based on this weight vector respectively serve as the input and output of the decision units.

All the sub-goals, including the power purchase costs, the comprehensive investment costs, the costs of network power loss, and the voltage stability margin, are normalized. That is $f^1/minf^1, f^2/minf^2, f^3/minf^3$, $(\min f^4 - f^4)/(-f^4)$. Since the value of the super efficiency output variable must be the maximum value, so the voltage stability margin is used as the output of the decision units. According to the characteristics of super efficiency DEA, if some decision variable has the highest efficiency, then the corresponding decision-making unit will be more effective. Since the total goal is to minimize the overall cost, the operating costs are maximized at the time of normalization, so we get the highest efficiency linear weighting factor, α_{i1}, α_{i2}, α_{i3}, β_{i1}. Therefore, the single-objective function based on linear weighting method can be shown as follow:

$$\min F = \alpha_{i1}\frac{f^1}{\min f^1} + \alpha_{i2}\frac{f^2}{\min f^2} + \alpha_{i3}\frac{f^3}{\min f^3} + \beta_{i1}\frac{\min f^4 - f^4}{-f^4}. \tag{19}$$

5.2. The IEEE-69 node distribution network system optimization

Here, take the IEEE-69 node distribution network system as an example. The DGs are processed as negative PQ nodes, so the power factor is set as 0.9. Let $P_s = 10$kW, if $X_i = 0$, then it indicate that there is no DG installed at the load node i; if $X_i \neq 0$, then it indicates that there is a DG installed at the load node i, and its installed capacity is $10X_i$kW, $X_i \in [0, M]$. The maximum value for this node is $M = \text{DG}(\max(i))/P_s$; the unit cost of network power loss is set as $C = \$0.5/\text{kW}h$; The system runs time in a year is set as 8700 hours; the unit installation fee of DG is set as $C_{D,i} = \$9000/\text{kW}$; the operation and maintenance cost is set as $C_{r,i} = \$2000/\text{kW}$; r is set as 0.068; the planning period is set as 20 years.

In this experiment, we choose 38 groups of weight coefficients, whose range is [0.1, 0.8]. The super efficiency DEA method is used to evaluate the 38 weighting coefficient combination schemes. The combination scheme with highest DEA value is the most effective one. The linear weighting method is used to transform the multi-objective optimization problem into a single-objective optimization problem. The above-mentioned various intelligent algorithms are used to solve this single-objective optimization problem.

As can be seen from Table 3, the super efficiency DEA values based on the weight coefficient combination scheme of decision units 12, 28 and 43 are all greater than 1. The relative efficiency of the decision unit 12 is higher, so the combination scheme of the decision unit 12 is adopted to serve as the weight coefficient of sub-objective function. Under the premise of satisfying the various constraints, the optimization results obtained by applying the intelligent algorithms mentioned above are shown in Table 4.

As can be seen from Table 3, the power purchase cost is reduced after the DGs were connected to the power grid. When the DGs with SBA were connected to the power grid, the system power loss is reduced from the original 336.75–183.22 kW, the power purchase cost was reduced by 42.95%. When the DGs with DES were connected to the power grid, the investment cost is slightly less than that when the DGs with SBA were connected to the power grid, but the system power loss and the power purchase cost are far more than the latter. When the DGs with LSHADE-cnEpSin were connected to the power grid, the power purchase cost is slightly less than that when the DGs with SBA were connected to the power grid, but the system power loss and the investment cost are far more than the latter. The results of these algorithms have their advantages, but overall, the search results SBA is better than other algorithms.

As can be seen from Figure 6, as long as access to DG, regardless of which DG configuration programme was adopted, the lowest voltage of each node in the power grid has been improved. Before connecting the distributed power supply, the minimum voltage was 0.9201 p.u. If the DGs with jSO were connected to the power grid, the minimum voltage was 0.9710 p.u. When the DES-DG configuration programme was adopted, the minimum voltage was 0.9508 p.u. The minimum voltages of the LSHADE-cnEpSin-DG configuration programme and the LSHADE-SPACMA-DG configuration programme were respectively 0.9522 and 0.9510 p.u. If the DGs with SBA were connected to the

Table 3. Combinational evaluation results of typical weight coefficient.

Decision unit	$[\alpha_i, \beta_i]$	f^1	f^2	f^3	f^4	DEA value
8	[0.1 0.2 0.2 0.5]	0.228	0.056	0.71	1.58	0.86
12	[0.2 0.2 0.2 0.4]	0.204	0.054	0.70	1.44	1.21
13	[0.1 0.3 0.2 0.4]	0.215	0.06	0.62	1.49	0.95
28	[0.3 0.1 0.1 0.5]	0.220	0.058	0.66	1.60	1.12
43	[0.5 0.1 0.2 0.2]	0.219	0.054	0.67	1.53	1.18
59	[0.2 0.2 0.1 0.5]	0.238	0.059	0.70	1.65	0.97

Table 4. The optimization results of different intelligent algorithms.

Algorithm name	DG	Power loss (kW)	Investment cost	Purchase cost
No DG	–	336.75	–	677.53
jSO	16(160)42(220)	201.24	198.18	410.28
DES	16(155)42(220)	254.08	140.37	453.18
SBA	16(160)42(220)	183.22	145.88	386.52
LSHADE-cnEpSin	16(160)42(220)	220.86	162.34	376.46
LSHADE-SPACMA	16(155)42(220)	243.05	160.09	397.76

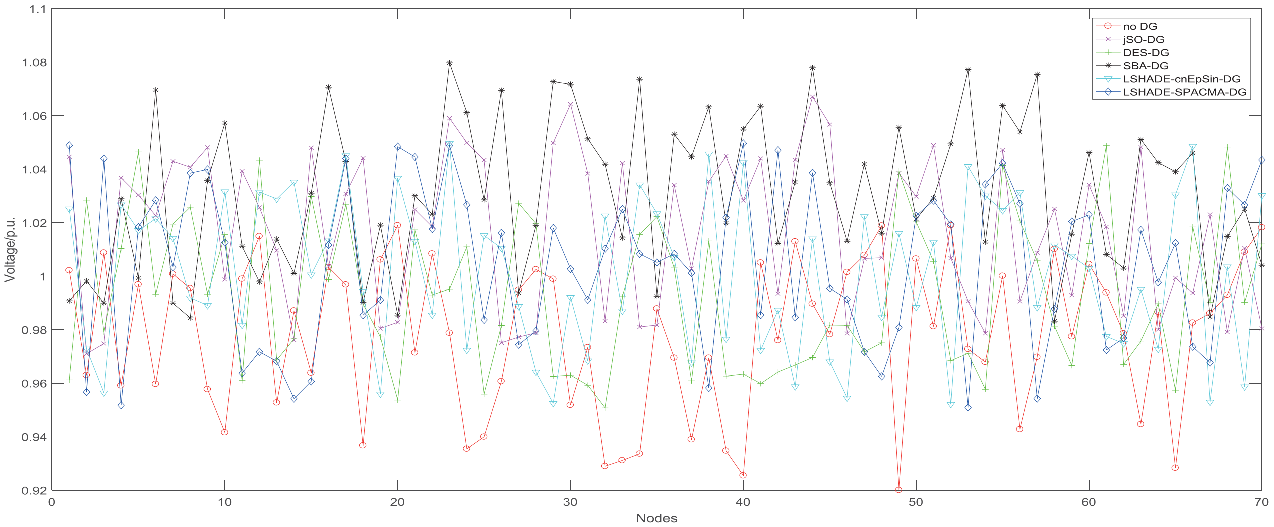

Figure 6. Voltage amplitude of IEEE-69 nodes with different DG configuration.

power grid, the minimum voltage reached 0.9832 p.u., the voltage of distribution network system was generally significantly improved, and the average voltage reached 1.0309 p.u.

6. Conclusion and outlook

The core innovation and contribution of this work has been made clear in the abstract. That is, the “intelligence” of the method proposed here is based on organizing tactics rather than probability rules. It broke the “hidden rules” that relying on the probability rules to balance the exploration and exploitation. As for the specific technical details, which have been adequately explained in the technical analyses section. In addition, although the different issues in different areas can be transformed into different optimization problems, and eventually which will be presented in different models (including physical models, conceptual models and mathematical models), they all can try to be solved by the algorithm model proposed in this paper. Essentially speaking, it is not just an algorithm, but a kind of methodology and/or a mode of thinking, which is between engineering technology and cognitive philosophy. So, it can be applied to many issues in science, engineering and management, like function optimization, combinatorial optimization, production scheduling, automatic control, robotics, image processing, artificial life, genetic programming, machine learning and data mining. Subject to file size limit, more simulation experiments on benchmarking functions were not included in this file. The goal of this paper is to provide an idea or a way of thinking, according to which readers can design their own optimization/search methods to adapt to specific problems. Of course, you can also download the algorithm code to make changes and adjustments. (http://www.mediafire.com/folder/7m1bfpflz4byl/Relevant_Data).

Similar to other IOAs, this method proposed here, it is easy to achieve through programming. And the actual effect is not bad as well. However, there are still two issues need further study. First, it lacks rigorous and comprehensive mathematical basis. For some problems, it is still necessary for more scientific research based on mathematical analysis. Like the individual density, the optimization probability, the convergence, the complexity and the precision, etc. Second, it is proposed mainly for the single-objective optimization problems with weak constraints. For multi-objective problems with complex and harsh restrictions in the field of science, engineering and management, how to coordinate these conflicting objectives and deal with those complex and harsh constraints, there is no discussion. It is also the focus of the next stage of the study.

Note

1. What is benchmarking, https://en.wikipedia.org/wiki/Benchmarking.

Disclosure statement

No potential conflict of interest was reported by the author.

Funding

This research is supported by the research fund from “Collaborative innovation centre for Transformation and Upgrading of Micro, Small and Medium Enterprises, Zhejiang University of Technology” [grant number 16JDGH048], “Zhejiang Provincial Key Research Base of Philosophy and Social Sciences-Research Centre for Technology Innovation and Enterprise Internationalization”.

ORCID

As Xie http://orcid.org/0000-0003-2544-8277

References

[1] Casanovas JMM, Montserrat X. A New Minkowski distance based on induced aggregation operators. Int J Comput Intel Syst. 2012;2011(2):123–133.

[2] Niedermeier R, Sanders P. On the Manhattan-distance between points on space-filling mesh-indexings. Technical Report IB 18/96, Karlsruhe University, Dept. of Computer Science; 1996.

[3] Krivulin N. Algebraic solutions to multidimensional minimax location problems with Chebyshev distance. WSEAS Trans Math. 2012;10(6):191–200.

[4] Hauke J, Kossowski T. Comparison of values of Pearson's and Spearman's correlation coefficients on the same sets of data. Quaest Geographicae. 2011;30(2): 87–93.

[5] Godden JW, Xue L, Bajorath J. Combinatorial preferences affect molecular similarity/diversity calculations using binary fingerprints and Tanimoto coefficients. J Chem Inf Comput Sci. 2000;40(1):163–166.

[6] Abdi H, coefficient TKrc. The Kendall rank correlation coefficient (Encyclopedia of Measurement and Statistics). Thousand Oaks (CA): Sage; 2007.

[7] Lavoie T, Merlo E. An accurate estimation of the Levenshtein distance using metric trees and Manhattan distance. In: 2012 6th international workshop on software clones (IWSC). 2012. p. 1–7.

[8] Rajabzadeh M, Tabibian S, Akbari A, et al. Improved dynamic match phone lattice search using Viterbi scores and Jaro Winkler distance for keyword spotting system. In: CSI International Symposium on Artificial Intelligence and Signal Processing; 2012. p. 423–427.

[9] Niwattanakul S, Singthongchai J, Naenudorn E, et al. Using of Jaccard coefficient for keywords similarity. Lecture Notes Eng Comput Sci. 2013;2202(1):13–15.

[10] Helwig S, Wanka R. Theoretical analysis of initial particle Swarm behavior. In: International conference on parallel problem solving from nature; 2008. p. 889–898.

[11] Srinivasa KG, Venugopal KR, Patnaik LM. A self-adaptive migration model genetic algorithm for data mining applications. (in English), Inf Sci (Ny). 2007; 177(20):4295–4313.

[12] Brest J, Maučec MS, Bošković B. Single objective real-parameter optimization: algorithm jSO. In: IEEE congress on evolutionary computation (CEC). IEEE; 2017. p. 1311–1318.

[13] Awad NH, Ali MZ, Suganthan PN. Ensemble sinusoidal differential covariance matrix adaptation with Euclidean neighborhood for solving CEC 2017 benchmark problems. In: 2017 IEEE congress on evolutionary computation (CEC). IEEE; 2017. p. 372–379.

[14] Mohamed AW, Hadi AA, Fattouh AM, et al. LSHADE with semi-parameter adaptation hybrid with CMA-ES for solving CEC 2017 benchmark problems. In: 2017 IEEE congress on evolutionary computation (CEC). IEEE; 2017. p. 145–152.

[15] Jagodziński D, Arabas J. A differential evolution strategy. In: 2017 IEEE congress on evolutionary computation (CEC). IEEE; 2017. p. 1872–1876.