

Research Article

An Improved Nondominated Sorting Genetic Algorithm for Multiobjective Problem

Ruihua Wang

Management Engineering Department of Zhengzhou University, Zhengzhou, Henan, China

Correspondence should be addressed to Ruihua Wang; wangrh0304@zzu.edu.cn

Received 18 July 2016; Revised 10 October 2016; Accepted 27 November 2016

Academic Editor: Ziran Wu

Copyright © 2016 Ruihua Wang. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In this paper, an improved NSGA2 algorithm is proposed, which is used to solve the multiobjective problem. For the original NSGA2 algorithm, the paper made one improvement: joining the local search strategy into the NSGA2 algorithm. After each iteration calculation of the NSGA2 algorithm, a kind of local search strategy is performed in the Pareto optimal set to search better solutions, such that the NSGA2 algorithm can gain a better local search ability which is helpful to the optimization process. Finally, the proposed modified NSGA2 algorithm (MNSGA2) is simulated in the two classic multiobjective problems which is called KUR problem and ZDT3 problem. The calculation results show the modified NSGA2 outperforms the original NSGA2, which indicates that the improvement strategy is helpful to improve the algorithm.

1. Introduction

The maximization of profit is always the pursuit of mankind, no matter in which field, especially the development of the energy issues and the population problems in recent years. For many social and economic problems, it is needed to find an optimal solution to reduce the input and increase the output. As a result, the optimization problem is more and more concerned by scholars. In recent years, many scholars have put forward a lot of optimization problem solving methods from many aspects. Many of them are to find the value of the control variables to minimize the single objective. However, many real world decision problems not only need to consider just one factor, but a variety of factors to get a decision. In other words, a lot of practical problems are the multiobjective programming problem. Therefore, in recent years, scholars have conducted in-depth research on multiobjective planning and got a lot of meaningful research results, which lay the foundation for the future research of multiobjective problem. In recent years, the heuristic search algorithm has a great development. Some researchers introduced and developed many heuristic search algorithm according to the behavior of the social animal in the nature and the characteristic of the optimization problems to be optimized like genetic

algorithm [1], particle swarm optimization algorithm [2], differential evolution algorithm [3], shuffle frog leap algorithm [4], artificial bee colony algorithm [5], and so on. In all of these heuristic search algorithms, the genetic algorithm is very representative. According to its good calculation performance and genetic idea, the genetic algorithm is largely used in many aspects [6–10]. The first genetic algorithm is used to deal with the single objective optimization problem, while as the time passes by, many researchers introduce a lot of multiobjective optimization algorithms according to the idea of the genetic algorithm. These multiobjective optimization algorithms contain vector evaluation genetic algorithm [11], multiobjective genetic algorithm [12, 13], strength Pareto evolutionary algorithm [14, 15], and nondominating sorting genetic algorithm [16]. After several years later, in order to modify each of these multiobjective genetic algorithms calculating performance, a lot of researchers also introduce the improved version of these multiobjective genetic algorithms like strength Pareto evolutionary algorithm 2 [17] which is the improved version of the strength Pareto evolutionary algorithm, strength Pareto evolutionary algorithm [18] which is also the improved version of the strength Pareto evolutionary algorithm, nondominated sorting genetic algorithm 2 which is the improved version of the nondominating sorting genetic

algorithm [19] and so on. In this way, many multiobjective genetic algorithms are proposed in these years, which made a great contribution to these multiobjective optimization problems.

Nondominated sorting genetic algorithm 2 (which we call it as the NSGA2 algorithm in the rest of this paper) is the improved version of the nondominating sorting genetic algorithm which is first proposed by Deb in 2001. During these years development and use, the NSGA2 algorithm has been proved as a very good multiobjective genetic algorithm which has been successfully used in many multiobjective optimization problems [20–26] because the calculation performance of one algorithm varies very much when dealing with different optimization problems. Beside this, the algorithm's calculation performance also depends on the optimizing strategy inside of the algorithm itself. So a lot of researchers also proposed some improvement strategies into the NSGA2 algorithm to deal with the shortcomings when optimizing some optimization problems. In this paper, in order to overcome the poor local search ability problem in the NSGA2 algorithm, a local search strategy is joined into the NSGA2 algorithm to enhance the local search ability. The general idea of the proposed improvement strategy is to search locally around the optimization solution found so far to find a better solution, such that the algorithm can get a better local search ability and can make a more detailed search. In order to see whether the local search ability can improve the calculation performance of the original NSGA2 algorithm or not, this paper also introduce two classic multiobjective optimization problems to let the modified NSGA2 algorithm to simulate. At last, the outcome of the simulation works is compared to the original NSGA2 algorithm. The comparison shows the local search ability can really improve the calculation performance of the NSGA2 algorithm.

The rest of this paper is organized as follows: Section 2 describes the mathematical model of the multiobjective problem. Section 3 shows the principles and steps of the NSGA2 algorithm, where we introduce three aspects to describe the original NSGA2 algorithm: fast nondominate sorting, crowding distance function, and environmental selection based on PCD. At the end of Section 3, the basic steps of the original NSGA2 algorithm are introduced to the readers. The simulation works and the results analysis of different multiobjective optimization problem are given in Section 4, where we introduced two classic multiobjective optimization problems. Section 5 draws the conclusion of this paper.

2. Multiobjective Problem

A multiobjective problem can be seen as a problem which should optimize at least two objectives while it satisfies a lot of constrains. A multiobjective optimization problem can be mathematically described as

$$\begin{aligned} \text{Min } & F = \{f_1(X), f_2(X), \dots, f_k(X)\} \\ \text{S.t. } & g_i(X) \leq 0, \quad i = 1, 2, \dots, n \\ & X = (x_1, x_2, \dots, x_m) \in R^m, \end{aligned} \quad (1)$$

where $f_i(x)$ represents the i -th objective function, X represents the control variables vector, n represents the number of constrains, and k is the total number of the objective to be optimized.

But the purpose of the decision scheme of the multiobjective optimization problem is not to make each objective to be optimized optimal, but to provide a series of solutions for the decision makers. In accordance with their own decision intention, the decision makers choose the decision scheme to make decisions. A series of optimal solution set of multiobjective optimization problems found in the end is called the Pareto optimal solution set. The mathematical explanation of the Pareto optimal solution set can be shown as follows:

$$P = \{x^* \in \Omega \mid \neg \exists x_k \in \Omega : x_k < x^*\}, \quad (2)$$

where the x^* represents the Pareto optimal solutions, whose mathematical expression can be shown as follows:

$$\neg \exists x_j \in \Omega : x^* < x_j \quad (3)$$

In (3), the expression $x_1 < x_2$ represents the individual x_1 dominating the individual x_2 . For a minimization optimization problem, a solution dominating another solution in a multiobjective can be mathematically is expressed as follows:

$$\begin{aligned} \forall i \in \{1, 2, \dots, N\}: & f_i(x_1) \leq f_i(x_2) \\ \exists j \in \{1, 2, \dots, N\}: & f_j(x_1) < f_j(x_2). \end{aligned} \quad (4)$$

From (4) we can conclude that if a solution x_1 's each objective value in a multiobjective problem is not bigger than a x_2 's corresponding objective value, and at least one objective can be found the x_1 's value is smaller than the x_2 's, then we can say that the solution x_1 dominates x_2 .

The multiobjective optimization problem is exactly according to seeking the nondominating solution and use some kind of evolutionary strategy to evolve the evolutionary population to finally find the Pareto optimization set of the multiobjective optimization problem to be optimized. In order to more intuitively demonstrate the value of each objective value in the Pareto optimal solution set and its distribution in the solution space, the Pareto front is introduced. The Pareto front is a line or a curved surface formed by the objective values of each solution in the Pareto optimal set whose mathematical expression is as follows:

$$\text{PF} = \{F(x) = (f_1(x), f_2(x), \dots, f_m(x)) \mid x \in P\}. \quad (5)$$

3. Improved NSGA2 Algorithm

3.1. NSGA2 Algorithm. The NSGA2 algorithm is the modified version of NSGA by Deb and other researchers; after several years of development and use, the NSGA2 algorithm is seen as a very good and mature multiobjective optimization algorithm at present. In NSGA2 algorithm, the crowding distance function is used to calculate each individual's fitness value in the algorithm. According to the fast nondominating sorting idea and environmental selection of PCD, the algorithm

uses the two strategies to evolve the evolutionary population to optimize the corresponding problem. In the rest of this paper, three important aspects of this algorithm is introduced to study the original NSGA2: fast nondominating sorting, which is used to evaluate each individual in the NSGA2 algorithm; crowding distance function which is used to calculate each individual's fitness value. Environmental selection based on PCD is used to select individuals to the next generation of the evolutionary population in the algorithm.

3.1.1. Fast Nondominated Sorting (FNS). In the evolutionary multiobjective optimization algorithm, the evolution of the population is to eliminate the inferior solution and select the excellent solutions by a certain strategy, such that the inferior solution is eliminated and the excellent solution is retained in the next generation of the evolutionary population so as to find the final excellent solution set through the step by step evolution. In NSGA2, the fast nondominated sorting strategy is used to evaluate each individual's dominating status in the evolutionary population, and according to the dominating status of each individual, the evolutionary population is divided into different subgroups, which is a necessary way to calculate the fitness of each individual in the crowd and to evaluate the quality of each individual. In the fast nondominated sorting strategy, the first step of this strategy is to calculate each individual's (e.g., individual i) corresponding dominating solutions number n_i and the individuals number n_j who dominate the individual i in the evolutionary population. When an individual's corresponding n_i equals 0, then send it to the first level of the nondominating individuals. In the fast nondominated sorting strategy, it declares that when the nondominating individuals' dominating grade in a level is determined already, the level of all individuals in the control of other individuals is no longer considered. Considering this, for an individual i 's dominating individual set s_i , each of the individual's (let us say individual j) corresponding n_j in the i 's dominating individual set s_i should subtract 1. After the subtraction, when $n_j = 0$, the individual j will be put into the second nondominating level. The rest of the individuals in the evolutionary population use the same way to find which nondominating level they belong to. After each of the individuals in the evolutionary population finds its own nondominating level, the lowest level's corresponding individuals are close to the real Pareto front of the multiobjective optimization problem to be optimized. So the lowest level's corresponding individuals are more superior than the other individuals in the evolutionary population of the NSGA2 algorithm. And the NSGA2 algorithm exactly uses a larger probability to select the lowest level compared to the high level into the next generation of the evolutionary population, such that the NSGA2 algorithm can evolve to the direction of the real Pareto optimization set of the multiobjective optimization problem to be optimized.

3.1.2. Crowding Distance Function. The purpose of the multiobjective optimization algorithm is not just only to find the Pareto optimal set which is close to the real Pareto optimal set, but also to find the solutions in the Pareto optimal set have a very good distribution, such that better and more reasonable

solutions can be provided to the decision maker. The purpose of the multiobjective optimization algorithm is to find a series of uniform distribution of the solution for decision makers to choose. In order to obtain the distribution of the solution set well, various evolutionary algorithms have used a lot of different kinds of treatment measures. Most algorithms evaluate the distribution of each individual by constructing a function that considers the distribution of individual distribution. In the NSGA2 algorithm, by introducing the crowding distance function, the congestion degree of each individual and other individuals is calculated, and the local aggregation of the solution set is avoided by a certain strategy. The crowding distance function of the NSGA2 algorithm can be expressed as

$$\text{PCD}(k) = \frac{1}{n} \sum_{j=1}^n \frac{f_i(k+1) - f_i(k-1)}{f_i^{\max} - f_i^{\min}}, \quad (6)$$

where $\text{PCD}(k)$ represents the k -th individual's crowding distance value, $f_i(k+1)$ represents the $k+1$ -th individual's the i -th objective function value, and the f_i^{\max} and the f_i^{\min} separately represents the i -th objective function value's maximum and minimum value in the solution set. The mathematical expression of the crowding distance function can be seen that the PCD is the mean value of the Euclidean distance and each objective function of the individual.

On calculating the PCD value of each individual, we need to notice that the boundary point's corresponding crowding distance function value of each objective function is set as 1. From the PCD mathematical expression we can see that the value of each individual's PCD value ranges from 0 to 1. The purpose of the introduction of PCD concept into the NSGA2 algorithm is to evaluate the congestion of each other in the same nondominated level. A bigger PCD value can tell the NSGA2 algorithm its corresponding individual is more sparse with the surrounding individuals. The individuals who have a small PCD value have a very good distribution in the evolutionary population in the NSGA2 algorithm. According to the introduction of the PCD function into the NSGA2 algorithm, the individuals in the Pareto front will have a great distribution in the Pareto optimal set, such that it can guarantee the evenness and diversity of the evolutionary population in the NSGA2 algorithm.

3.1.3. Environmental Selection Based on PCD. In multiobjective evolution algorithm, the individuals evolve towards the optimal solution set depending on a certain evolution strategy. In the NSGA2 algorithm, the population evolutionary strategy can be concluded as according to fast nondominated sorting strategy to determine each individual's nondominated level. By comparing the PCD value and each individual's nondominated level to determine the elite individuals from the current population and former population and save the elite individuals to the next generation, and then the evolution of the dominating individuals in the NSGA2 algorithm is completed. In the NSGA2 algorithm, the selection of the individuals to the next generation for evolution follows the following two rules: (1) if the nondominating level of an individual p is higher than q , then the former individual is chosen to the next generation; (2) if the nondominating level

of an individual p is equal to q , in this situation, the PCD value of these two individuals should be compared. In the NSGA2 algorithm, the one with a smaller PCD value will be selected to the next generation evolution population. In the NSGA2 algorithm, $P < Q$ is defined as the following mathematical form to express the environmental selection strategy:

$$p < q \begin{cases} p_{\text{rank}} < q_{\text{rank}} \\ \text{or} \\ p_{\text{rank}} < q_{\text{rank}}, \text{PCD}(p) > \text{PCD}(q). \end{cases} \quad (7)$$

Through the NSGA2 algorithm's environment selection strategy, we can see when two individuals' nondominating level is different, in this situation the individual with smaller nondominating level, such that the NSGA2 algorithm can ensure the elite population; on the contrary, when two individuals' nondominating level is the same, choosing the individual with larger crowding distance value can ensure the diversity of population, such that the evolution speed of the NSGA2 algorithm can be accelerated.

3.1.4. NSGA2 Algorithm Calculation Steps

Step 1. Randomly initialize the initial population and each parameter's value in the NSGA2 algorithm, and then set the iteration number as 1; then start the algorithm's iteration calculation.

Step 2. Determine if the iteration number right now attains the biggest iteration times set at Step 1; if the former is bigger than the latter, then end the iteration calculation and output the Pareto optimal set; if the former is not bigger than the latter, then continue the iteration calculation until the two numbers are equal to each other.

Step 3. Use the fast nondominated sorting strategies to perform the evolutionary population. And then perform the genetic operation like selection, crossover, and mutation to the evolutionary population to form the new generation which is called Q in this paper.

Step 4. Combine the P population and the Q population as the new mixture population which is called R , and perform the fast nondominated sorting strategy to each individual in the R population; then calculate the crowding distance of each individual in the R population and sort them.

Step 5. Retain the elite individuals as the new generation of the evolutionary population which we call it as P_{t+1} in our paper.

Step 6. Add 1 to the iteration number $i = i + 1$, and one iteration calculation is done; then go to Step 2 and continue the rest of the iteration calculation.

3.2. Improvement Strategy. Although the NSGA2 algorithm is largely and successfully used in many fields as an excellent algorithm, it still has many drawbacks to be settled. In recent years, with the heuristic search algorithm being largely used

in many aspects, many researchers have found many evolutionary algorithms having poor local search ability problem which is very important for the algorithm's calculation performance. The algorithm with a better local search ability can use a more precise way to seek the optimal solution of the optimization problem to be settled. A better individual can be found by searching around a nondominated individual. So, searching around the nondominated individuals for better individual is needed. The studies show that the local search ability of the original NSGA2 algorithm is poor; especially when it is close to the Pareto frontier at the end of the evolutionary process, the evolutionary efficiency is greatly reduced. To enhance the local search ability, the local search strategy is added into the NSGA2 algorithm after the next evolutionary population is generated. The local search strategy is exploring an individual of the evolutionary population to determine whether a better performing solution can be found. There are two types of the mentioned solution with better performance: (1) the new individual dominates the original; (2) the new and the original individuals do not dominate each other, but the fitness of the new individual is better than the original one. So for an individual x , search for a new individual x' in its neighborhood, if x' which is a new individual searched out from individual x 's neighborhood dominates x or its corresponding fitness is better, replace the latter; otherwise proceed with the local search.

The specific steps of the local search strategy is as follows.

Step 1. Set the local search iteration time $i = 1$ and start local search calculating.

Step 2. For each solution x' in the Pareto optimal set, generate a new feasible solution x around it, and calculate its fitness.

Step 3. If x' dominates x or $\text{fitness}(x') < \text{fitness}(x)$, then replace x with x' , if not, maintain the x unchanged.

Step 4. $i = i + 1$.

Step 5. If $i = Nc$, stop local search calculation and output the Pareto optimal set to the NSGA2 algorithm; if not, continue the local search calculation.

4. Simulation Works

4.1. Parameter Selection. There are a lot of parameters in NSGA2 algorithms such as population size, iteration times, cross probability, and mutation probability. The value of these parameters has a great influence on the NSGA2 algorithm's calculation performance. So when we use the NSGA2 algorithm, the parameter value selection becomes a very important problem. For some complicated multiobjective problem, a small population size cannot help NSGA2 algorithm achieve a very ideal Pareto optimal set. A very large population size can slow down the calculation speed of the NSGA2 algorithm, although it can get a good Pareto optimal set. So how to select the population size value according to the complexity of the multiobjective problem is a problem to be further researched. Right now, the population size value is selected by the experience of the users. The researcher adjusts

the population value step by step, until the Pareto optimal set is stable. The NSGA2 algorithm is a variation of the GA algorithm; they have many things in common although the first one is multiobjective algorithm while the second one is single objective algorithm. So the NSGA2 is also uses a iterative way to calculate the proposed problem. Then a very important parameter must be valued first which is iteration times. A very large iteration time can help the NSGA2 algorithm gain a very good Pareto optimal set, but it can also enlarge the calculation time which is bad to the algorithm users. While a small iteration times cannot guarantee that the NSGA2 algorithm achieves an ideal Pareto optimal set it can help the NSGA2 algorithm calculate very fast. Many researchers now also use a step by step method to adjust the iteration time's value to select it. If the Pareto optimal set will not change a lot in the NSGA2 algorithm, the iteration value can be used as the final value of this parameter of the NSGA2 algorithm. The genetic manipulation includes selection, crossover, and mutation operations [6, 7] which is very important in the NSGA2 algorithm. In the NSGA2 algorithm the genetic operations are the main way of evolution which directly determine the outcome performance. In the genetic operations, there are a lot of parameters which can change the genetic evolution performance of the NSGA2 algorithm. So the genetic parameters is also very important and is key to the NSGA2 algorithm's calculation performance. In the NSGA2 algorithm, the tournament selection is used in the selection process, whose parameter is called tournament value. The tournament value determines the selection pressure; the bigger the value is, the bigger the selection pressure is. In the NSGA2 algorithm, the tournament value is suggested to set 2. The NSGA2 algorithm crossover process uses a kind of simulated binary crossover method to cross-over two individuals. There is also a parameter in the crossover process which is called crossover distribution parameter. The value of the crossover distribution parameter determines the probability of the likelihood of offspring and father individual. The bigger the crossover distribution parameter, the bigger the probability of the likelihood of offspring and father individual. This paper sets the value of the crossover distribution parameter as 20. In the NSGA2 algorithm, a kind of polynomial mutation is used in the mutation process, whose parameter inside is called mutation distribution parameter. The mutation distribution parameter is more or less the same as the crossover distribution parameter. In our paper, the value of the mutation distribution parameter is set as 20. As the local search strategy is added in the NSGA2 algorithm, the local search iteration time should be valued. In this paper, the value of the local search iteration time is set as 10.

4.2. Simulation. In order to research the improvement effect to NSGA2, this paper selects some classic multiobjective problem for simulation. The first problem is called KUR which is MISA problem. The mathematical formulation is as follows:

$$f_1 = -10 \exp(-0.2 * \text{sqrt}(x^2 + y^2)) - 10 \exp(-0.2 * \text{sqrt}(y^2 + z^2))$$

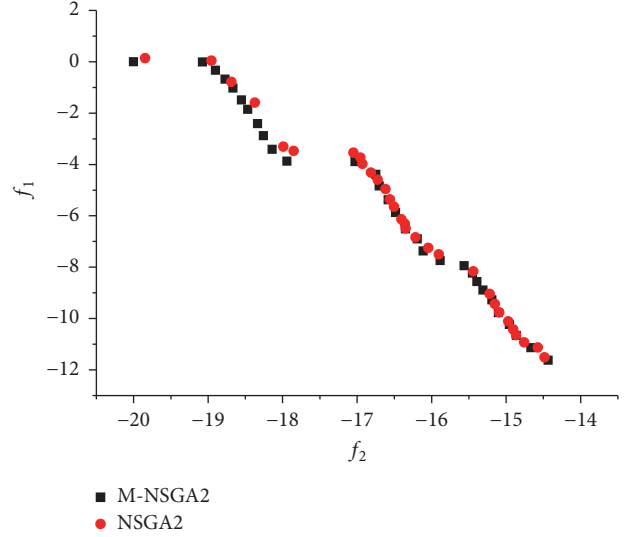


FIGURE 1: Two Pareto fronts of KUR problem.

$$f_2 = |x|^{0.8} + 5 * \sin(x^3) + |y|^{0.8} + 5 * \sin(y^3) + |z|^{0.8} + 5 * \sin(z^3), \quad (8)$$

where x , y , and z are the control variables, and their domains are all $[-5, 5]$.

Figure 1 shows the NSGA2's and the I-NSGA2's Pareto fronts of this problem. From Figure 1 we can see the difference of the two algorithms Pareto fronts: the distribution of I-NSGA2's Pareto fronts is more uniform than the NSGA2 algorithm. Furthermore, the I-NSGA2's Pareto fronts is closer to the coordinate origin. So the I-NSGA2 can get a better Pareto optimal solution than the original one.

To further study and explain the difference of the two outcomes, we can conclude that just because of the local search strategy, when the NSGA2 algorithm finds a Pareto set solution in each iterative calculation, the local search can begin a deeper and more accurate search for better solution, and this is why the M-NSGA2's Pareto set is more close to the coordinate origin point than the original algorithm.

The second simulation case is also a classic multiobjective problem called ZDT3 problem. The ZDT3 problem can be mathematically expressed as follows:

$$\begin{aligned} \min \quad & f_1(x) = x_1 \\ \min \quad & f_2(x) = g \left[1 - \sqrt{\frac{f_1}{g}} - \frac{f_1}{g} \sin(10\pi f_1) \right] \\ & g = 1 + 9 \sum_{i=2}^n \frac{x_i}{n-1}, \end{aligned} \quad (9)$$

where $0 < x_i < 1$, and $n = 30$.

The comparison of the two different NSGA2 algorithms in the ZDT3 problems is shown in Figure 2. From Figure 2 we can clearly see the difference of the two Pareto fronts.

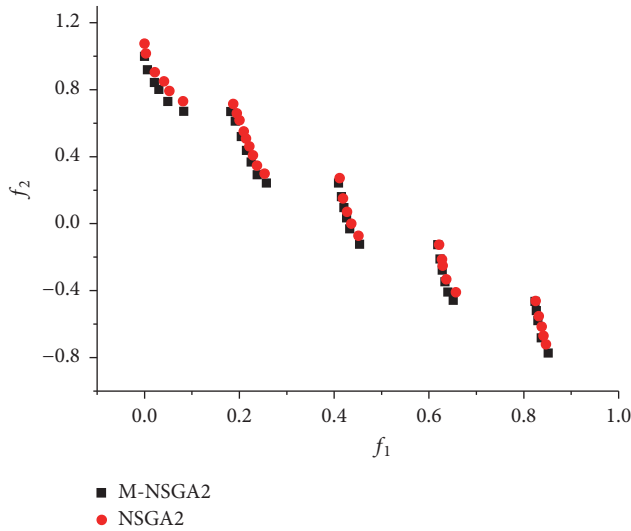


FIGURE 2: Two Pareto fronts of ZDT3 problem.

The Pareto front of the M-NSGA2 algorithm can conclude the second one, which means the first Pareto front is closer to the real Pareto front of the ZDT3 problem. By comparing this case's results with the first case, we also can see the more complex the problem is, the more the superiority of the M-NSGA2 will be obvious. Seeing this comparison we can conclude that the modified strategy in M-NSGA2 algorithm can improve the optimization effect of the NSGA2 algorithm.

5. Conclusion

This paper introduced a new improvement strategy to NSGA2 algorithm, which uses a local search way to search a better solution around the optimized solution of NSGA2 algorithm. In this paper, we firstly researched the basic NSGA2 algorithm, from three different but important parts: the fast nondominating sorting, the crowding distance function, and the environmental selection based on PCD. After that, we introduced the NSGA2 algorithm calculation steps for the multiobjective problem. After the introduction of NSGA2 algorithm, we also introduced the local search improvement for the basic NSGA2 problem and formed the M-NSGA2 algorithm. In the end, in order to test effect of the improvement in NSGA2 algorithm, we use two multiobjective problems to simulate the two algorithms. From the comparison figures we can see the difference of the two algorithms calculation results.

Disclosure

Ruihua Wang is a Ph.D. candidate in School of Traffic and Transportation of Beijing Jiaotong University and a lecturer in Management Engineering Department of Zhengzhou University.

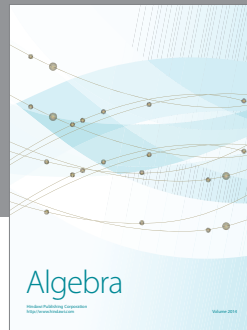
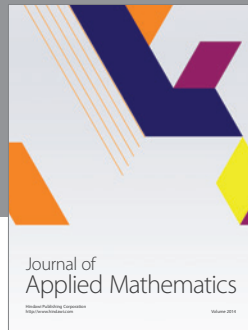
Competing Interests

The author declares that they have no competing interests.

References

- [1] C. Razvan, G. Lucian, and M. Ioan, "An OOP MATLAB extensible framework for the implementation of genetic algorithms. Part I: the framework," in *Proceedings of the 8th International Conference Interdisciplinarity in Engineering (INTER-ENG '14)*, vol. 19, pp. 193–200, Târgu Mureş, Romania, 2015.
- [2] J. Kennedy, "Particle swarm optimization," in *Encyclopedia of Machine Learning*, pp. 760–766, Springer, New York, NY, USA, 2011.
- [3] A. K. Qin, V. L. Huang, and P. N. Suganthan, "Differential evolution algorithm with strategy adaptation for global numerical optimization," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 2, pp. 398–417, 2009.
- [4] M. Eusuff, K. Lansey, and F. Pasha, "Shuffled frog-leaping algorithm: a memetic meta-heuristic for discrete optimization," *Engineering Optimization*, vol. 38, no. 2, pp. 129–154, 2006.
- [5] D. Karaboga and B. Basturk, "A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm," *Journal of Global Optimization*, vol. 39, no. 3, pp. 459–471, 2007.
- [6] Y.-B. Park, J.-S. Yoo, and H.-S. Park, "A genetic algorithm for the vendor-managed inventory routing problem with lost sales," *Expert Systems with Applications*, vol. 53, no. 1, pp. 149–159, 2016.
- [7] P. Ghosh, M. Mitchell, J. A. Tanyi, and A. Y. Hung, "Incorporating priors for medical image segmentation using a genetic algorithm," *Neurocomputing*, vol. 195, pp. 181–194, 2016.
- [8] Y. Liu, H. Dong, N. Lohse, and S. Petrovic, "A multi-objective genetic algorithm for optimisation of energy consumption and shop floor production performance," *International Journal of Production Economics*, vol. 179, pp. 259–272, 2016.
- [9] M. A. Salido, J. Escamilla, A. Giret, and F. Barber, "A genetic algorithm for energy-efficiency in job-shop scheduling," *The International Journal of Advanced Manufacturing Technology*, vol. 85, no. 5, pp. 1304–1314, 2016.
- [10] O. Abdelkhalak, S. Krichen, and A. Guitouni, "A genetic algorithm based decision support system for the multi-objective node placement problem in next wireless generation network," *Applied Soft Computing Journal*, vol. 33, no. 8, pp. 278–291, 2015.
- [11] M. Salami and T. Hendtlass, "A fitness estimation strategy for genetic algorithms," *Lecture Notes in Computer Science*, vol. 2358, pp. 502–513, 2002.
- [12] E. E. Korkmaz, "Multi-objective genetic algorithms for grouping problems," *Applied Intelligence*, vol. 33, no. 2, pp. 179–192, 2010.
- [13] M. Kaya, "Autonomous classifiers with understandable rule using multi-objective genetic algorithms," *Expert Systems with Applications*, vol. 37, no. 4, pp. 3489–3494, 2010.
- [14] M. A. Abido, "Multiobjective optimal VAR dispatch using Strength Pareto Evolutionary Algorithm," in *Proceedings of the IEEE Congress on Evolutionary Computation*, pp. 730–736, 2002.
- [15] M. Eghbal, N. Yorino, E. E. El-Araby, and Y. Zoka, "Optimal VAR expansion considering ATC using strength pareto evolutionary algorithm," in *Proceedings of the 40th North American Power Symposium (NAPS '08)*, pp. 550–555, IEEE, Calgary, Canada, 2008.
- [16] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.

- [17] E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2: improving the strength Pareto evolutionary algorithm," *Eurogen*, vol. 3242, no. 103, pp. 95–100, 2001.
- [18] M. Kim, T. Hiroyasu, M. Miki et al., "SPEA2+: improving the performance of the strength Pareto evolutionary algorithm 2," in *Parallel Problem Solving from Nature—PPSN VIII: 8th International Conference, Birmingham, UK, September 18–22, 2004. Proceedings*, vol. 3242, pp. 742–751, Springer, Berlin, Germany, 2004.
- [19] K. Deb, S. Agrawal, A. Pratap et al., "A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II," in *Parallel Problem Solving from Nature PPSN VI: 6th International Conference Paris, France, September 18–20, 2000 Proceedings*, vol. 1917 of *Lecture Notes in Computer Science*, pp. 849–858, Springer, Berlin, Germany, 2000.
- [20] B. Milošević and M. Begović, "Nondominated sorting genetic algorithm for optimal phasor measurement placement," *IEEE Transactions on Power Systems*, vol. 18, no. 1, pp. 69–75, 2003.
- [21] S. Kuriakose and M. S. Shunmugam, "Multi-objective optimization of wire-electro discharge machining process by non-dominated sorting genetic algorithm," *Journal of Materials Processing Technology*, vol. 170, no. 1-2, pp. 133–141, 2005.
- [22] D. Kanagarajan, R. Karthikeyan, K. Palanikumar, and J. P. Davim, "Optimization of electrical discharge machining characteristics of WC/Co composites using non-dominated sorting genetic algorithm (NSGA-II)," *International Journal of Advanced Manufacturing Technology*, vol. 36, no. 11-12, pp. 1124–1132, 2008.
- [23] S. Panda, "Multi-objective PID controller tuning for a FACTS-based damping stabilizer using non-dominated sorting genetic algorithm-II," *International Journal of Electrical Power & Energy Systems*, vol. 33, no. 7, pp. 1296–1308, 2011.
- [24] S. Panda and N. K. Yegireddy, "Automatic generation control of multi-area power system using multi-objective non-dominated sorting genetic algorithm-II," *International Journal of Electrical Power & Energy Systems*, vol. 53, no. 1, pp. 54–63, 2013.
- [25] S. R. M. Yandamuri, K. Srinivasan, and S. Murty Bhallamudi, "Multiobjective optimal waste load allocation models for rivers using nondominated sorting genetic algorithm-II," *Journal of Water Resources Planning and Management*, vol. 132, no. 3, pp. 133–143, 2006.
- [26] L. Wang, T.-G. Wang, and Y. Luo, "Improved non-dominated sorting genetic algorithm (NSGA)-II in multi-objective optimization studies of wind turbine blades," *Applied Mathematics and Mechanics*, vol. 32, no. 6, pp. 739–748, 2011.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

