International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 10
Article Received: *20 August 2023* ***Revised:*** *12 October 2023* ***Accepted:*** *24 October 2023*

An Advanced High Dimensional Power System of Optimization Problems Algorithm

Hayder H. Safi[1], Eman. Abbas Ali Al-Khaffaf[2], Hussein Faris Saeed[3]
[1]Department of Computer, College of Basic Education
Mustansiriyah University
Baghdad, Iraq
hayder.h.safi@uomustansiriyah.edu.iq
[2]College of Basic Education-Mustansiriyah University, Baghdad, Iraq
Iman3w@yahoo.com
[3]College of Basic Education-Mustansiriyah University, Baghdad, Iraq.
huseenfars036@gmail.com

Abstract

New Evolutionary Algorithm-based for High-Dimensional Power System Optimization Problems and Modern power system optimization problems face growing challenges as there are numerous decision variables and unbiased functions. Multi-objective evolutionary algorithms (MOEAs) basically are popularly applied for solving and dealing with such complicated problems in substantial power systems that have several objective functions. However, customary MOEAs works appropriately when the number of objective features comes to be less than three. Generally, the effectiveness of MOEAs begins to decrease apparently with an increase in the number of unbiased functions. This study is devoted to applying an authentic gradable evolutionary algorithm for the purpose of solving power system optimization issues which manifest numerous objective functions. In particular, this seeks an algorithm based upon NSGA-II algorithm dedicated for an efficient solution of multi-objective optimization problems. Here, we attempt to modify NSGA-II algorithm to effectively solve many target optimization problems. A novel effective grouping method is applied in the proposed algorithm in similar to the non-dominant grouping method of NSGA-II algorithm to accelerate decision convergence action in POF. The recommended algorithm is compared with modern numerous objective optimization algorithms following three test problems. The findings reveal that as the number of unbiased functions is noticeably large, the algorithm that we propose is significantly superior to other algorithms.

Keywords— NSGA-II; MOEA, optimization problems; evolutionary algorithms; objective problems;

I. INTRODUCTION

Generally, many applications, simply, have to apply simultaneous optimization of multiple objectives, termed as multi-objective optimization problems (MOPs). Setting various goals, which typically include a number of conflicting objectives, demands looking for a number of optimal solutions (called Pareto Optimal Bounds, or POFs) rather than one optimal solution. A method to do that; solving MOP, is to use an evolutionary algorithm (EA). Evolutionary algorithms can be viewed as one of the influential methods that was applied to find solutions for all kinds of optimization issues [1]. The influence of evolutionary algorithms stems from the idea that they use a set of solutions on the account of having one solution. In EA, the number of individuals stands for the group of solution candidates. Any individual is given merit that indicates how good the generated solution is. When maximizing or minimizing the fitness function, the solution set can be randomly initialized to form the initial population. Individuals within the population can then mate and produce offspring. Parents and children, basically, work hard and compete to be part of the generation that follows, so, accordingly, only highly qualified individuals may survive. Hence, the total population is repetitively improved. EA often applies mechanisms supported by biologically-oriented evolution in addition to crossover and alteration to improve the solution group [2]. When EAs were first introduced, they have been commonly followed to attempt solving optimization problems with a single goal. After that, researchers started using EAs to provide solutions to multi-objective optimization problems. EA algorithms can be implemented to attain the optimal group of solutions for a number of goals. That is because the EA works on the solution population [3].

A lot of work has been done using and following evolutionary algorithms to MOPs in order to provide solutions to various true optimization issues. Marked Multi-Objective Evolutionary Algorithms (MOEAs) consist of Vector Evaluated Genetic Algorithm VEGA [10], Multi-objective Genetic Algorithm MOGA [9], Strong Pareto Evolutionary algorithm SPEA [6], Multi-objective Particle Swarm Optimization MOPSO [7,8], Non-dominating Sorting Genetic Algorithm NSGA [4] and NSGA-II [5].

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 10
Article Received: *20 August 2023* ***Revised:*** *12 October 2023* ***Accepted:*** *24 October 2023*

NSGA-II is considered a prominent algorithm for solving MOP. NSGA-II tends to merge antecedent parent and child populations to attain 2N population. N, here denotes the solution number of population. Thus, population, then, is grouped through manipulating a non-dominant grouping algorithm. The first group of solutions represents the highly-distinguished solution in population and are carried over to the next following generation. Members of the following generation population are chosen in terms of the solution number in each rank. A strategy of diversity called the comparison operator which is described in [5] is applied to bring about portioning out the solutions over a large area over POF.

NSGA-II and the majority of MOEAs are produced to perform excellently as the number of objective function is not large. Scholars have, therefore, begun to explore solutions by applying new methods to MOEA in order to tackle MOPs with noticeably a large number of functions, which is termed as multi-objective problems. Multi-objective problems (MaOPs), traditionally, are viewed as MOPs with a minimum of four objectives [11], or MOPs with three objectives [12]. Recently, MaOP has attracted the scholars' attention because many true optimization problems involve objective functions of more than three. MaOP exists in a number of areas such as portfolio planning [13], controller optimization [14], and engineering design [15].

Here, NSGA-II algorithm is modified with the aim of solving MaOS efficiently. According to this, this study proposes a non-dominant grouping algorithm, a novel sorting strategy which is different from the non-dominant sorting algorithm. The suggested grouping algorithm puts into account the number of dominant solutions separately for any single objective function. The proposed grouping strategy increases the priority among solutions within the population. This produces more portioned solutions and reduces the solution number for any single rank. Moreover, the newly suggested algorithm applies an effective local mechanism to optimize the search function. The suggested algorithm is compared to other algorithms through applying three test problems. For calculating IGD metric, 500 points were created on the POF for any single test problem and generated an objective figure for each case. The points that were created include the total real POF for any problem and form the optimal solution for the MOP. Thus, it is therefore a very important process.

This study is arranged in terms of sections. Section two is dedicated to the related work while section three deals with the suggested sorting strategy . Section four focuses on the results obtained from the experiment. Finally, the study comes out with a number of conclusions included in section 5.

II. RELATED WORK

The current section is an attempt to review the major research for discussing multi-objective optimization problems marked with at least two objective functions. It is found that the recorded number of articles addressing problems with more than four objectives increased substantially from 23 articles in 2008 to reach 54 articles in 2013 [16]. The growing interest in evolutionary algorithms with many objectives is attributed to the problems which are originated as the objective number is seemingly large. The related literature considers many issues with MaOP such as: numerous non-dominant solutions, time of computation, diversity loss, difficulties related to visualization and performance [16].

Different approaches have been developed to address these issues, depending on the issues that may arise when solving the MaOP. In [17], two diversity mechanisms for management are suggested to highly improve and promote diversity of convergence MaOP's overall population. The study tested the two methods individually and all-inclusively with reference to a group of test objectives. The results revealed that the use of only one technique of diversity enhanced and improved MOEA's performance in solving the MaOP. In another study [18], the scholars tried to analyze the NSGA-II algorithm in order to solve MaOP and found that the congestion distance applied by the algorithm to secure diversity was optimal with many other goals. We have previously proposed other methods, arguing that they are not suitable for the optimization problem. It can effectively interchange or replace the operator of congestion distance in the NSGA-II algorithm. The study analyzed a number of DTLZ problems, the author suggested surrogate assignment distance, which he uses in NSGA-II, to improve the performance of MaOP's algorithm.

Performance of evolutionary algorithms in solving MaOP can be improved by changing the crossover operator [19]. This can be done through the use of a new effective crossover operator which is chosen out a group of 10 variable crossover operators. [19] also suggested an authentic dominance concept and an adaptively dynamic population size method for promoting and improving MaOP's evolutionary algorithm performance.

For the aim of improving the process of selecting regarding the Pareto Optimal Front (POF), a number of newly-adopted techniques were suggested to improve or entirely alternate non-dominant sorting. A study in this respect suggested a new terminology of dominance termed as ε-dominance [19]. The authentic dominant ratio causes the dominant space a bit larger and, then, easier to attain. In [20], the study suggested an effective stable-state algorithm for the purpose of processing MaOP depending on ε-dominance which is termed as ε-MOEA [21]. The performance of ε-MOEA was put to the test in [22].

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 10
Article Received: *20 August 2023* ***Revised:*** *12 October 2023* ***Accepted:*** *24 October 2023*

Findings reveal that the algorithm can, to a large extent, solve MaOP, in a way, better than the conventional MOEA.

Lately, an evolutionary algorithm based upon reference point groups was suggested to tackle MaOPs. Such approaches typically are based on measuring solution quality through implementing a group of reference solutions. So, the so called reference points customarily will control the searching process very properly. That's why Deb and Jain [23] suggested a multi-objective NSGA-II algorithm and termed it as NSGAIII. This algorithm applies a reference strategy point to distinguish solutions which are not dominant but, seemingly more or less, close to a given group of reference points. NSGA-III was, then, tested in a group of multi-objective problems. Also, the findings reveal that NSGA-III performs effectively on all kinds of problems.

Adding to the above, Wang et al. suggest an authentic algorithm through applying two groups of data archives (TAA). They got their second version, namely, (Two Arch2). The suggested algorithm divides the origination of non-dominant solutions through creating two archives: the diversity archive and the convergence archive. Then, according to this, the second archive is applied and based for online and real reference groups. It traditionally includes non-dominant solutions. However, as the solutions number regarding the two archives overreach a recommended size by following the shortest distance technique, the solutions in the archive of diversity are thus cleared and removed. Experimental findings when testing this algorithm reveal that TAA in addition to its pre-based second version predominate other evolutionary MaO algorithms with respect to convergence in addition to diversity [24].

Another method to solving MaOP is to use dimensionality reduction method [25]. A dimensionality reduction method is applied to solve MaOP through reducing the copious target number. Advantages of the dimensionality reduction method may, to some degree, come to include decreasing the cost of computation of the MaO evolutionary algorithm in addition to result visualization through removing copious target technique. Additionally, it is necessary to highlight that the approach could also be used similarly with other methods of performance enhancement of evolutionary algorithms [26].

III. PROPOSED ALGORITHM

This study focuses on non-dominant solutions with a large number in MaOP. Normally, as a MOEA begins to evolve, there will, traditionally, be non-dominant solutions with a small number, an ever-increasing number of non-dominant solutions, and almost all population solutions becoming non-dominant. This process should be done step by step to ensure that the best solution is selected. This might be accepted or possible only when the number of objective functions is seemingly small or, let's say, less than four. A relationship was found between the number of non-dominant solutions with regard to numbers of objective function generations. It is definitely clear that when the goal number gets high, so do the non-dominant solutions. For example, if the number of objectives is the same as or more than three, all solutions are thought to be out of control in about 10 generations. Accordingly, such a situation may lead to limit the convergence tendency of evolutionary algorithms. Therefore, a new sorting mechanism is demanded as it can effectively classify solutions even in cases of a large objective number.

A. The Proposed Sorting Algorithm

In the current study, a new sorting method is suggested to effectively classify MaOP solutions. This method, the sorting one, tends to raise the priority average among solutions, making the process of ranking possibly reasonable including non-dominant solutions. The new sorting algorithm specifies the rank of every solution by separately transforming the dominance relations of every problem objective function. Additionally, the relationship, here, is deemed as an integer to denote the number of objectives being controlled by the function. The entire dominance relationship of the solution is, then, calculated by collecting the dominance values of any single objective function.

The difference of values recorded between NSGA-II sorting mechanism and the suggested one are shown in table (1). V1, V2, and V3 are the objective function values, and C1 through C5 refer to 5 population solutions. As it is clear in Table 1, the population has three goals and 5 solution candidates. The NSGA-II sorting mechanism relies basically on candidate solution number dominated by those under consideration. So, when there is no solution which controls the solution under study, it is ranked first, as are her first four solutions (C1-C4). If there is only one solution which controls the solution under study, it is in rank 2 and only dominated by C4, as indicated by the row for C5.

Table I. NSGA-II Non-Domination Sorting and the Suggested One

	V1	V2	V3	NSGA-II Ranking	Proposed Ranking	
C1	6	2	3	1	3+0+2=5	4
C2	7	4	2	1	4+3+0=7	3
C3	3	5	5	1	1+4+3=8	2
C4	5	4	6	1	2+3+4=9	1
C5	3	3	3	2	1+1+2=4	5

The sorting mechanism of the new method depends on the total of solutions controlled by every solution objective, as presented in the equations 1 and 2. The suggested sorting mechanism can,

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 10
Article Received: *20 August 2023* ***Revised:*** *12 October 2023* ***Accepted:*** *24 October 2023*

to a large extent, break up the solution into more distributed ranks as shown in table 1. This speeds up the convergence process.

$$d_i = \sum_{k=0}^{n} \partial_k \quad \dots (1)$$

Where $\partial_k = 1$, if $f_{ij} \geq f_{ik}$ and $\partial_k = 0$ otherwise, n is the size of population

$$S_t = \sum_{i=0}^{m} d_i \quad \dots (2)$$

To further illustrate, let's use the equation to the S_1 solution presented in the above table. After that, we get $S_1 = d_1 + d_2 + d_3$. As it is clear from the table, it can be calculated as: $S_1 = 3 + 0 + 2 = 5$. Noticeably, It might also be significant to say that the non-dominant algorithm is confined to classify the population into two ranks, whereas the sorting algorithm that we propose can classify it into 5 ranks. For this, it means that there is significantly more preference between solutions. Moreover, the suggested sorting algorithm has no impact on non-dominant sorting, as for both sorting methods S5 remains the worst solution.

B. Combining the New Sorting Algorithm with NSGA-II

We combine a new sorting algorithm with the NSGA-II algorithm in so it could efficiently improve the NSGA-II algorithm to solve MaOP without struggling with the problem of large numbers of non-dominant solutions. The modified algorithm couldn't be used directly by substituting the non-dominant sorting algorithm. Because we need to make sure that the solutions chosen for the following generation are not dominant. Henceforth,, a specified sorting algorithm can be applied as long as the number of non-dominant solutions overreaches population size, as shown in Figure 2. In such a state, we run the specified algorithm to partition the non-dominant solutions (rank 1 solutions) over a wider level in order the algorithm chooses the highly suitable solution from the non-dominant solutions.

Thus, our suggested sorting algorithm provides a bit novel average of NSGA-II sorting algorithm, allowing MaOP to handle large numbers of objectives. Additionally, the sorting algorithm that we propose does surely not play a role in affecting NSGA-II algorithm performance. Furthermore, no evaluation is required which will be performed if the non-dominant solution overtakes or exceeds normal limits which might be a standard input of the algorithm and could also be adapted depending on environment. Then, after evaluating the proposed reordering algorithm, the remaining steps of the NSGA-II algorithm are applied in a normal way.

```
1.  // Merge parent and offspring population
2.      Pop(t) = Combine (Parents(t), Children(t))
3.  // Sort using non-dominating technique
4.   SortedPop = NonDominatedSort (Pop(t))
5.  // Check number of candidate  solutions in rank 1
6.  if  | Rj | > PopSize
7.      SortedPop = NewRanking (Pop(t))
8.  end
9.   Parents (t+1) = Φ and j = 1
10. // Till parent pop.  not filled
11. while | Fathers (t+1) | + | Rj | < PopSize
12.      Fathers (t+1) = Parents (t+1) + Rj
13.      j = j +1
14. end
15. // Sort using the crowding density
16.    Parents (t+1) = Parents (t+1) + Rj[1:(PopSize – | Parents (t+1)|)]
17.     Children(t+1) = NormalNSGAGenetic(Fathers (t+1))
```

Fig. 1. Modified NSGA-II algorithm

IV. EXPERIMENT AND DISCUSSION

Here, the item is dedicated to the use of a group of test functions taken from the known problem group DTLZ [23] to provide a comprehensive framework for measuring the performance improvement of our new algorithm against three other multi-objective evolutionary algorithms. provides a meaningful comparison. The suggested algorithm is then quantified with one of the most suitable algorithms for solving MOP, the non-dominant genetic permutation algorithm version 2 (NSGA-II).

A. Test Problems

Three test problems were manipulated to check the performance of the novel algorithm. The three test problems are chosen from the DTLZ [23] group of multi-target optimization problems. The DTLZ problem is gradable and could be given to any suggested objective functions. DTLZ1 is, conventionally, a linear Pareto optimal front problem because the values lie in the linear hyper-plane. So, when this is the case, DTLZ2 and DTLZ3 are nonlinear problems, and DTLZ3 is specifically suggested to measure the performance of MOEA on local Pareto-optimal fronts. This problem involves a series of local Pareto-optimal fronts that are similar to the global Pareto-optimal front and the applicable algorithms can, to an extent, be connected at any one of the local Pareto-optimal fronts, and before the time when converging to the global Pareto-optimal front.

$$Minimize \quad f1(x) = \frac{1}{2} x1x2 \dots x_{M-1}(1 + g(x_M)),$$
$$Minimize \quad f2(x) = \tfrac{1}{2} x1x2 \dots (1 - x_{M-1})(1 + g(x_M)), Minimize \quad fM - 1(x) = \tfrac{1}{2} x1(1 - x2)(1 + g(x_M)),$$
$$Minimize \quad fM(x) = \frac{1}{2} (1 - x1)(1 + g(x_M),$$
$$subject\ to\ 0 \leq xi \leq 1, \quad for\ i = 1,2,3, \dots \dots n.$$
$$g(Xm) = 100 \left[|Xm| + \sum_{xi \in X_M}^{\square} (xi - 0.5)^2 - \cos(20\pi(xi - 0.5)) \right]$$

The Pareto-optimal solution is parallel to x_M = 0 while objective functions should be in the linear hyper-plane, as it is presented below:

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 10
Article Received: *20 August 2023* ***Revised:*** *12 October 2023* ***Accepted:*** *24 October 2023*

$$\sum_{m=1}^{M} fm = 0.5$$

B. Performance Metrics

According to the current study, the Inverted Generational Distance (IGD) [26] was applied for estimating performance of our suggested algorithm and NSGA-II algorithms. IGD is used to estimate diversity and convergence. This metric could be calculated using the following two formulas:

$$IGD_t = \frac{\sum_{t=1}^{|PF_t^*|} d_i}{|PF_t^*|} \quad \ldots (3)$$

$$d_i = \min_{k=1}^{|PF_t^*|} \sqrt{\sum_{j=1}^{M} (f_j^{*(i)} - f_j^{(k)})^2} \quad \ldots (4)$$

Where $f_j^{(k)}$ refers to the value of the j-th objective function for the k-th member of the non-dominated solutions, and PF_t^* is Optimal Pareto Front, Thus, the number of objective functions is M while the Euclidean distance is d_i. Accordingly, the IGD index is estimated for the ultimate population solution. Eventually, for IGD metric to be applied, it is necessary to create a large number of random and evenly allotted points in POF. That's why, in our endeavor, we created 500 points for any problems and every number of objective functions.

C. Experimental Results

To implement an unbiased comparison framework for experimentation, NSGA-II parameters were set for the two algorithms tested. 100 population, crossover probability of 0.7, and mutation probability of 1/n are applied. Here, n refers to variable number. Real-valued representations are applied for algorithms and simulated binary crossovers (SBX) as well.

These generations were estimated with three different numbers (100, 200, and 300). It allows to examine performance of the algorithm at different steps of implementation. The variable number was specified to 10. We applied 5 different numbers of objective functions to examine the algorithm's performance of a large number of objective functions.

Table II, Table III, and Table IV demonstrate the performance of NSGA-II and suggested algorithms for all DTLZ1, DTLZ2, and DTLZ3 problems, respectively. The best results to any single experiment have been displayed in bold. The values shown in the table refer to the average of 10 independent runs for each instance of the experiment. Better than algorithm. According to table 1, which shows DTLZ1results, it can be seen that NSGA-II algorithm outperforms in only 3 out of 15 cases. Moreover, our algorithm outperforms the NSGA-II algorithm in 13 cases. Subsequently, the DTLZ1 problem results reveal that when NSGA-II is run over a large number of generations, it gives better results as objectives look like 3 or 4.

According to table 2, DTLZ2 results reveal that the suggested algorithm shows nearly better performance in 10 cases, while NSGA-II algorithm acts well in 5 cases. On the other hand, the table also shows that the proposed algorithm outperforms the NSGA-II algorithm in the majority of the cases as the number of objectives is greater than 4. But overall the algorithm provides better results on DTLZ1 problem than in DTLZ2. That's because of conditions of the two problems. Along with, DTLZ1 problem is characterized with a linear POF equation.

Evidently, DTLZ3 results are similar to those of the above two test problems, but our new algorithm outperforms the NSGA-II algorithm. As long as there are many local optima to this problem, therefore, NSGA-II performance degrades significantly, making the gap with regard to how the two algorithms perform more pronounced.

Table 2: The Suggested Algorithm Results and NSGA Test Problem

gen.	obj.	NSGA-II	Proposed
100	3	4.9345	**3.3312**
	4	26.0980	**19.5001**
	5	54.3154	**42.3622**
	6	65.9997	**57.4752**
	7	56.2094	**46.6145**
200	3	1.0395	**0.9608**
	4	10.4708	**7.3221**
	5	31.7161	**27.1683**
	6	41.6462	**28.8672**
	7	48.0566	**25.7695**
300	3	**0.3235**	0.4420
	4	**4.3427**	18.8477
	5	26.6527	**20.2164**
	6	**20.6279**	22.9584
	7	22.7482	**20.7092**

Table 3: The Suggested Algorithm and NSGA for DTLZ2 Test problem

gen.	obj.	NSGA-II	Proposed
100	3	0.07195	**0.07065**
	4	0.18906	**0.18187**
	5	**0.29388**	0.30692

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 10
***Article Received:** 20 August 2023 **Revised:** 12 October 2023 **Accepted:** 24 October 2023*

	6	0.40228	**0.40040**
	7	0.47126	**0.45961**
200	3	0.07047	**0.06886**
	4	0.18905	**0.18389**
	5	**0.29207**	0.31470
	6	**0.40026**	0.41230
	7	0.46842	**0.46440**
300	3	0.06989	**0.06910**
	4	0.18758	**0.18410**
	5	**0.30076**	0.31243
	6	**0.39287**	0.41732
	7	0.46694	**0.46326**

The study conducted another experiment to highlight the performance enhancement that the new proposed algorithm can achieve. In this, we estimated the algorithms with 200 generation and the objective functions reach seven. Values of the same size are used for the remaining parameters in this experiment. Figures 2, 3 and 4 show us the results mentioned. It is clear that the suggested algorithm converges to POF quicker than NSGA-II algorithm. The new sorting mechanism that we use in our algorithm is capable of selecting the best solutions from the non-dominating solutions and because the population will be guided to POF quicker than NSGA-II algorithm.

Table 4: The Suggested Algorithm and NSGA-II for DTLZ3 Test Problem

gen.	obj.	NSGA-II	Proposed
100	3	11.25580	**10.48939**
	4	**52.62811**	57.24726
	5	**59.93428**	69.05014
	6	**63.62725**	73.28039
	7	62.58267	**58.21793**
200	3	2.82619	**2.07531**
	4	38.24524	**25.55183**
	5	96.02376	**82.09266**
	6	95.62635	**76.36449**
	7	73.33984	**65.38907**
300	3	0.56393	0.70619
	4	24.16421	**18.65761**
	5	**50.83064**	50.34360
	6	**58.62893**	61.64765
	7	51.39962	**48.99231**

V. CONCLUSION

The present study suggests an authentic multi-objective algorithm for efficiently manipulating objective optimization issues. The suggested algorithm is applied and used by specifying and improving the reordering mechanism of the NSGA-II algorithm. The suggested algorithm follows an authentic reordering mechanism similarly with the non-dominant reordering scheme of NSGA-II algorithm to accelerate the process of solution convergence. For this, a group of experiments were conducted to examine the act of the suggested algorithm. Most importantly, the experiments apply the IGD metric to compare the algorithm's performance to the NSGA-II algorithm. The study applies a group of three test problems from DTLZ set with various numbers of objective functions. The findings reveal that, normally, when the function number becomes higher, the suggested algorithm clearly outperforms the NSGA-II algorithm. For future work, a comparison might be executed for this algorithm with reference to more versatile algorithms. Additionally, the suggested algorithm creates and solves a group of true energy system problems .

REFERENCES

[1] Eiben, Agoston E., and James E. Smith. Introduction to evolutionary computing. Vol. 53. Heidelberg: springer, 2003.
[2] Lee, Kwang Y., and Mohamed A. El-Sharkawi, eds. Modern heuristic optimization techniques: theory and applications to power systems. Vol. 39. John Wiley & Sons, 2008.
[3] Haupt, Randy L., and Sue Ellen Haupt. Practical genetic algorithms. John Wiley & Sons, 2004.
[4] Srinivas N. and Deb K.: Multiobjective optimization using nondominated sorting in genetic algorithms. IEEE Transactions on Evolutionary Computation. 2 (3), 221–248 (1994)
[5] Deb K., Pratap A., Agarwal S., and Meyarivan T.: A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Transactions on Evolutionary Computation. 6 (2), 182–197 (2002)
[6] Zitzler E. and Thiele L.: Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach. IEEE Transactions on Evolutionary Computation. 3 (4), 257–271 (1999)
[7] Hu X. and Eberhart R. C.: Multiobjective optimization using dynamic neighborhood particle swarm optimization. in Proc. Congr. Evol. Comput., Honolulu, HI, 1677–1681 (2002)
[8] Zhang L. B., Zhou C. G., Liu X. H., Ma Z. Q., Ma M., and Liang Y. C.: Solving multi objective problems using particle swarm optimization. Proc. Congr. Evol. Comput., Canberra, Australia, 2400–2405 (2003).
[9] Fonseca, C. M. and Fleming, P. J. (1993). Genetic Algorithms for Multiobjective Optimization: Formulation, Discussion and Generalization. In Forrest, S., editor, Genetic Algorithms: Proceedings of the Fifth International Conference, pages 416–423. Morgan Kaufmann.
[10] Schaffer, J. D. (1985). Multiple Objective Optimization with Vector Evaluated Genetic Algorithms. In Proceedings of the 1st International Conference on Genetic Algorithms, pages 93–100, Mahwah, NJ, USA. Lawrence Erlbaum Associates, Inc.

[11] M. Farina and P. Amato. 2002. On the optimal solution definition for many-criteria optimization problems. In Fuzzy Information Processing Society, 2002. Proceedings. NAFIPS. 2002 Annual Meeting of the North American. IEEE, 233–238.

[12] R.C. Purshouse, k. Deb, M.M. Mansor, S. Mostaghim, and R. Wang. 2014. A review of hybrid evolutionary multiple criteria decision making methods. COIN Report,(2014005), January (2014).

[13] G. Fu, Z. Kapelan, J.R. Kasprzyk, and P.M. Reed. 2012. Optimal design of water distribution systems using many-objective visual analytics. Journal of Water Resources Planning and Management (2012).

[14] K. Narukawa and T. Rodemann. 2012. Examining the Performance of Evolutionary Many-Objective Optimization Algorithms on a Real-World Application. In Genetic and Evolutionary Computing (ICGEC), 2012 Sixth International Conference on. IEEE, 316–319.

[15] P.J. Fleming, R.C. Purshouse, and R.J. Lygoe. 2005. Many-objective optimization: An engineering design perspective. In Evolutionary Multi-criterion Optimization. Springer, 14–32.

[16] Li, B., Li, J., Tang, K., & Yao, X. (2015). Many-objective evolutionary algorithms: A survey. ACM Computing Surveys (CSUR), 48(1), 13.

[17] S. F. Adra and P. J. Fleming, "Diversity management in evolutionary many-objective optimization," IEEE Transactions on Evolutionary Computation, vol. 15, no. 2, pp. 183–195, 2011.

[18] M. Koppen and K. Yoshida, "Substitute distance assignments in nsga-ii for handling many-objective optimization problems," in Proceedings of Fourth International Conference on Multi-Objective Optimization, EMO 2007 (LNCS 4403). Heidelberg,Germany:Springer-Verlag, 2007,pp. 727–741.

[19] D. Hadka and P. Reed, "Borg: An auto-adaptive many-objective evolutionary computing framework," Evolutionary Computation, p. in press, 2011.

[20] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler. 2002. Scalable multi-objective optimization test problems. In Evolutionary Computation (CEC), 2002 IEEE Congress on. IEEE, 825–830.

[21] K. Deb, M. Mohan, and S. Mishra. 2005. Evaluating the "ε-domination based multi-objective evolutionary algorithm for a quick computation of Pareto-optimal solutions. Evolutionary Computation 13, 4 (2005), 501–525.

[22] D. Hadka and P.M. Reed. 2012. Diagnostic assessment of search controls and failure modes in manyobjective evolutionary optimization. Evolutionary Computation 20, 3 (2012), 423–452.

[23] Deb, Kalyanmoy, et al. Scalable test problems for evolutionary multiobjective optimization. Springer London, 2005.

[24] K. Praditwong and X. Yao. 2006. A new multi-objective evolutionary optimisation algorithm: the two-archive algorithm. In Computational Intelligence and Security, 2006 International Conference on, Vol. 1. IEEE, 286–291.

[25] Goh, C., Tan, K.: A competitive-cooperative coevolutionary paradigm for dynamic multiobjective optimization. IEEE Transactions on Evolutionary Computation. 13 (1), 103–127 (2009)

[26] Zhou, Y. Jin, Q. Zhang, "A population prediction strategy for evolutionary dynamic multiobjective optimization," IEEE Trans. Cybern., vol. 44, no. 1, pp. 40–53, 2014.