■Ngày nhận bài: 19/4/2024 ■Ngày sửa bài: 15/5/2024 ■Ngày chấp nhận đăng: 06/6/2024

NSGA-II and SPEA2+SDE: A comparative study

NSGA-II và SPEA2+SDE: Một nghiên cứu so sánh

> DR XUAN-BINH LAM[1], DR ANH-THANG LE[2]
[1]Department of Mechanics, Faculty of Civil Engineering, Ho Chi Minh City University of Technology and Education
Email: binhlx@hcmute.edu.vn
[2]Faculty of Civil Engineering, Ho Chi Minh City University of Technology and Education

ABSTRACT

Design Optimization is research area which was carried out a lot on the world, with many applications in the specifications of mechanical engineering, aerospace engineering, civil engineering. The global optimization methods are more effective than local optimization methods. In this article, the authors will deal with 2 multi-objective global optimization methods NSGA-II and SPEA2+SDE. The authors will do novel research about comparing the efficiencies of these multi-objective optimization methods. For the sample problems in this article, it is investigated that SPEA2+SDE can produce optimal solutions very close to exact Pareto front. In these circumstances, SPEA2+SDE is better than NSGA-II in finding multi-objective optimization and optimal front. It is also investigated that SPEA2+SDE will cost more computational time than NSGA-II. In summary, SPEA2+SDE is better than NSGA-II though SPEA2+SDE is more time-consuming.

Keywords: Design optimization; multi-objective optimization Methods; SPEA2+SDE; NSGA-II.

TÓM TẮT

Thiết kế tối ưu là lĩnh vực nghiên cứu được thực hiện nhiều trên thế giới, với nhiều ứng dụng trong các chuyên ngành kỹ thuật cơ khí, kỹ thuật hàng không vũ trụ, kỹ thuật xây dựng. Các phương pháp tối ưu hóa toàn cục hiệu quả hơn các phương pháp tối ưu hóa cục bộ. Trong bài báo này, các tác giả sẽ nghiên cứu 2 phương pháp tối ưu toàn cục đa mục tiêu NSGA-II và SPEA2+SDE. Các tác giả sẽ nghiên cứu mới về so sánh các hiệu năng của các phương pháp tối ưu đa mục tiêu này. Đối với những bài toán mẫu trong bài báo này, chúng ta khám phá ra rằng SPEA2+SDE có thể xuất ra các lời giải tối ưu rất gần với mặt Pareto chính xác. Trong những tình huống này, SPEA2+SDE thì tốt hơn NSGA-II trong việc tìm tối ưu hóa đa mục tiêu và mặt tối ưu. Chúng ta cũng khám phá ra rằng SPEA2+SDE sẽ tốn nhiều thời gian tính toán hơn NSGA-II. Tổng kết lại, SPEA2+SDE thì tốt hơn NSGA-II cho dẫu SPEA2+SDE tốn nhiều thời gian hơn.

Từ khóa: Thiết kế tối ưu; các phương pháp tối ưu hóa đa mục tiêu; SPEA2+SDE; NSGA-II.

1. INTRODUCTION

Let's compare the Multi-objective Optimization (MOO) [1,2,3] and single objective optimization. Single objective optimization just has only one global optimum, but multi-objective optimization has a set of global solutions, this set of global optimizations is called Pareto-optimal set. So multi-objective optimization can find many optima at one time. Almost every real-world problem involves simultaneous optimization of several incommensurable and often competing objectives. While in single-objective optimization, the optimal solution is usually clearly defined, this does not hold for multi-objective optimization problems. Instead of a single optimum, there is rather a set of alternative trade-offs, generally known as *Pareto-optimal* solutions. These solutions are optimal in the wider sense that no other solutions in the search space are superior to them when *all* objectives are considered.

Multi-objective optimization problems (MOPs) [1,2,3] are common. For example, consider the design of a complex hardware/software system as it can be found in mobile phones, cars, etc. Often the cost of such systems is to be minimized, while maximum performance is desired. Depending on the application, further objectives may be important such as reliability and power dissipation. They can be either defined explicitly as separate optimization criteria or formulated as constraints, e.g., that the size of the system must not exceed given dimensions.

In this article, there are several multi-objective optimization algorithms will be considered, such as Shift-Based Density Estimation Strength Pareto Evolutionary Algorithm 2 (SPEA2+SDE) [4], Non-dominated Sorting Genetic Algorithm II (NSGA-II) [5,6,7]. SPEA [8] (Strength Pareto Evolutionary Algorithm) uses a mixture of established and some techniques in order to approximate the Pareto-optimal set. On one hand, similarly to other Multi-objective Optimization Evolutionary Algorithms (MOEAs), it stores those individuals externally that represent a nondominated front among all solutions considered so far; it uses the concept of Pareto dominance in order to assign scalar fitness values to individuals; it performs clustering to reduce the number of individuals externally stored without destroying the characteristics of the trade-off front. On the other hand, SPEA [8] is unique in four respects: it combines the above three techniques in a single algorithm; the fitness of a population member is determined only from the individuals stored in the external set; whether individuals in the population dominate each other is irrelevant; all individuals in the external set participate in selection; a new Pareto-based niching method is provided in order to preserve diversity in the population. SPEA [8] can outperform non-elitist methods (in mid-1980s, the researchers proposed several Pareto-based optimization methods, these methods did not incorporate elitism explicitly). The SPEA2 [9,10,11] can try to eliminate weaknesses of SPEA [8]. There are several differences between SPEA and SPEA2: 1) SPEA2 can enhance fitness assignment; 2) SPEA2 can use a technique of neighbor density estimation; 3) SPEA2 proposed a new method of archive truncation. Besides, it is investigated that NSGA-II [5,6,7] and SPEA2 [9,10,11] cannot make sufficient pressure of selection towards Pareto front. In this article, a MOO method called SPEA2+SDE [4] will be considered. The SPEA2+SDE will be the SPEA2 that integrates Shift-Based Density Estimation (SDE). Moreover, this article also deals with Non-dominated Sorting Genetic Algorithm II (NSGA-II). The NSGA [12] was one of the first Evolutionary Algorithms. There are several main criticisms of NSGA such as: 1) The nondominated sorting has high computational complexity: The nondominated sorting algorithm has a computational complexity of $O(MN^3)$ (where M is the number of objectives and N is the population size), this makes NSGA computationally expensive for large population sizes; 2) NSGA does not have elitism: the effectiveness of elitism is that elitism can speed up the performance of the Genetic Algorithm (GA) significantly, which also can help preventing the loss of good solutions once they are found; 3) NSGA must identify the sharing parameter σ_{share}: traditional mechanisms of ensuring diversity in a population so as to get a wide variety of equivalent solutions have relied mostly on the concept of sharing, the main problem with sharing is that it requires the specification of a sharing parameter. NSGA-II is an improvement of NSGA. NSGA-II will replace sharing function with crowded-comparison approach. No user-defined parameter for maintaining the diversity among members of population is needed. NSGA-II can have better computational complexity than NSGA.

In this article, the authors will do novel research of comparative study of SPEA2+SDE and NSGA-II. The novel part of this article is that SPEA2 integrates SDE in comparative study. In previous researches, the other authors just compared the performances between SPEA2 and NSGA-II (not SPEA2+SDE and NSGA-II). The article will compare the results of SPEA2+SDE and NSGA-II on 2-objective problems DTLZ1, DTLZ2, DTLZ3, DTLZ5, DTLZ7 [13,14]. The computational time for 2 methods is also taken into account.

The ultimate goal of developing any optimization algorithm is to solve real-world optimization problems reliably and efficiently. However, since in real-world optimization problems the nature of landscape and optimization solution(s) are not usually known beforehand, at the end of a simulation run, it becomes difficult to test how well an algorithm has performed. For this purpose, there is a need to develop test problems for testing optimization algorithms. Since the landscape and corresponding optimum solution(s) of such problems will be known, they allow to test an algorithm's ability to overcome the difficulties posed by the landscape and ability to converge near the optimum solution(s). Keeping in mind the ultimate goal of solving real-world problems, it then becomes important to construct test problems which are representative to real-world problems. In the reference [13], Deb proposed 5 optimization problems DTLZ1, DTLZ2, DTLZ3, DTLZ5, DTLZ7 to test the multi-objective optimization algorithms. The optimum solutions of these 5 optimization problems are known clearly. The optimum solutions of these 5 optimization problems are different. Deb utilized 2 approaches (bottom-up approach and constraint surface approach) to construct these 5 optimization problems. These 5 optimization problems will have big meaning and are very popular in specification of design optimization. Almost every research scientist in the specification of design optimization utilizes these 5 optimization problems to test their multi-objective optimization algorithms.

2. NONDOMINATED SORTING GENETIC ALGORITHM II (NSGA-II)

2.1. Elitist Nondominated Sorting Genetic Algorithm

In the first step, we will describe a Fast Nondominated Sorting Approach [5,12]. To determine the first nondominated front in a population of size N, the algorithm will compare each solution with every other solution in the population to investigate if it is dominated. If M is the number of objectives, the algorithm will request $O(MN)$ comparisons for each solution. Then the total complexity can be obtained as $O(MN^2)$. At this step, the algorithm can find all individuals in the first nondominated front. To search for the individuals in the next nondominated front, the algorithm will discount the solutions of the first front temporarily, then the algorithm will repeat the above procedure.

The algorithm will mention about the next notion: Preservation of Diversity. NSGA [5,12] will utilize the sharing function method. In this sharing function method, the sharing parameter σ_{share} is included, the sharing parameter will establish the extent of sharing desired in a problem. The sharing parameter is the biggest value of distance metric in estimating proximity measure between 2 population members.

This is the Fast Nondominated Sorting Approach [5,12] (table 1 and figure 1):

Table 1. Fast Nondominated Sorting Approach

for each $p \in P$	
$S_p = \varnothing$	
$n_p = 0$	
for each $q \in P$	
if $(p \prec q)$ then	If q is dominated by p
$S_p = S_p \cup \{q\}$	Add q to the set of solutions dominated by p
else if $(q \prec p)$ then	
$n_p = n_p + 1$	The domination counter of p is increased
if $n_p = 0$ then	p is in the first front
$p_{rank} = 1$	
$F_1 = F_1 \cup \{p\}$	
$i = 1$	The front counter is initialized
while $F_i \neq \varnothing$	
$Q = 0$	To store the members in the next front
for each $p \in F_i$	
for each $q \in S_p$	
$n_q = n_q - 1$	
if $n_q = 0$ then	q is in the next front
$q_{rank} = i + 1$	
$Q = Q \cup \{q\}$	
$i = i + 1$	
$F_i = Q$	

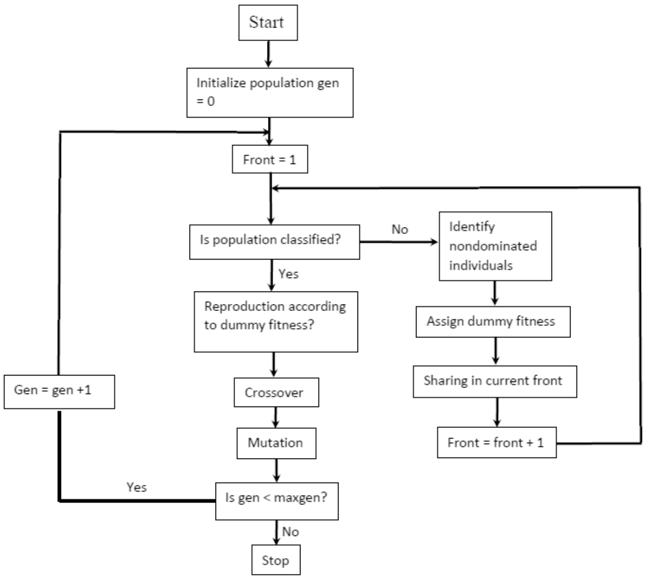

Figure 1. Flow chart of NSGA

NSGA-II [5,6,7] replaces the sharing function approach with a crowded-comparison approach. This approach does not require *any* user-defined parameter for maintaining diversity among population members. Also, this approach has a better computational complexity than NSGA. To describe this approach, we first define a density-estimation metric and then present the crowded-comparison operator.

The NSGA-II [5,6,7] will pay attention to the next notion: Density Estimation. To calculate the density of solutions surrounding a particular solution in population, the algorithm will estimate the average distance between 2 points on either side of this point along each objective. The algorithm will calculate the quantity i_{distance} (the calculation of the cuboid's perimeter formed by utilizing the nearest neighbors as the vertices.

2.2. Crowded-comparison Operator

This is the crowed-comparison operator:

1) rank of nondomination (i_{rank})

2) calculate crowding distance $(i_{distance})$

A partial order $\prec_n$ is identified as

$$i \prec_n j \text{ if } (i_{rank} < j_{rank}) \text{ or } ((i_{rank} = j_{rank}) \text{ and } (i_{dis\tan ce} > j_{dis\tan ce})) \quad (1)$$

This is crowding distance assignment (table 2):

Table 2. Crowding distance assignment

$l = \|I\|$	Number of solutions in *I*
for each *i*, set $I[i]_{dsi\tan ce} = 0$	Initialize the distance
for each objective *m*	
$I = sort(I,m)$	Sort using each objective value
$I[1]_{dis\tan ce} = I[l]_{dsi\tan ce} = \infty$	So that boundary points are always selected
for $i = 2$ $i = 2$ to $(l - 1)$	For all other points

$$I[i]_{dsi\tan ce} = I[i]_{dsi\tan ce} + (I[i+1] - I[i-1].m).m / \left(f_m^{\max} - f_m^{\min}\right) \quad (2)$$

3. STRENGTH PARETO EVOLUTIONARY ALGORITHM 2 + SHIFT-BASED DENSITY ESTIMATION (SPEA2+SDE)

In general, an Evolutionary Algorithm [8] is characterized by facts: a set of solution candidates is maintained, which undergoes a selection process and is manipulated by genetic operators, usually recombination and mutation. By analogy to natural evolution, the solution candidates are called *individuals* and the set of solution candidates is called the *population*. Each individual represents a possible solution, i.e., a decision vector, to the problem at hand where, however, an individual *is not* a decision vector but rather encodes it based on an appropriate structure. Without loss of generality, this structure is assumed to be a vector here, e.g., a bit vector or a real-valued vector.

3.1. SPEA2

The SPEA [8] can be shown as follows:

Input: N

$\overline{N}$

T

p_c

p_m

Output: A

Step 1: Initialization: Create initial population P_0 and generate external set $\overline{P}_0 = \varnothing$. Set $t = 0$.

Step 2: Update of the external set: Set the temporary external set $\overline{P}' = \overline{P}_t$.

This step has 3 smaller steps a + b + c.

a) Copy individuals whose decision vectors are nondominated regarding $m(P_t)$ to $\overline{P}' : \overline{P}' = \overline{P}' + \{i \mid i \in P_t \wedge m(i \in p(m(P_t)))\}$.

b) Remove individuals from $\overline{P}'$ whose corresponding decision vectors are weakly dominated regarding $m(\overline{P}')$, i.e., as long as there exists a pair (i, j) with $i, j \in \overline{P}'$ and $m(i) \succeq m(j)$ do $\overline{P}' = \overline{P}' - \{j\}$.

c) Reduce the number of individuals externally stored by means of clustering…

Step 3: Fitness assignment: Calculate fitness values of individuals in P_t and $\overline{P}_t$.

Step 4: Selection: Set $P' = \varnothing$. For $i = 1, \dots N$ do

a) Select 2 individuals $i, j \in P_t + \overline{P}_t$ at random.

b) If $F(i) < F(j)$ then $P' = P' + \{i\}$ else $P' = P' + \{j\}$.

Step 5: Recombination.

Step 6: Mutation.

Step 7: Termination. Set $P_{t+1} = P'''$ and $t = t + 1$. If $t \geq T$ or another stopping criterion is satisfied then set $A = p(m(\overline{P}_t))$, else go to step 2.

In contrast to SPEA, SPEA2 [9,10,11] uses a fine-grained fitness assignment strategy which incorporates density information. Furthermore, the archive size is fixed, i.e., whenever the number of nondominated individuals is less than the predefined archive size, the archive is filled up by dominated individuals; with SPEA, the archive size varies over time. In addition, the clustering technique, which is invoked when the nondominated front exceeds the archive limit, has been replaced by an alternative truncation method which has similar features but does not loose boundary points.

The SPEA2 algorithm [9,10,11] is as follows:

Input:

N: size of population

$\overline{N}$: archive size

T: max number of generations

Output:

$\boldsymbol{A}$: nondominated set

Step 1:

Initialization: Create initial population $\boldsymbol{P}_0$ and generate external set (empty archive) $\overline{\boldsymbol{P}}_0 = \varnothing$. Set $t = 0$.

Step 2:

Assignment of fitness: Estimate the individual's fitness values in $\boldsymbol{P}_t$ and $\overline{\boldsymbol{P}}_t$.

Step 3:

Selection of environment: All nondominated individuals in $\boldsymbol{P}_t$ and $\overline{\boldsymbol{P}}_t$ will be copied to $\overline{\boldsymbol{P}}_{t+1}$. If size of $\overline{\boldsymbol{P}}_{t+1} > \overline{N}$ then $\overline{\boldsymbol{P}}_{t+1}$ is reduced by means of truncation operator. If size of $\overline{\boldsymbol{P}}_{t+1} < \overline{N}$ then nondominated individuals in $\boldsymbol{P}_t$ and $\overline{\boldsymbol{P}}_t$ are filled to $\overline{\boldsymbol{P}}_{t+1}$.

Step 4:

Terminating: we will establish $\boldsymbol{A}$ to the set of decision vectors expressed by nondominated individuals in $\overline{\boldsymbol{P}}_{t+1}$ if $t \geq T$ or another stopping criterion is satisfied. Stop.

Step 5:

Selection of mating: Binary tournament selection is done with replacement on $\overline{\boldsymbol{P}}_{t+1}$ so as to fill the pool of mating.

Step 6:

Variation: Recombination and mutation are applied to pool of mating and $\boldsymbol{P}_{t+1}$ is established to the resulting population. Set counter $(t = t + 1)$ and go to step 2.

3.2. Shift-Based Density Estimation

The positions of some individuals in population will be shifted by Shift-Based Density Estimation (SDE) while calculating the density of individual p. If an individual operates better than p for an objective, it will be shifted to the same position of p on this objective; otherwise, it will not change. Without any loss of generality, it is assumed that a minimization of MOP will be considered, we will have expression of density $D'(p,P)$ of individual p in population P:

$$D'(p,P) = D(dist(p,q_1'), dist(p,q_2'), ..., dist(p,q_{N-1}')) \quad (3)$$

in which N is size of P, $dist(p,q_i')$ is the index of similarity between p and q_i', q_i' is the shift of q_i $(q_i \in P; q_i \neq p)$:

$$q_{i(j)}' = \begin{cases} p_{(j)}, \text{if } q_{i(j)} < p_{i(j)} \\ q_{i(j)}, \text{otherwise} \end{cases}, j \in (1,2,...,m) \quad (4)$$

If we have individuals p, q_i, q_i', then the respective j-th objective value of these individuals are $p_{(j)}, q_{i(j)}, q_{i(j)}'$; m is number of objectives.

SPEA2+SDE [4] is combination of SPEA2 and SDE.

4. COMPARATIVE CASE STUDY

The article will compare the efficiencies of 2 multi-objective optimization methods: SPEA2+SDE and NSGA-II. The 2 methods will apply to find multi-objective optimization of 5 problems: DTLZ1, DTLZ2, DLTZ3, DTLZ5, DTLZ7. SPEA2+SDE and NSGA-II are in C programming language. The 2 methods will be run by using CPU Intel Core i7 12700H, 16GB RAM.

Table 3. Performance comparison between SPEA2+SDE and NSGA-II regarding the mean and standard deviation values on 5 problems.

Problem	SPEA2+SDE	NSGA-II
DTLZ1	0.2501 (0.1526)	8.1813 (13.5954)
DTLZ2	0.6660 (0.2349)	0.7316 (0.6317)
DTLZ3	0.6653 (0.2443)	466.9296 (722.6675)
DTLZ5	0.6615 (0.2517)	0.7316 (0.6317)
DTLZ7	1.8582 (1.5222)	1.8588 (1.4113)

If the 2 quantities have similar mean, then the quantity which has smaller standard deviation will be more stable. From table 3, for DTLZ2 and DTLZ5, it is investigated that SPEA2+SDE has much smaller standard deviation than NSGA-II, so SPEA2+SDE is much more stable than NSGA-II. So, in these circumstances, SPEA2+SDE is much better than NSGA-II in finding multi-objective optimization.

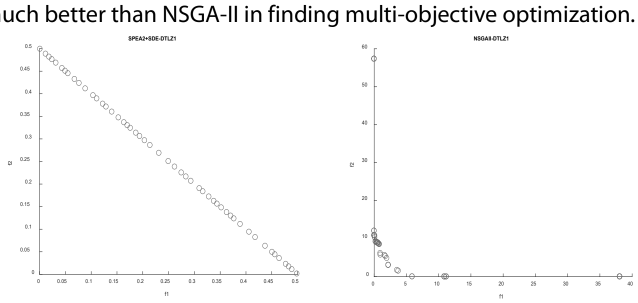

Figure 2. Result comparison between SPEA2+SDE and NSGA-II on 2-objective DTLZ1 problem. The final solutions are shown regarding the 2-dimensional space f_1 and f_2.

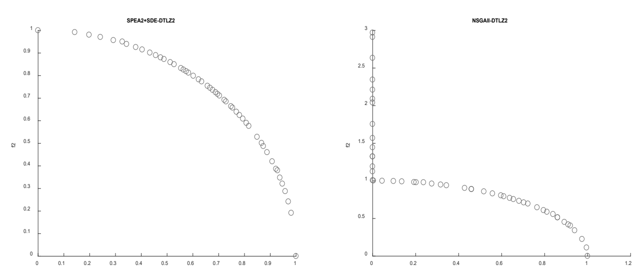

Figure 3. Result comparison between SPEA2+SDE and NSGA-II on 2-objective DTLZ2 problem. The final solutions are shown regarding the 2-dimensional space f_1 and f_2.

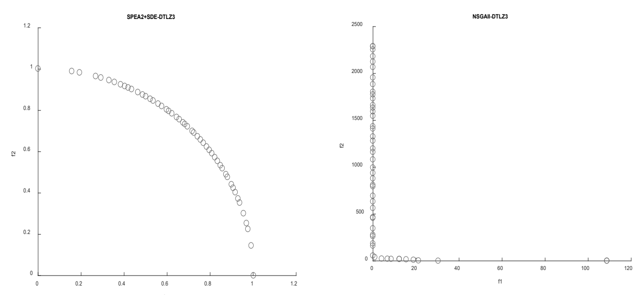

Figure 4. Result comparison between SPEA2+SDE and NSGA-II on 2-objective DTLZ3 problem. The final solutions are shown regarding the 2-dimensional space f_1 and f_2.

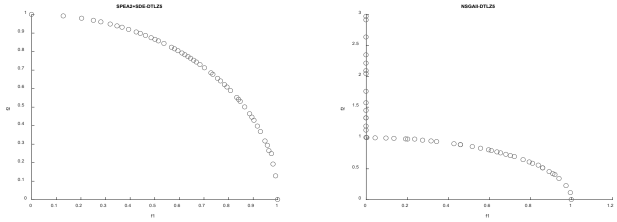

Figure 5. Result comparison between SPEA2+SDE and NSGA-II on 2-objective DTLZ5 problem. The final solutions are shown regarding the 2-dimensional space f_1 and f_2.

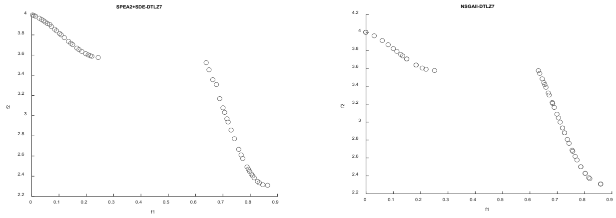

Figure 6. Result comparison between SPEA2+SDE and NSGA-II on 2-objective DTLZ7 problem. The final solutions are shown regarding the 2-dimensional space f_1 and f_2.

From figure 1 to figure 5, the compared results between SPEA2+SDE and NSGA-II will be shown regarding the 2-dimensional space f_1 and f_2. In these figures, the fronts of optimal solutions for 5 problems DTLZ1, DTLZ2, DTLZ3, DTLZ5, DTLZ7 obtained from SPEA2+SDE and NSGA-II will be drawn. The readers can also refer to the exact Pareto fronts of 5 problems DTLZ1, DTLZ2, DTLZ3, DTLZ5, DTLZ7 in reference [13]. From these figures, it is shown that SPEA2+SDE can produce optimal solutions very close to exact Pareto front. In these circumstances, SPEA2+SDE is better than NSGA-II in finding multi-objective optimization and optimal front.

Table 4. Computational time of SPEA2+SDE and NSGA-II

Problem	SPEA2+SDE	NSGA-II
DTLZ1	9.713s	2.723s
DTLZ2	9.286s	2.743s
DTLZ3	8.887s	2.693s
DTLZ5	9.331s	2.719s
DTLZ7	9.091s	2.717s

It is investigated that SPEA2+SDE will cost more computational time than NSGA-II.

In short, SPEA2+SDE is better than NSGA-II though SPEA2+SDE is more time-consuming.

Now, a practical example in civil engineering will be paid attention: test durability of concrete material. Several cylinders of concrete will be created, diameter 10cm, height 20cm. 1 m^3 of concrete will be mixed, maintaining sand, altering 3 variables. The method of creating experimental points is simplex centroid design method. Minitab will be utilized to find regressions. Method of regression is ordinary least square regression. There are 3 variables: silicafume, cement, additive. There are 4 objectives: maximize flexural test, compressive strength, splitting, slump flow.

This is the table 5: Experiments for testing durability of concrete material.

Table 5: Experiments for testing durability of concrete material.

Number	Silicafume (volume/m^3)	Cement (volume/m^3)	Additive (volume/m^3)	Compressive strength (MPa)	Slump flow (cm)	Flexural test on mortars (MPa)	Splitting (MPa)
1	0.12	0.2	0.012	87.4255	57	5.294	1.839912
2	0.12	0.212	0	88.991	60	12.456	3.808835
3	0.0674	0.2482	0.0164	94.3445	59	18.673	4.282457
4	0	0.332	0	61.3085	54	10.195	2.677842
5	0.097	0.2	0.035	76.937	55	13.973	2.434109
6	0	0.297	0.035	35.3725	0	6.647	1.553846

Regression Equations in Coded Units:

Flexural test on mortars (MPa)= 20.57 + 10.12 *Silicafume
+ 10.34 *Cement
- 13.07 *Silicafume*Silicafume
- 2.482 *Cement*Cement
+ 4.950 *Additive*Additive

Compressive strength (MPa)= 99.68 + 54.42 *Silicafume
+ 44.04 *Cement
- 25.37 *Silicafume*Silicafume
+ 3.314 *Cement*Cement
- 5.932 * Additive*Additive

Slump flow (cm)= 66.84 + 65.26 *Silicafume
+ 63.78 *Cement - 37.20 *Silicafume*Silicafume
+ 25.89 *Cement*Cement
- 0.04206*Additive*Additive

Splitting (MPa)= 4.970 + 3.923 *Silicafume
+ 3.916 *Cement - 2.009 *Silicafume*Silicafume
- 1.222 *Cement*Cement
+ 0.9464*Additive*Additive

SPEA2+SDE results:

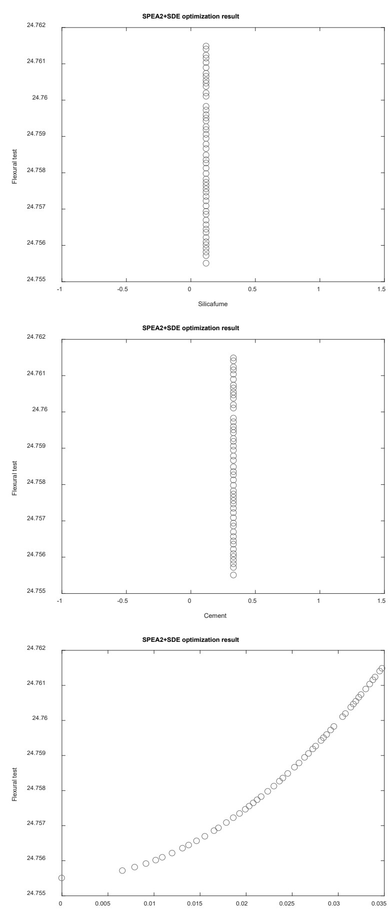

Figure 7. Optimization results for SPEA2+SDE flexural test with 3 variables.

NSGA-II results:

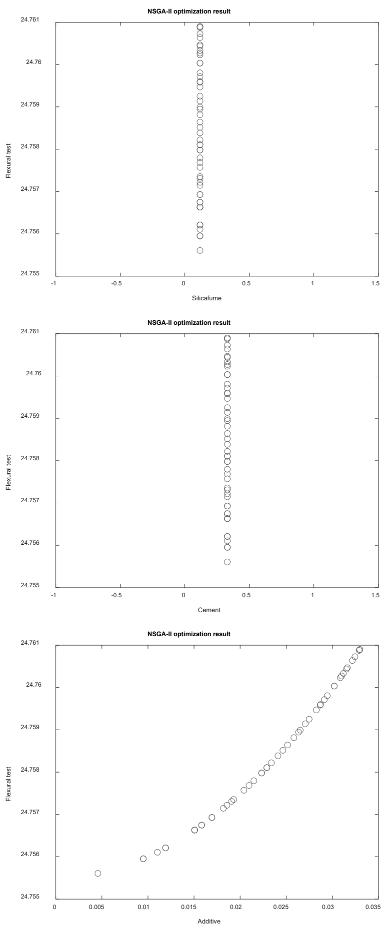

Figure 8. Optimization results for NSGA-II flexural test with 3 variables.

For the practical example, 2 methods SPEA2+SDE and NSGA-II give similar multi-objective optimization results. For the practical example, 2 methods SPEA2+SDE and NSGA-II are equally good.

5. CONCLUSION

In this article, the authors successfully retrieved the C source codes of 2 multi-objective optimization methods including NSGA-II and SPEA2+SDE. NSGA-II is an improvement of NSGA. NSGA-II will replace sharing function with crowded-comparison approach. No user-defined parameter for maintaining the diversity among members of population is needed. NSGA-II can have better computational complexity than NSGA. Besides, SPEA2 can try to eliminate weaknesses of SPEA. There are several differences between SPEA and SPEA2: 1) SPEA2 can enhance fitness assignment; 2) SPEA2 can use a technique of neighbor density estimation; 3) SPEA2 proposed a new method of archive truncation. In this article, a MOO method called SPEA2+SDE will be considered. For the sample problems in this article, it is investigated that SPEA2+SDE can produce optimal solutions very close to exact Pareto front. In these circumstances, SPEA2+SDE is better than NSGA-II in finding multi-objective optimization and optimal front. It is also investigated that SPEA2+SDE will cost more computational time than NSGA-II. In summary, SPEA2+SDE is better than NSGA-II though SPEA2+SDE is more time-consuming.

ACKNOWLEDGMENT

This work belongs to the project grant No: T2023-134 funded by Ho Chi Minh City University of Technology and Education, Vietnam.

REFERENCES

[1] S. Sharma, and V. Chahar (2022), "A Comphrehensive Review on Multi-objective Optimization Techniques: Past, Present and Future," Archives of Computational Methods in Engineering, vol. 29, pp. 5605-5633.

[2] A. A. H. Ahmadini, U. M. Modibbo, A. A. Shaikh, and I. Ali (2021), "Multi-objective Optimization Modelling of Sustainable Green Supply in Inventory and Production Management," Alexandria Engineering Journal, vol. 60, issue 6, pp. 5129-5146.

[3] W.-J. Hong, P. Yang, and K. Tang (2021), "Evolutionary Computation for Large-scale Multi-objective Optimization: A Decade of Progresses," International Journal of Automation and Computing, vol. 18, pp. 155-169.

[4] M. Li, S. Yang, and X. Liu (2014), "Shift-Based Density Estimation for Pareto-Based Algorithms in Many-Objective Optimization", IEEE Transactions on Evolutionary Computation, vol. 18, issue 3, pp. 348-365.

[5] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan (2002), "A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II," IEEE Transactions on Evolutionary Computation, vol. 6, no. 2, pp. 182-197.

[6] H. Ma, Y. Zhang, S. Sun, T. Liu, and Y. Shan (2023), "A Comprehensive Survey on NSGA-II for Multi-objective Optimization and Applications," Artificial Intelligence Review, vol .56, pp. 15217-15270.

[7] L. Lv, and W. Shen (2023), "An Improved NSGA-II with Local Search for Multi-objective Integrated Production and Inventory Scheduling Problem," Journal of Manufacturing Systems, vol. 68, pp. 99-116.

[8] E. Zitzler (1999), Evolutionary Algorithms for Multiobjective Optimization: Methods and Applications, Ph.D thesis, Swiss Federal Institute of Technology Zurich.

[9] E. Zitzler, M. Laumanns, and L. Thiele (2001), "SPEA2: Improving the Strength Pareto Evolutionary Algorithm," TIK Report, ETH Zurich, https://doi.org/10.3929/ethz-a-004284029.

[10] J. Singh (2023), "A Review and Comparison of Two Archive Based Algorithm: SPEA2 and PAES," AIP Conference Preoceedings, vol. 2819, issue 1, https://doi.org/10.1063/5.0137491.

[11] M. Kim, T. Hiroyasu, M. Miki, and S. Watanabe (2004), "SPEA2+: Improving the Performance of the Strength Pareto Evolutionary Algorithm 2," Parallel Problem Solving from Nature – PPSN VIII, Lecture Notes in Computer Science, pp. 742-751.

[12] N. Srinivas, and K. Deb (1994), "Multiobjective Optimization Using Nondominated Sorting in Genetic Algorithms," Evolutionary Computation, vol. 2, issue 3, pp. 221-248.

[13] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler (2005), "Scalable Test Problems for Evolutionary Multi-objective Optimization," Part of Book Series: Advanced Information and Knowledge Processing, pp. 105-145.

[14] M.R. Sharifi, S. Akbarifard, K. Qaderi,and M.R. Madadi (2021), "A new optimization algorithm to solve multi-objective problems," Scientific Reports, vol. 11, https://doi.org/10.1038/s41598-021-99617-x.