

## AN EVALUATION OF A CONSTRAINED MULTI-OBJECTIVE GENETIC ALGORITHM

*Youssef ALIOUI\* & Reşat ACAR\*\**

*\* Ph.D. Student, Atatürk University,  
TURKEY, e-mail: [youssef.alioui14@ogr.atauni.edu.tr](mailto:youssef.alioui14@ogr.atauni.edu.tr)  
ORCID: <https://orcid.org/0000-0003-2915-6917>*

*\*\* Prof. Dr. Atatürk University,  
TURKEY, e-mail: [racar@atauni.edu.tr](mailto:racar@atauni.edu.tr)  
ORCID: <https://orcid.org/0000-0002-0653-1991>*

Received: 30 December 2019, Accepted: 12 May 2020

---

### **ABSTRACT**

*Real world optimization problems involve multiple conflicting objectives (such as minimizing cost while maximizing the quality of a product) and are subject to constraints (such as physical feasibility or budget limitations) which makes them interesting to solve. Over the last decades, evolutionary algorithms have been largely used in solving optimization problems in various fields of science. The aim of this study is to evaluate the performance of a constrained version of the Non-dominated Sorting Genetic Algorithm 2 (NSGA 2), a multi-objective evolutionary optimization algorithm, written in MATLAB. The developed NSGA 2 is compared, in terms of convergence and diversity of the obtained solutions, to a number of popular constrained multi-objective evolutionary algorithms from the literature. Widely used four benchmark problems (including CONSTR, OSY, SRN, and TNK problems) with varying difficulty and type of constraints are reviewed and used. The NSGA 2 obtained the lowest values of inverse generational distance (IGD) values for almost all the problems. These results show that the developed constrained NSGA 2 is an effective technique and is competitive to the other optimization methods in the literature.*

**Keywords:** *Genetic algorithms, Evolutionary algorithms, Non-dominated Sorting Genetic Algorithm 2, Multi-objective optimization, constrained multi-objective optimization.*

---

## 1. INTRODUCTION

Solving multi-objective optimization problems have interested researchers in all the fields of science over the last decades. Real world optimization problems usually involve multiple objectives and are subject to constraints. Multi-objective optimization problems have multiple solutions which produce trade-offs between the objectives. A multi-objective optimization problem can be mathematically formulated, without loss of generality, as follows:

$$\begin{aligned} & \text{minimize } F(x) = (f_1(x), \dots, f_m(x)) \\ & \text{subject to } g_i(x) \geq 0, \quad i = 1, \dots, q \\ & x \in \mathbb{R}^n \end{aligned}$$

A solution  $x$  is feasible if it satisfies all the constraints simultaneously.

A solution  $x_1$  is said to dominate a solution  $x_2$  if both of the following conditions are true:

- For all the objectives  $f_i(x_1) \leq f_i(x_2)$
- There exists at least one objective where  $f_i(x_1) < f_i(x_2)$

A solution is said to be non-dominated if it is not dominated by any other solution. The set of non-dominated solutions is referred to as the Pareto optimal set, the image of this set in the objective space is called the Pareto front

In multi-objective optimization our goal is to find as many pareto optimal solutions as possible. classical methods work with a single solution which will require multiple runs while expecting to find a different solution each time. Evolutionary algorithms work with a population of solutions so multiple pareto solutions can be found on a single run which make them suitable to solve this type of problems. Over the last decades a variety of constrained evolutionary algorithms have been presented. We can cite the following algorithms: Adaptive tradeoff model (ATM) by (Wang, Cai, Zhou, & Zeng, 2008), Infeasibility Driven Evolutionary Algorithm (IDEA) by (Ray, Singh, Isaacs, & Smith, 2009), Self- adaptive penalty (SP) by (Woldesenbet, Yen, & Tessema, 2009), the MOEA/D-IEpsilon by (Fan et al., 2016) and the non-dominated sorting genetic algorithm (NSGA 2) by (Deb, Pratap, Agarwal, & Meyarivan, 2002) which we will be using in our study.

In this study a constrained NSGA 2 was written in Matlab. A sensitivity analysis was performed to determine the best parameters for the algorithm. In the end the performance of the algorithm was evaluated against other algorithms from the literature.

## 2. MATERIALS AND METHODS

Before describing the main loop of the NSGA 2, we are going to present the main procedures of the algorithm: non-dominated sorting, crowding distance, the crowded comparison operator and the constraint handling technique.

### 2.1 Non-Dominated Sorting

This method was first introduced by (Deb et al., 2002). The objective of this procedure is to sort the population into different nondomination levels. In the beginning we define and calculate two substances:

- For each solution  $p$  we calculate the number of solutions dominating  $p$  and we note it as  $np$  the domination count.
- All the solutions that the solution  $p$  dominates are grouped in a set  $S_p$  noted as domination set of  $p$ .

Solutions with a domination count  $n_p=0$  will form the first nondominated front. For each solution  $p$  of this front we reduce the domination count of the members of its domination set by one. Elements whose domination count became zero are put in separate set  $Q$ , this set is the second nondominated front.  $Q$  undergoes the same procedure to identify the third front. this operation is repeated until all the fronts are determined.

## **2.2 Crowding Distance (Diversity Preservation)**

Solutions obtained by the evolutionary algorithm are desired to be diverse and well dispersed along the Pareto front. to preserve diversity in the obtained solutions (Deb et al., 2002) introduced a parameterless diversity preservation technique. In this technique each solution is assigned a crowding distance which can be calculated as follows:

- For each objective function we sort the population in an ascending order, and assign an infinite distance value to the boundary solutions,
- For the in-between solutions we attribute the absolute normalized difference of function values of the two neighboring solutions. This is repeated for all the objective functions,
- The crowding-distance is the sum of distances calculated for each objective function.

It is beneficial to note that a solution with a bigger crowding-distance is less crowded by other solutions.

After the assignment of the crowding distance, every two solutions can be compared using the crowded-comparison operator as follows:

- If they have different nondomination ranks, the one with smaller rank is selected,
- If they have the same nondomination rank, the one with the bigger crowding distance is selected.

## **2.3 Constraint Handling**

constraint handling is critical in solving constrained optimization problems. To handle the constraints a penalty function based technique is widely used, it converts the problem to an unconstrained optimization problem by adding a penalty to the objective function value for every constraint violation. The quality of obtained results depends highly on the choice of the penalty values. Other efficient methods have been successfully applied such as the Ray-Tai-Seow's method (Ray, Tai, & Seow, 2001) and the Constrained tournament method (Deb et al., 2002). In our study we are going to use the latter, this method is applied in the binary tournament selection process where the following rules are followed:

- If both solutions are feasible select the solution with better fitness,
- If one is feasible and the other is not select the feasible one,
- If both solutions are infeasible select the solution with smaller constraint violation,

## **2.4 Genetic Operators (Selection, Crossover, Mutation)**

Selection operator chooses the chromosomes that will generate the next generation. Various selection methods have been presented in the literature: Roulette Wheel Selection, Tournament Selection, Rank Selection.

chromosomes selected by the selection process undergo crossover to provide child chromosomes, which might have better fitness values. In the crossover process two individuals are selected and some parts of these individuals are exchanged to create new individuals called child chromosomes. A variety of crossover techniques have been presented in the literature: one-point crossover, uniform crossover, two point crossover for the binary representation of chromosomes. random crossover, arithmetic crossover, BLX-alpha crossover, simulated binary

crossover for real presentation of chromosomes.

Mutation operator modify a gene of a chromosome randomly, it is applied with a low probability. This operator introduces new material to the population and prevents premature convergence.

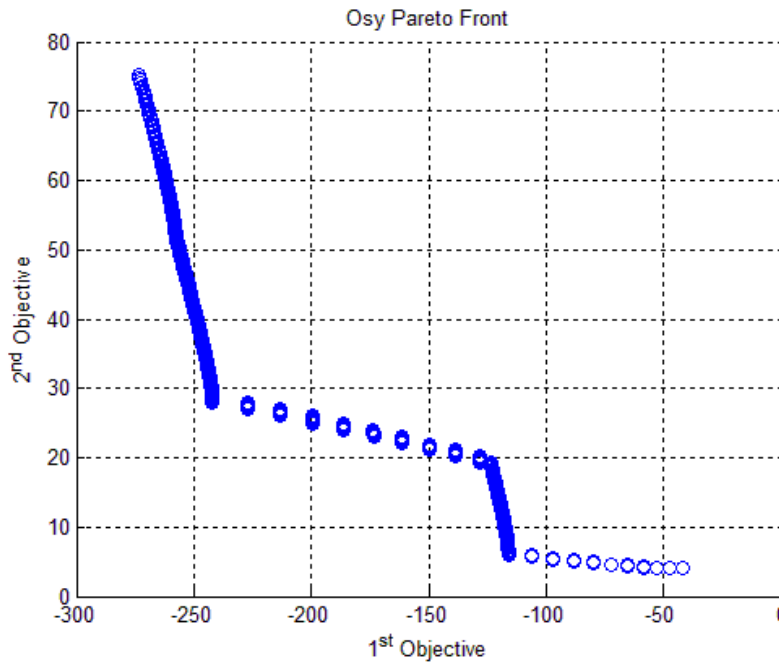
These genetic operators have been detailed by (Kalyanmoy, 2001) and (Goldberg, 1989).

### 2.5 Test problems

To test the performance of the elaborated algorithm we are going to use the following benchmark problems:

Osyc proposed by (Osyczka & Kundu, 1995), the Pareto front is a concatenation of five regions which lies on the intersection of certain constrains, which requires the algorithm to maintain solutions at the different regions which makes the problem difficult to solve. The Pareto front of the problem is shown in figure 1.

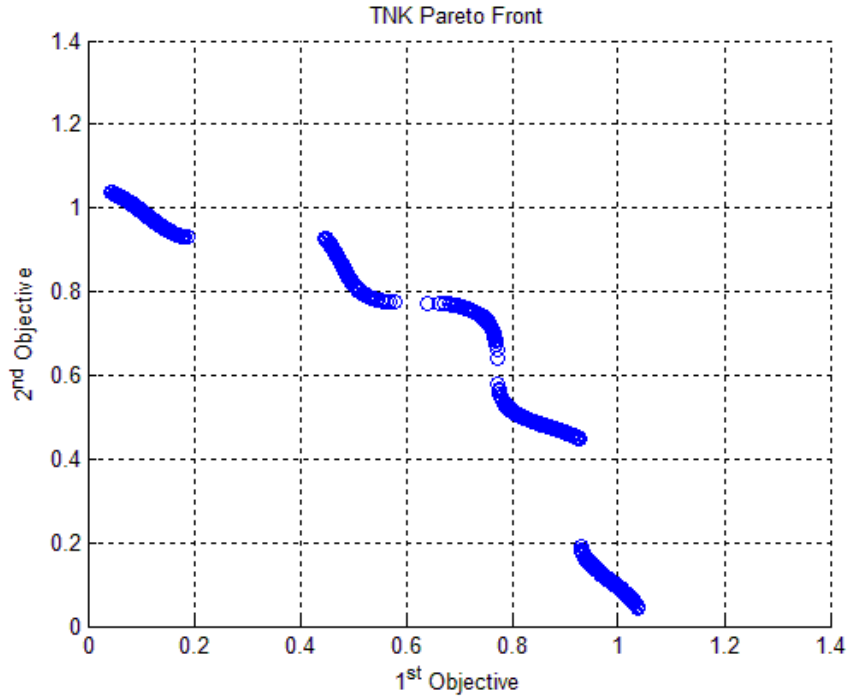
Figure 1. Osy problem's Pareto Front



$$\begin{cases}
 \text{OSY:} \left\{ \begin{array}{l}
 \text{Minimize } f_1(x) = -(25(x_1 - 2)^2 + (x_2 - 2)^2 + (x_3 - 1)^2 + (x_4 - 4)^2 + (x_5 - 1)^2) \\
 \text{Minimize } f_2(x) = x_1^2 + x_2^2 + x_3^2 + x_4^2 + x_5^2 + x_6^2 \\
 \text{Subject to } x_1 + x_2 - 2 \geq 0 \\
 6 - x_1 - x_2 \geq 0 \\
 2 - x_1 + x_1 \geq 0 \\
 2 - x_1 + 3x_2 \geq 0 \\
 4 - (x_3 - 3)^2 - x_4 \geq 0 \\
 (x_5 - 3)^2 + x_6 - 4 \geq 0 \\
 0 \leq x_1, x_2, x_6 \leq 10 \\
 1 \leq x_3, x_5 \leq 5 \\
 0 \leq x_4 \leq 6
 \end{array} \right.
 \end{cases}$$

TNK proposed by (Tanaka, Watanabe, Furukawa, & Tanino, 1995), the fact that the Pareto front is composed of disconnected sets and lies on a non linear constraint surface makes it difficult to find solutions spread across the entire Pareto front. The Pareto front of the problem is shown in figure 2.

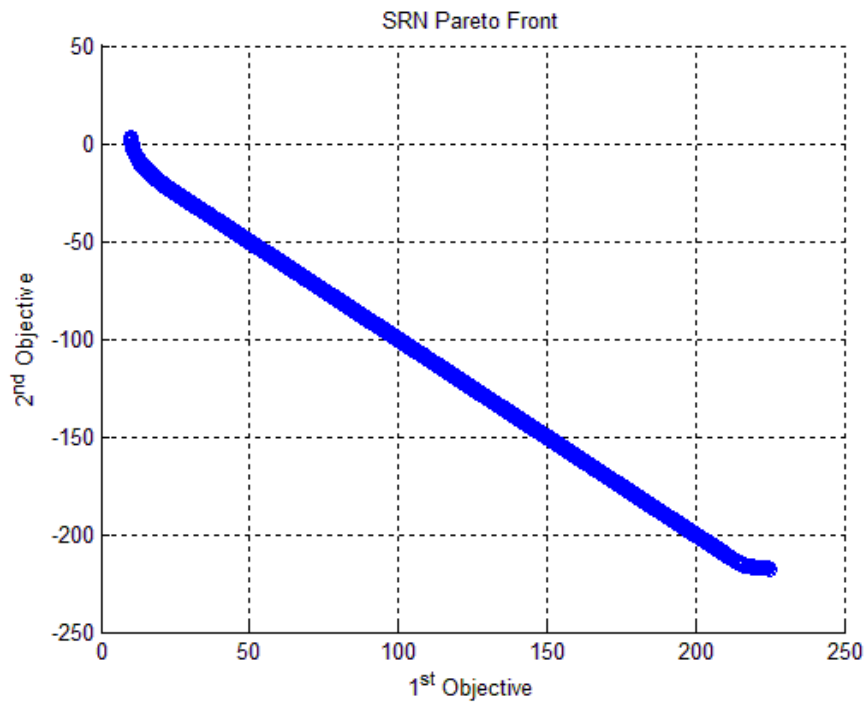
Figure 2. TNK problem's Pareto Front



$$\text{TNK:} \left\{ \begin{array}{l} \text{Minimize } f_1(x) = x_1 \\ \text{Minimize } f_2(x) = x_2 \\ \text{Subject to } x_1^2 + x_2^2 - 1 - 0.1 \cos \left( 16 \arctan \left( \frac{x_1}{x_2} \right) \right) \geq 0 \\ \quad \quad \quad (x_1 - 0.5)^2 + (x_2 - 0.5)^2 \leq 0.5 \\ \quad \quad \quad 0 \leq x_1 \leq \pi \\ \quad \quad \quad 0 \leq x_2 \leq \pi \end{array} \right.$$

SRN proposed by (Srinivas & Deb, 1994) , the constraints of this problem eliminate a part of the unconstrained Pareto-Optimal front which make it difficult to solve. The Pareto front of the problem is shown in figure 3.

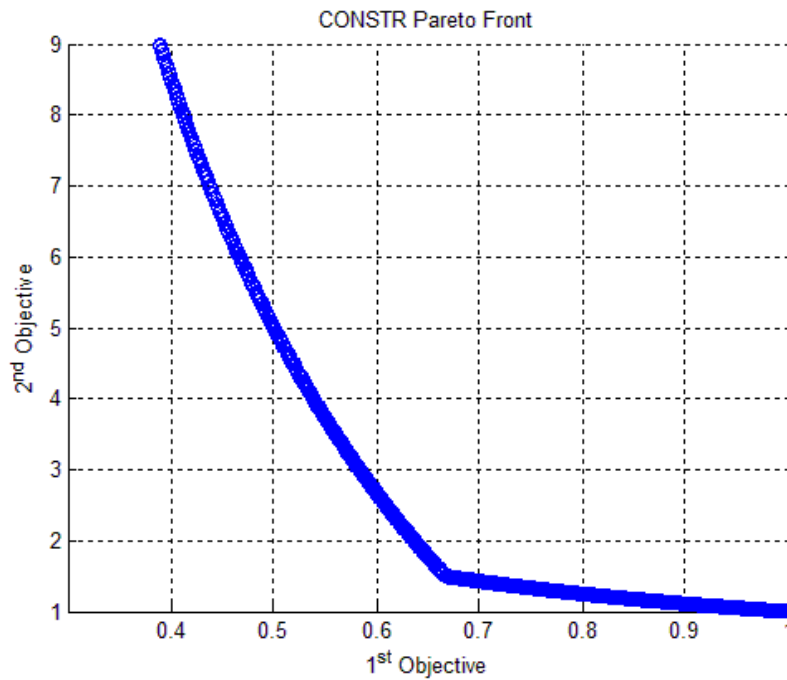
**Figure 3** SRN problem's Pareto Front



$$\text{SRN: } \left\{ \begin{array}{l} \text{Minimize } f_1(x) = 2 + (x_1 - 2)^2 + (x_2 - 1)^2 \\ \text{Minimize } f_2(x) = 9x_1 - (x_2 - 1)^2 \\ \text{Subject to } x_1^2 + x_2^2 \leq 225 \\ \quad \quad \quad x_1 - 3x_2 + 10 \leq 0 \\ \quad \quad \quad -20 \leq x_1 \leq 20 \\ \quad \quad \quad -20 \leq x_2 \leq 20 \end{array} \right.$$

CONSTR-Ex proposed by (Deb et al., 2002), in this problem the Pareto front is a concatenation of the first constraint boundary and a some part of the unconstrained pareto front. The Pareto front of the problem is shown in figure 4.

Figure 4 CONSTR-EX problem's Pareto Front



$$\text{CONSTR: } \left\{ \begin{array}{l} \text{Minimize } f_1(x) = x_1 \\ \text{Minimize } f_2(x) = \frac{1 + x_2}{x_1} \\ \text{Subject to } x_2 + 9x_1 \geq 6 \\ \quad \quad \quad -x_2 + 9x_1 \geq 1 \\ \quad \quad \quad 0.1 \leq x_1 \leq 1 \\ \quad \quad \quad 0 \leq x_2 \leq 5 \end{array} \right.$$

### 2.6 Performance Criteria

In multi-objective optimization we aim to have solutions as close to the Pareto optimal front and spread as possible along the obtained nondominated front. To evaluate the performance of our algorithm we have used the inverse generational distance (IGD). This metric is able to assess the convergence and the diversity of the obtained solutions simultaneously. It is defined as the average of the minimum Euclidean distance between elements of the obtained non dominated front and the optimal Pareto front. A small value of the generational distance means a better convergence toward the Pareto optimal front and a diverse set.

### 3. RESULTS AND DISCUSSION

To select parameters of the algorithm (crossover probability, crossover technique, mutation probability, population size) a sensitivity analysis has been performed and the following parameter values have been selected:

- Crossover probability: 0.9
- Crossover technique: Blx- $\alpha$  and  $\alpha=0.5$
- Mutation probability: 0.05
- Population size: 150
- Iterations number: 1000

Real Pareto front values of the used problems were obtained from (Coello, Lamont, & Van Veldhuizen, 2007). The algorithm was run 30 independent times. Results obtained by the NSGA 2 are illustrated in figure 5 we can conclude that the algorithm converged for all the four problems.

Figure 5 Real Pareto Front vs Obtained Pareto Front

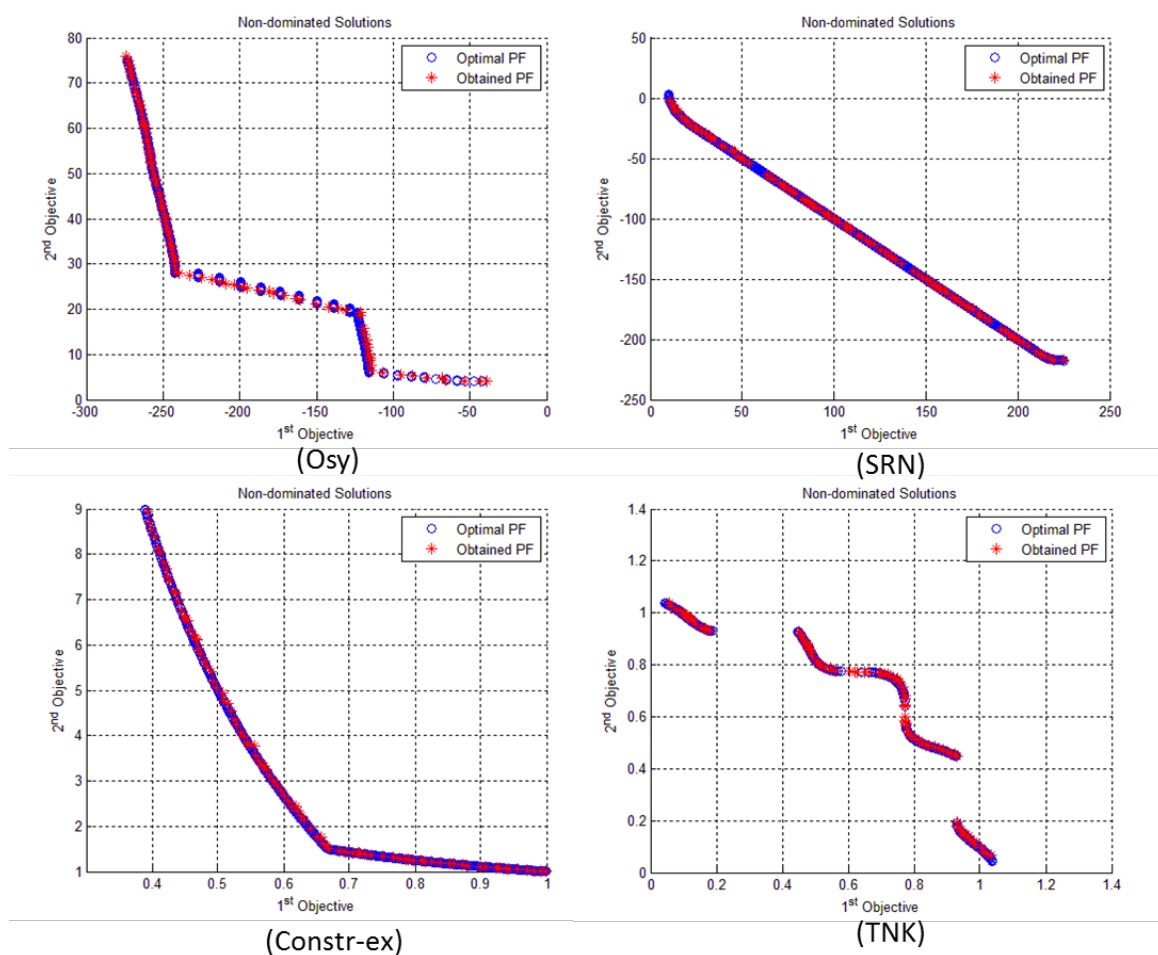


Table 1 shows the mean and variance of the inverse generational distance obtained by (Fan et al., 2017) and those we obtained by the algorithm we developed for the resolution of the four problems.

Table 1 Performance of the algorithms in terms of the mean and standard deviation values of IGD

		ATM	CMOEA/D	IDEA	MOEA/D-CDP	MOEA/D-SR	NSGA-II-CDP	SP	MOEA/D-Epsilon	Our NSGA II
CONTR	mean	1.22E-02	2.33E-02	1.09E-02	2.33E-02	4.52E-02	1.36E-02	1.22E-02	1.28E-02	<b>5.18E-03</b>
	variance	3.69E-03	5.28E-05	2.72E-04	5.70E-05	7.17E-03	7.74E-03	6.31E-03	1.29E-03	<b>1.78E-07</b>
OSY	mean	1.16E+01	1.16E+01	2.75E+00	1.20E+01	1.92E+01	1.14E+01	1.15E+01	4.31E+00	<b>1.10E+00</b>
	variance	1.04E+00	3.79E+00	1.73E+00	4.77E+00	4.13E+00	1.35E+00	8.60E-01	8.67E-01	<b>1.19E-02</b>
SRN	mean	5.08E-01	1.05E+00	6.53E-01	1.08E+00	1.06E+00	5.13E-01	5.11E-01	3.56E-01	<b>1.73E-01</b>
	variance	1.15E-02	5.59E-02	2.37E-02	4.58E-02	5.71E-02	1.78E-02	1.31E-02	8.26E-03	<b>1.11E-03</b>
TNK	mean	4.33E-03	2.52E-03	2.52E-03	2.53E-03	2.69E-02	8.09E-03	4.12E-03	<b>1.82E-03</b>	3.18E-03
	variance	1.01E-03	4.13E-05	7.43E-05	3.60E-05	3.25E-03	4.47E-03	9.48E-04	4.47E-05	<b>6.07E-08</b>



The NSGA 2 outperformed the other algorithms for all the problems except for the TNK where the MOEA/D-IEpsilon (Fan et al., 2016) had the best IGD. This indicates that the NSGA 2 is still a powerful tool in constrained optimization problems.

## REFERENCES

- COELLO, C. A. C., LAMONT, G. B., & VAN VELDHUIZEN, D. A. (2007). *Evolutionary algorithms for solving multi-objective problems* (Vol. 5): Springer.
- DEB, K., PRATAP, A., AGARWAL, S., & MEYARIVAN, T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2), 182-197. doi:Pii S 1089-778x(02)04101-2 Doi 10.1109/4235.996017
- FAN, Z., LI, H., CAIMIN, W., LI, W., HAN, H., CAI, X., & CAI, Z. (2016, 6-9 Dec. 2016). *An improved epsilon constraint handling method embedded in MOEA/D for constrained multi-objective optimization problems*. Paper presented at the 2016 IEEE Symposium Series on Computational Intelligence (SSCI).
- FAN, Z., YI, F., LI, W., JIEWEI, L., CAI, X., & CAIMIN, W. (2017, 5-8 June 2017). *A comparative study of constrained multi-objective evolutionary algorithms on constrained multi-objective optimization problems*. Paper presented at the 2017 IEEE Congress on Evolutionary Computation (CEC).
- GOLDBERG, D. E. (1989). *Genetic Algorithms in Search, Optimization and Machine Learning*: Addison-Wesley Longman Publishing Co., Inc.
- KALYANMOY, D. (2001). *Multi-Objective Optimization Using Evolutionary Algorithms*: John Wiley & Sons, Inc.
- OSYCZKA, A., & KUNDU, S. (1995). A New Method to Solve Generalized Multicriteria Optimization Problems Using the Simple Genetic Algorithm. *Structural Optimization*, 10(2), 94-99. doi:Doi 10.1007/Bf01743536
- RAY, T., SINGH, H. K., ISAACS, A., & SMITH, W. (2009). Infeasibility Driven Evolutionary Algorithm for Constrained Optimization. In E. Mezura-Montes (Ed.), *Constraint-Handling in Evolutionary Optimization* (pp. 145-165). Berlin, Heidelberg: Springer Berlin Heidelberg.
- RAY, T., TAI, K., & SEOW, K. C. (2001). MULTIOBJECTIVE DESIGN OPTIMIZATION BY AN EVOLUTIONARY ALGORITHM. *Engineering Optimization*, 33(4), 399-424. doi:10.1080/03052150108940926
- SRINIVAS, N., & DEB, K. (1994). Multiobjective Optimization Using Nondominated Sorting in Genetic Algorithms. *Evolutionary Computation*, 2(3), 221-248. doi:10.1162/evco.1994.2.3.221
- TANAKA, M., WATANABE, H., FURUKAWA, Y., & TANINO, T. (1995). GA-based decision support system for multicriteria optimization. *1995 Ieee International Conference on Systems, Man and Cybernetics, Vols 1-5*, 1556-1561.
- WANG, Y., CAI, Z., ZHOU, Y., & ZENG, W. (2008). An Adaptive Tradeoff Model for Constrained Evolutionary Optimization. *IEEE Transactions on Evolutionary Computation*, 12(1), 80-92. doi:10.1109/TEVC.2007.902851
- WOLDESENBET, Y. G., YEN, G. G., & TESSEMA, B. G. (2009). Constraint Handling in Multiobjective Evolutionary Optimization. *IEEE Transactions on Evolutionary Computation*, 13(3), 514-525. doi:10.1109/TEVC.2008.2009032