International Journal of Innovative Computing, Information and Control
Volume **18**, Number **4**, August **2022**

ICIC International ©2022 ISSN 1349-4198
pp. 1089–1102

A NOVELTY-BASED APPROACH TO CAPTURING UNIQUE SOLUTIONS IN CONTINUOUS MULTIOBJECTIVE PROBLEMS

Mahmoud Mohamed Nabil[1], Haitham Seada[2], Magdi Zakaria Rashad[3] and Amro Abd El Latif[1]

[1]Computer Science Department
Zagazig University
Zagazig, Egypt
{ mahmoudnabil20164589; amro43210 }@gmail.com

[2]Lead Data Scientist
Ford Motor Company
Michigan, United States
hseada@ford.com

[3]Faculty of Computers and Information Systems
Mansoura University
Mansoura, Egypt
Magdi_z2011@yahoo.com

Received January 2022; revised May 2022

Abstract. *Novelty search is used in this study to reduce the half of selection space in U-NSGA-III algorithm. We modify the selection mechanism to consider the novelty of solutions during selection. And in this way, we reduce time and lose important unique solutions because their uniqueness is only exhibited in the changing space (not the objective space) which is not taken into consideration by any of these earlier algorithms. We use U-NSGA-III to implement our algorithm after some modifications to it. The most important modification is introducing additional constraints restricting the feasible region of the problem. The new constraints deem half the originally feasible objective space infeasible, except for a narrow region that – due to the new setup – has become separated from the rest of the feasible space.*
Keywords: Novelty search, Unique solution, Evolutionary multi-objective optimization, Evolutionary algorithm, NSGA-III

1. **Introduction.** Finding a unique solution is quite challenging using standard methods of optimization, for example, EAs (evolutionary algorithms) are algorithms that have been created to tackle single-objective problems, more than 3 objectives, and more than 15 objectives optimization problems. Thus, over the last years, some efforts in using many objectives evolutionary algorithms are done. However, we use U-NSGA-III in this study to improve our search about novelty and add novelty algorithm in the selection step in EvoLib library [12,15-21].

In the last twenty years, evolutionary multi-objective optimization (EMO) algorithms have shown that they can be used to solve multi-objective issues with optimization so we use U-NSGA-III made by Seada and Deb and add novelty search as the third constraint and see how the algorithm goes better or not and after having bad result, we add novelty in selection step in Seada and Deb's U-NSAG-III and we find better result [15,22-25].

DOI: 10.24507/ijicic.18.04.1089

We will implement our use of U-NSGA-III in this paper in part three and how we change in EvoLib implementation of Seada and Deb by adding the novelty search code to the selection step and every step we do.

We add novelty search as the third constraint in ZDT problems and we make many runs to see if we improve our problems' time and how much time it takes finding a rare solution but we did not find a rare solution in our test space. The new constraints deem half the originally feasible objective space infeasible, with the exception of a narrow region that – due to the new setup – has become separated from the rest of the feasible space. Equation (3) shows the definition of the modified ZDT-1 (ZDT-1-restricted).

$$0 \leq x_1 \leq 0.5$$
$$0.9 \leq x_1 \leq 0.901$$

Finally, in Section 4.2 we report our result table and explain how and how many populations and runs we use and how much better the new algorithm is working well, and find our rare solution.

A. This section is useful for comparing several rare solutions finding in algorithms with and without novelty search.
B. We calculate the mean of generation in the algorithm before and after adding novelty search and how much it gets better.
C. We can find mean IGD gets better before and after adding novelty search in our algorithm.

The remainder of the paper is organized as follows. In Section 2, we will introduce background about evolutionary algorithms and their use in our research. Section 3 exposes the proposed method. The empirical results and discussion are introduced in Section 4. Finally, the paper is concluded in Section 5.

2. Background.

Evolutionary algorithm (EA) is an expression used to refer to population-based stochastic direct search algorithms that resemble natural evolution in some ways. Prominent representatives of such algorithms are genetic algorithms, evolution strategies, evolutionary programming, and genetic programming. Using the evolutionary cycle as a guide, EA has been used in ways. Reproduction, mutation, recombination, and selection are all examples of biological evolution. Individuals in a population are represented as candidate solutions to the optimization problem, and the fitness function determines the solution's uniqueness. Meta-heuristic methods are applied successfully in several applications such as virtual machine placement [34,35].

In traditional GAs, cross-pollination and mutation are the primary mechanisms for producing progeny. These two operators have a direct effect on specific solutions (parents). To make children, the crossover operator did a pair of parents and swapped portions of these two parents. To produce offspring, the mutation operator alters a portion of a parent solution at random. Scatter search generates new solutions by combining selected solutions in linear combinations, with heuristic improvement and a rounding process.

The NSGA-II does not rise to the occasion effectively solving two objective optimization problems, and they have been discovered to be effective at solving single-objective optimization issues. According to the NSGA-II framework, to solve single- and multi-objective optimization issues, a single-optimizer algorithm was proposed. This is because the NSGA-II technique of selection's domination operator is turned into an ordinal comparison operator, this is a necessary step in arriving at the best single-objective optimization issue solution. As a result, these multi-objective optimization methods can be regarded as a collection of unified ways to handle single- and multi-objective optimization problems. However, for many objective difficulties, a single-optimizer was plainly insufficient.

NSGA-III is in charge of a large, random population S as well as a series of widely dispersed predetermined Q-dimensional reference points H on a unit hyperplane with a one-dimensional normal vector covering the entire RM+ area. The hyper-plane is positioned so that it crosses each objective axis at the same point. The approach of Das and Dennis [37] is used to place objects

$$\mathrm{H} = \begin{pmatrix} Q + p - 1 \\ p \end{pmatrix}$$

On the hyperplane, reference points with $(p + 1)$ points along each border. S is chosen as higher than the lowest multiple of four H for the population size, with the premise that for each point of reference, one member of the population is expected to be discovered.

The following operations are performed at any generation. To begin, the entire population is separated into non-domination levels, it is done the same manner in NSGA-II, following the non-dominated sorting principle. Using standard recombination and mutation operators, an offspring population is formed. Because only one member of the population is anticipated to be located for each and every reference point. In NSGA-III, there will be no selecting operation that is required.

Like any other selection operator, it is possible that there will be competition that can be arranged between different reference points. After that, a composite population is established. Thereafter, starting with the first non-dominated front, points are chosen one by one until no more solutions from that front may be included. This procedure is also the same as the NSGA-II procedure. Let us draw a line through the final front that could not be totally chosen. In most cases, only a few options must be chosen in order to use. Next, we will talk about a niche-preserving operator. First, the current population spread is used to normalize each individual in the population so that all objective vectors and reference points have similar values. To associate each member with a given reference point, the shortest perpendicular distance between each population member and a reference line constructed by joining the origin with a selected reference point is employed. The members connected with the least represented reference locations are then chosen using a rigorous niching method. The niching strategy focuses on choosing a population member for as many provided reference points as possible.

If the genetic variation operators (recombination and mutation) are capable of producing respective solutions, the entire process should find one population member corresponding to each supplied reference point close to the Pareto-optimal front, with continuous stress for highlighting non-dominated individuals. The use of a widely spread reference point assures that the end result is a well-distributed set of trade-off points.

The initial NSGA-III studies have been proved to be effective with problems ranging from three to fifteen objectives. The fact that NSGA-III does not require any additional settings is a major feature. Without adding any new parameters, the function was also enhanced to handle restrictions. The reference point set is adaptively updated on the fly depending on the association status of each reference point throughout numerous generations, according to the findings of that study.

The nondominated sorting genetic algorithm II is abbreviated as NSGA-II. A selection operator is described that generates a combination between the collecting parent and offspring populations and pick the best, which alleviates the issues of (MOEA) (with respect to fitness and spread).

In the NSGA-II, we apply a crowded-comparison technique in the stade of sharing function approach, which eliminates the issues of another way. To discuss this approach, we establish a density-estimation metric and then offer the crowded-comparison operator, which does not require any user-defined parameters for maintaining variety among population members.

- Density Estimation: We calculate the average distance of two points on either side of this point along each of the objectives to get a rating of the density of solutions surrounding a specific solution in the population. This number is a rating of the cuboid's perimeter when the nearest neighbors are used as vertices (call this the crowding distance). The average side length of the cuboid is the crowding distance of the solution at its front (shown with solid circles). The crowding-distance calculation necessitates sorting the population in ascending order of magnitude according to each objective function value. The boundary solutions (solutions with the smallest and biggest function values) are then given an infinite distance value for each objective function. The absolute standardized difference in the function values of two close solutions is used to assign a distance value to all other medium solutions. Other objective functions are added to the equation. The total of individual distance values corresponding to each objective is used to calculate the overall crowding-distance value. Before computing the crowding distance, each objective function is straightened, the process for computing the crowding-distance of all solutions in a nondominated set β. The parameters are related to the maximum and minimum values of the objective function, as well as the objective function value of the person in the set. The sorting algorithm determines the degree of difficulty of this procedure. The aforementioned approach has O(MNLogN) computational complexity since it involves independent sortings of N at most solutions (where all population members are in the same front). We can compare two solutions for their extent of proximity with other solutions after all population members in the set have been assigned a distance measure. In some ways, a solution with a less value of this distance measure is crowded by other answers. This is precisely what the suggested crowded-comparison operator (detailed below) compares. Although this example shows how to compute the crowding-distance for two objectives, the approach may be applied to any number of objectives.
- Crowded-Comparison Operator: The crowded-comparison operator directs the algorithm's selection process toward a uniformly spread-out Pareto optimum front at various phases. Assume that each person in the population has two characteristics:
  1) nondomination rank
  2) crowding distance

  A partial order is currently defined as if or and. That is, we choose the solution with the lower (better) nondomination rank for two solutions with different nondomination ranks. If both solutions are located on the same front, we favour the one that is located in a less congested area. The paper is with these three innovations – a fast nondominated sorting procedure, a fast crowded distance estimation procedure, and a simple crowded comparison operator.

Some work was made in the last few years using NSGA-III and we will explain five ways of use:

- In water distribution systems (WDS), a number of issues have arisen, resulting in significant harm, financial loss, and long-term societal consequences. As a result, the most effective technique is to use a water quality sensor to monitor WDS in real time. The location of such a sensor in a water distribution network (WDN) has become critical all over the world. The purpose of this system is to install a water quality sensor and monitor probable pollution incidents to reduce the danger of contamination [26].

- The main idea of this is to minimize the emission of fossil fuel-fired in power plants. This way is called the many objectives combined economic emission dispatch (MO-CEED) problem. The main benefit of fossil fuel-fired in power plant management is optimal arranging of active power to committed units so as to achieve the least possible generation fuel cost as well as emission value. The main goal of electricity generation utilities is with huge consciousness toward environmental protection and the clean air [27].
- Software engineering problems dealing with many objectives improvements are required. One of these issues is software restructuring, which necessitates a restructuring sequence in order to optimize multiple software standards. A set of 15 quality standards is used to evaluate automated refactoring solutions [28].
- Nowadays the effective allocation of shipments is very important for many companies which deliver goods around the world. The most important thing is a multi-objective optimization of an optimal shipment allocation. In the field of scheduling, there are limiting the number of vehicles employed, maximizing vehicle usage, and maximizing operator route familiarity, as well as minimizing cost and time. Human resource management manufacturing and scheduling in the logistic sector is how to develop ways to distribute goads to clients, which has turned into a multi-objective optimization problem since numerous elements must be taken into account, and companies must create solutions that benefit from all angles [29].
- Wireless sensor networks (WSN) are important for now and tomorrow, so some researchers developed and adopted the most important approaches for wireless sensor networks deployment (WSND) optimization. Depending on guiding searching it is NSGA-III under the connectivity limitation, the best strategy to provide maximum coverage while consuming the least amount of energy is to have the longest network lifetime possible. WSNs are classified into monitoring and tracking. Monitoring uses in the defense, agriculture, health care, and intelligent home. Pursuit in some industries and armed in this search the authors used deployment to find the best place the sensor can give him the best result. MOASA (Multi-objective optimization approach for sensor arrangement) algorithm is developed in this paper to (i) maximize the binary coverage, (ii) minimize the redundancy of coverage, and (iii) minimize the deployed sensors number, and developed multi-objective GA (FD-MOGA) to solve three-dimensional deployment problems where the objectives are maximizing detection level, and minimizing energy consumption [30].

3. **Proposed Approach.** NSGA-III uses conventional tournament selection to pick the parents that will undergo genetic operations. As shown in Algorithm 1, conventional tournament selection picks solutions based on the following criteria in order:

1) Feasibility
2) Constraint violation

If these three criteria are not enough to differentiate solutions (e.g., two feasible solutions on the same front), one solution is picked randomly. Since ranking is not among the used selection criteria in NSGA-III, the algorithm applies notably less selection pressure compared to its predecessor, NSGA-II. This intentional mild selection pressure allows NSGA-III the opportunity to explore the exponentially increasing search space as the dimensionality of the problem (the number of objectives) increases. On the single-objective

```
Algorithm 1: conventionalTournamentSelection (s1, s2)
Inputs: Two randomly picked solutions s1 and s2
Output: either s1 or s2
If is Feasible(s1) AND ~is Feasible(s2)
        return s1
Else If ~is Feasible(s1) AND is Feasible(s2)
        return s2
Else If ~is Feasible(s1) AND ~is Feasible(s2)
        If constraint Violation(s1) > constraint Violation(s2)
                return s1
        Else
                return s2
        End
Else If
        return random Pick (s1, s2)
End
```

```
Algorithm 2: nichingBasedTournamentSelection (s1, s2)
Inputs: Two randomly picked solutions s1 and s2
Output: either s1 or s2
If dir(s1) = dir(s2)
    If rank(s1) < rank(s2)
        return s1
    Else If rank(s2) < rank(s1)
        return s2
    Else If perpendicularDist(s1) < perpendicularDist(s2)
        return s1
    Else
        return s2
    End
Else
    random Pick (s1, s2)
End
```

level, this mild-pressure selection mechanism was shown to degrade the performance of the algorithm [5].

In 2015, Seada and Deb proposed U-NSGA-III (Unified NSGA-III) which resolved the single-objective issues faced by NSGA-III while maintaining the same outstanding performance in problems of multi-objective optimization [15]. U-NSGA-III adopts a slightly different section approach where only solutions within the same niche (attached to the same reference direction) are compared to each other, hence prioritizing niching over domination (ranking) as shown in Algorithm 2.

Neither NSGA-II nor NSGA-III nor U-NSGA-III pays any attention to the superiority of a solution in the design (variable) space. They focus solely on the objective space. In our proposed algorithm, we modify the selection mechanism to consider the novelty of solutions during selection. As outlined in Algorithm 3, 50% of the selections are made based on novelty while the other half use the same selection mechanism of U-NSGA-III. Consequently, the more unique the solution is from a design (variable) point of view, the higher the probability of it being selected as a parent solution. Earlier algorithms tend

Algorithm 3: noveltyTournamentSelection (s1, s2)
Inputs: Two randomly picked solutions s1 and s2
Output: either s1 or s2

```
If random () < 0.5
    If novelty(s1) > novelty(s2)
        return s1
    Else
        return s2
    End
Else
    return nichingBasedTournamentSelection (s1, s2)
End
```

to lose these unique solutions because their uniqueness is only exhibited in the variable space (not the objective space) which is not taken into consideration by any of these earlier algorithms.

In this study, we use Euclidean distance (Equation (1)) as our base proximity metric for measuring the novelty of a solution:

$$dist(p,q) = \sqrt{\sum_{i=0}^{n} (p_i - q_i)^2} \tag{1}$$

Hence, novelty can be defined as the sparseness of the area surrounding the solution as shown in Equation (2).

$$novelty(x) = 1 \Big/ k \sum_{i=0}^{k} dist(x, u_i) \tag{2}$$

where u_i is the j-the-nearest neighbor of i within the population and in the archive of novel individuals. Distance $dist(x, u_i)$ is a domain-dependent metric that evaluates the "difference" between individuals x and i. Equation (1) is the standard Euclidean distance equation, while Equation (2) calculates the mean distance of the solution under consideration to its K-nearest neighbors. The main contribution of this study is encapsulated in Algorithm 3 which shows how any distance-based novelty metric can be used within the framework of an evolutionary multiobjective optimization algorithm, to emphasize unique solutions. This notion can be generalized over other distance formulas based on the domain of the study.

4. Results.

4.1. Experimental setup.

Our implementation is a modification of EvoLib (an open-source implementation of NSGA-II, NSGA-III, and Unified NSGA-III (U-NSGA-III)) [5]. NSGA-II is a well-known evolutionary multiobjective optimization (EMO) algorithm that can handle up to two objectives efficiently. NSGA-III is a widely used algorithm for handling multi (2, 3) and many (3 to 20) objectives. However, NSGA-III was shown to be less efficient with single objective optimization problems. U-NSGA-III is an improved version of NSGA-III that enhances the performance of the original algorithm in one objective optimization problem while maintaining its high performance in more than one objective (from 3 to 15) optimization problems [15]. Most existing test problems [6,7,14] are not designed to examine the capability of the algorithm to attain rare solutions. In this study, we modified some of the problems of the well-known ZDT test suite to serve as testbeds for the proposed approach (with the exception of ZDT-5 whose peculiar nature excludes it from the scope of this study). The original problems have a continuous objective space. In ZDT-1, ZDT-2, ZDT-3 and ZDT-4, $f_1(x) = x_1$. This implies that the difficulty to reach a Pareto point does not change from one point to another in these problems. In other words, these problems do not need novelty search to attain particularly difficult sections of the Pareto front. Even ZDT-6, where the objective space becomes less dense towards the lower end of $f_1(x)$, still has a continuous objective space. This continuity allows traditional evolutionary algorithms to reach less dense regions gradually as optimization proceeds.

In our version of the problems, we introduced additional constraints restricting the feasible region of the problem. The new constraints deem half the originally feasible objective space infeasible, with the exception of a narrow region that – due to the new setup – has become separated from the rest of the feasible space. Equation (3) shows the definition of the modified ZDT-1 (ZDT-1-restricted).

$$\begin{aligned} & g(x) = 1 + \frac{9\left(\sum_{i=2}^{n} x_i\right)}{n-1} \\ & f_1(x) = x_1 \\ & f_2(x) = g(x)\left[1 - \sqrt{\frac{x_1}{g(x)}}\right], \quad x \in [0,1] \end{aligned} \tag{3}$$

where

$$\begin{gathered} 0 \leq x_1 \leq 0.5 \\ 0.9 \leq x_1 \leq 0.901 \end{gathered}$$

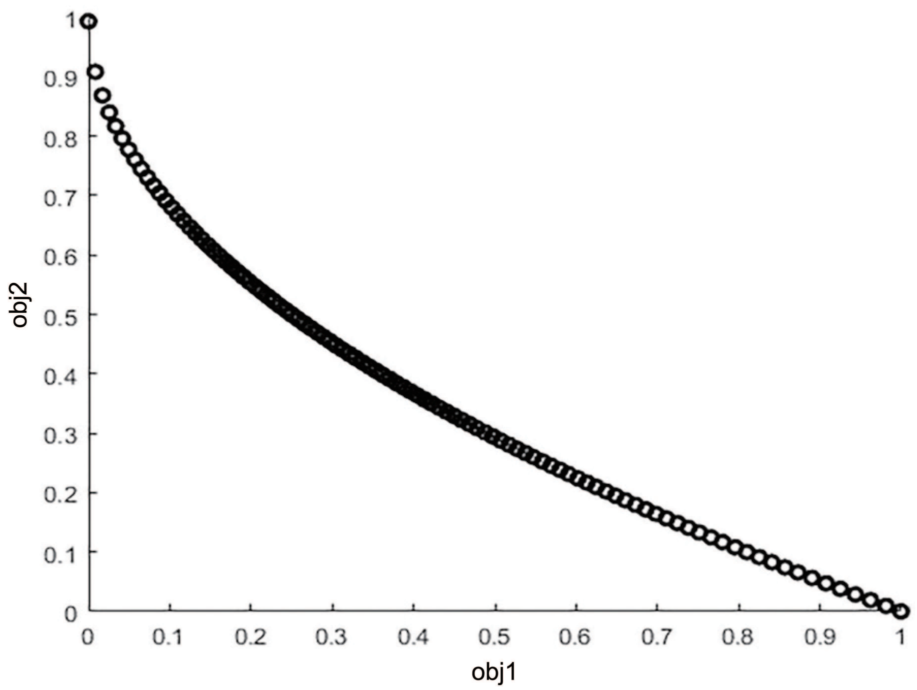

FIGURE 1. ZDT-1 before adding novelty constraints

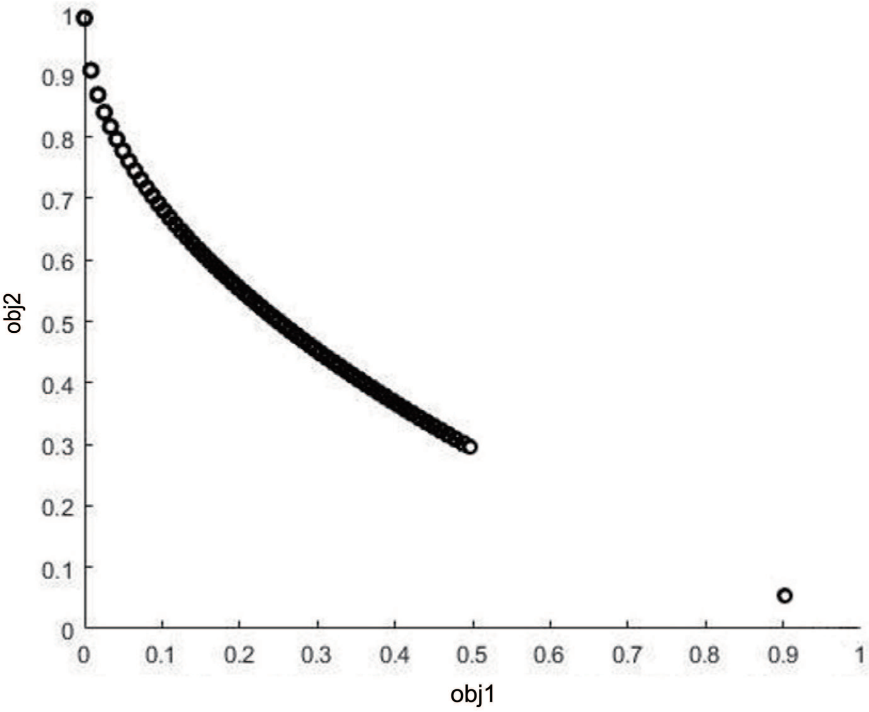

Figure 2. ZDT-1 after adding novelty constraints

4.2. **Table of results.** We find in our lab test that is in ZDT-1 problem rare solution before and after adding novelty search that the number of rare solutions is different and is better and after calculating statistical significance "the p-value" we prove that is better.

	Success count		Gen index (mean)			IGD		
	novelty	standard	novelty	standard	p-value	novelty	standard	p-value
ZDT-1	12	4	35.9	31	0.9681	0.0424	0.0430	1.1747e-04
ZDT-2	12	7	51.75	38.571429	0.5217	0.0011	0.0045	0.1958
ZDT-3	3	1	1	1	1	0.0265	8.0672e-04	3.0199e-11
ZDT-4	5	0	1	0	0.3333	4.4621	10.7861	1.1674e-05

If we look at that table, we will find that all results before we add our algorithm to EvoLib and make new generation we will take a long time to find rare solutions in ZDT-1 and the number of rare solutions will increase with novelty and will be more than standard run so it looks so good to adding novelty search algorithm to our EvoLib to make better.

After all of that we believe that in our future work in OSY and BNH problems we will find a good rare solution "OSY is function used as a multi-objectives test function for the purpose of estimating the performance of optimization algorithms".

In Figures 3 and 4, we use 100 generations and 100 populations and 99 steps to solve this test problem and without adding our constraint and novelty algorithm and after adding our constraint.

In Figures 5 and 6, we use 100 generations and 100 populations and 99 steps to find rare solutions to test the problems.

5. **Conclusion and Future Work.** This paper develops a new algorithm adding to EvoLib (an open-source implementation of NSGA-II, NSGA-III and Unified NSGA-III (U-NSGA-III)) to find a rare solution in continuous problem ZDT-1 ZDT-2, ZDT-3, ZDT-4,

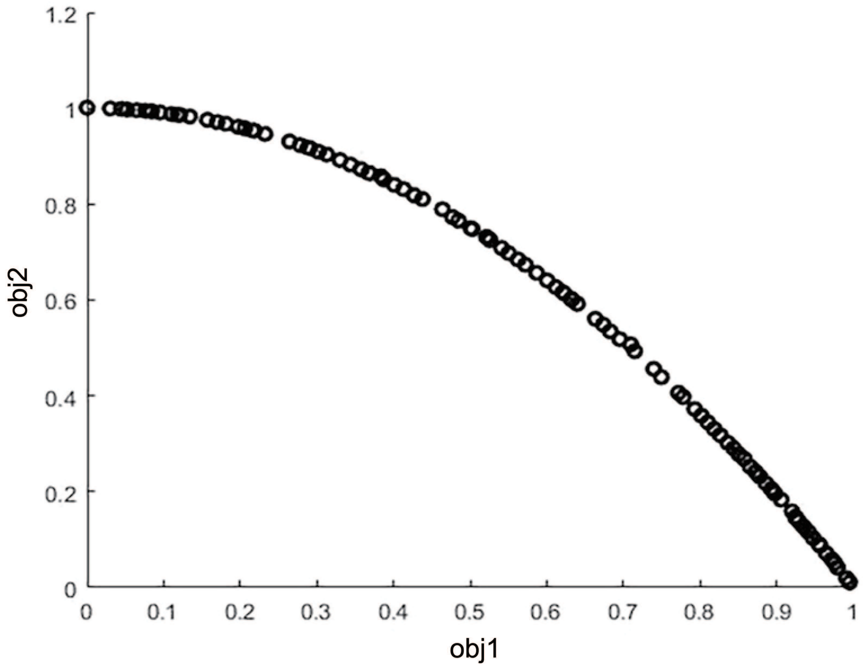

Figure 3. ZDT-2 before adding novelty and constraint

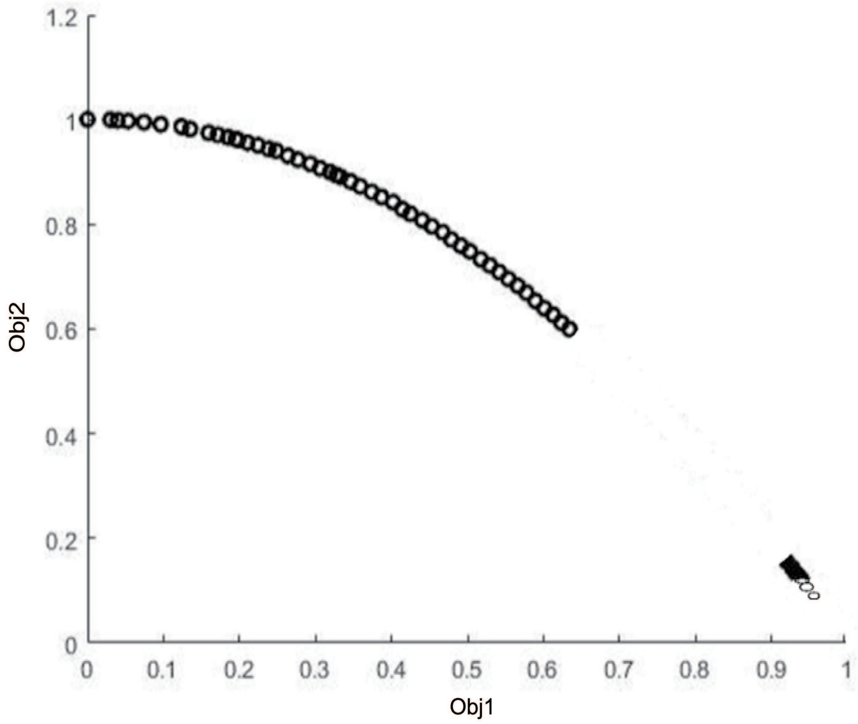

Figure 4. ZDT-2 after adding novelty and constraint

ZDT-6 by adding novelty search in the selection step of NSGA-III and then, we try to find if NS provide a statistically significant improvement on top of the base level performance of already existing EMO techniques but finally we do it and make it better and find many solutions and apply our algorithm. In ZDT-1 problem rare solution before and after adding novelty search that number of rare solutions is different and be better after calculating statistical significance.

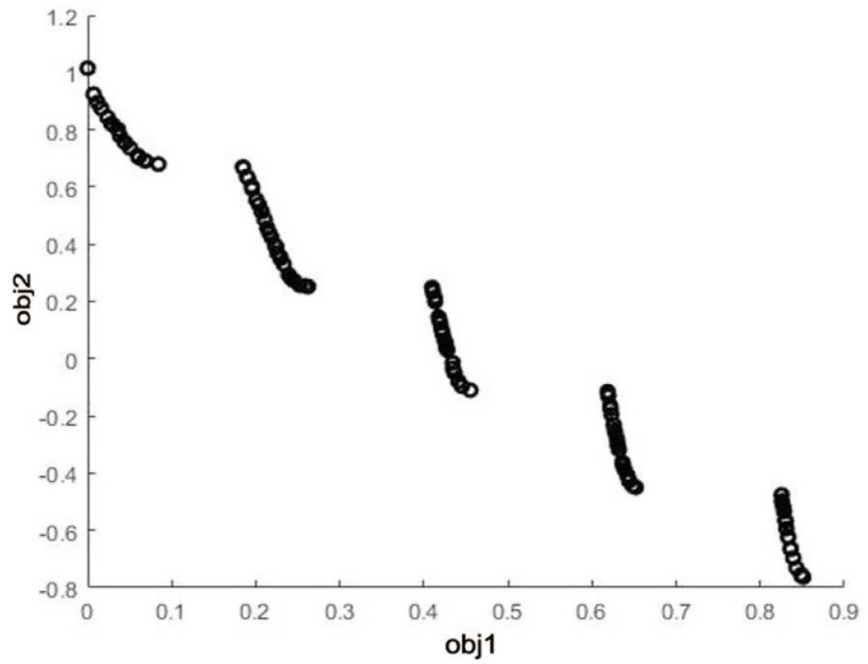

Figure 5. ZDT-3 before adding novelty and constraint

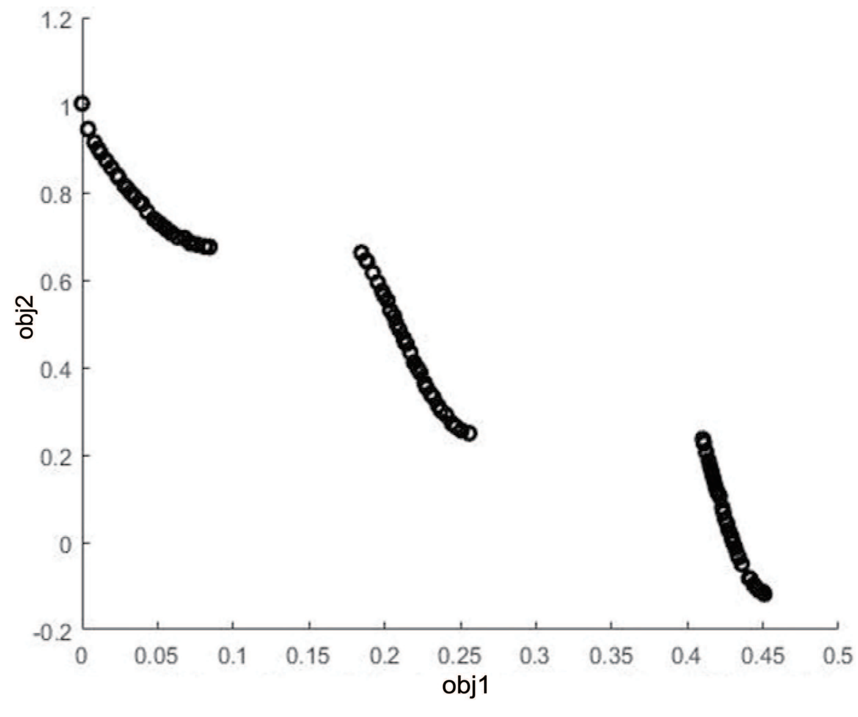

Figure 6. ZDT-3 after adding novelty and constraint

We found that in all problems in our bed test more rare solutions are found. For instance, in ZDT-1 the new mechanism performance is better by 33% and in ZDT2 the performance is better by 58%, and in ZDT-3 performance is better by 33%. We can improve our performance by increasing the generation count by 20% every time making a run to our search. We proposed using U-NSGA-III implementation to degenerate to an equal and work well for each class in a population-based optimization approach, simply relying on the number of problem objectives that have been defined. The algorithm uses the standard evolutionary algorithm settings as well as some new ones. Extensive simulations on both unrestricted and restricted tests issues with single, two, more than three, more than 15 objectives, as well as two other engineering optimization problems, are undertaken to illustrate the functionality of U-NSGA-III.

Our work will be applying to other problems such as OSY, and BNH, and find it work the same as it does in constraints problems and improve more power in NS to work in more small time.

REFERENCES

[1] J. Lehman and K. O. Stanley, Abandoning objectives: Evolution through the search for novelty alone, *Evolutionary Computation*, vol.19, no.2, pp.189-223, 2011.

[2] J.-B. Mouret, Novelty-based multiobjectivization, in *New Horizons in Evolutionary Robotics. Studies in Computational Intelligence*, S. Doncieux, N. Bredèche and J.-B. Mouret (eds.), Berlin, Heidelberg, Springer, 2011.

[3] J. Gomes, P. Urbano and A. L. Christensen, Introducing novelty search in evolutionary swarm robotics, in *Swarm Intelligence. ANTS 2012. Lecture Notes in Computer Science*, M. Dorigo et al. (eds.), Berlin, Heidelberg, Springer, 2012.

[4] D. E. Goldberg and J. Richardson, Genetic algorithms with sharing for multimodal function optimization, *Proc. of the 2nd International Conference on Genetic Algorithms*, Morgan, Kaufmann, pp.148-154, 1987.

[5] K. Deb, S. Agrawal, A. Pratap and T. Meyarivan, A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II, in *Parallel Problem Solving from Nature PPSN VI. PPSN 2000. Lecture Notes in Computer Science*, Berlin, Heidelberg, Springer, 2000.

[6] C. M. Fonseca and P. J. Fleming, Genetic algorithms for multiobjective optimization: Formulation, discussion, and generalization, *Proc. of the 4th International Conference on Evolutionary Programming*, pp.416-423, 1993.

[7] R. C. Purshouse and P. J. Fleming, On the evolutionary optimization of many conflicting objectives, *IEEE Trans. Evolutionary Computation*, vol.11, no.6, pp.770-784, 2007.

[8] S. Risi, S. D. van der Beek, C. E. Hughes and K. O. Stanley, How novelty search escapes the deceptive trap of learning to learn, *Proc. of the Genetic and Evolutionary Computation Conference (GECCO'09)*, New York, NY, 2009.

[9] K. O. Stanley and R. Miikkulainen, Evolving neural networks through augmenting topologies, *Evolutionary Computation*, vol.10, no.2, pp.99-127, 2002.

[10] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, MIT Press, 1998.

[11] A. Toffolo and E. Benini, Genetic diversity as an objective in multi-objective evolutionary algorithms, *Evolutionary Computation*, vol.11, no.2, pp.151-167, 2003.

[12] E. Zitzler, M. Laumanns and L. Thiele, SPEA2: Improving the strength Pareto evolutionary algorithm, *Proc. of EUROGEN2011*, pp.95-100, 2001.

[13] T. Bäck, *Evolutionary Algorithms in Theory and Practice: Evolution Strategies, Evolutionary Programming, Genetic Algorithms*, Oxford University Press, Oxford, 1995.

[14] D. A. Van Veldhuizen and G. B. Lamont, Multiobjective evolutionary algorithms: Analyzing the state-of-the-art, *Evolutionary Computation*, vol.8, no.2, pp.125-147, 2000.

[15] H. Seada and K. Deb, U-NSGA-III: A unified evolutionary algorithm for single, multiple, and many-objective optimization, *Computer Science*, 2014.

[16] N. Srinivas and K. Deb, Multi-objective function optimization using non-dominated sorting genetic algorithms, *Evolutionary Computation*, vol.2, no.3, pp.221-248, 1994.

[17] K. Deb, S. Agrawal, A. Pratap and T. Meyarivan, A fast and elitist multi-objective genetic algorithm: NSGA-II, *IEEE Trans. Evolutionary Computation*, vol.6, no.2, pp.182-197, 2002.

[18] D. W. Corne, J. D. Knowles and M. Oates, The Pareto envelope-based selection algorithm for multiobjective optimization, *Proc. of the 6th International Conference on Parallel Problem Solving from Nature VI (PPSN-VI)*, pp.839-848, 2000.

[19] C. A. C. Coello and G. T. Pulido, A micro-genetic algorithm for multi-objective optimization, *Proc. of the 1st International Conference on Evolutionary Multi-Criterion Optimization (EMO2001)*, Zurich, Switzerland, 2001.

[20] M. L. Ryerkerk, R. C. Averill, K. Deb and E. D. Goodman, Optimization for variable-size problems using genetic algorithms, *Proc. of the 12th AIAA Aviation Technology, Integration, and Operations (ATIO) Conference and the 14th AIAA/ISSMO*, Indianapolis, IN, 2012.
[21] E. F. Khor, K. C. Tan and T. H. Lee, Tabu-based exploratory evolutionary algorithm for effective multi-objective optimization, *Proc. of the 1st Evolutionary Multi-Objective Optimization (EMO-01)*, pp.344-358, 2001.
[22] K. Deb, *Multi-Objective Optimization Using Evolutionary Algorithms*, Wiley, Chichester, UK, 2001.
[23] C. A. C. Coello, D. A. van Veldhuizen and G. Lamont, *Evolutionary Algorithms for Solving Multi-Objective Problems*, Kluwer, Boston, MA, 2002.
[24] C. K. Goh and K. C. Tan, *Evolutionary Multi-Objective Optimization in Uncertain Environments: Issues and Algorithms*, Springer-Verlag, 2009.
[25] K. C. Tan, E. F. Khor and T. H. Lee, *Multiobjective Evolutionary Algorithms and Applications*, Springer-Verlag, London, UK, 2005.
[26] C. Hu, L. Dai, X. Yan, W. Gong, X. Liu and L. Wang, Modified NSGA-III for sensor placement in water distribution system, *Information Sciences*, vol.509, pp.488-500, DOI: 10.1016/j.ins.2018.06.055, 2020.
[27] R. H. Bhesdadiya, I. N. Trivedi, P. Jangir, N. Jangir and A. Kumar, An NSGA-III algorithm for solving multi-objective economic/environmental dispatch problem, *Cogent Engineering*, vol.3, no.1, DOI: 10.1080/23311916.2016.1269383, 2016.
[28] W. Mkaouer, M. Kessentini, S. Bechikh, K. Deb and M. Ó Cinnéide, High dimensional search-based software engineering: Finding tradeoffs among 15 objectives for automating software refactoring using NSGA-III, *Genetic and Evolutionary Computation Conference*, Vancouver, 2014.
[29] K. Lavangnananda and P. Wangsom, Multi-objective shipment allocation using extreme nondominated sorting genetic algorithm-III (E-NSGA-III), *2019 18th IEEE International Conference on Machine Learning and Applications (ICMLA)*, DOI: 10.1109/ICMLA.2019.00247, 2019.
[30] A. Metiaf and Q. Wu, Wireless sensor network deployment optimization using reference-point-based non-dominated sorting approach (NSGA-III), *2019 3rd International Conference on Data Mining, Communications and Information Technology (DMCIT2019)*, Beijing, China, 2019.
[31] K. Deb and M. Goyal, A robust optimization procedure for mechanical component design based on genetic adaptive search, *Transactions of the ASME: Journal of Mechanical Design*, vol.120, no.2, pp.162-164, 1998.
[32] J. J. Liang, T. P. Runarsson, E. Mezura-Montes, M. Clerc, P. N. Suganthan, C. A. C. Coello and K. Deb, *Problem Definitions and Evaluation Criteria*, Special Session on Constrained Real-Parameter Optimization, Technical Report, Nanyang Technological University, Singapore, 2006.
[33] K. Deb and H. Jain, An evolutionary many-objective optimization algorithm using reference-point based non-dominated sorting approach, Part I: Solving problems with box constraints, *IEEE Trans. Evolutionary Computation*, vol.18, no.4, pp.577-601, 2014.
[34] A. Al-Moalmi, J. Luo, A. Salah and K. Li, Optimal virtual machine placement based on grey wolf optimization, *Electronics*, vol.8, no.3, DOI: 10.3390/electronics8030283, 2019.
[35] A. Al-Moalmi, J. Luo, A. Salah, K. Li and L. Yin, A whale optimization system for energy-efficient container placement in data centers, *Expert Systems with Applications*, vol.164, DOI: 10.1016/j.eswa.2020.113719, 2021.
[36] Y. Ngamsanroaj, N. Kumyaito, K. Ngamsanroaj and K. Tamee, Optimal reservoir operation for water supply using genetic algorithm: A case study of Bhumibol Dam, Thailand, *ICIC Express Letters, Part B: Applications*, vol.11, no.1, pp.41-49, 2020.
[37] I. Das and J. E. Dennis, Normal-boundary intersection: A new method for generating the Pareto surface in nonlinear multicriteria optimization problems, *SIAM Journal on Optimization*, vol.8, no.3, pp.631-657, 1998.

Author Biography

Mahmoud Mohamed Nabil is computer science engineer graduated from Zagazig university 2010. Mr. Nabil is working in second paper to obtain a master's degree in computer science "A Novelty-based Approach to Capture Unique Solutions in Continuous Multiobjective Problems".

Haitham Seada is the lead data scientist on Ford Motor Company's vehicle routing solution "Symphony". He is major in computer science with a master degree in bioinformatics and a Ph.D. in evolutionary multiobjective optimization. After his Ph.D., Dr. Seada joined Kim Birkenhead's lab at Brown University as a postdoctoral research associate, where he developed computational pipelines to process and analyze high throughput cell culture imaging data. His current role at Ford involves creating optimization models and algorithms for vehicle routing problems and architecting software wrappers to the developed techniques. His research interests involve vehicle routing, non-linear optimization, computational intelligence, linear modeling and machine learning, as exhibited in his list of publications. For more than 8 years, Seada used to teach a diverse selection of computer science topics to both undergraduate and graduate students at several private and public sector universities and institutions. He is also the author of several software libraries, both closed and open source, spanning a wide range of technologies.

Magdi Zakaria Rashad is a professor of computer science at Mansoura University, Egypt. Prof. Rashad holds a Ph.D. in computer science from Faculty of Engineering, Cairo University in Egypt and is the author of over 160 papers published in refereed international journals. He has served as a head of Computer Science Department and a vice dean of Faculty of Computers and Information Systems, Mansoura University. He has also served as a reviewer for various international journals, such as IEEE Transactions in Internet of Things (IoT), Elsevier and he is interested in the following fields: artificial intelligence, pattern recognition, machine learning, image processing, cloud computing and Internet of Things (IoT).

Amro Abd El Latif received the Ph.D. degree from Faculty of Computer Science, Cairo University in 2016. He is currently a lecturer at Faculty of Computer Science, Zagazig University. His research interests include compiler construction, parsers, automata, parallel processing natural language processing and artificial intelligence.