




# Multi-objective Non-overlapping Front Generation: A Pivot-Based Deterministic Non-dominated Sorting Approach

Sourab Mandal<sup>(✉)</sup>  and Paramartha Dutta 

Department of Computer & System Sciences, Visva-Bharati University, Santiniketan,  
Bolpur, West Bengal, India  
[smbv.rs@gmail.com](mailto:smbv.rs@gmail.com)

**Abstract.** Multiobjective/Many-objective Optimization Genetic Algorithm (MOGA) is a method that utilizes a stochastic search inspired by nature to find solutions in a multiobjective optimization framework. Examples of MOGA include NSGA-II and NSGA-III, both of which rely on fast non-dominated sorting during execution. Non-dominated sorting is a difficult task in this field. This paper presents an alternative deterministic approach to fast non-dominated sorting that is both correct and complete. The approach focuses on pivots, which enables it to handle data with multiple or many objectives. The proposed method has been applied to relevant benchmark data sets and has been proven effective.

**Keywords:** NSGA-II · NSGA-III · Pivot · Non-dominated Sorting · Decision Space · Objective Space · Multiobjective Optimization

## 1 Introduction

MOGA is extensively employed in diverse fields such as engineering, science, medicine, management, finance, etc. A couple of examples [1, 2] showcase its effectiveness in addressing crucial multi-objective optimization problems in recent times. The widespread adoption of MOGA in industrial applications has significantly elevated its significance, making it a highly valued domain.

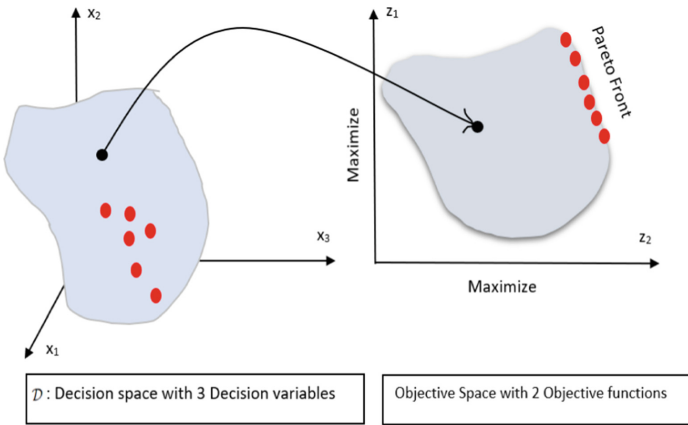
### 1.1 Prerequisites

This portion presents an exploration of the essential terminologies and pertinent concepts related to multi-objective optimization problems.

**Definition 1 (Multi-objective Optimization Problem(MOP)).** *Multiobjective optimization is a method utilized to improve multiple objectives concurrently, considering their mutual influence and any existing constraints. It aims to find optimal solutions that enhance two or more objectives simultaneously. The formal definition of the MOP is as follows:*

$$\begin{aligned}
 & \text{Min/Max } z_m(x), \quad m = 1, 2, \dots, M; \\
 \text{s.t., } & w_j(x) \geq 0, \quad j = 1, 2, \dots, J; \quad y_k(x) = 0, \quad k = 1, 2, \dots, K; \\
 & x_i^{(L)} \leq x_i \leq x_i^{(U)} \quad i = 1, 2, \dots, n;
 \end{aligned}$$

A solution<sup>1</sup>, denoted as  $x \in \mathbb{R}^n$ , is a vector consisting of  $n$  decision variables:  $x = (x_1, x_2, \dots, x_n)^T$ . In this context,  $M$ ,  $J$ , and  $K$  represent the number of objectives that require optimization, denoted by  $z_m$ , the number of inequality constraints denoted by  $w_j$ , and the number of equality constraints denoted by  $y_k$ , respectively. The lower and upper bounds of  $x_i$  are denoted as  $x_i^{(L)}$  and  $x_i^{(U)}$ , respectively, where  $i = 1, 2, \dots, n$  [3]. The mapping of decision space points to objective space points is illustrated in Fig. 1.



**Fig. 1.** Decision Space to Objective Space Mapping. The red points found in the objective space represent the members of the Pareto front. (Color figure online)

**Definition 2 (Domination Relation).** A solution  $\alpha \in \mathcal{D}$  (Decision Space) is said to dominate another solution  $\beta \in \mathcal{D}$  if both the following conditions hold together:

1. In terms of all the objective functions, the solution  $\alpha$  can not be worse than the solution  $\beta$ .
2. There exists at least one objective function for which  $\alpha$  is strictly better than  $\beta$  [3].

**Definition 3 (Pareto Optimal Set).** A solution is deemed Pareto optimal when there is no other solution that can improve one objective without worsening another. The set of all Pareto optimal solutions is the “Pareto optimal set”, and their corresponding objective points form the “Pareto front” [3].

<sup>1</sup> Here “solution” refers to the elements of the decision space and “point” refers to the elements of the objective space respectively.

For example, in Fig. 1, the red solutions in the decision space form the Pareto optimal set, while the red points in the objective space depict the Pareto front.

## 1.2 Literature Survey

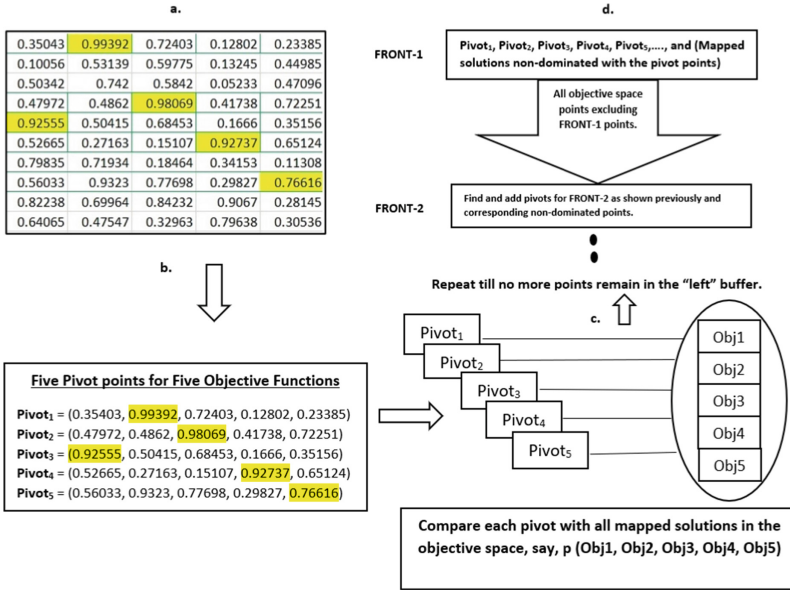
There have been several mechanisms to solve the MOPs, including SPEA2 [4], MOGA [5], NSGA-II [6], IBEA [7], NSGA-III [8], and more. Among these, NSGA-II stands out as one of the most popular approaches. NSGA-II comprises two main constituents: Fast Non-dominated Sorting (FNDS) and Crowding Distance. FNDS having the computational complexity of  $O(MN^2)$ , generates non-dominated fronts, with  $M$  and  $N$  being the number of objective functions and population size respectively. Although other non-dominated sorting approaches exist, for example, a recursive mechanism having a computational complexity of  $O(N \log^{M-1} N)$  by Jensen et al. [9], they become ineffective when two different solutions are with the same objective function value. Tang et al. introduced a fast approach for generating the non-dominated set using the arena's principle [10]. Zhang et al. introduced an important method called Efficient Non-dominated Sorting (ENS), which encompasses two variations: one employing sequential search (ENS-SS) and the other utilizing binary search (ENS-BS) [11]. However, no existing algorithm in the literature achieves better time complexity in the worst-case scenario than  $O(MN^2)$  for non-dominated sorting while remaining complete.

## 1.3 Gap Identification and Motivation

To address the issues discussed in Sect. 1.2, we have offered a pivot-based non-dominated sorting that intends to provide a complete, correct, and time-efficient algorithm that generates non-dominated fronts.

## 1.4 Structure of the Paper

This paper is structured into the following four sections. First, “Introduction” in Sect. 1, covers the background information, literature review, and gap analysis. Followed by “Methodology” in Sect. 2, which explains the proposed technique and provides information about the relevant dataset. Section 3 is dedicated to “Result and Analysis”, where the completeness and accuracy of the proposed method are demonstrated and discussed. Finally, the “Conclusion” is presented in Sect. 4.



**Fig. 2.** Flow Diagram of the Pivot-based Deterministic Non-dominated Sorting Approach. **a.** Snippet of a demo data set in an objective space with each column representing five different objective functions. **b.** Five Pivot points for five objective functions. **c.** Comparing all points in the objective space with the pivot point(s) to check whether they are worse or in a non-dominated relation with the pivot point(s). **d.** The recursive process of generating fronts until the “left” buffer, used to hold all other elements excluding the current front’s pivot points and its non-dominated points, is empty.

## 2 Methodology

For a clear understanding of the methodology, we introduce the concept of *Pivot*.

**Definition 4 (Pivot).** *Pivot points refer to the mapped solutions from a decision space to an objective space where there is at least one objective function that achieves the optimal value. This optimal value can be either the maximum value if the objective function aims to maximize or the minimum value if the objective function aims to minimize. In a typical multi-objective optimization, there can be a combination of possibilities for both maximization and minimization. Figure 2 depicts the flow diagram of the mechanism, with an explanation of pivot points.*

### 2.1 Benchmark Data-Set Specification

**Zitzler-Deb-Thiele-1 (ZDT-1) Test Suite.** The construction of the ZDT problem suite is rooted in a 30-variable problem ( $n = 30$ ) that exhibits a convex Pareto-optimal set [14].

$$\min z_1(x); \min z_2(x) = g(x)h(z_1(x), g(x));$$

where two objectives need minimization, the function  $g(x)$  is typically used to represent convergence, and it is common for  $g(x) = 1$  for Pareto-optimal solutions. The definition of the ZDT-1 test suite is as follows:

$$z_1(x) = x_1; \quad g(x) = 1 + \frac{9}{n-1} \sum_{i=2}^n x_i; \quad h(z_1, g) = 1 - \sqrt{\frac{z_1}{g}}; \quad 0 \leq x_i \leq 1$$

where,  $i = 1, \dots, n$ .

**The Vehicle Crashworthiness Problem.** This real-life application-based benchmark involves optimizing the level of safety in the event of a crash and is characterized as a problem with three objectives and no constraints [13]. This problem involves five decision variables,  $x_1, x_2, x_3, x_4$ , and  $x_5$ . Three objective functions:  $z_1, z_2$ , and  $z_3$  are depicted as follows:

$$z_1(x) = 1640.2823 + 2.3573285x_1 + 2.3220035x_2 + 4.5688768x_3 + 7.7213633x_4 + 4.4559504x_5$$

$$z_2(x) = 6.5856 + 1.15x_1 - 1.0427x_2 + 0.9738x_3 + 0.8364x_4 - 0.3695x_1x_4 + 0.0861x_1x_5 + 0.3628x_2x_4 - 0.1106x_1x_1 - 0.3437x_3x_3 + 0.1764x_4x_4$$

$$z_3(x) = -0.0551 + 0.0181x_1 + 0.1024x_2 + 0.0421x_3 - 0.0073x_1x_2 + 0.024x_2x_3 - 0.0118x_2x_4 - 0.0204x_3x_4 - 0.008x_3x_5 - 0.0241x_2x_2 + 0.0109x_4x_4$$

**Deb-Thiele-Laumanns-Zitzler-1 (DTLZ-1).** This involves  $n$  decision variables and is defined as follows [15]:

Minimize:

$$f_1(\vec{x}) = \frac{1}{2}x_1x_2 \cdots x_{M-1}(1 + g(\vec{x}))$$

$$f_i(\vec{x}) = \frac{1}{2}x_1x_2 \cdots (1 - x_{i-1}) \cdots (1 - x_{M-1})(1 + g(\vec{x})), \quad i = 2, \dots, M - 1$$

$$f_M(\vec{x}) = \frac{1}{2}(1 - x_1)(1 + g(\vec{x}))$$

where  $\vec{x} = (x_1, x_2, \dots, x_K)$  depicts the decision variables, and  $g(\vec{x})$  is a non-linear function given by:

$$g(\vec{x}) = \sum_{i=K-M+1}^K (x_i - 0.5)^2 - \cos(20\pi(x_i - 0.5))$$

The decision variables are subject to the following constraints:

$$0 \leq x_i \leq 1, \quad i = 1, \dots, K$$

**OneMinMax and Leading Ones and Trailing Zeros (LOTZ).** The OneMinMax benchmark was originally suggested by Giel and Lehre (2010) as a two-objective version of the well-known ONEMAX benchmark. Laumanns, Thiele, and Zitzler (2004) introduced a benchmark called LOTZ. The benchmarks mentioned here contain objective functions that are of maximization type [12].  
 $z : \{0, 1\}^n \rightarrow \mathbb{N} \times \mathbb{N}$

OneMinMax Function is defined by:

$$z(x) = (z_1(x), z_2(x)) = (n - \sum_{i=1}^n x_i, \sum_{i=1}^n x_i); \forall x = (x_1, x_2, \dots, x_n) \in \{0, 1\}^n$$

The LOTZ function is defined as follows:

$$z(x) = (z_1(x), z_2(x)) = (\sum_{i=1}^n \prod_{j=1}^i x_j, \sum_{i=1}^n \prod_{j=1}^n (1 - x_j)); \forall x \in \{0, 1\}^n$$

## 2.2 Pivot-Based Deterministic Non-dominated Sorting

The algorithm takes input as the mapped solutions from the decision space and outputs non-overlapping decomposed Front(s), where the first Front represents the Pareto Front. Algorithm 1 shows the Pivot-based Deterministic Non-dominated Sorting.

## 3 Result and Analysis

The result yielded for different benchmark functions are shown in Fig. 3. The result of the suggested Pivot-based algorithm is evaluated using the ZDT-1 bi-objective benchmark function to compare the fast non-dominated sorting and the proposed approach. In the sample of 500 elements in the objective space, both methods successfully position all elements on the Pareto Front. This indicates the presence of a single front, where all 500 elements in the sample are mutually non-dominated. This information is visualized as a diamond shape in Fig. 3. The analysis extends to the ‘‘Vehicle Crashworthiness’’ benchmark, which involves three objective functions and is examined using both fast non-dominated sorting and the proposed approach. In this case, the entire objective space is divided into eight fronts, with each front containing an equal number of elements for both methods. These fronts are represented by a brown-coloured asterisk sign. The consistency of this outcome is observed across all other benchmark functions showcased in Fig. 3 where yellow square, green triangle, and brown cross are used respectively for DTLZ - 1, LOTZ, and OneMinMax front representation, thereby affirming the validity of the proposed algorithm.

### 3.1 Completeness and Correctness of the Algorithm

Formation of pivots followed by placement of an objective space element in a suitable front in respect of the pivot undergo exhaustive exploration in a deterministic sense. Hence, assigning a newly explored element in an appropriate Front can not but be procedurally complete.

**Algorithm 1** Pivot-based Deterministic Non-dominated Sorting Algorithm

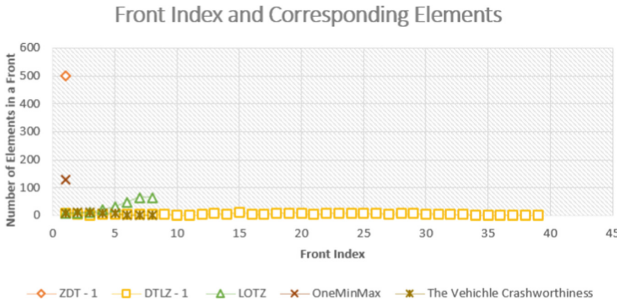
---

```

1: Find the pivot points (min/max depending on the objective function(s)) and assign
   them to the head of the current level
2: Save the remaining tuples (or points) in a buffer named left
3: for each  $Tmp_{tpl}$  from left fetch one by one and do
4:    $swap_{f_{lg}} \leftarrow 0$ ;  $move_{f_{lg}} \leftarrow 0$ 
5:   for each  $i^{th}$  element of both  $Tmp_{tpl}$  and  $Head_{tpl}$  do
6:      $lower_{f_{lg}} \leftarrow 0$ ;  $greater_{f_{lg}} \leftarrow 0$ ;  $equi_{f_{lg}} \leftarrow 0$ 
7:     if  $Tmp_{tpl} == Head_{tpl}$  then
8:       Insert the  $Tmp_{tpl}$  into current Head and go to step-3.
9:     end if
10:    for  $i \leftarrow 0$  to  $M$  do ▷  $M$ : Number of objective functions
11:      if  $(Tmp_{tpl})_i > Head_{tpl}$  then
12:         $greater_{f_{lg}} + = 1$ 
13:      end if
14:      if  $(Tmp_{tpl})_i < Head_{tpl}$  then
15:         $lower_{f_{lg}} + = 1$ 
16:      end if
17:      if  $(Tmp_{tpl})_i == Head_{tpl}$  then
18:         $equi_{f_{lg}} + = 1$ 
19:      end if
20:    end for
21:    if  $(greater_{f_{lg}} + equi_{f_{lg}}) == M$  then
22:      move  $Head_{tpl}$  from head list to left
23:       $swap_{f_{lg}} + = 1$ 
24:    end if
25:    if  $(greater_{f_{lg}} == 0$  and  $lower_{f_{lg}} \neq 0)$  then
26:      move  $Temp_{tpl}$  to left
27:       $swap_{f_{lg}} \leftarrow 0$ ,  $move_{f_{lg}} \leftarrow 0$ 
28:      goto step-3
29:    end if
30:    if  $(greater_{f_{lg}} \neq 0$  and  $lower_{f_{lg}} == 0)$  then
31:       $move_{f_{lg}} + = 1$ 
32:    end if
33:    if  $(swap_{f_{lg}} > 0$  or  $move_{f_{lg}} > 0)$  then
34:      move/add  $Tmp_{tpl}$  in the  $Head$ (current Head)
35:      goto step-3
36:    end if
37:  end for
38: end for
39: if left is not empty then
40:   Create next level and goto step-1
41: end if

```

---



**Fig. 3.** Number of Fronts and corresponding elements generated using the proposed method and the Fast Non-dominated Sorting.

### 3.2 Complexity of the Proposed Algorithm

The Algorithm proposed in Algorithm 1 consists of two nested *for* loops in steps 3 and 5, which iterate  $N$  times, where  $N$  represents the number of mapped solutions in the objective space. In the worst-case scenario, this algorithm requires  $O(N^2)$  time to finish. Additionally, step 10 includes another loop that iterates  $M$  times, where  $M$  is the number of objective functions. Step 1, on the other hand, takes  $O(MN)$  time to complete. Consequently, the algorithm’s best-case, average-case, and worst-case time complexity is respectively,  $\Omega(MN^2)$ ,  $\Theta(MN^2)$ ,  $O(MN^2)$ , which is similar to respective cases of the Fast Non-dominated Sorting phase in NSGA-II and NSGA-III. Furthermore, no worse than any existing non-dominated sorting algorithm, which is complete, in the worst-case scenario.

## 4 Conclusion

The pivot-based framework has a number of benefits. First off, the idea is simple and straightforward to put into practice. Second, it covers all the solutions in the decision space by effectively dividing them into non-overlapping fronts using the mapped objective space values, needless to mention which ensures the categorization of the quality of the solutions. Last but not least, it accomplishes these advantages in a time-efficient way that isn’t worse than current state-of-the-art methods. The effectiveness of the suggested approach is validated by comparisons with multiple benchmark multi-objective optimization functions that exist in the literature. Despite being a novel, complete, and correct approach, the algorithm cannot handle streamed or online data. Our efforts in the future will be directed towards addressing this particular issue in the algorithm.

## References

1. Mei, Y., Wu, K.: Application of multi-objective optimization in the study of anti-breast cancer candidate drugs. *Sci. Rep.* **12**, 19347 (2022). <https://doi.org/10.1038/s41598-022-23851-0>



2. Winiarczyk, R., Kaleta, A., Górnicki, K.: Application of a MOGA algorithm and ANN in the optimization of apple drying and rehydration processes. *Processes* **9**(8), 1415 (2021). <https://doi.org/10.3390/pr9081415>
3. Deb, K.: *Multiobjective Optimization Using Evolutionary Algorithms*. Wiley, New York (2001)
4. Zitzler, E., Laumanns, M., Thiele, L.: SPEA2: improving the strength pareto evolutionary algorithm. In: *Evolutionary Methods for Design, Optimization and Control With Applications to Industrial Problems. Proceedings of the EUROGEN 2001, Athens, Greece, 19–21 September 2001* (2001)
5. Fonseca, C.M., Fleming, P.J.: Genetic algorithms for multiobjective optimization: Formulation, discussion and generalization. In: Forrest, S. (Ed.), *Proceedings of the Fifth International Conference on Genetic Algorithms*, pp. 416–423. Morgan Kaufman Publishers (1993)
6. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans. Evol. Comput.* **6**(2), 182–197 (2002). <https://doi.org/10.1109/4235.996017>
7. Zitzler, E., Künzli, S.: Indicator-based selection in multiobjective search. In: Yao, X., et al. (eds.) *PPSN 2004. LNCS*, vol. 3242, pp. 832–842. Springer, Heidelberg (2004). [https://doi.org/10.1007/978-3-540-30217-9\\_84](https://doi.org/10.1007/978-3-540-30217-9_84)
8. Deb, K., Jain, H.: An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, Part I: solving problems with box constraints. *IEEE Trans. Evol. Comput.* **18**(4), 577–601 (2014)
9. Jensen, M.T.: Reducing the run-time complexity of multiobjective EAs: the NSGA-II and other algorithms. *IEEE Trans. Evol. Comput.* **7**(5), 503–515 (2003)
10. Tang, S., Cai, Z., Zheng, J.: A fast method of constructing the non-dominated set: Arena’s principle. In: *2008 Fourth International Conference on Natural Computation*, pp. 391–395. IEEE Computer Society Press (2008)
11. Zhang, X., Tian, Y., Cheng, R., Jin, Y.: An efficient approach to nondominated sorting for evolutionary multiobjective optimization. *IEEE Trans. Evol. Comput.* **19**(2), 201–213 (2015)
12. Zheng, W., Liu, Y., Doerr, B.: A first mathematical runtime analysis of the non-dominated sorting genetic algorithm II (NSGA-II). In: *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, no. 9, pp. 10408–10416 (2022). <https://doi.org/10.1609/aaai.v36i9.21283>
13. Liao, X., Li, Q., Yang, X., Zhang, W., Li, W.: Multiobjective optimization for crash safety design of vehicles using stepwise regression model. *Struct. Multidiscip. Optim.* **35**(6), 561–569 (2008)
14. Zitzler, E., Deb, K., Thiele, L.: Comparison of multiobjective evolutionary algorithms: empirical results. *Evol. Comput.* **8**(2), 173–195 (2000). <https://doi.org/10.1162/106365600568202>
15. Deb, K., Thiele, L., Laumanns, M., Zitzler, E.: Scalable multi-objective optimization test problems. In: *Proceedings of the 2002 Congress on Evolutionary Computation. CEC 2002 (Cat. No.02TH8600)*, vol. 1, pp. 825–830. IEEE, Honolulu, HI, USA (2002). <https://doi.org/10.1109/CEC.2002.1007032>