Adaptive Use of Innovization Principles for a Faster Convergence of Evolutionary Multi-Objective Optimization Algorithms*

Abhinav Gaur and Kalyanmoy Deb
Computational Optimization and Innovation (COIN) Laboratory
Department of Electrical and Computer Engineering
Michigan State University
East Lansing, MI 48824, USA
{gauabhi,kdeb}@msu.edu, http://www.egr.msu.edu/~kdeb/

COIN Report Number 2016003

March 27, 2016

Abstract

"Innovization" is a task of learning common principles that exist among some or all of the Pareto-optimal solutions of a multi-objective optimization problem. Except a few earlier studies, most innovization related studies were performed on the final non-dominated solutions found by an EMO algorithm. Since the innovization principles are properties of good and near-optimal solutions, an early identification of them can help improve the evolving population to converge quicker to the Pareto-optimal set. This paper advocates the discovery of innovization principles through machine learning methods during an evolutionary multi-objective optimization run and then using these principles to repair the population adaptively to achieve a faster convergence. Using the proposed idea, the paper presents encouraging results on four modified ZDT-series problems and an engineering design problem. The results show not only an improvement in convergence rate but also in the diversity of non-dominated solutions. The results are encouraging and spurs a number of interesting immediate future studies.

1 Introduction

All multi-objective optimization (MOO) problems possess a unique property of having not one, but a set of "equally good" or Pareto-optimal (PO) solutions. The idea of learning from the PO solutions and deciphering design principles that are unique to the optimization problem was first presented in [7]. This concept is called *innovization* and it is pictorially shown in Figure 1. Since an innovization task involves learning from a set of equivalent solutions, it is appealing to couple this idea to population based optimization methods such as *evolutionary multi-objective optimization* or EMO algorithms. There are examples in literature where the innovization task has been conducted on the results of an EMO algorithm to decipher interesting knowledge about the problem. For example, [3] uses innovization to discover new design principles in three engineering case studies. Another example[8] proposes a novel recommendation tool for software refactoring based on innovization. [1] suggests a way to automate the finding of salient problem design principles for engineering problems. [9] uses the innovization task in a simulation based MOO problem and suggest different ways to couple the innovization with the EMO that can throw light on important principles of the problem.

*Accepted in GECCO-2016 Conference (20-24 July 2016), Denver, Colorado, USA.

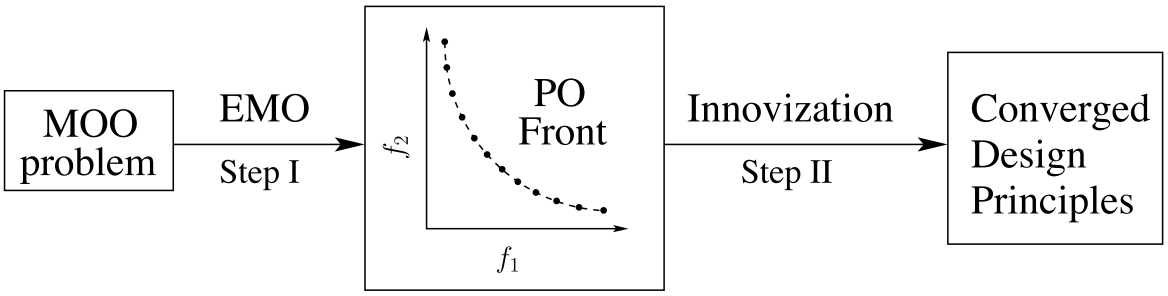

Figure 1: The concept of innovization.

However, some recent publications point to a growing trend of coupling the ideas of EMO with that of innovization in a way, that makes innovization a an integral part of the optimization process itself and not a post optimality analysis tool . This situation is similar to what is shown in Figure 2. We will refer to this new method as '*evolutionary multi-objective optimization with innovization*' or EMO/I method in this paper. The idea was first suggested in [5] where the authors initially discover the salient principles by innovization task and then use those principles as heuristic for local search and obtain a faster convergence than the EMO applied alone. [10] suggests interleaving the optimization and innovization tasks to expedite convergence to a preferred region of the PO front.

This work is similar in concept to [10] in terms of the end goal, i.e. achieving faster convergence by coupling EMO with innovization, but it has a different way of achieving the same goal. In Figure 2, the part named 'Active Intervention' differs greatly for our method, EMO/I, and the 'interleaving' method proposed in [10]. Authors of [10] advocate learning the innovized rules using decision trees and then adding those learned rules as if-else statement type of constraints during the EMO run. Whereas, this work advocates directly repairing the solutions, once a rule is learned.

In the remainder of the paper, we describe the proposed EMO/I method in Section 2. Thereafter, we present the different MOO problems that were used to test the EMO/I method in Section 3. The results are discussed for all the test problems in Section 4. Finally, our conclusions are given in Section 5. We have used NSGA-II [6] algorithm as the EMO for this work.

2 Methodology

Figure 3 gives an overview of the proposed EMO/I method, i.e. an EMO algorithm in conjunction with innovization. The six component blocks that differentiate it from a regular EMO algorithm are labeled 1 to 6 in gray circles. They are, (a) one input/output block labeled 1, (b) two decision blocks labeled 2 and 5, and (c) three process blocks labeled 3,4 and 6. The following sections explain the innovization part of proposed EMO/I method shown in Figure 3.

2.1 Block-1 : Rule Info

This block represents the information on the rules or principles that the user supplies in the beginning. This section first explains what is meant by a rule in this work and then explains the information that the user supplies as part of block-1 of Figure 3.

A rule, Λ, can be any relation that comprises of some or all of design variables, objectives, constraints or any function of the design variables of the MOO problem. A set of M such rules

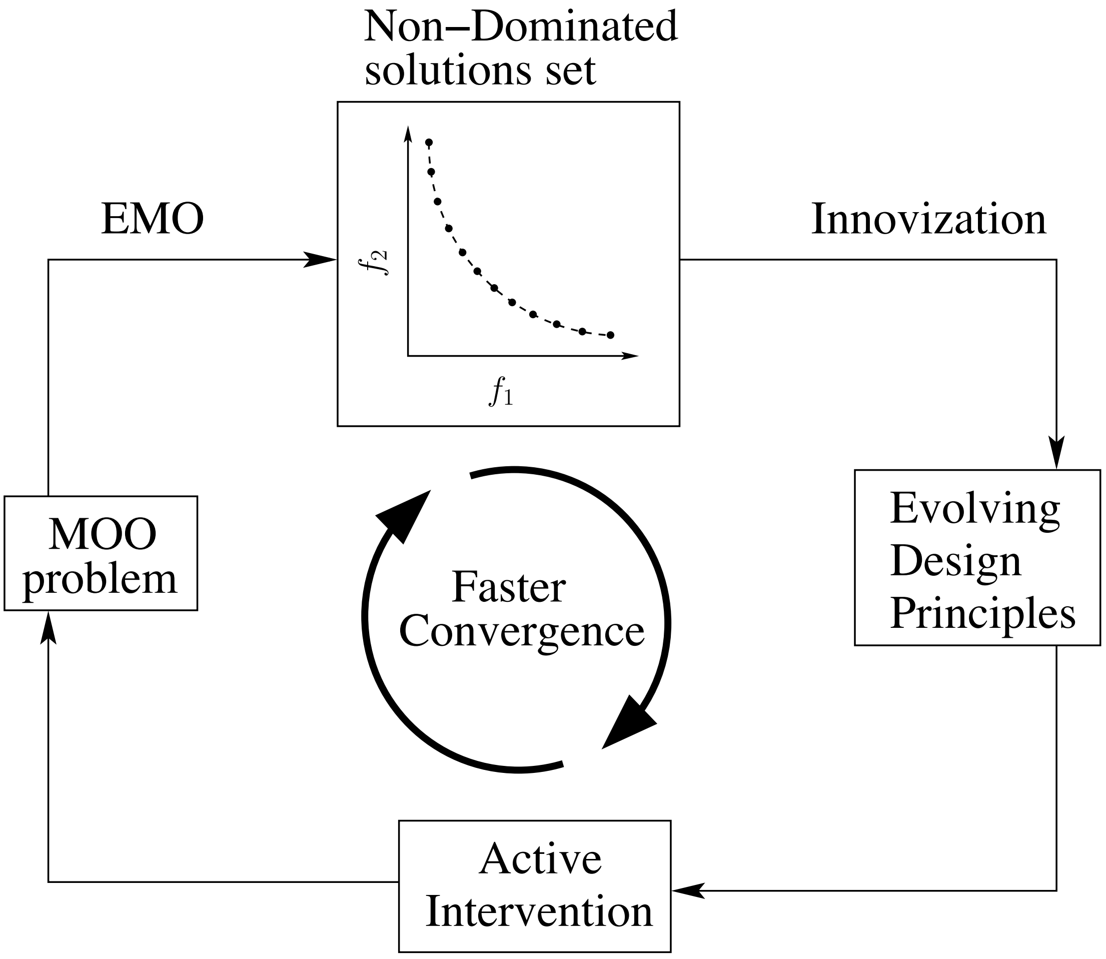

Figure 2: The concept of EMO/I.

are shown in (1).

$$\begin{aligned} &\Lambda_i \equiv \prod_j \lambda_j(\mathbf{x})^{a_{ij}b_{ij}} = c_i \text{ where,} \\ &a_{ij} \in \{0,1\}\ ,\ b_{ij} \in \mathbb{R}, \\ &i \in \{1,2,\cdots,M\} \text{ and } j \in \{1,2,\cdots,N\}. \end{aligned} \tag{1}$$

As per (1), the left hand side of i^{th} rule Λ_i is composed of a product of N '$\lambda_j(\mathbf{x})$' terms which we choose to call *basis functions.* A basis function can be any of the design variables, objectives, constraints or any scalar function of the design variables of the problem. The right hand side of i^{th} rule can be some real constant c_i. With Boolean variables, a_{ij}'s, and real exponents, b_{ij}'s, any rule of the power-law form involving the pre-decided basis functions can be generated. The b_{ij}'s can be transformed as per (2) to obtain an equivalent form and avoid different forms of the same rule structure. The reason for assuming such a power-law form is that earlier studies [7, 3] on innovization have found such design rules existing in the PO set in many engineering design problems.

$$b_{ij} = \frac{b_{ij}}{\{b_{ip} \mid p = (\text{argmax}_p \mid a_{ip}b_{ip})\}}\ \forall\ i. \tag{2}$$

The information supplied by the user to the learning algorithm is listed in Table 1. Note that the the parameters b_{ij} and c_i are learned by the EMO/I during an optimization run. As an

Table 1: Information provided by user related to the rules

Parameter	Description
M	Number of rules
N	Number of basis functions
$\lambda_j(\mathbf{x})$	basis functions (total N)
a_{ij}	$M \times N$ boolean variables
ρ_{max}	Max allowed Error Ratio

example of a rule, lets say that for some problem, the user provides the information given in (3);

$$\begin{aligned} &M = 1, N = 3, \lambda_1 = x_1, \lambda_2 = x_2, \lambda_3 = f_1, \\ &a_{11} = 1, a_{12} = 1, a_{13} = 0, \rho_{max} = 0.1, \end{aligned} \tag{3}$$

where x_1, x_2 are design variables and f_1 is first objective for the MOO problem. This will make the EMO/I algorithm to learn rules of the form $x_1^{b_{11}} \cdot x_2^{b_{12}} = c_1$ during the algorithm run. The parameters, b_{ij}, c_i can be learned by a some suitable machine learning algorithm. The parameter ρ_{max} is explained in coming sections.

2.2 Block-2 : Decision C2

This decision block checks the condition on whether the entire population is part of the non-dominated solutions set or not. If not, then the algorithm passes the control to the 'selection' block of the EMO/I, else the control passes to block-3.

2.3 Block-3 : Machine Learning

Once the entire population of EMO/I algorithm is in the non-dominated solutions set, block-3 receives these solutions from block-2 (refer Figure 3). Block-3 can be any suitable machine learning algorithm to learn the rules and it shall return two aspects for all the learned rules;

- quantitative aspects of the learned rule (e.g. b_{ij} and $c_i \forall i$ in this case) and,
- qualitative aspects of goodness for the learned rules (e.g. Sum of squares of Errors or SSE for linear regression).

For simplicity, all test problems chosen in this paper involve rules with only two variables, i.e. (4) is followed in all the test problems in this paper. Combining (1) and (4) results in rules of form given in (5).

$$\sum_{j=1}^{N} a_{ij} = 2, \ \forall \ i \in \{1, 2, \cdots, M\} \tag{4}$$

$$\begin{aligned} &\Lambda_i \equiv \lambda_{j_1}(\mathbf{x})^{b_{ij_1}} \cdot \lambda_{j_2}(\mathbf{x})^{b_{ij_2}} = c_i \ \forall \ i \text{ where,} \\ &i \in \{1, 2, \cdots, M\} \ , \ j_1 \in \{1, 2, \cdots, N \mid a_{ij_1} = 1\}, \\ &j_2 \in \{1, 2, \cdots, N \mid a_{ij_2} = 1\}. \end{aligned} \tag{5}$$

Using normalization of exponents as per (2), rules of form (5) are transformed to (6).

It is important to point it out here that in this paper, the authors have devised a repair strategy based on machine learning only for cases where the basis functions (λ_j) are all decision variables of the problems. If the basis functions are something else, e.g. objective values or constraint values, that would require a different strategy and is not addressed in this work.

$$\begin{aligned} &\Lambda_i \equiv \lambda_{j_1}(\mathbf{x}) \cdot \lambda_{j_2}(\mathbf{x})^{b_i} = c_i \ \forall \ i, \text{ where,} \\ &i \in \{1, 2, \cdots, M\} \ , \ j_1 \in \{1, 2, \cdots, N \mid a_{ij_1} = 1\}, \\ &j_2 \in \{1, 2, \cdots, N \mid a_{ij_2} = 1\}. \end{aligned} \tag{6}$$

Note that each of the M rules of (6) has two unknowns namely, b_i and c_i. To learn rules of the form (6), *linear-regression* is used as a machine learning tool of choice in this work. The log-linear model is used to compute estimates of $\hat{b}_i$ and $\hat{c}_i$, as well as the sum of squares due to errors in parameter estimates or sse, for all M rules.

Figure (4) takes a closer look at block-3 of Figure 3 using an example. Lets say that a user attempts to solve an unconstrained bi-objective optimization problem with objectives f_1 and f_2. Let the problem have three design variables, namely x_1, x_2 and x_3. Suppose that the user provides the information given in (3) to block-1 of EMO/I method. Block-3 of Figure 3 then passes on the parameter estimates ($\hat{b}_i$ and $\hat{c}_i \ \forall\ i$) and sum of squared errors or ($sse_i \ \forall\ i$) to block-4 of Figure 3.

2.4 Block-4 : Evaluate $\rho_i \ \forall\ i$

As shown in Figure 4, this block receives the learned parameter estimates ($\hat{b}_i$ and $\hat{c}_i \ \forall\ i$) and sum of squared errors or ($sse_i \ \forall\ i$) for all M rules from block-3 of Figure 3. It then estimates for all M rules, a metric we refer to as the *error ratio* or ρ and it is defined in (7). The term $sse_{i,max}$ in (7) is the maximum sse found for i^{th} rule till current generation of EMO/I. The value of $sse_{i,max}$ is not set initially. When block-3 (Figure 3) is invoked for the first time in EMO/I, $sse_{i,max}$ is set to their respective sse_i for all rules, and ρ_i is set to the value of one in that generation.

$$\rho_i = sse_i/sse_{i,max} \ \forall\ i \in\ \{1, 2, \cdots, M\}. \tag{7}$$

This block then passes all the estimated rule parameters ($\hat{b}, \hat{c}$) and error ratio (ρ) for all M rules under consideration to block-5.

2.5 Block-5 : Decision C3

After receiving the rule parameter estimates and error ratio for all rules, this block checks the condition given by (8).

$$\rho_i > \rho_{max} \ \forall\ i \in\ \{1, 2, \cdots, M\}. \tag{8}$$

The parameter ρ_{max} is provided by the user initially. If the condition (8) evaluates to true, EMO/I passes control to regular EMO method. If the condition (8) evaluates to false, that means that there is at least one rule which has converged fairly well. In that case, the control passes to block-6 of Figure 3 for innovization based repairs.

The advantages of using max error ratio, ρ_{max}, as a parameter instead of $sse_{i,max}$ are;

- user has to set one parameter value, ρ_{max}, and not M $sse_{i,max}$ values and;
- sse of each rule is compared to its own worst value at each generation. This way rules which converge slower compared to others can be filtered prior to the repair stage, where this information is vital.

2.6 Block-6 : Innovized Repair

This block performs the function of repairing the EMO/I solutions as per the rules learned in block-3. If condition (8) in block-5 of Figure 3 evaluates to false, that means that there is at least one rule which has converged fairly well and solution repairs can be made based on the same.

As explained in section 2.3, we only deal with rules of form (6) in this paper and the basis functions, λ_{j_1} and λ_{j_2} of (6) are decision variables of the problem.

For each rule, a repair is randomly chosen from (9a) or (9b). If (9a) is chosen, then the value of basis function λ_{j_1} is repaired using $\lambda_{j_2}, \hat{b}_i$ and $\hat{c}_i$. Whereas, if (9b) is chosen, then the value of basis function λ_{j_2} is repaired using $\lambda_{j_1}, \hat{b}_i$ and $\hat{c}_i$.

$$\breve{\lambda}_{j_1} = \left(\hat{c}_i/\lambda_{j_2}^{\hat{b}_i}\right) \ \forall\ i \text{ , or apply} \tag{9a}$$

$$\breve{\lambda}_{j_2} = (\hat{c}_i/\lambda_{j_1})^{1/\hat{b}_i} \ \forall\, i \text{ , where} \tag{9b}$$
$$i \in \{1, 2, \cdots, M \mid \rho_i < \rho_{max}\},\ j_1 \in \{1, 2, \cdots, N \mid a_{ij_1} = 1\},$$
$$j_2 \in \{1, 2, \cdots, N \mid a_{ij_2} = 1\}.$$

Once all legitimate repairs ($\{i \mid \rho_i < \rho_{max}\}$) are made, algorithm passes the control to mutation operator and flow control is again taken over by the EMO algorithm. Let us now look at the test problems considered in this work.

Note that in all the test examples in this paper and in most engineering design problems, the decision variables are always non-negative. Hence, this avoids the possibilities of getting undefined results from (9a) or (9b). However, future development of EMO/I needs to take this point into account too, i.e. how to respect the domain and range of entities being repaired or changed.

3 Problems Considered

This section describes five bi-objective optimization problems that were used to test the proof of concept for the proposed EMO/I method to expedite convergence of EMO algorithms. All problems considered in this work are three variable problems and have a design principle of the form (1) instilled in the problem definition in such a way that all PO solutions of the bi-objective optimization problem must adhere to this rule.

3.1 Modified ZDT Problems (ZDTm)

Of the five problems considered in this work, four are derived from the ZDT [11] problem test suite. These are described in following sections.

3.1.1 ZDT1m

The mathematical description of ZDT1m problem is given by (10).

$$\begin{aligned}
\text{Minimize} \quad & f_1(\mathbf{x}) = x_1, \\
& f_2(\mathbf{x}) = g(\mathbf{x})\ h(f_1(\mathbf{x}), g(\mathbf{x})), \\
\text{Where} \quad & g(\mathbf{x}) = 1 + 9\left(x_2\sqrt{x_3} - 0.5\right)^2 \ , \\
& h(f_1, g) = 1 - \sqrt{f_1/g}, \\
& x_i \in [0, 1],\ \forall\, i \in \{1, 2, 3\}.
\end{aligned} \tag{10}$$

The Pareto-optimal region for this problem is given by (11a) and (11b).

$$0.0 \le x_1^* \le 1.0, \tag{11a}$$

$$x_2^*\sqrt{x_3^*} - 0.5 = 0 \ . \tag{11b}$$

The Pareto front of ZDT1m is continuous, convex and same as that of ZDT1 problem. The rule information input (Table 1) for this problem is shown in (12).

$$\begin{aligned}
& M = 1, N = 2, \phi_1 = x_2, \phi_2 = x_3, \\
& a_{11} = 1, a_{12} = 1, \rho_{max} = 0.1.
\end{aligned} \tag{12}$$

3.1.2 ZDT2m

The mathematical description of ZDT2m problem is given by (13).

$$\begin{aligned}
\text{Minimize}\quad & f_1(\mathbf{x}) = x_1, \\
& f_2(\mathbf{x}) = g(\mathbf{x})\; h(f_1(\mathbf{x}), g(\mathbf{x})), \\
\text{Where}\quad & g(\mathbf{x}) = 1 + 9\left(x_2\sqrt{x_3} - 0.5\right)^2, \\
& h(f_1, g) = 1 - (f_1/g)^2, \\
& x_i \in [0,1],\ \forall\, i \in \{1,2,3\}.
\end{aligned} \tag{13}$$

The Pareto-optimal region for this problem is given by (11a) and (11b). The Pareto front of ZDT2m is continuous, non-convex and same as that of ZDT2 problem. The rule information input (Table 1) for this problem is shown in (12).

3.1.3 ZDT3m

The mathematical description of ZDT3m problem is given by (14).

$$\begin{aligned}
\text{Minimize}\quad & f_1(\mathbf{x}) = x_1, \\
& f_2(\mathbf{x}) = g(\mathbf{x})\; h(f_1(\mathbf{x}), g(\mathbf{x})), \\
\text{Where}\quad & g(\mathbf{x}) = 1 + 9\left(x_2\sqrt{x_3} - 0.5\right)^2, \\
& h(f_1, g) = 1 - \sqrt{f_1/g} - (f_1/g)\sin(10\pi f_1), \\
& x_i \in [0,1],\ \forall\, i \in \{1,2,3\}.
\end{aligned} \tag{14}$$

This problem has a number of disconnected Pareto-optimal fronts, same as that of the ZDT3 problem. The Pareto-optimal region for this problem is given by (11b) and (15).

$$\begin{aligned}
& x_1^* \in \mathbb{F} \quad \text{where}, \\
& \mathbb{F} = [0, 0.0830] \cup [0.1822, 0.2577] \cup [0.4093, 0.4538] \\
& \quad \cup [0.6183, 0.6525] \cup [0.8233, 0.8518].
\end{aligned} \tag{15}$$

The rule information input (Table 1) for this problem is shown in (12).

3.1.4 ZDT4m

The mathematical description of ZDT4m problem is given by (16).

$$\begin{aligned}
\text{Minimize}\quad & f_1(\mathbf{x}) = x_1, \\
& f_2(\mathbf{x}) = g(\mathbf{x})\; h(f_1(\mathbf{x}), g(\mathbf{x})), \\
\text{Where}\quad & g(\mathbf{x}) = 1 + 10 + (x_2 \cdot x_3 - 15)^2 \\
& \qquad\quad - 10\cos(4\pi(x_2 \cdot x_3 - 15)^2), \\
& h(f_1, g) = 1 - \sqrt{f_1/g}, \\
& x_1 \in [0,1], \quad x_2, x_3 \in [\sqrt{10}, \sqrt{20}].
\end{aligned} \tag{16}$$

This problem has a convex Pareto-optimal set. The problem has a total of 11 distinct Pareto-optimal fronts in the objective space, of which only one is global. The global PO region corresponds to (17a,17b) whereas, the local PO region is given by (17a,17c).

$$0.0 \le x_1^* \le 1.0, \tag{17a}$$

$$x_2^* \, x_3^* - 15 = 0, \tag{17b}$$

$$\begin{aligned} &x_2 \, x_3 - 15 = 0.5m \text{ where,} \\ &m \text{ is any integer} \in [-10, 10]. \end{aligned} \tag{17c}$$

The rule information input (Table 1) for this problem is shown in (12).

3.2 Two Bar Truss Design (TBT)

In this problem, a two bar truss structure is designed to carry a certain load without elastic failure. Its mathematical formulation is given by (18). The reader is referred to [7] for complete description of the problem with figure.

$$\begin{aligned} \text{Minimize} \quad & V = x_1\sqrt{16+y^2} + x_2\sqrt{1+y^2}, \\ & S = \max(\sigma_{AC}, \sigma_{BC}), \\ \text{Subject to} \quad & g_1 \equiv \max(\sigma_{AC}, \sigma_{BC}) \leq 10^5, \\ & g_2 \equiv V \leq 0.044, \\ \text{Where} \quad & \sigma_{AC} = \frac{20\sqrt{16+y^2}}{yx_1}, \sigma_{BC} = \frac{80\sqrt{1+y^2}}{yx_2}, \\ & x_1, x_2 \in [0, 0.01], \quad y \in [1.0, 3.0]. \end{aligned} \tag{18}$$

The design variables x_1 and x_2 are the cross sectional areas (measured in m^2) of the two truss members and, design variable y is the vertical distance (measured in m) between the point of loading to the base where the two trusses are mounted. The Pareto-optimal region for this problem is given by (19a,19b). The Pareto-optimal front of this problem is continuous and convex.

$$x_1^*/x_2^* - 0.5 = 0, \tag{19a}$$

$$y^* - 2 = 0 \; . \tag{19b}$$

The rule information input (Table 1) for this problem is shown in (20).

$$\begin{aligned} &M = 3, N = 2, \phi_1 = x_1, \phi_2 = x_2, \phi_3 = x_3, \\ &a_{11} = 1, a_{12} = 1, a_{13} = 0, \\ &a_{21} = 1, a_{22} = 0, a_{23} = 1, \\ &a_{31} = 0, a_{32} = 1, a_{33} = 1 \text{ and } \rho_{max} = 0.25. \end{aligned} \tag{20}$$

4 Results

This section presents the results obtained for problems discussed in section 3 with the proposed EMO/I methodology given in section 2. The two algorithms under study are : (a) NSGA-II , a popular EMO algorithm and (b) NSGA-II with innovization, which is an example of the proposed EMO/I algorithm. For all the problems, performance on convergence rate is measured using *generational distance* (GD) [2] and performance on diversity is measured using *spread* [2]. We use one more metric to to compare the algorithms. We call it the *'median absolute distance from Pareto-solutions in decision space'* or $e_{\mathbf{x}^*}$ and it quantifies the convergence level of an algorithm in the design space. It is possible to calculate in all test problems under consideration because the equation(s) of Pareto-optimal solutions in design space are known a priori for all of them. If a function $\mathbb{O}(\mathbf{x})$ defines the Pareto-optimal solutions of a problem, such that $\mathbb{O}(\mathbf{x}^*) = 0$, then for any population based MOO method with population set $\mathbf{P}$, we define $e_{\mathbf{x}^*}$ by (21).

$$e_{\mathbf{x}^*} = \underset{\mathbf{x} \in \mathbf{P}}{\text{median}} \{|\mathbb{O}(\mathbf{x})|\}. \tag{21}$$

A *lower value* of $e_{\mathbf{x}^*}$ signifies that the solutions of an EMO are closer to the PO solutions. Also, a *lower variability* in $e_{\mathbf{x}^*}$ values signifies more solutions of an EMO being closer to the PO solutions. If the Pareto-optimal solutions are defined by more than one equation, e.g. (19a and 19b) for TBT, then we can calculate $e_{\mathbf{x}^*}$ for each equation separately.

Table 2 shows the NSGA-II parameters used for the different problems studied in this work. Other parameters, common to all cases are (a) # of generations = 100, (b)# of runs = 100, (c) probability of crossover (η_c) =0.9 and (d) probability of mutation (η_m) =0.1.

Table 2: List of NSGA-II parameters used. η_c, η_m are Distribution indices for SBX and Polynomial Mutation respectively [2].

Problem	Pop	η_c	η_m
ZDT1m	8	15	20
ZDT2m	8	15	20
ZDT3m	8	15	20
ZDT4m	16,40	15	20
Truss	8	10	50

Solving ZDT1m problem with EMO (NSGA-II here) and EMO/I (NSGA-II with innovization here) yields Figures 5 and 6 as results in GD and Spread respectively. Clearly, the proposed method gives better results on both metrics. The good results in spread are unexpected and a bonus, as the idea of EMO/I was originally conceived for achieving faster convergence by possibly making some compromise in diversity. But these results show that by combining the right innovization strategy with an EMO, good results in both convergence and diversity are possible.

Furthermore, using (11b), we can calculate $e_{\mathbf{x}^*}$ for both algorithms in ZDT1m case. Table 3 gives a comparison of the two algorithms' performance on ZDT1m in terms of the median and inter quartile range (IQR) of the $e_{\mathbf{x}^*}$ metric for four generation cases and over all 100 runs. A lower value of median and IQR (in all generation cases) of $e_{\mathbf{x}^*}$ for EMO/I than that of EMO shows that, not only do the solutions of EMO/I reached closer to the PO solutions but also, more solutions of EMO/I are closer to PO solutions than that of EMO.

Similar results are seen for ZDT2m (see Figures 7, 8 and Table 4) and, ZDT3m (see Figures 9, 10 and Table 5). It should be noted that these advantages were recorded even with a low population size of 8 individuals.

With ZDT4m, at low populations, the EMO/I method shows poor convergence. This can be seen in Figure 11 which shows the median GD plot for ZDT4m problem with population of 16 individuals. This is because ZDT4m has multiple local fronts and all the local fronts of ZDT4m adhere to the rule (11b). Hence in low population, the EMO/I method diminishes diversity of its solutions very quickly and converges to a local PO front. Upon increasing the population size to 40, EMO/I yields marginally better results over EMO (NSGA-II) (see Figures 12, 13 and Table 6).

In TBT problem, it is clear from Figures 14 and 15 that even with a low population of just 8 individuals, the EMO/I method performs better than EMO in both convergence and spread. Since this problem has two equations defining the PO solutions (see 19a,19b), thus $e_{\mathbf{x}^*}$ is calculated for each of them separately. Tables 7 and 8 show the $e_{\mathbf{x}^*}$ metric for rules (19a) and (19b) respectively, and they show again that EMO/I performs better than EMO in not only reaching closer to the PO solutions (low median of $e_{\mathbf{x}^*}$ over all runs) but also, in having more number of solutions in that close range as compared to the EMO solutions (low IQR of $e_{\mathbf{x}^*}$ over all runs).

Table 3: Median and IQR of $e_{\mathbf{x}^*}$ for ZDT1m using (11b) (for 100 runs).

	NSGA-II		NSGA-II + INNOV.	
Gen	median $e_{\mathbf{x}^*}$	IQR $e_{\mathbf{x}^*}$	median $e_{\mathbf{x}^*}$	IQR $e_{\mathbf{x}^*}$
025	0.0104	0.0121	0.0032	0.0072
050	0.0044	0.0076	0.0005	0.0011
075	0.0024	0.0042	0.0001	0.0005
100	0.0012	0.0028	0.0001	0.0002

Table 4: Median and IQR of $e_{\mathbf{x}^*}$ for ZDT2m using (11b) (for 100 runs).

	NSGA-II		NSGA-II + INNOV.	
Gen	median $e_{\mathbf{x}^*}$	IQR $e_{\mathbf{x}^*}$	median $e_{\mathbf{x}^*}$	IQR $e_{\mathbf{x}^*}$
025	0.0077	0.0126	0.0041	0.0068
050	0.0043	0.0058	0.0004	0.0013
075	0.0020	0.0043	0.0002	0.0005
100	0.0015	0.0029	0.0001	0.0002

5 Conclusions

In this paper, we worked on the problem of coupling EMO algorithms with the innovization task to expedite the convergence rate of a standalone EMO algorithm. This problem has only recently caught attention of the EMO community and appears to be a very promising direction of research. We argued that EMO algorithms, by virtue of being population based, are brilliantly positioned for development of optimization-cum-innovization tools such as, EMO/I suggested in this paper and SBI tool suggested in [10]. These methods exploit both, the bottoms-up approach of EMO algorithms as well as the top-down approach of statistical machine learning algorithms.

Using an EMO/I method (NSGA-II with innovization) on four modified ZDT series problems and an engineering design problem, we showed that EMO/I performed better than a standalone EMO (NSGA-II) algorithm in both convergence and diversity in four of the five problems. EMO/I did not perform well in one case, where we used very low population size to solve ZDT4m problem. This is because ZDT4m has multiple local fronts adhering to the same innovized principles. Hence for such problems, the population size should be set slightly higher than recommended, so that its unlikely for all the points to be near a local front at the same instant. When ZDT4m was solved with higher population size in EMO and EMO/I, it yielded results similar in quality to the ones obtained using the EMO method. It should be noted that the EMO/I method outperformed standalone EMO method on both convergence and diversity under the case of using very low population , i.e. just 8 population members.

Now, the EMO/I method suggested here needs to be made more adaptive in generating and selecting appropriate rules for evaluation. As the number of variables, objectives and constraints increase, so does the number of possible rules between them. Hence, for future we have to come

Table 5: Median and IQR of $e_{\mathbf{x}^*}$ for ZDT3m using (11b) (for 100 runs).

	NSGA-II		NSGA-II + INNOV.	
Gen	median $e_{\mathbf{x}^*}$	IQR $e_{\mathbf{x}^*}$	median $e_{\mathbf{x}^*}$	IQR $e_{\mathbf{x}^*}$
025	0.0114	0.0177	0.0043	0.0112
050	0.0037	0.0062	0.0005	0.0014
075	0.0016	0.0034	0.0002	0.0005
100	0.0009	0.0020	0.0001	0.0002

Table 6: Median and IQR of $e_{\mathbf{x}^*}$ for ZDT4m using (17b) (for 100 runs).

	NSGA-II		NSGA-II + INNOV.	
Gen	median $e_{\mathbf{x}^*}$	IQR $e_{\mathbf{x}^*}$	median $e_{\mathbf{x}^*}$	IQR $e_{\mathbf{x}^*}$
025	0.0021	0.0035	0.0016	0.0087
050	0.0007	0.0011	0.0006	0.0013
075	0.0004	0.0007	0.0002	0.0009
100	0.0002	0.0003	0.0002	0.0006

Table 7: Median and IQR of $e_{\mathbf{x}^*}$ for TBT problem using (19a) (for 100 runs).

	NSGA-II		NSGA-II + INNOV.	
Gen	median $e_{\mathbf{x}^*}$	IQR $e_{\mathbf{x}^*}$	median $e_{\mathbf{x}^*}$	IQR $e_{\mathbf{x}^*}$
025	0.1304	0.1150	0.1097	0.1519
050	0.1050	0.0858	0.0808	0.0823
075	0.1125	0.0807	0.0765	0.0689
100	0.1126	0.0923	0.0678	0.0621

up with a strategy that generate large number of candidate design rules automatically and filters them to keep only those rules which hold promise at that instant or are expected to become important in subsequent generations. Here, the concept of temporal innovization [4] may be helpful in identifying such rules. Also, we need to devise a more generic EMO/I method that can handle discrete variables, as well as variables that can take negative values.

Finally, for future EMO/I methods, we may also have to develop a strategy about using relations involving objective values and constraints. We can use them as additional constraints as was first done in [5] and [10]. Another possible area of work is using innovization to direct the EMO/I method for something other than convergence or diversity, but for other metrics such as robustness. One more direction for developing EMO/I methods would be to learn from the best as well as worst solutions and avoid learning rules which have a foothold in both ends of the population. The fast growing body of efficient machine learning algorithms, along with EMO methods being population based MOO methods makes a good case for expanding the body of EMO/I methods.

References

[1] S. Bandaru and K. Deb. Automated discovery of vital knowledge from pareto-optimal solutions: First results from engineering design. In *Evolutionary Computation (CEC), 2010 IEEE Congress on*, pages 1–8. IEEE, 2010.

[2] K. Deb. *Multi-objective optimization using evolutionary algorithms*, volume 16. John Wiley & Sons, 2001.

Table 8: Median and IQR of $e_{\mathbf{x}^*}$ for TBT problem using (19b) (for 100 runs).

	NSGA-II		NSGA-II + INNOV.	
Gen	median $e_{\mathbf{x}^*}$	IQR $e_{\mathbf{x}^*}$	median $e_{\mathbf{x}^*}$	IQR $e_{\mathbf{x}^*}$
025	0.4838	0.3731	0.4894	0.3262
050	0.5131	0.3630	0.4838	0.3363
075	0.4798	0.3683	0.4739	0.3254
100	0.5130	0.3916	0.4908	0.2957

[3] K. Deb, S. Bandaru, D. Greiner, A. Gaspar-Cunha, and C. C. Tutum. An integrated approach to automated innovization for discovering useful design principles: Case studies from engineering. *Applied Soft Computing*, 15:42–56, 2014.

[4] K. Deb, S. Bandaru, and C. C. Tutum. Temporal evolution of design principles in engineering systems: Analogies with human evolution. In *Parallel Problem Solving from Nature-PPSN XII*, pages 1–10. Springer, 2012.

[5] K. Deb and R. Datta. Hybrid evolutionary multi-objective optimization and analysis of machining operations. *Engineering Optimization*, 44(6):685–706, 2012.

[6] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: Nsga-ii. *Evolutionary Computation, IEEE Transactions on*, 6(2):182–197, 2002.

[7] K. Deb and A. Srinivasan. Innovization: Innovating design principles through optimization. In *Proceedings of the 8th annual conference on Genetic and evolutionary computation*, pages 1629–1636. ACM, 2006.

[8] M. W. Mkaouer, M. Kessentini, S. Bechikh, K. Deb, and M. Ó Cinnéide. Recommendation system for software refactoring using innovization and interactive dynamic optimization. In *Proceedings of the 29th ACM/IEEE international conference on Automated software engineering*, pages 331–336. ACM, 2014.

[9] A. Ng, K. Deb, and C. Dudas. Simulation-based innovization for production systems improvement: An industrial case study. In *Proceedings of The International 3rd Swedish Production Symposium, SPS'09, Goteborg, Sweden, 2-3 December 2009*, pages 278–286. The Swedish Production Academy, 2009.

[10] A. H. Ng, C. Dudas, H. Boström, and K. Deb. Interleaving innovization with evolutionary multi-objective optimization in production system simulation for faster convergence. In *Learning and Intelligent Optimization*, pages 1–18. Springer, 2013.

[11] E. Zitzler, K. Deb, and L. Thiele. Comparison of multiobjective evolutionary algorithms: Empirical results. *Evolutionary computation*, 8(2):173–195, 2000.

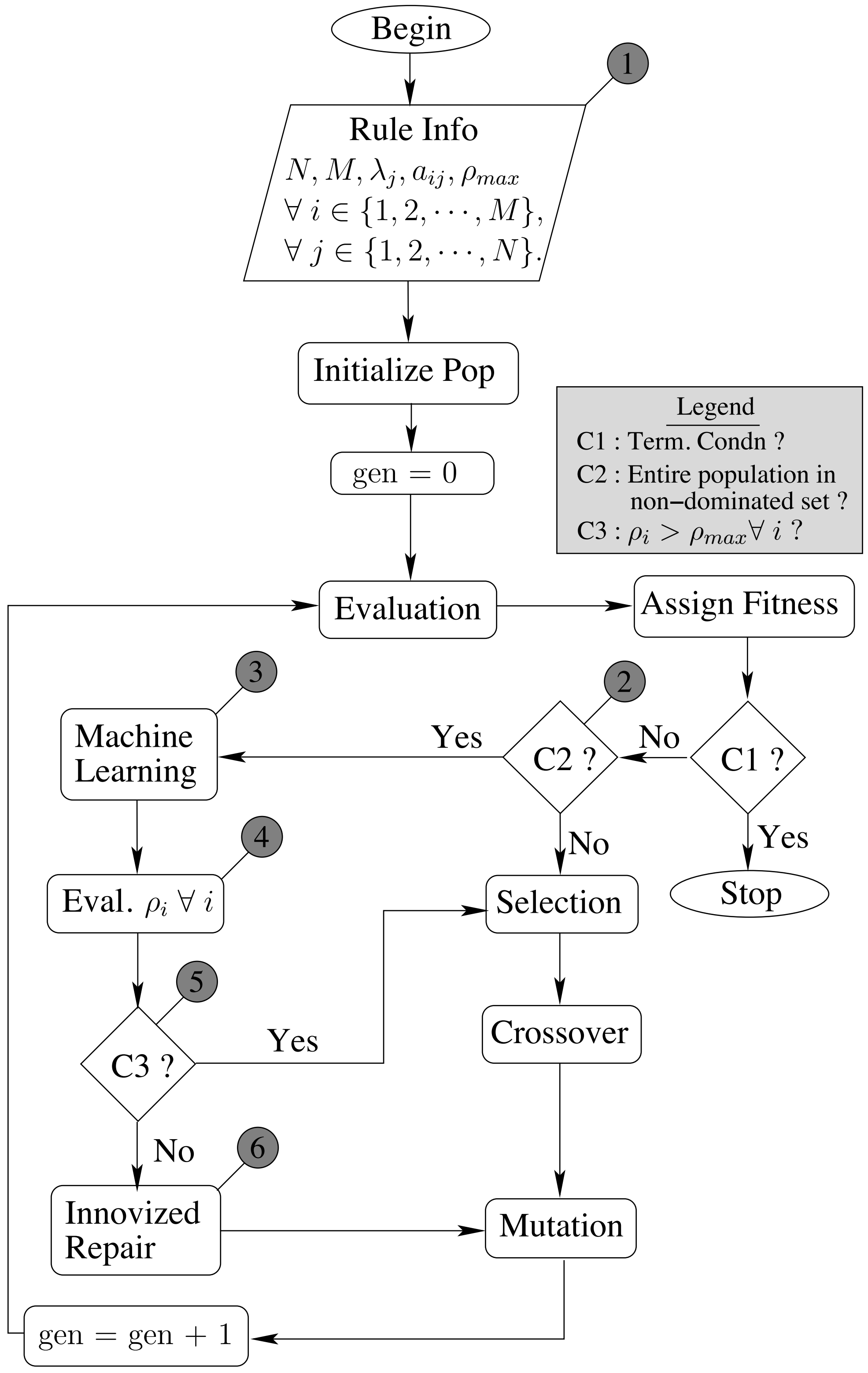
Begin
1
Rule Info
$N, M, \lambda_j, a_{ij}, \rho_{max}$
$\forall\, i \in \{1, 2, \cdots, M\}$,
$\forall\, j \in \{1, 2, \cdots, N\}$.
Initialize Pop
Legend
C1 : Term. Condn ?
C2 : Entire population in non–dominated set ?
C3 : $\rho_i > \rho_{max} \forall\, i$?
gen = 0
Evaluation
Assign Fitness
3
2
Machine Learning
Yes
C2 ?
No
C1 ?
No
Yes
4
Stop
Eval. $\rho_i\ \forall\, i$
Selection
5
C3 ?
Yes
Crossover
No
6
Innovized Repair
Mutation
gen = gen + 1

Figure 3: The proposed EMO/I algorithm, which is a combination of a regular EMO algorithm with innovization. The algorithm blocks related to innovization are labeled with numbers in Grey circles.

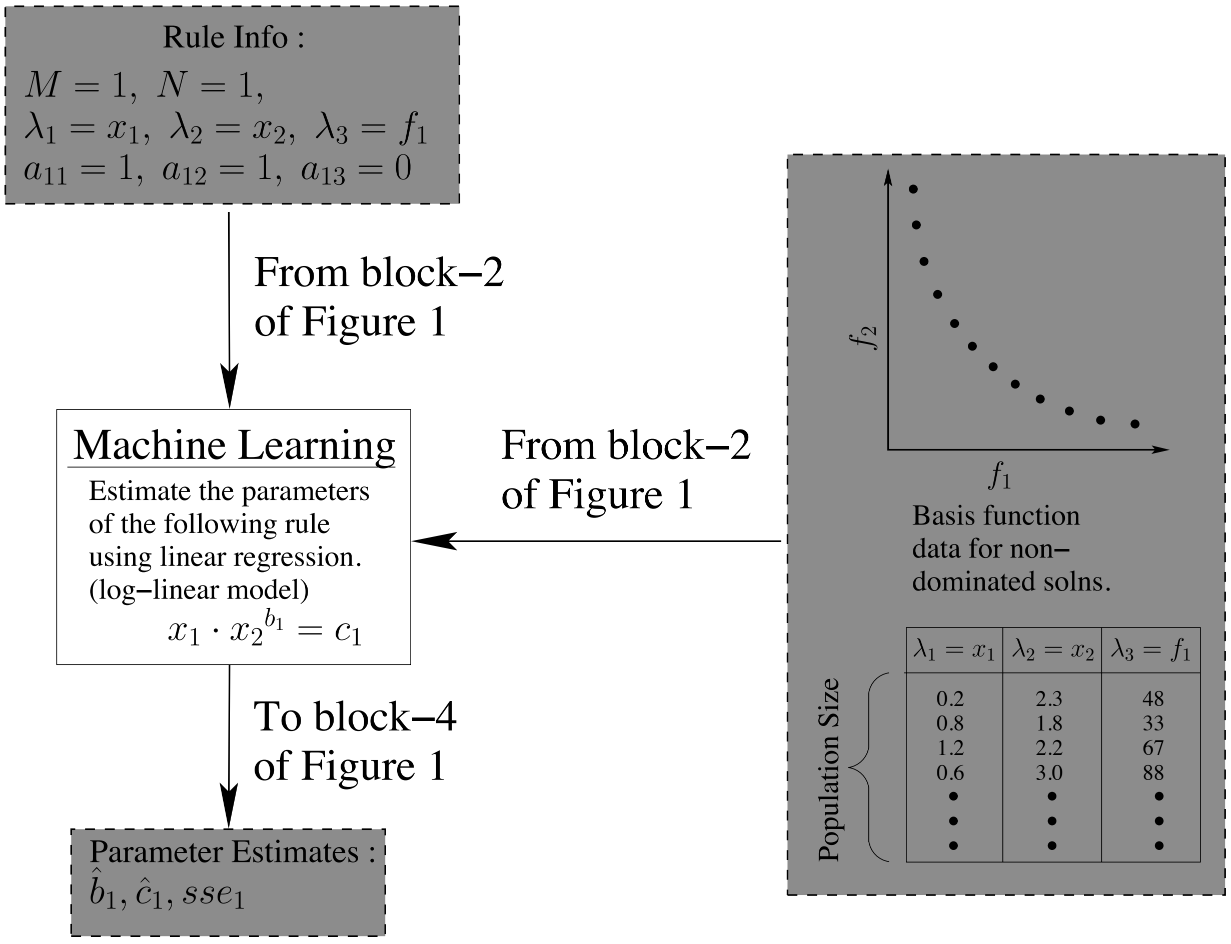
Rule Info :
$M = 1,\ N = 1,$
$\lambda_1 = x_1,\ \lambda_2 = x_2,\ \lambda_3 = f_1$
$a_{11} = 1,\ a_{12} = 1,\ a_{13} = 0$
From block−2 of Figure 1
Machine Learning
Estimate the parameters of the following rule using linear regression. (log−linear model)
$x_1 \cdot {x_2}^{b_1} = c_1$
To block−4 of Figure 1
Parameter Estimates :
$\hat{b}_1, \hat{c}_1, sse_1$
From block−2 of Figure 1
f_2
f_1
Basis function data for non−dominated solns.
Population Size
$\lambda_1 = x_1$ | $\lambda_2 = x_2$ | $\lambda_3 = f_1$
0.2 2.3 48
0.8 1.8 33
1.2 2.2 67
0.6 3.0 88

Figure 4: Details of block-3 of Figure 3.

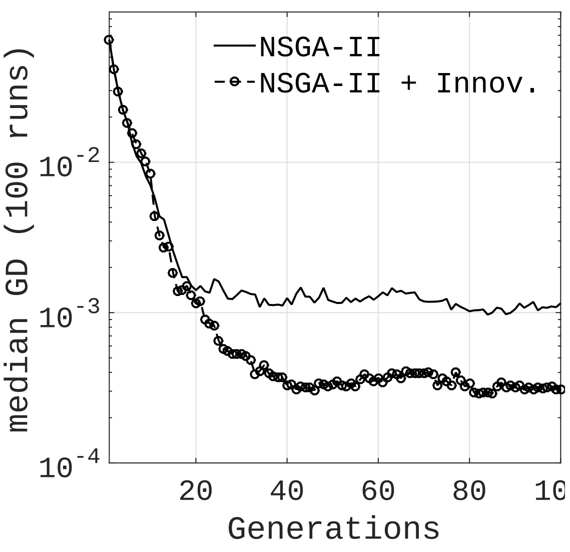

Figure 5: GD versus generations for ZDT1m

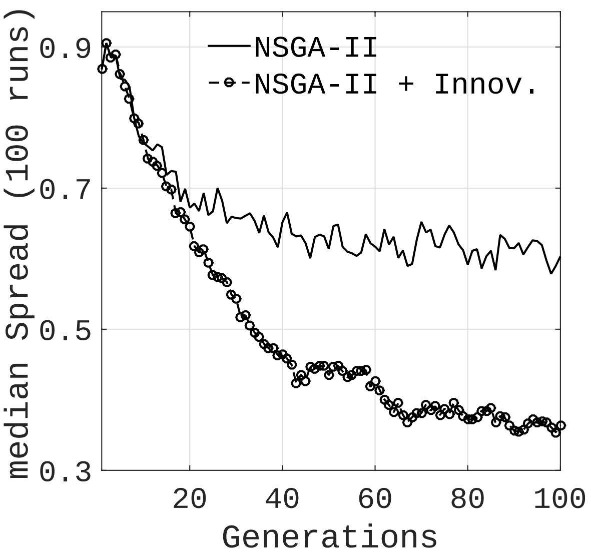

Figure 6: Spread plot for ZDT1m

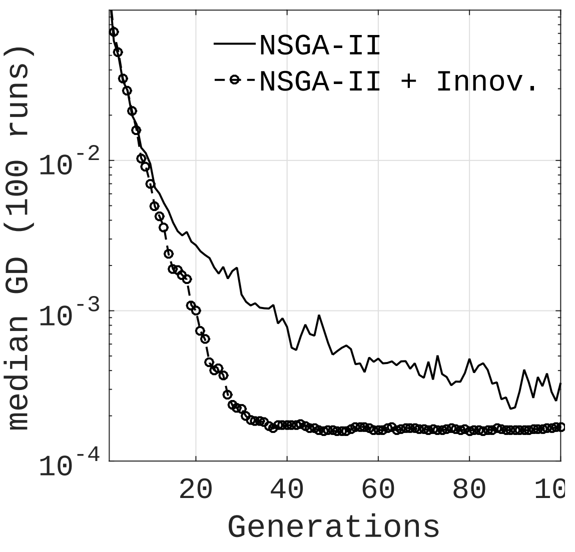

Figure 7: GD versus generations for ZDT2m

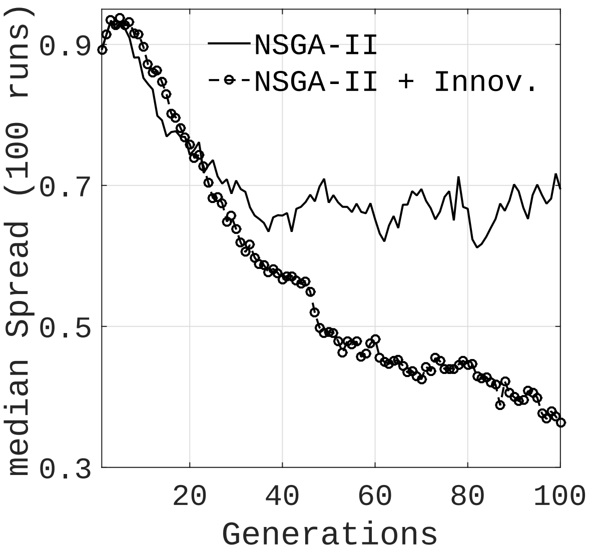

Figure 8: Spread plot for ZDT2m

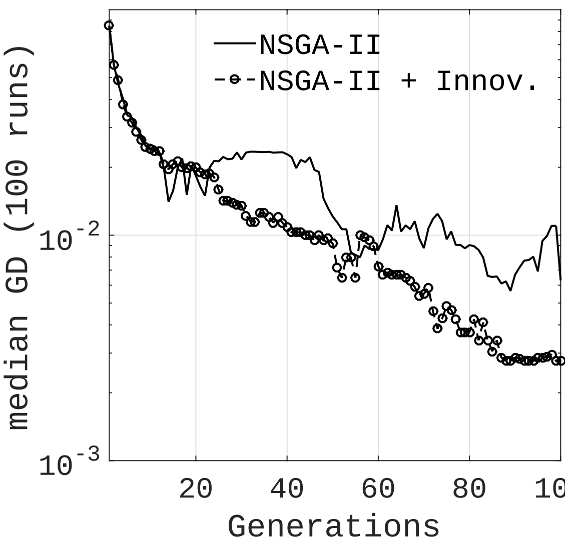

Figure 9: GD versus generations for ZDT3m

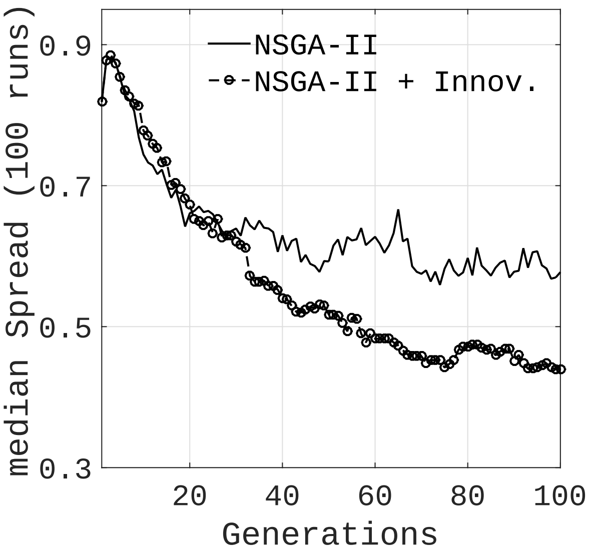

Figure 10: Spread plot for ZDT3m

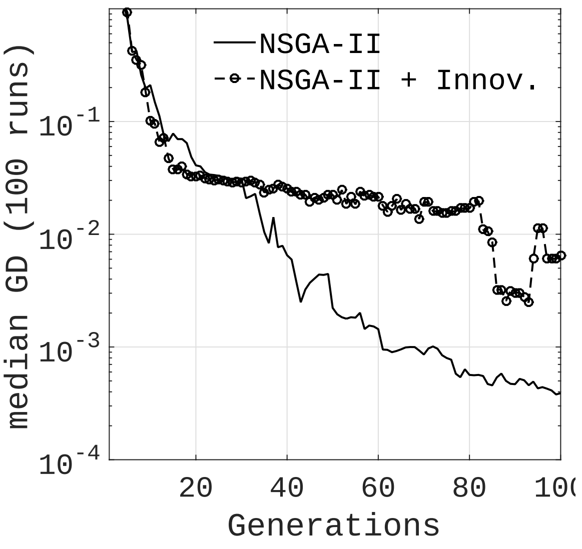

Figure 11: GD semilog-plot for ZDT4m (Pop = 16)

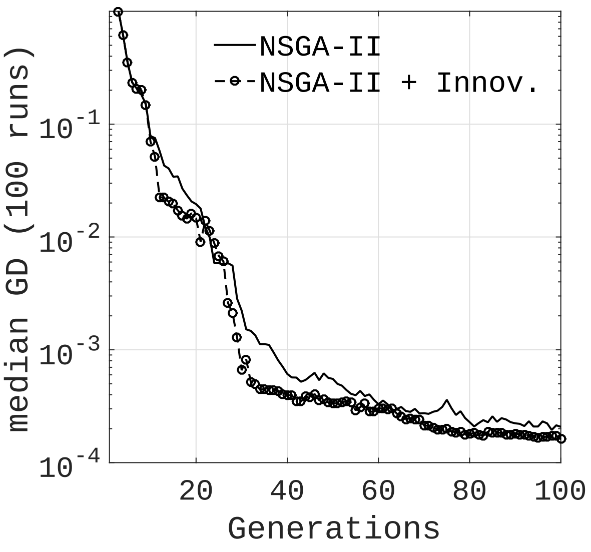

Figure 12: GD versus generations for ZDT4m (Pop = 40)

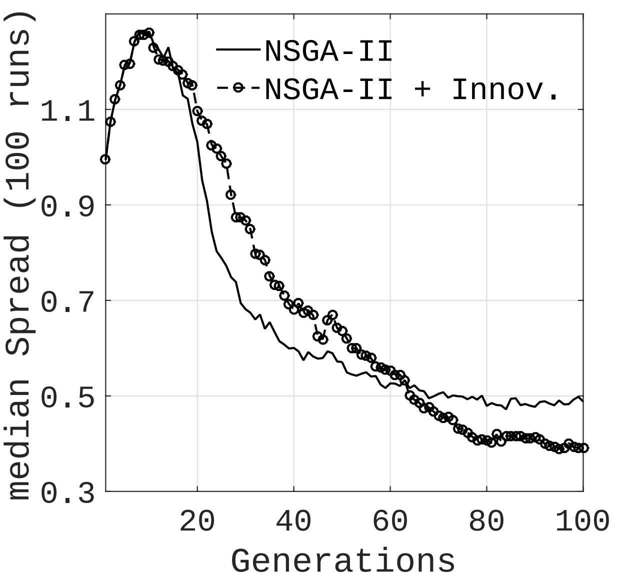

Figure 13: Spread plot for ZDT4m (Pop = 40)

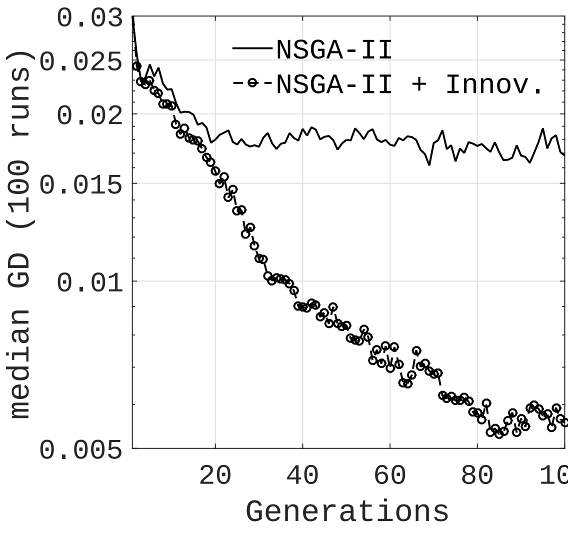

Figure 14: GD versus generations for TBT problem

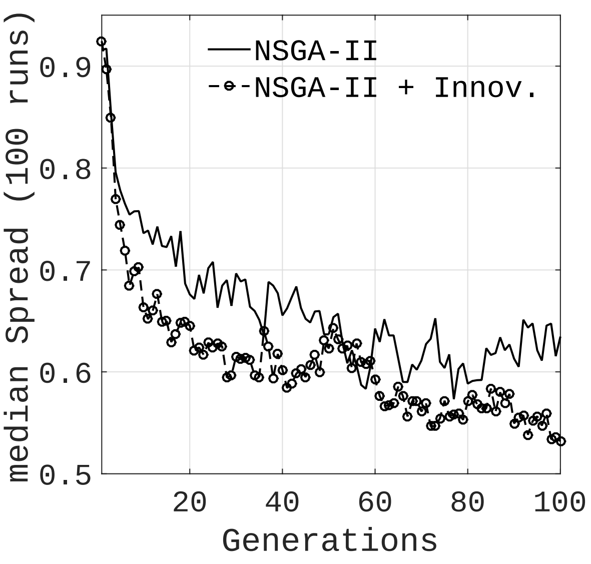

Figure 15: Spread plot for TBT problem