Scalable Polynomial RegEM(a)O for Multi-/Many-objective Platform-based Design Optimization Problems

Ritam Guha, *Student Member, IEEE* and Kalyanmoy Deb, *Fellow, IEEE*

COIN Report 2024004

***Abstract*—The goal of a generic evolutionary multi- or many-objective algorithm is to explore a search space and find the trade-off optimal solutions for two or more conflicting objectives. In platform-based practical design optimization problems, it is not sufficient to just find a set of trade-off optimal solutions, certain regularity properties are expected in the entire fleet of trade-off solutions. For this purpose, we propose a scalable regularity-based optimization framework – RegEM(a)O – which automatically extracts polynomial regularity principles from the resulting Pareto-optimal front of multi- or many-objective problems. Thereafter, it attempts to search for a *regular front* of trade-off solutions following a similar form of polynomial regularity principles. Despite being slightly worse than the true Pareto-optimal solutions, regular solutions possess simple properties among them, making them easily interpretable, their inventory easily maintainable, and easily scalable. In this paper, we apply RegEM(a)O to a number of small and large-scale real-world engineering design problems to demonstrate its practical advantage.**

***Index Terms*—Regularity, Platform-based Design, Multi-/Many-objective Optimization, Pareto Front, Bi-level Search**

I. Introduction

Evolutionary multi- or many-objective (EMO/EMaO) algorithms are special class of optimization algorithms that are efficient in solving real-world multi (for two and three objectives) [1] or many-objective (> 3 objectives) [2] problems. As the number of contradicting objectives increases, the complexity of the problem increases, as well. So, we need special operators to solve multi-/many-objective problems in a reasonable time frame. This led to the development of various EMO and EMaO algorithms. A special trait of multi-/many-objective problems is the presence of multiple Pareto-optimal (PO) solutions based on multiple contradicting objectives. These optimal solutions at the end provide the user with multiple options to select from. Moreover, if the user wants to implement different solutions for different applications, it could be achieved by selecting multiple solutions from the final Pareto-optimal front (PF). This category of design problems is known as Platform-based design [3] where different implementations are used in different underlying platforms or applications.

When multiple solutions are implemented parallelly, it becomes hard if all the solutions are completely different in design. Some regularity principles in the final set of solutions can help the users implement, scale, and maintain multiple solutions at the same time. The idea is described more clearly in Figure 1. Even though the regular set of solutions (called the regular front or RF) is a little bit away from the PF, all of them have the same shape, differing only by dimension, which makes them easier to implement compared to the PO set that may have completely different structures for different solutions. Realizing the need for regularity in EMO, Guha and Deb [4] proposed a regularity-based EMO (RegEMO) algorithm confined to find linear regularity properties. The basic idea of the proposed algorithm was to find a regular set of solutions following some common design principles, instead of working with the original PO solutions. Later, the process of obtaining this regular set of solutions has been simplified and successfully applied over multiple real-life engineering design problems in [5]. The application proved that it is possible to find some regularity or commonly occurring patterns among solutions of a typical real-life multi-objective optimization problem. The only other work in the literature that addresses this concern is [6], where the authors introduced the concept of shared variables to support modularity in products. But, the process did not attempt to find and enforce variable relationships in its solutions.

In this paper, we extend the RegEMO algorithm with the following updates:

- We propose a scalable version of RegEMO, where we replace the computationally complex non-fixed variable categorization process with a simplified version to reduce the computational complexity of the entire approach.
- The new version allows generic polynomial relationships as regularity principles, thereby enhancing the range of trade-off solutions compared to the previous version.
- The upper level search is performed using a multi-objective optimization algorithm, instead of using an exhaustive search, thereby making the new version computationally tractable and scalable.
- The modified version of RegEM(a)O is applied to mul-

Ritam Guha is with the Computer Science and Engineering Department at Michigan State University, East Lansing, USA, e-mail: guharita@msu.edu

Kalyanmoy Deb is with the Electrical and Computer Engineering Department at Michigan State University, East Lansing, USA, e-mail: kdeb@msu.edu

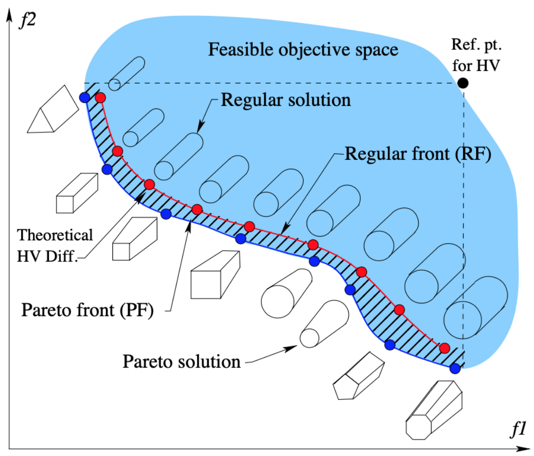


Fig. 1: A regular front contains solutions with common features but may be dominated by the Pareto front. It is worth sacrificing Pareto-optimal solutions having complex features (different shapes, for example) for solutions with a certain regularity (circular cross-section, for example). In platform-based designs, regular solutions have the same cross-sectional shape but vary based on the diameter. This figure is taken from [5].

tiple small and large-scale multi- and many-objective optimization problems.

The rest of the paper is organized as follows: Section II explains the proposed RegEM(a)O algorithm in detail. The results obtained by the algorithm over multiple small and large-scale multi- and many-objective problems are discussed in Section III. Finally, the paper is concluded in Section IV.

II. Proposed RegEM(a)O Procedure

The entire process of RegEM(a)O consists of a bi-level optimization approach where the upper level is a bi-objective search and the lower level can be a multi- or many-objective optimization problem. On the upper level, the approach tries to find a trade-off front between two objectives: the complexity of the regularity principles (denoting interpretability) and the deviation of the RF from the original PF (denoting deviation). On the other hand, the lower level of the procedure extracts regularity principles based on the hyper-parameters provided by the upper level and finds the corresponding RF.

A. Bi-objective upper level Search

Let us consider that there exists a procedure called Reg that takes a set of hyper-parameters $\mathbf{p}$ and the original PF ($\mathbf{x}^*$) obtained by an EM(a)O as its input. The output is expected to be the corresponding regularity principles $\mathbf{R}^\mathbf{p}$ which results in an innovative RF ($\mathbf{x}^{\mathbf{R}^\mathbf{p}}$), as shown in Equation (1).

$$[\mathbf{R}^\mathbf{p}, \mathbf{x}^{\mathbf{R}^\mathbf{p}}] = Reg(\mathbf{x}^*, \mathbf{p}). \tag{1}$$

Corresponding to each regularity principle and RF, we can compute the complexity ($\mathbf{C_p}$) of the principles and the hypervolume deviation ($\mathbf{D_p}$) of the RF from the original PF. A conflict is expected between the complexity and the deviation terms because a more complicated rule may bring the RF closer to the original PF. Thus, the upper level optimization problem becomes

$$\begin{aligned} \underset{(\mathbf{p}, \mathbf{x}^{\mathbf{R}^\mathbf{p}})}{\text{Minimize}} \quad & \{\mathbf{C_p},\ \mathbf{D_p}\}, \\ \text{subject to} \quad & \mathbf{x}^{\mathbf{R}^\mathbf{p}} = \text{argmin}_\mathbf{x} \left(\text{LLP}\left(\mathbf{p}, \mathbf{x}\right)\right), \end{aligned} \tag{2}$$

where LLP() is the lower level optimization problem described in the next subsection.

1) Complexity Score of Regularity Principles: The regularity principles are represented as a set of rules ($\mathbf{R}^\mathbf{p}{}_k, k = 1, 2, \ldots, K$) involving non-fixed variables. The fixed variables are those which have more or less the same value (with respect to a hyper-parameter in $\mathbf{p}$) in the entire PF. We follow a complexity scoring scheme here, which can be modified to investigate other schemes as well. We assign a complexity score of 0.5 for each of the n_f fixed variables. For each polynomial equation, we simply count the degree ($\eta_{k,t}$) of each term (t) in the k-th equation and add them up. This makes a large score (being worse) for higher-degree polynomials. Finally, we also set a precision (λ, dependent on one of the hyper-parameters in $\mathbf{p}$) to each of the real-valued coefficients in the rule. A higher precision signifies less interpretation, so we also add the precision to the complexity score. The final complexity of $\mathbf{R}^\mathbf{p}$ is represented as follows:

$$\mathbf{C_p} = 0.5 n_f + \lambda + \sum_{k=1}^{K} \sum_{t \in \mathbf{R}^\mathbf{p}{}_k} \eta_{k,t} \tag{3}$$

2) Deviation of the Regular Front from Pareto-optimal Front: The deviation of the RF from the PF can be computed using the difference in hypervolumes of the two fronts as follows:

$$\mathbf{D_p} = \frac{HV(\mathbf{f}(\mathbf{x}^*)) - HV(\mathbf{f}(\mathbf{x}^{\mathbf{R}^\mathbf{p}}))}{HV(\mathbf{f}(\mathbf{x}^*))}.$$

Each hyper-parameter vector $\mathbf{p}$ leads to a pair of complexity and deviation terms ($\mathbf{C_p}, \mathbf{D_p}$). In the previous work, the upper level search was performed by specifying some categorical values for each of the variables and exhaustively searching all possible combinations of the hyper-parameters. In this work, the upper level search is done using a mixed-variable version of NSGA-II [7] algorithm. Even though NSGA-II has been used in this work for demonstration due to its enormous popularity in the multi-objective literature, any multi-objective optimization algorithm can be employed. It finds a trade-off front considering the complexity and the deviation.

B. Multi-/Many-Objective lower level Search

The lower level procedure of the optimization framework collects the hyper-parameter vectors $\mathbf{p}$ from the upper level and finds the corresponding rules $\mathbf{R}^\mathbf{p}$ and the respective RF. There are five different hyper-parameters used in this study: $\mathbf{p} = (\zeta, n_b, \eta, \lambda, \tau)$ that get consumed by the lower level optimization. The meaning and use of the hyper-parameters are explained below, except λ which was already explained in the previous subsection. Essentially, the lower level search is expected to find the underlying structure of the Reg function mentioned in Equation (1).

1) Fixed Variable Identification: We have used the same fixed variable identification process as proposed in [5]. First, we measure the degree of variation (Δ_i) in all the variables. The variables with Δ_i less than a threshold $\zeta \in \mathbf{p}$ are declared as fixed variables ($\mathbf{x}^{\text{fx}}$). For the rest of the variables, we divide the range ($x_i^U - x_i^L$) in a certain number of bins (n_b). If a variable has representations in equal or more than 50% bins, we declare it as a non-fixed variable, else it is added to the fixed variable set. This second step ensures that we are not declaring the variables having all the values concentrated towards the edges as non-fixed variables.

2) Regularity in Fixed Variables: After completing the fixed and non-fixed variable identification process, the entire variable set gets divided into fixed and non-fixed variables. The fixed variables have small variations in the Pareto front. So, we fix these variables to their average values.

3) Regularity in Non-fixed Variables: Finding regularity in non-fixed variables is the most difficult part of the process. First, we need to identify the non-fixed dependent ($\mathbf{x}^{\text{nf},d}$) and independent variables ($\mathbf{x}^{\text{nf},i}$) and then fit a polynomial function $\mathcal{P}_k(\mathbf{x}^{\text{nf},i})$ with a maximum degree of $\eta \in \mathbf{p}$ for each dependent variable ($k = 1, 2, \ldots, |\mathbf{x}^{\text{nf},d}|$) in each term ($t$) of the independent variables using the PF data.

In [5], a correlation matrix was formed for all non-fixed variables to extract the level of dependency across the variables which was used to classify the non-fixed variables into dependent and independent variables. The computational complexity of the process restricts the scalability of the entire approach for high-dimensional problems ($n_{\text{nf}} \geq 100$).

In this paper, we replace this process with another approach based on cosine similarity score [8]. The process is presented in Algorithm 1. We start by including the non-fixed variable that corresponds to the maximum coefficient of variation ($\mathbf{x}_{\text{idx}}$ in Algorithm 1) in the independent variable set. Then, we keep on adding the variable with the least cumulative cosine similarity with respect to the independent variables already chosen until the required number of independent variables ($\tau \in \mathbf{p}$) is met.

4) Re-optimization at the Lower Level: After finding the regularity rules in both fixed and non-fixed variables, we then re-formulate the problem by making the non-fixed independent variables the only variables of the new problem. But, for evaluating each solution based on the objective functions of the problem, we need to fetch all variable values ($\mathbf{x} = (\mathbf{x}^{\text{fx}}, \mathbf{x}^{\text{nf,d}}, \mathbf{x}^{\text{nf,i}})$). Essentially, the new lower level problem (LLP()) is equivalent to the original problem under the restriction of the regularity principle and having reduced number of variables, given as follows:

$$\begin{array}{ll} \underset{\mathbf{x}^{\text{nf},i}}{\text{Minimize}} & \mathbf{f}(\mathbf{x}), \\ \text{subject to} & \mathbf{x}^{\text{fx}} = \mathbf{c}^{\text{fx}}, \\ & \mathbf{x}_k^{\text{nf},d} = \mathcal{P}_k(\mathbf{x}^{\text{nf},i}), \forall k, \\ & (x_k^{\text{nf}})^L \leq x_k^{\text{nf}} \leq (x_k^{\text{nf}})^U, \forall k. \end{array}$$

Algorithm 1: Non-fixed Variables Regularity Extraction.

Data: $\mathbf{x}$, τ, λ, x^L, x^U, η
Result: $\mathbf{x}^{\text{nf},i}$, $\mathbf{x}^{\text{nf},d}$, $\mathcal{P}_{1,\ldots,|\mathbf{x}^{\text{nf},d}|}$

$m, n = \mathbf{x}.shape$
/* Add the non-fixed variable with the maximum coefficient of variation to the independent variable set */
for $i = (1, \ldots, n)$ **do**
  $\text{CV}_i = \dfrac{SD(\mathbf{x}_i)}{Mean(\mathbf{x}_i)}$
$\text{idx} \leftarrow \arg\max(\text{CV})$
$\mathbf{x}^{\text{nf},i} \leftarrow \mathbf{x}_{\text{idx}}$
while $|\mathbf{x}^{\text{nf},i}| < \tau$ **do**
  for $j = (1, \ldots, |\mathbf{x}|)$ **do**
    /* At each iteration, the variable with the least cosine similarity w.r.t. all the selected independent variables become the new independent variable */
    $\text{CS}_j \leftarrow 0$
    for $k = (1, \ldots, |\mathbf{x}^{\text{nf},i}|)$ **do**
      $\text{CS}_j \leftarrow \text{CS}_j + cosine\_similarity(\mathbf{x}_j, \mathbf{x}_k^{\text{nf},i})$
  $\text{idx} \leftarrow \arg\min(\text{CS})$
  $\mathbf{x}^{\text{nf},i} \leftarrow [\mathbf{x}^{\text{nf},i}, \mathbf{x}_{\text{idx}}]$
  $\mathbf{x} \leftarrow \mathbf{x} \setminus \mathbf{x}_{\text{idx}}$
$\mathbf{x}^{\text{nf},d} \leftarrow \mathbf{x}$
/* fit polynomial equations for the dependent variables through the independent variables */
for $k = (1, \ldots, |\mathbf{x}^{\text{nf},d}|$ **do**
  $\mathcal{P}_k \leftarrow polynomial\_fit(\mathbf{x}^{\text{nf},i}, \mathbf{x}^{\text{nf},d}, \eta)$
  /* apply λ-precision to the coefficients of the equation */
  $\mathcal{P}_k \leftarrow set\_precision(\mathcal{P}, \lambda)$

C. Decision-Making

After performing the bi-level search process, we expect to obtain a set of non-dominated trade-off points with respect to complexity and hyper-volume deviation which we term as the RegEM(a)O's non-dominated front (NDF). Each point on NDF represents a regular front (RF) on the original objective space along with a rule set $\mathbf{R}^{\mathbf{p}}$ for a specific a hyper-parameter vector $\mathbf{p}$. Finally, we select the most preferred point from NDF using the same decision-making approach used in [5].

III. Results and Discussion

In this section, we present results obtained by our proposed RegEM(a)O algorithm on a test problem and a number of small and large-scale multi- and many-objective engineering design problems. We begin by illustrating the proof-of-principle results on a simple test problem BNH. Then, we apply to other problems.

1) Illustration on BNH: BNH is a two-variable, two-objective test problem. After performing the upper level search using the NSGA-II procedure, we obtain an NDF shown in Figure 2. The NDF represents a trade-off between the complexity of the regularity principles and the hyper-volume deviation of corresponding RFs from the PF. As we go from left to right in the plot, the complexity of the associated rules increases, indicated by the degree of the polynomials and other factors mentioned in Equation (3). If we select three points from the NDF, denoted by '1', '10', and '13', based on their positions along the complexity direction, we observe that they have one, six and nine-degree polynomial equations in them. These number are also marked in Figure 4. The corresponding RFs in the original objective space are displayed in Figure 3.

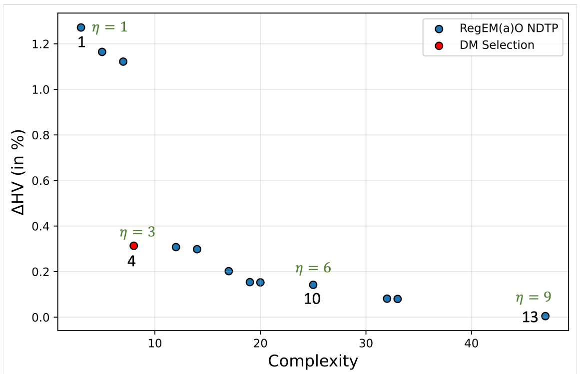
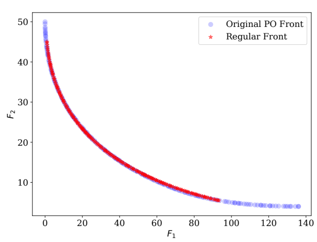
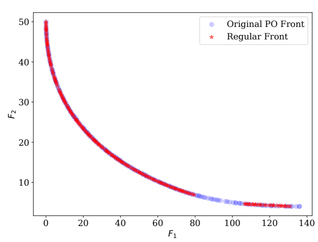


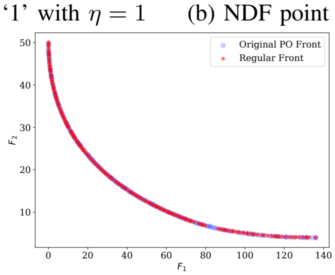
Fig. 2: RegEM(a)O NDF for the BNH problem.



(a) NDF point '1' with $\eta = 1$



(b) NDF point '10' with $\eta = 6$



(c) NDF point '13' RF with $\eta = 9$

Fig. 3: Three RFs from RegEM(a)O NDF set: '1', '10', '13'. While the complexity of them increases in the following order: '1' $<$ '10' $<$ '13', the HV difference reduces in the same order.

Applying the decision-making principle as described in [5] on the obtained NDF points, we find the point denoted by '4' having $\eta = 3$ as the winner. The corresponding RF and regularity principles are shown in Figure 5. This RF has cubic relationships between the dependent non-fixed variable x_2 as a function of non-fixed independent variable x_1, as shown in Figure 5b. The RF covers most of the original PF, but leaves out a part of PF to make a simple cubic relationship between the two variables among RF solutions. We argue here that practitioners may be willing to sacrifice a part of the original PF solutions with not-so-interpretable relationships among variables to obtain slightly sacrificed RF solutions with an interpretable relationship. This illustrates the crux of our proposed approach.

2) Low-dimensional Engineering Design Problems: After showing the results of the proposed approach over BNH, we now perform a similar analysis over multiple low-dimensional engineering design problems outlined in Table I. Instead of showing the RFs and regularity principles for all problems, we show the effect of both the upper level search and the lower level search for the problems by displaying the NDF point with the two extreme complexity values and the point selected by the decision-making process in Figures 6 and 7. For brevity, the corresponding regularity principles are not mentioned in the manuscript[1]. As shown in Table I, the preferred RF discovers low-order polynomials to relate the variables with a small deviation in HV from PF. As the complexity of the regularity principles increases, it becomes possible to overfit to the original PF. So, the RFs in the right side columns in Figures 6 and 7 are always closest to the PF in our experiments. Sometimes, these complex solutions also got selected by our automated decision-making process.

TABLE I: Overview of the low-dimensional engineering design problems used for the regularity search.

Problem Name	#**Vars** (n)	#**Objs** (m)	#**Cons** (c)	Preferred RF $\eta^{\max}$	Preferred RF $\mathbf{D_P}$
Constrained design problems					
Two-member truss	3	2	1	2	2.31×10^{-5}
Crashworthiness	5	3	0	5	2.44
Disc brake design	3	2	3	2	0.88
Speed reducer	7	2	11	2	0.07
Car side impact	7	3	10	3	1.40
Unconstrained design problems					
Reinf. concrete	3	2	0	2	0.13
Pressure vessel	4	2	0	2	0.01
Rocket injector	4	3	0	10	2.60

3) High-dimensional Engineering Design Problems: To demonstrate the scalability of our proposed RegEM(a)O approach, we now apply it to a scalable truss design problem [9]. The truss design problem has cross-sectional size variables and shape variables. The dimension of the problem can be scaled according to the number of shape variables. For testing the scalability of the proposed method, the number of variables has been scaled from 279 to 879 variables. Similar to the past trend, the extreme complexity points along with the preferred point are displayed in Figure 8.

4) Many-objective Engineering Design Problems: In addition to scalable multi-objective problems, we now apply RegEM(a)O to two many-objective (>3 objectives) problems [10]. It becomes difficult to visualize the original PF and RF in

[1]Please find the regularity principles on Google Drive.

Fixed Variables: None
Orphan Variables: None
Non-fixed Independent Variables:
$x_1 \in [0.0, 5.0]$
Non-fixed Dependent Variables:
$x_2 = 1.11x_1 + 0.16$

(a) Regularity principle for NDF '1'.

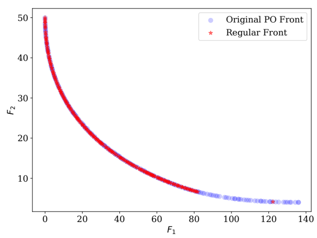
Fixed Variables: None
Orphan Variables: None
Non-fixed Independent Variables:
$x_1 \in [0.0, 5.0]$
Non-fixed Dependent Variables:
$x_2 = 2.3551x_1 - 7.6712x_1^2 + 30.4024x_1^3 - 50.6135x_1^4 + 34.214x_1^5 - 7.6701x_1^6 - 0.0002$

(b) Regularity principle for NDF '10'.

Fixed Variables: None
Orphan Variables: None
Non-fixed Independent Variables:
$x_1 \in [0.0, 5.0]$
Non-fixed Dependent Variables:
$x_2 = 3.07x_1 - 36.31x_1^2 + 373.11x_1^3 - 1948.2x_1^4 + 5720.41x_1^5 - 9808.71x_1^6 + 9716.55x_1^7 - 5148.27x_1^8 + 1129.35x_1^9 - 0.0$

(c) Regularity principle for NDF '13'.

Fig. 4: Regularity principles for the three selected RegEM(a)O NDF points.



(a) RegEM(a)O NDF selected point '4' RF.

Fixed Variables: None
Orphan Variables: None
Non-fixed Independent Variables:
$x_1 \in [0.0, 5.0]$
Non-fixed Dependent Variables:
$x_2 = 1.64x_1 + 0.74x_1^2 - 1.45x_1^3 - 0.0$

(b) RegEM(a)O NDF selected point '4' regularity principle.

Fig. 5: RF and regularity principle for the point selected through decision-making in RegEM(a)O NDF for BNH.

many-objective space. So, we represent the RFs using RadViz plots [11]. It is clear that the minimum complexity solution is too simplistic to cover the entire original PF (shown with blue points). For the water resources problem, linear equations (η = 1) are selected for all points in the NDF, but the number of non-fixed independent variables changes. The minimum complexity point uses only one non-fixed independent variable in the equation, while the maximum complexity point uses two non-fixed independent variables and the maximum complexity point is chosen by the decision-making process. On the other hand, for the machining problem, the minimum complexity point uses linear rules, while the maximum complexity point uses a polynomial of degree nine. In this case also, the maximum complexity point gets selected through the decision-making process.

TABLE II: Overview of the many-objective engineering design problems used for the regularity search.

Problem Name	**#Vars** (n)	**#Objs** (m)	**#Cons** (c)	Preferred RF $\eta^{\max}$	$\mathbf{D_P}$
Water res. planning	3	5	7	1	2.60
Machining problem	3	4	3	9	3.06

IV. Conclusions and Future Work

In platform-based engineering design problems, finding a regularity or commonly occurring pattern is very important across various products for better maintenance, scalability and cost reduction. The definition and complexity of such regularity principles can be different in different contexts. The precursor to this study was able to prove the importance of such a process over multiple engineering design problems but the principles were restricted to only linear equations. In this study, the algorithm has been made more flexible in terms of the degree of such polynomial equations. The process has been also made more scalable by replacing one of the most resource-intensive module of the algorithm with a low-cost module. Moreover, the process has been made more automated and computationally tractable by using a mixed-variable multi-objective optimization algorithm in the upper level search.

These practical improvements have made the process more flexible, scalable and automated. In this paper, we have proposed the newer version of the process and applied it over multiple low and high-dimensional multi-objective engineering design problems and some many-objective design problems, as well. Currently, the process is only applicable to numerical optimization problems. In the future, this work can be extended to problems other than numerical optimization problems like neural architecture search, encryption problems, large language models, etc. In this work, we have used our definition of complexity which might not be applicable in every scenario. The complexity metric might need some fine-tuning based on different application scenarios.

References

[1] N. Srinivas and K. Deb, "Muiltiobjective optimization using nondominated sorting in genetic algorithms," *Evolutionary Computation*, vol. 2, no. 3, pp. 221–248, 1994.

[2] K. Deb and H. Jain, "An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, Part I: Solving problems with box constraints," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 4, pp. 577–601, 2013.

[3] K. Keutzer, A. R. Newton, J. M. Rabaey, and A. Sangiovanni-Vincentelli, "System-level design: Orthogonalization of concerns and platform-based design," *IEEE Transactions on Computer-aided Design of Integrated Circuits and Systems*, vol. 19, no. 12, pp. 1523–1543, 2000.

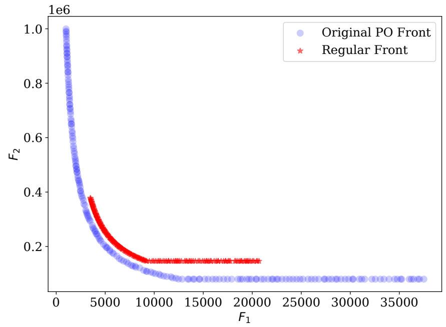
[4] R. Guha and K. Deb, "RegEMO: Sacrificing pareto-optimality for regularity in multi-objective problem-solving," in *International Conference on Evolutionary Multi-Criterion Optimization*. Springer, 2023, pp. 29–42.

[5] R. Guha and K. Deb, "Compromising pareto-optimality with regularity in platform-based multi-objective optimization," *IEEE Transactions on Evolutionary Computation*, 2023.

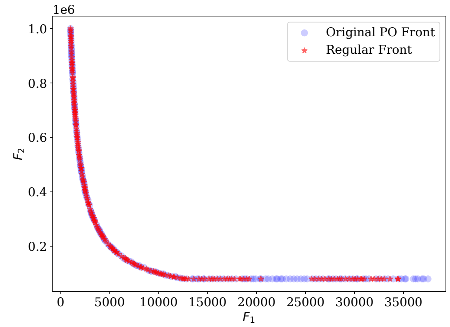
[6] P. Guo, Q. Zhang, and X. Lin, "Approximation of a pareto set segment using a linear model with sharing variables," in *International Conference on Evolutionary Multi-Criterion Optimization*. Springer, 2023, pp. 247–259.

[7] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.

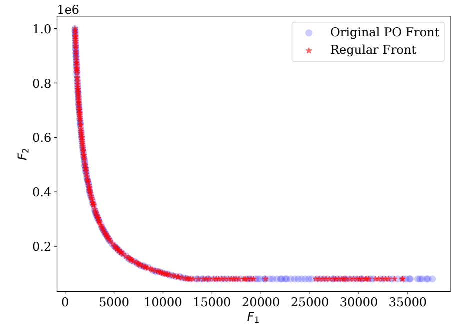
[8] A. R. Lahitani, A. E. Permanasari, and N. A. Setiawan, "Cosine similarity to determine similarity measure: Study case in online essay assessment," in *2016 4th International Conference on Cyber and IT Service Management*. IEEE, 2016, pp. 1–6.



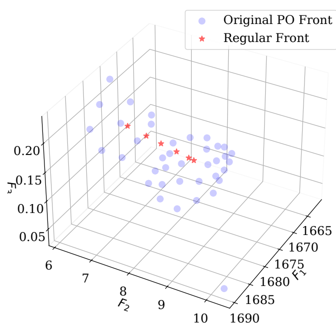
(a) RegEM(a)O RF for two-member truss with minimum complexity.



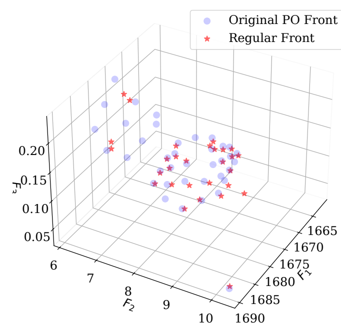
(b) RegEM(a)O RF for the preferred two-member truss problem.



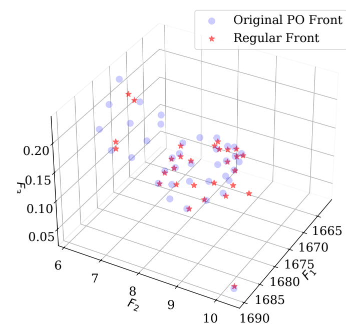
(c) RegEM(a)O RF for two-member truss with maximum complexity.



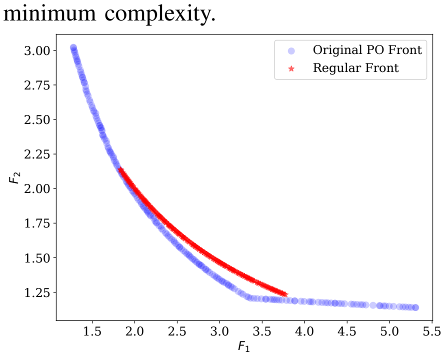
(d) RegEM(a)O RF for crashworthiness with minimum complexity.



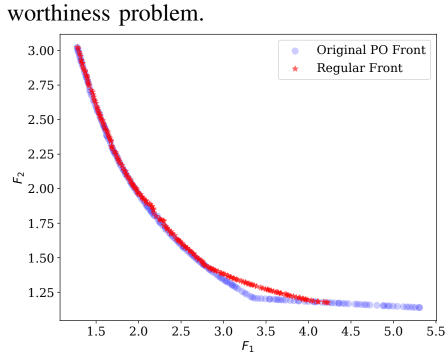
(e) RegEM(a)O RF for the preferred crashworthiness problem.



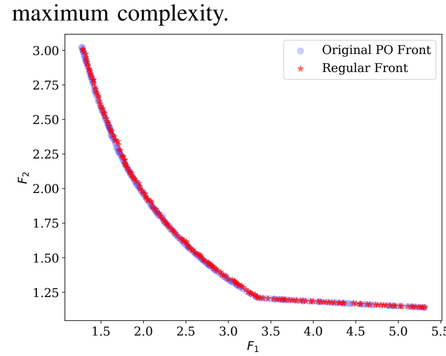
(f) RegEM(a)O RF for crashworthiness with maximum complexity.



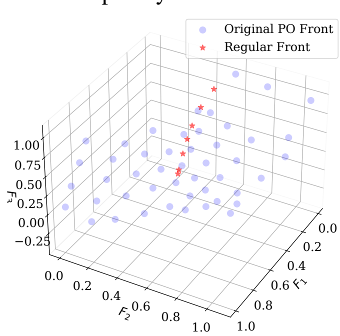
(g) RegEM(a)O RF for disc brake design with minimum complexity.



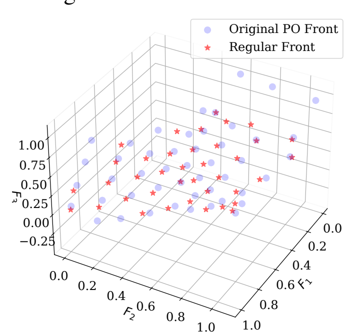
(h) RegEM(a)O RF for the preferred disc brake design.



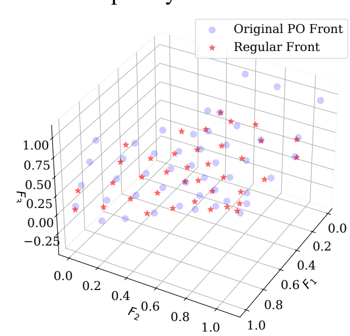
(i) RegEM(a)O RF for disc brake design with minimum complexity.



(j) RegEM(a)O RF for rocket injector spring design with minimum complexity.

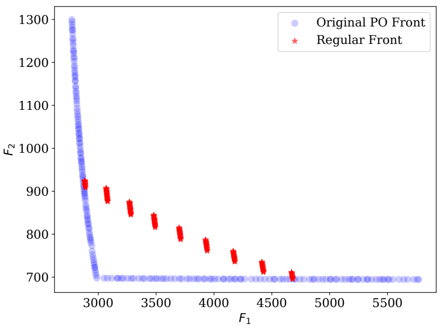


(k) RegEM(a)O RF for the preferred rocket injector spring design.

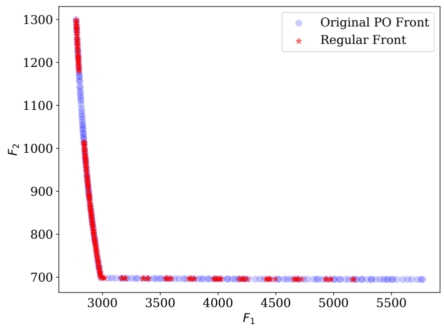


(l) RegEM(a)O RF for rocket injector spring design with maximum complexity.

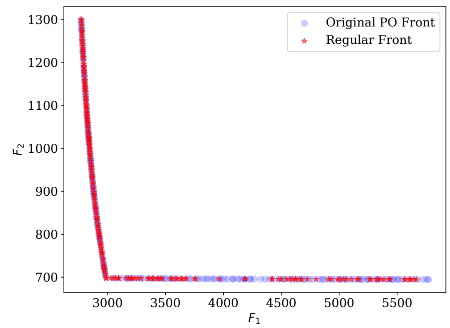
Fig. 6: RFs for three RegEM(a)O NDF points for a number of multi-objective engineering design problems.



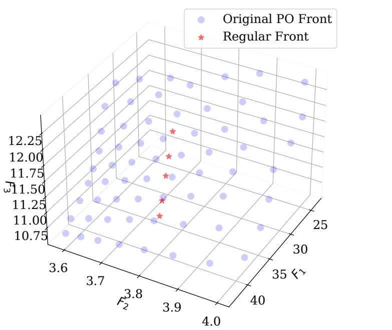
(a) RegEM(a)O RF for speed reducer design with minimum complexity.



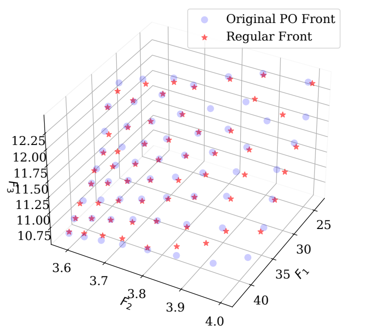
(b) RegEM(a)O RF for the preferred speed reducer design.



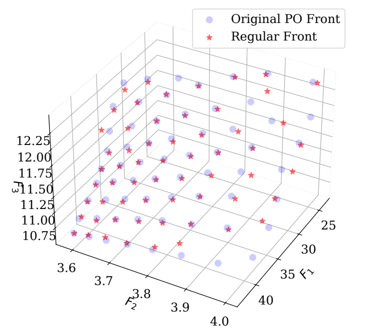
(c) RegEM(a)O RF for speed reducer design with maximum complexity.



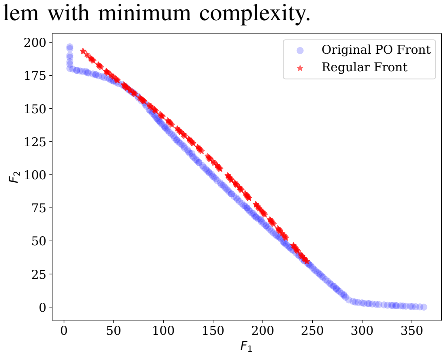
(d) RegEM(a)O RF for car side impact problem with minimum complexity.



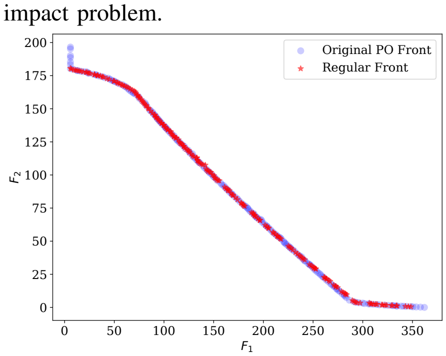
(e) RegEM(a)O RF for the preferred car side impact problem.



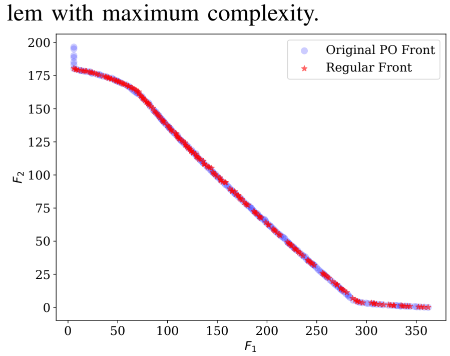
(f) RegEM(a)O RF for car side impact problem with maximum complexity.



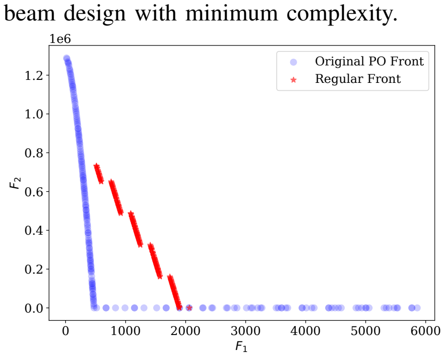
(g) RegEM(a)O RF for reinforced concrete beam design with minimum complexity.



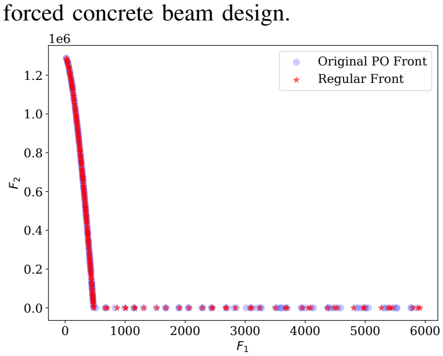
(h) RegEM(a)O RF for the preferred reinforced concrete beam design.



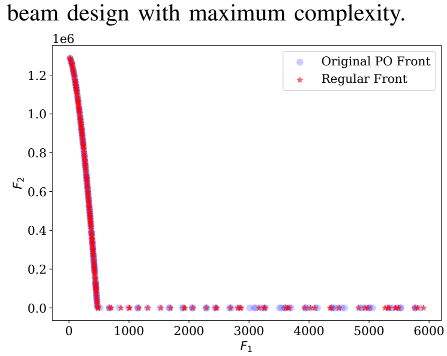
(i) RegEM(a)O RF for reinforced concrete beam design with maximum complexity.



(j) RegEM(a)O RF for pressure vessel design with minimum complexity.

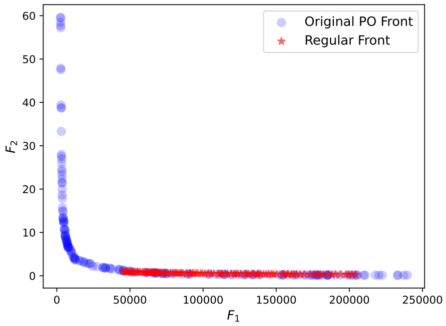


(k) RegEM(a)O RF for the preferred pressure vessel design.

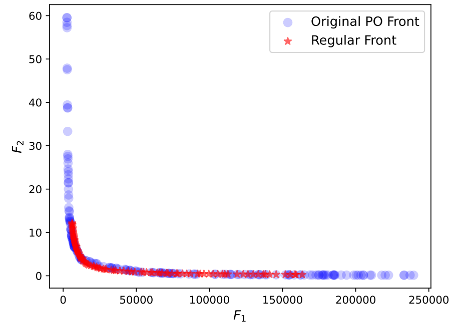


(l) RegEM(a)O RF for pressure vessel design with maximum complexity.

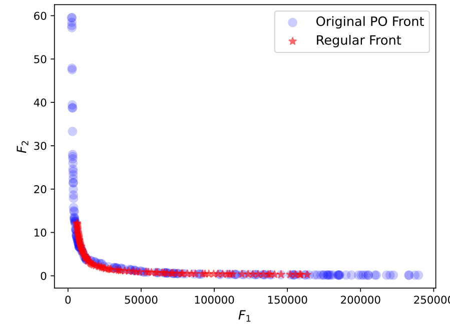
Fig. 7: RFs for three RegEM(a)O NDF points for a number of multi-objective engineering design problems.



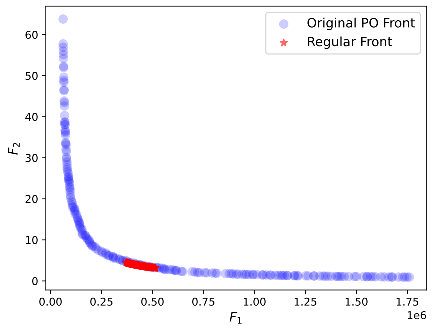
(a) RegEM(a)O RF for 279-variable scalable truss design with minimum complexity.



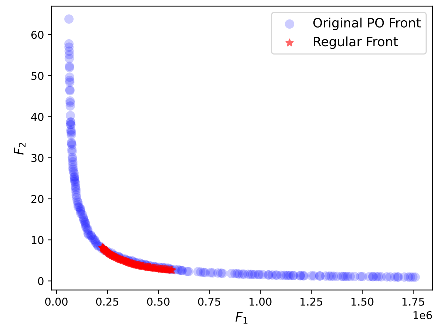
(b) RegEM(a)O RF for the preferred 279-variable scalable truss design.



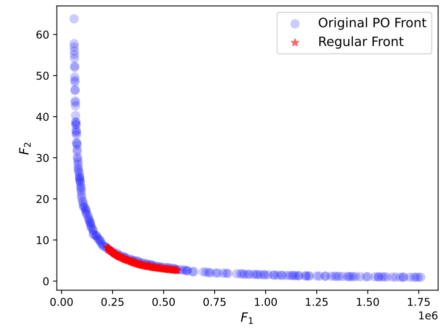
(c) RegEM(a)O RF for 279-variable scalable truss design with maximum complexity.



(d) RegEM(a)O RF for 879-variable scalable truss design with minimum complexity.

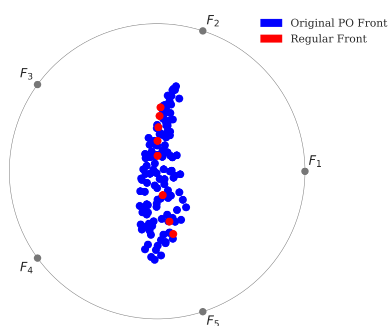


(e) RegEM(a)O RF for the preferred 879-variable scalable truss design.

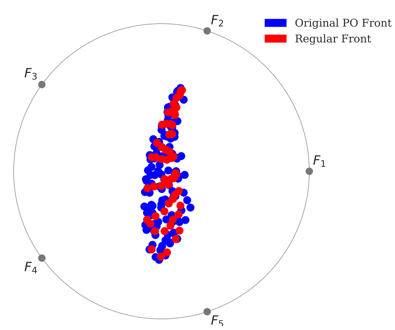


(f) RegEM(a)O RF for 879-variable scalable truss design with maximum complexity.

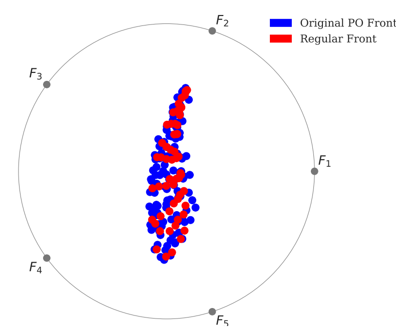
Fig. 8: RFs for three RegEM(a)O NDF points for scalable truss design problems.



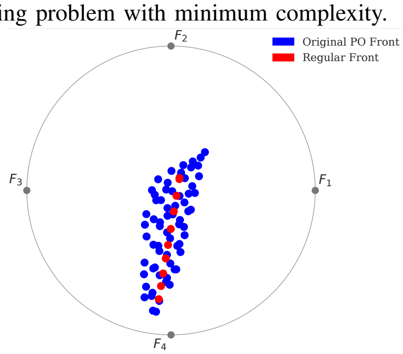
(a) RegEM(a)O RF for water resource planning problem with minimum complexity.



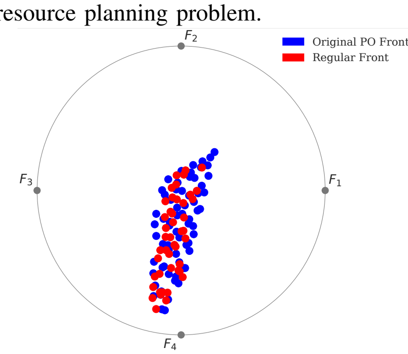
(b) RegEM(a)O RF for the preferred water resource planning problem.



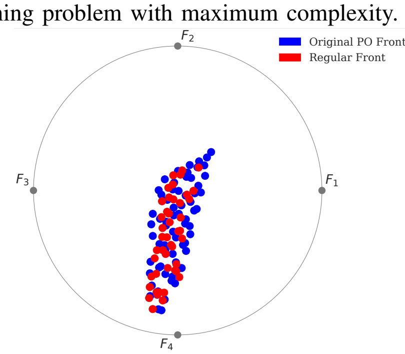
(c) RegEM(a)O RF for water resource planning problem with maximum complexity.



(d) RegEM(a)O RF for machining problem with minimum complexity.



(e) RegEM(a)O RF for the preferred machining problem.



(f) RegEM(a)O RF for machining problem with maximum complexity.

Fig. 9: RFs for three RegEM(a)O NDF points for two many-objective engineering design problems.

[9] A. Ghosh, K. Deb, E. Goodman, and R. Averill, "A user-guided innovization-based evolutionary algorithm framework for practical multi-objective optimization problems," *Engineering Optimization*, vol. 55, no. 12, pp. 2084–2096, 2023.
[10] V. Palakonda and J.-M. Kang, "Many-objective real-world engineering problems: A comparative study of state-of-the-art algorithms," *IEEE Access*, 2023.
[11] B. Bengfort and R. Bilbro, "Yellowbrick: Visualizing the scikit-learn model selection process," *Journal of Open Source Software*, vol. 4, no. 35, p. 1075, 2019.