A Unified Automated Innovization Framework Using Threshold-based Clustering

Sukrit Mittal
Dept. of Mechanical & Industrial Engg.
Indian Institute of Technology Roorkee
Roorkee, India
smittal1@me.iitr.ac.in

Dhish Kumar Saxena
Dept. of Mechanical & Industrial Engg.
Indian Institute of Technology Roorkee
Roorkee, India
dhish.saxena@me.iitr.ac.in

Kalyanmoy Deb
Dept. of Electrical & Computer Engg.
Michigan State University
East Lansing, MI, USA
kdeb@egr.msu.edu

***Abstract*—Automated Innovization procedure aims to extract hidden, non-intuitive, closed-form relationships from a design task without human intervention. Existing procedures involve the application of an Evolutionary Multi-objective Optimization (EMO) Algorithm in two phases. The first phase of EMO algorithm leads to a set of Pareto-optimal (PO) solutions, while the second phase helps identify the implicit relationships. The latter involves clustering which in turn enables the evaluation of innovization-driven objective function. The existing procedures for Automated Innovization differ in their clustering technique and objective formulation. Unlike any existing study, this paper proposes a Unified Automated Innovization (UAI) framework which can deal with both continuous and discrete variable problems, and identify the inherent single- or multiple-cluster rules, as the case may be. The scope and efficacy of the proposed UAI, demonstrated through some benchmark design problems, is rooted in the novel contributions made in the clustering technique, and innovization-driven objective function formulation(s).**

***Index Terms*—Innovization, Design Principles, Discrete Space, Knowledge Mining, Optimization, One-Dimensional Clustering.**

I. Introduction

Automated Innovization procedure was introduced to extract hidden and non-intuitive relationships from a design task without human intervention. The goal has been to obtain closed-form and simple-to-understand relationships which describe the complete set or subset(s) of the Pareto-optimal (PO) solutions. This procedure is initiated by first obtaining a set of PO solutions for a given multi-objective optimization problem through an EMO algorithm. Subsequently, the PO solutions are clustered, contributing to evaluation of the innovization-driven objective function in the second application of the EMO algorithm, leading to revelation of the innovized rules.

Some other studies have reported achieving a similar goal using data mining techniques in the domain of post-optimality analysis. However, some human interaction is required to complete the analysis defying the motive of having an automated procedure in the first place. This can be seen in [1], where k-means clustering is used on the PO solutions to simplify the data-visualization. Similarly, kriging and self-organizing maps have been used in [2] to visualize the structure of decision variables from the PO solutions, and the idea of *Pareto-shells* have been used in [3] to analyze the PO solutions visually.

The first proposition of automated innovization [4] involved a GA-based unsupervised learning procedure with a newly developed grid-based clustering approach. However, this procedure could only deal with the design-optimization tasks having continuous variables. Subsequently, the complexities associated with discrete search spaces were tackled, and a procedure to deal with design tasks involving discrete variables was proposed in [5]. While these frameworks have laid the foundations for automated innovization, some associated challenges are obvious, including:

- despite some similarities between these frameworks, they are not compatible for integration, which mandates their separate use depending on the continuous or discrete nature of the problem.
- the Innovization problem formulation in [4] does not have a *balanced* objective function which results in deviation of the innovized rules from the true relationships.
- in [5], candidate rules above a significance-threshold are reported, but their true significance can not be evaluated. Furthermore, though design problems with multiple-cluster design rules may exist, this framework is capable of finding only single-cluster rules.

This paper is rooted in the motivation to overcome the above challenges and pitfalls. Towards it, a unified framework for automated innovization (UAI) has been proposed, that is not just capable of tackling both continuous and discrete search spaces, but is also equipped to identify single- or multiple-cluster innovized rules which conform with true implicit relationships. The unprecedented scope and empirically demonstrated efficacy of this framework are ingrained in the novel proposition of a clustering technique and innovization-driven objective function formulation(s). In that, *UAI: Single-objective Innovization Problem* (UAI-SIP), and *UAI: Multi-objective Innovization Problem* (UAI-MIP) formulations are proposed. The results demonstrate the trade-off between lower computational expense associated with UAI-SIP, and a better exploration & more knowledge associated with UAI-MIP aided with a trade-off-based decision-making analysis.

The remaining paper is structured, as follows. Section II

This research is conducted under the MHRD sponsored SPARC project (P66), titled, 'INNOVIZATION: Innovation through Optimization & Machine Learning.'

978-1-7281-6929-3/20/$31.00 ©2020 IEEE

discusses the existing automated innovization procedures, following which a new clustering procedure along with UAI (both formulations) are presented in Section III. Further, Section IV presents the results on benchmark problems, and the paper is concluded with Section V.

II. Related Studies

The existence of patterns of variable values among Pareto-optimal solutions were first observed in 2003 [6]. Deb and Srinivas [7] named the task of identifying patterns from Pareto-optimal solutions as a task of "innovization". Until 2010, an innovization task was performed manually by plotting variables pair-wise and observing for any obvious relationships. In 2010, Bandaru and Deb [4] proposed an automated innovization procedure by finding the clusters of non-dominated points exhibiting a relationship using an optimization problem. The proposition was restricted for continuous variables only. We call that study as "Existing Automated Innovization" or EAI study here and use it as the base framework. The capability of automated innovization process was further extended to deal with discrete search space in [5], [8], [9]. We call the study in [5] as "Existing Automated Innovization in Discrete" or EAID. Here, we provide a more detail description of EAI in the following subsection, as we propose an extension of EAI in this paper.

A. Existing Automated Innovization (EAI)

Given that an EMO algorithm can be used to find m Pareto-optimal solutions for an optimization problem, the existing automated innovization (EAI) procedure tries to find the hidden relationships between various variable-function-constraint combinations. However, the relationships were restricted to follow a fixed mathematical structure $\psi(\phi(x))$ given in equation (1).

$$\psi(\phi(x)) \equiv \Pi_{j=1}^{N} \phi_j(x)^{b_j} = c. \tag{1}$$

Here, ϕ_j's are the *basis functions* which can be design variables, objective functions, constraints, or any other functions of interest which can be derived from the design variables (such as, $(x_1 + x_2)$, if the designer considered it as a potentially important information for the problem). With b_j's as real numbers, this forms a power-law rule. With a b_j-vector denoted by a population member, the respective c-value (right side of the power law) is computed as the left side of the power law for clustered points obtained from Pareto-optimal dataset, described in next subsection. The real nature of b_j values can result in redundant solutions like $\phi_1^1\phi_2^2 = c_1$ and $\phi_1^2\phi_2^4 = c_2$, where both represent the same rule with $c_2 = c_1^2$. To avoid such cases, a transformation is applied as given in Equation 2 to limit $b_j \in [-1, 1]\ \forall\ j = 1$ to N:

$$b_j \leftarrow \frac{b_j}{\{b_p | p = \text{argmax}_i |b_i|\}}. \tag{2}$$

For any candidate rule, two points are to be focused say, (a) How many Pareto-solutions adhere to this rule, and (b) How closely they satisfy the rule. A grid-based clustering was proposed to answer the aforementioned questions. Using this technique, the process deployed a GA to obtain the candidate rule by solving an optimization problem, provided in the next section. For this GA, the chromosome has $N+1$ variables out of which N are b_j's and a parameter deciding on the number of divisions d which is further explained below.

1) Grid-based Clustering of EAI Procedure: The clustering procedure fits a grid defined by given number of divisions d on each objective axis on the already obtained m Pareto data-points by an EMO. Thus, for an M-objective problem, there are a total of 2^M subdivisions. Further, the subdivisions containing more data-points than a threshold value (m/d) are marked as sub-clusters. Once all the sub-clusters are identified, the adjacent ones are combined together, thereby forming the clusters with high density regions. The GA chromosome provides the b_j's which are used to obtain a set of c-values from m input data points. These c-values are clustered using this technique to reveal a few parameters: total number of clusters ($\mathcal{C}$), the coefficient of variation of c-values in each cluster (c_v), and the number of unclustered points ($\mathcal{U}$). These parameters are used to evaluate the fitness of each individual in the Innovization-based GA. There is another parameter ϵ which is used to redefine the sub-cluster threshold to $(m/d+\epsilon)$ when clustering for the last time. This removes the clustered points which barely meet the threshold requirement. The value of ϵ is fixed at 3 which is another user-defined parameter.

2) The Optimization Problem: As mentioned earlier, a GA is used in the EAI procedure to find an appropriate rule which is present in the Pareto-optimal dataset. The optimization formulation is given below:

$$\begin{aligned} \text{Minimize} \quad & \mathcal{C} + \mathcal{U} + \Sigma_{i=1}^{\mathcal{C}} c_v^{(i)}, \\ \text{Subject to} \quad & -1 \le b_j \le 1, \quad \forall j, \\ & |b_j| \ge 0.1, \quad \forall j, \\ & 1 \le d \le m, \\ & \mathcal{U} \le 0, \\ & d \text{ is an integer and } b_j\text{'s are real.} \end{aligned} \tag{3}$$

For every cluster, c_v is a parameter which is evaluated from the c-values belonging to that particular cluster. If the mean and standard deviation of the c-values are μ and σ, then c_v is computed as follows:

$$c_v = \frac{\sigma}{\mu} \times 100\%. \tag{4}$$

The GA used to solve the above problem included *selection*, *crossover*, and *mutation* operators, but no *survival-selection* operator was used, making the search process slow. The final solution of the optimization process resulted in an *innovized* rule. A later study [10] implemented a multi-modal GA to find multiple innovized rules for a problem.

B. Extension of EAI to handle Discrete Variables (EAID)

We have restricted this paper to single-rule discovery in one run, hence the description of handling discrete variable problems presented in [5] has been explained here in context

of single-rule discovery only. In EAID, The grid-based clustering has been replaced with a new window-based clustering procedure to handle discrete variables better, along with some changes in optimization problem definition as further described in this section.

For a given Pareto-dataset with discrete variables, the evaluated m c-values (to-be-clustered) are also discrete in nature. In such cases, grid-based clustering results in rules with multiple clusters and very less c_v value. This is not a preferred solution when it is known that the theoretical relationship has lesser number of clusters. However, this *window-based clustering* procedure requires minimum desired rule significance S_w as an input parameter. The step-wise clustering process is given in the steps below.

- Evaluate the number of points to form a window P_w using $P_w = m \times S_w$.
- Evaluate total number of windows $\mathcal{W}$ using $\mathcal{W} = m - P_w + 1$.
- Assume each window to be a cluster and evaluate $c_v^{(k)}$ $\forall$ $k = 1, 2, ..., \mathcal{W}$, using equation (4).
- Find c_v of the cluster with least variation, i.e., c_{vw} using equation (5).

$$c_{vw} = \{c_v^{(p)} | p = \text{argmin}_i |c_v^{(i)}|\} \tag{5}$$

The optimization process in EAID is similar to the one described in EAI but the optimization problem definition is different. However, EAID can not reveal the true significance of the rule but can ensure some minimum significance as per S_w. Another limitation of this procedure is that only the rules with $\mathcal{C} = 1$ can be derived. The formulation is shown in equation (6). There is another term $c_v^{\max}$, which is a user-defined parameter to put an upper limit to the intra-cluster variation of any candidate rule.

$$\begin{aligned} \text{Minimize} \quad & c_{vw}, \\ \text{Subject to} \quad & -1 \leq b_j \leq 1, \quad \forall j, \\ & |b_j| \geq 0.1, \quad \forall j, \\ & c_{vw} \leq c_{vw}^{\max}, \\ & b_j\text{'s are real.} \end{aligned} \tag{6}$$

III. Proposed Algorithm

In this section, we propose a novel *threshold-based clustering* procedure, followed by the two variants of the UAI, namely, UAI-SIP and UAI-MIP. The similarity between these variants is that both utilize the EAI as the base framework, and the newly proposed clustering procedure. Their difference lies in the fact that while UAI-SIP pursues identification of innovized rules through a single innovization-driven objective function, UAI-MIP pursues it through two innovization-driven objective functions.

A. Threshold-based Clustering

The clustering procedure has a vital role in the innovization framework since the clustering results act as direct inputs for the innovization objective function. Like EAI, UAI also optimizes objective function(s) which is(are) derived from the parameters obtained by clustering the c-values. However, instead of using the grid-size parameter d from EAI, a new parameter *cluster-threshold parameter* ($\mathcal{T}$) is introduced. For any individual in any generation of the GA procedure in innovization with N basis functions, the total number of variables become $N + 1$, i.e., N number of b_j values and one parameter $\mathcal{T}$.

The proposed clustering procedure (algorithm 1) is based on the concept of slope. Consider a list of single-valued data points (c) of length m, indexed as c = [c_1, c_2,..., c_m]. These values are normalized and sorted in ascending order. The plot of normalized index-values (X-axis) and normalized c-values (Y-axis) is shown in Figure 1. If the c-values follow an arithmetic progression (red crosses), then the line joining them will have $slope = 1$. On the other hand, if a few points come closer to each other to form a cluster (as depicted by green circled points between black dotted boundaries), the remaining points will get distant from each other during normalization. In other words, the points forming a cluster will have a lower slope value than 1, whereas the unclustered points will have a slope value greater than 1. This enables us to create a clear demarcation between the clustered and unclustered points with the slope threshold, here referred to as $\mathcal{T}$. In Algorithm 1, it can be observed that the normalization has been done with limits on c-values increased up to twice the range, thus making the ideal slope threshold ($\mathcal{T}$) to be 0.5 or less. Moreover, there is another threshold used to differentiate between a clustered and unclustered points given by $0.005 \times m$, i.e., 0.5% of the entire dataset. Since, the standard dataset size used in this paper is around 1,000, the threshold suggests that any cluster with < 5 will not be considered. This measure can be changed as per designer's requirements or restrictions.

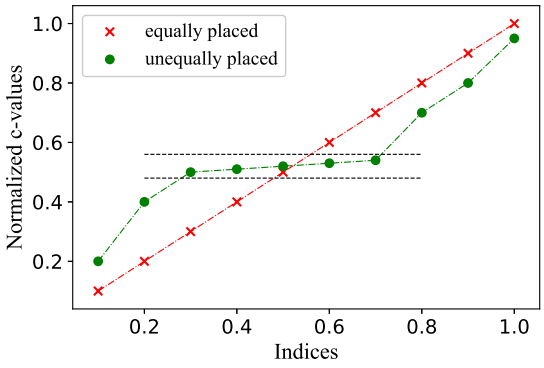

Fig. 1. Illustration of c-values in the normalized space.

Defining $\mathcal{T}$ has a secondary motive as well. This parameter can be used as an extra property of the discovered design principle in addition to the significance of the rule. The significance ($S\%$) reflects the fraction of total number of points which follow the innovized rule, whereas the cluster-threshold parameter $\mathcal{T}$ reflects how well those $S\%$ points follow the obtained rule. This enables the designer to compare

two differently converged rules as well as comment on how well the candidate-rule represents the Pareto-front. The full account of this parameter on the outcome of optimization is discussed in Section III-B.

Algorithm 1 Threshold-based clustering of c-values.

Input: Sorted c-values c, Pareto-dataset size m, Cluster-threshold parameter $\mathcal{T}$

Output: No. of clusters ($\mathcal{C}$), No. of unclustered points ($\mathcal{U}$) and Clustered Points (CP)

$count \leftarrow 0$
$\mathcal{C} \leftarrow 0$
$CP \leftarrow \emptyset$
$c_{upperlim} \leftarrow \texttt{max}(c) + (\texttt{max}(c) - \texttt{min}(c))/2$
$c_{lowerlim} \leftarrow \texttt{min}(c) - (\texttt{max}(c) - \texttt{min}(c))/2$
$e \leftarrow (c_{upperlim} - c_{lowerlim})/(m-1)$
$I_{start}, I_{end} \leftarrow 0$
for $i = 1$ to $m - 1$ **do**
  $slope_i \leftarrow (c_{i+1} - c_i)/e$
  if $slope_i \geq \mathcal{T}$ **or** $i = m - 1$ **then**
    $I_{start} \leftarrow I_{end}$
    $I_{end} \leftarrow i$
    if $i = m - 1$ **then**
      $I_{end} \leftarrow I_{end} + 1$
    end if
    if $I_{end} - I_{start} \geq 0.005 \times m$ **then**
      $CP \leftarrow CP + c[I_{start} : I_{end}]$ `%new cluster`
      $count \leftarrow count + (I_{end} - I_{start})$
      $\mathcal{C} \leftarrow \mathcal{C} + 1$
    end if
  end if
end for
$\mathcal{U} \leftarrow m - count$
return No. of clusters ($\mathcal{C}$), No. of unclustered points ($\mathcal{U}$) and Clustered Points (CP)

B. UAI: Single-objective Innovization Problem (UAI-SIP)

The innovization objective function in the EAI framework comprises of three different terms: (i) number of clusters $\mathcal{C}$, (ii) number of unclustered points $\mathcal{U}$, and (iii) sum of coefficient of variation for clusters $\sum_{i=1}^{\mathcal{C}} c_v^{(i)}$, as given in equation (3).

Minimizing c_v ensures minimizing the intra-cluster variation of the c-values for a cluster, while minimizing $\mathcal{T}$ can create pressure to reduce the difference between `max` and `min` of the c-values of that cluster, eventually complimenting each other towards better convergence. Also, it is interesting to notice that the sum of all c_v(s) can reduce even if one of the clusters converges, whereas $\mathcal{T}$ reduces only when all the clusters improve together. These two measures, put together in the objective function, can account for the accuracy of the candidate-rule, i.e., how well the obtained rule represents or explains the set of clustered points. More c_v or $\mathcal{T}$ means more deviation of the innovized rule from the true relationship or an indication that there may not even exist any relationship.

There is another modification we propose w.r.t. the intra-cluster variation measure, that is, the sum of all c_v(s) have now been replaced with their mean, which can be calculated using equation (7). It is known that forcing data-points to be in a cluster, i.e., reducing $\mathcal{C}$ leads to increase in the c_v value. While using the sum of all c_v(s), it is not certain whether $\Sigma_{clusters} c_v$ will increase due to increasing c_v(s) or decrease due to decreasing $\mathcal{C}$. In order to make this term independent of change in $\mathcal{C}$, the mean of all c_v(s), i.e., $\overline{c_v}$ is used. This also enables us to have an apple-to-apple comparison between two candidate-rules in terms of intra-cluster variation.

$$\overline{c_v} = \frac{\Sigma_{i=1}^{\mathcal{C}} c_v^{(i)}}{\mathcal{C}}. \tag{7}$$

In EAI and associated literature, the innovization results have been demonstrated with PO dataset size $m = 1000$ and minimum desirable significance ($\tau_s = 70\%$ or 80%). With these values, $\mathcal{U}$ can vary in range [0,300], while c_v lies in range [0,100] (from equation 4) and $\mathcal{C}$ remains a small integer value. Though towards convergence of innovization process, the value of latter two terms reduce but the unclustered points can still take any value depending upon m and τ_s. Thus, it is an *unbalanced* objective formulation. Moreover, $\mathcal{U}$, having the maximum value, dominates the innovization process to yield a rule with maximum possible significance despite having lower accuracy. Hence, $\mathcal{U}$ has been eliminated in the proposed objective function, and the constraint $\mathcal{U} = 0$ is also removed. To ensure the minimum desirable significance (τ_s), a new constraint is put. The proposed problem definition for UAI-SIP is given in equation (8).

$$\begin{aligned} \text{Minimize} \quad & (\mathcal{C} + \overline{c_v} + \alpha\mathcal{T}), \\ \text{Subject to} \quad & -1 \leq b_j \leq 1, \quad \forall j, \\ & |b_j| \geq 0.1, \quad \forall j, \\ & 0.1 \leq \mathcal{T} \leq 10, \\ & \frac{(m - \mathcal{U})}{m} \times 100\% \geq \tau_s, \\ & \mathcal{T} \text{ and } b_j\text{'s are real.} \end{aligned} \tag{8}$$

There are two unexplained additions in equation (8): α and the bounds on $\mathcal{T}$. The lower bound on $\mathcal{T}$ was kept as 0.1 since $\mathcal{T} = 0$ can result in no clustering at all, while the upper bound was kept at 10 to allow some clustering in early generations of GA despite with less accuracy. To tackle the unbalanced formulation till some extent, α is introduced to match the scales of $\overline{c_v}$ and $\mathcal{T}$. Since $\overline{c_v}$ can vary from 0 to 100 and $\mathcal{T}_{\text{max}} = 10$, $\alpha = 10$ is an appropriate choice for this formulation. However, it may be noted that only the scales of two terms (out of three) are met, $\mathcal{C}$ still lies on a different scale leaving the objective function unbalanced.

C. UAI: Multi-objective Innovization Problem (UAI-MIP)

As mentioned earlier, UAI-MIP has been proposed to have a *balanced* learning objective in the innovization GA, and find the most applicable relationship. $\mathcal{T}$ and $\overline{c_v}$ are properties of the input dataset while $\mathcal{C}$ is the property of the design problem,

which makes any kind of weight distribution restricted, i.e., it can not be generalised. Hence, a multi-objective formulation can be a more generalised proposition, keeping the terms with different scales as separate objectives. However, the creates a need for a mechanism which can choose one solution from the obtained multiple solutions.

Extending this argument to the proposed framework, the UAI-SIP problem definition in equation (8) is converted into a multi-objective formulation as given in equation (9). UAI-MIP can explore the trade-off between multi-cluster rules and their respective accuracy (how well they represent the input PO dataset). The trade-off analysis, which explore the optimal solutions and choose one, is explained later with an example.

$$\begin{aligned} \text{Minimize} \quad & f_1 = \mathcal{C}, \\ \text{Minimize} \quad & f_2 = \alpha\mathcal{T} + \overline{c_v}, \\ \text{Subject to} \quad & -1 \leq b_j \leq 1, \quad \forall j, \\ & |b_j| \geq 0.1, \quad \forall j, \\ & 0.1 \leq \mathcal{T} \leq 10, \\ & \frac{(m-\mathcal{U})}{m} \times 100\% \geq \tau_s, \\ & \mathcal{T} \text{ and } b_j\text{'s are real.} \end{aligned} \tag{9}$$

However, the advantages of MIP are computationally more expensive. Considering M-objectives and N population-size, the GA framework has complexity of $O(MN^2)$ (from NSGA-II [11]). Increasing M from 1 to 2 directly affects the overall complexity of the algorithm. Also, it was observed during initial experimentation that it takes more generations for UAI-MIP to converge than UAI-SIP due to the absence of survival-selection operator in the base framework. But the authors also realize that even though UAI-MIP framework requires more generations, those solutions may not be achieved by UAI-SIP framework at all.

1) An Illustrative Example: Let us consider two-bar truss problem from [4], as it's theoretical relationships are already known. Using UAI-MIP, the results (only non-dominated ones) for rule-discovery between V and S are in Table I and plotted in Figure 2, where the red crosses represent all population members (at generation 200) obtained while the green dots represent the four non-dominated solutions. It is clear that all four solutions represent a similar power law rule ($S \times V = 400$) with slight variation in the respective c-values.

It can be observed that the coefficients b_j(s) of V and S have almost converged to their theoretical values. As we move down in the table, the accuracy of rule increases (since $\mathcal{T}$ decreases). Thus, the user can differentiate between the different rules based on their accuracy since the $\mathcal{C} = 1$ rule is easier to explain but it represents the input PO dataset with least accuracy. It can be observed in the last column of Table I that the c-values in case of multiple clusters aren't much different from each other. From the four points in the two-objective space in Figure 2, we notice the $\mathcal{C} = 1$ solution (the left-most point) has the best trade-off value, calculated by using average trade-off of

TABLE I
RULES DISCOVERED IN TRUSS PROBLEM (SORTED BY CLUSTER-THRESHOLD PARAMETER $\mathcal{T}$). MULTIPLE CLUSTERS REPRESENT SIMILAR POWER LAW RULES WITH SLIGHT CHANGES IN THE c VALUES.

b_j-**V**	b_j-**S**	$\mathcal{T}$	**Significance**	$\mathcal{C}$	c-**Values**
1.00000	0.99999	0.35862	90.3%	1	400.596
1.00000	0.99999	0.35857	90.3%	2	400.583 401.785
1.00000	0.99999	0.26917	89.2%	3	400.523 401.569 401.769
1.00000	0.99999	0.24976	89.2%	4	400.522 401.551 401.669 401.769

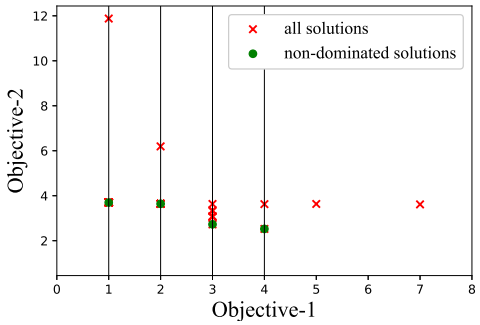

Fig. 2. Obtained front from UAI-MIP on TRUSS (S,V) rule.

moving to a neighboring point. The trade-off for a point $\mathbf{x}^{(i)}$ with a neighboring point $\mathbf{x}^{(j)}$ is defined as follows [12]:

$$R(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}) = \frac{w_{\text{loss}} \times \text{Loss}_f(\mathbf{x}^{(i)} \rightarrow \mathbf{x}^{(j)})}{w_{\text{gain}} \times \text{Gain}_f(\mathbf{x}^{(i)} \rightarrow \mathbf{x}^{(j)})}. \tag{10}$$

For evaluating trade-off(s), the individual objectives are normalized and the weight vector $\mathbf{w}$ used here is $(2, 1)^T$, providing more importance for choosing a small cluster solution. Labeling the four non-dominated solutions in Figure 2 as $\mathbf{x}^{(1)}$, $\mathbf{x}^{(2)}$, $\mathbf{x}^{(3)}$ and $\mathbf{x}^{(4)}$, the average trade-off values as per equation (10) for moving to left and right are presented in Table II.

TABLE II
TRADE-OFF VALUES FOR UAI-MIP IN TRUSS PROBLEM

Point	Obj-1	Obj-2	$R(\mathbf{x}^{(i)}, \mathbf{x}^{(i+1)})$	$R(\mathbf{x}^{(i+1)}, \mathbf{x}^{(i)})$	Average
$\mathbf{x}^{(1)}$	**1**	**3.698**	**13.949**	-	**13.949**
$\mathbf{x}^{(2)}$	2	3.642	0.856	0.072	0.464
$\mathbf{x}^{(3)}$	3	2.728	3.849	1.169	2.509
$\mathbf{x}^{(4)}$	4	2.525	-	0.259	0.259

Our proposed single and multi-objective approaches result in the following power law rules:

$$\text{(UAI-SIP):} \quad V^{1.00000} S^{0.99999} = 400.604, \tag{11}$$

$$\text{(UAI-MIP):} \quad V^{1.00000} S^{0.99999} = 400.596. \tag{12}$$

UAI-SIP solution has $\mathcal{T}$ = 0.35898 and Significance = 90.3% and is close to the best trade-off solution obtained by UAI-MIP.

D. Differences with Existing Automated Innovization framework

The key differences of the proposed innovization framework with respect to the existing innovization framework are presented below.

1) *Relevance of Clustering-Parameter*: Two candidate-rules can be compared with each other on the basis of their $\mathcal{T}$-values, i.e., the rule with lesser $\mathcal{T}$ will have lesser intra-cluster variation. On the other hand, there is no reference measure for no. of divisions d in EAI.
2) *Contribution of* c_v: As explained in Section III-B, the nature of $\Sigma_{i=1}^{\mathcal{C}} c_v^{(i)}$ is uncertain, i.e., it is difficult to predict whether is supports or have a conflict with $\mathcal{C}$. Whereas, $\overline{c_v}$ shows a pure conflict with $\mathcal{C}$, thus reducing the uncertainty in prediction.
3) *Contribution of* $\mathcal{U}$: Minimization of unclustered points ($\mathcal{U}$) has been removed, eliminating the dominance of the maximum significance rule at the cost of accuracy. Though, desired significance has been ensured by adding a separate constraint.
4) *Multi-Objective*: UAI-MIP provides a balanced objective function formulation to explore the search space better, but at a higher computational cost than SIP.
5) *Application*: The proposed formulation can work with data-sets of both continuous and discrete nature while two separate frameworks EAI and EAID have been proposed earlier in literature.

IV. Results

In this section, we present and discuss the results from the proposed framework on several benchmark problems with both continuous and discrete variables and with single- and multi-cluster rules. These results are then compared with EAI framework. The problem formulation for the benchmark optimization problems can be found in [4]. For all results produced here, the significance threshold (τ_s) is set to 80% (which completes the constraint function) and the input PO dataset size m to 1,000. For innovization-GAs (UAI-SIP and UAI-MIP) and for all problems, a population of size 100, SBX operator with probability of 0.9 and distribution index of 10, polynomial mutation with probability of 0.05 and distribution index of 50 are used. However, in case of UAI-MIP, the distribution indices are changed from 10 and 50 to 15 and 30, respectively.

A. Single-objective (UAI-SIP) Results

Table III presents results of UAI-SIP on three problems. Best performing results are emboldened. The number of clusters ($\mathcal{C}$) are mentioned in brackets with the c-values. In cases with $\mathcal{C} \geq 2$, the c-value of the largest cluster is mentioned. However, overall significance is reported in the last column. Theoretically, the rules in TRUSS (continuous and discrete) and BEAM problems have $\mathcal{C} = 1$, while the SPRING problem has $\mathcal{C} = 7$. For more rules, results of UAI-SIP is better.

For the discrete-TRUSS problem, the input PO solutions have been obtained by taking 1,000 discrete values between each variable bound. The innovized results are presented in Table IV. It can be observed that the rules obtained have converged well near the values obtained from the continuous version of the problem. Also, UAI-SIP can evaluate the significance of the rule which EAID could not.

TABLE III
Result comparison between UAI-SIP (first row) & EAI (second row) for different rules mentioned in column-2. Better results ($\mathcal{C}$ and b_j's convergence) are marked in bold.

Problem	Basis $\mathbf{f}_n$	Coefficient b_j		c-value ($\mathcal{C}$)	Sig.
		ϕ_1	ϕ_2		
TRUSS	(S,V)	**1.00000**	**0.99999**	**400.60 (1)**	**90.3%**
		1.00000	0.99988	400.77 (3)	91.8%
	(x_1,x_2)	**-0.99999**	**1.00000**	**2.0035 (1)**	**89.1%**
		-0.99869	1.00000	1.9838 (1)	86.5%
	(x_1,V)	**1.00000**	**-0.99827**	**0.1128 (1)**	**87.8%**
		1.00000	-0.99738	0.1105 (3)	87.0%
	(x_2,V)	**1.00000**	**-0.99531**	**0.2296 (1)**	**89.4%**
		1.00000	**-0.99995**	**0.2236 (7)**	**88.0%**
BEAM	(b, D)	**1.00000**	**0.99998**	**0.002195 (1)**	**89.1%**
		1.00000	0.99987	0.002197 (1)	94.6%
	(C, D)	**1.00000**	**0.93464**	**0.02457 (1)**	**92.8%**
		1.00000	0.90924	0.02179 (16)	78.6%
	(D, P_c)	**1.00000**	**0.33337**	**0.17011 (1)**	**89.3%**
		1.00000	**0.33334**	**0.16993 (1)**	**94.6%**
	(D, σ)	1.00000	-0.99994	4.3e-07 (1)	89.5%
		1.00000	**-0.99999**	**3.9e-07 (1)**	**94.6%**
SPRING	(D, N)	**1.00000**	**0.33333**	**4.54 (7)**	**100%**
		1.00000	**0.33333**	**4.54 (7)**	**100%**

TABLE IV
Innovization results on Discrete-TRUSS problem (Single-objective Analysis).

Rule	Coefficient b_j		c-Value ($\mathcal{C}$)	Significance
	ϕ_1	ϕ_2		
(S, V)	0.99992	1.00000	400.127 (1)	89.1%
(x_1, x_2)	-0.99272	1.00000	2.062 (3)	85.9%
(x_1, V)	1.00000	-0.99369	0.1128 (2)	87.8%
(x_2, V)	1.00000	-0.99998	0.2231 (2)	90.7%

It can be observed that UAI-SIP is able to converge equivalent or better to EAI framework in most of the cases, and can be used for design tasks involving discrete variables as well. However, for discrete-TRUSS problems, multiple rules are found in some cases, but each conforming to the correct b_j(s) and c-values.

Apart from the obtained rules, the convergence of the UAI-SIP framework was observed with respect to the EAI for the same GA settings. Since it is a single-objective formulation, the average distance of the population from their respective ideal solution is recorded generation-wise, and are shown in Figures 3 and 4 for rules (x_1,x_2) and (x_1,V), respectively, for the TRUSS problem. These convergence results are median-values of 21 independent runs. It is interesting to note that EAI converges faster in the initial generations but then, UAI-SIP takes over and maintains a better performance till the end, which is coherent with the results shown in Table III.

B. Multi-objective (UAI-MIP) Results

The results for the proposed UAI-MIP framework (at the end of 200 generations) are shown in Table V, which are the

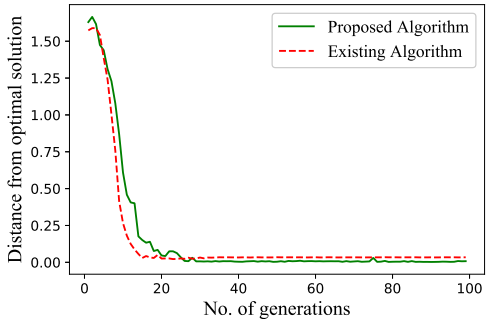

Fig. 3. Convergence of (x_1,x_2) rule in TRUSS problem.

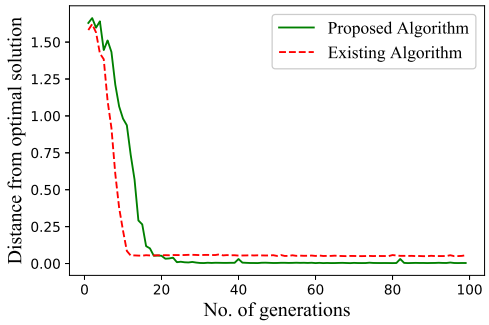

Fig. 4. Convergence of (x_1,V) rule in TRUSS problem.

minimum-cluster solutions from the results achieved in each case.

TABLE V
RESULT COMPARISON BETWEEN UAI-MIP (FIRST ROW) & EAI (SECOND ROW) FOR DIFFERENT RULES MENTIONED IN COLUMN-2. BETTER RESULTS ($\mathcal{C}$ AND b_j'S CONVERGENCE) ARE MARKED IN BOLD.

Problem	**Basis** f_n	**Coefficient** b_j ϕ_1	ϕ_2	c**-value** ($\mathcal{C}$)	**Sig.**
TRUSS	(S,V)	**1.00000**	**0.99999**	**400.59 (1)**	**90.3%**
		1.00000	0.99988	400.77 (3)	91.8%
	(x_1,x_2)	-0.99204	1.00000	2.0999 (1)	91.5%
		-0.99869	**1.00000**	**1.9838 (1)**	**86.5%**
	(x_1,V)	**1.00000**	**-0.99642**	**0.1097 (1)**	**89.3%**
		1.00000	-0.99738	0.1105 (3)	87.0%
	(x_2,V)	**1.00000**	**-0.99531**	**0.2296 (1)**	**89.4%**
		1.00000	-0.99995	0.2236 (7)	88.0%
BEAM	(b, D)	**1.00000**	**0.99998**	**0.002195 (1)**	**89.1%**
		1.00000	0.99987	0.002197 (1)	94.6%
	(C, D)	**1.00000**	**0.93464**	**0.02457 (1)**	**92.4%**
		1.00000	0.90924	0.02179 (16)	78.6%
	(D, P_c)	**1.00000**	**0.33333**	**0.17004 (1)**	**89.1%**
		1.00000	0.33334	0.16993 (1)	94.6%
	(D, σ)	**1.00000**	**-0.99999**	**4.3e-07 (1)**	**89.2%**
		1.00000	**-0.99999**	**3.9e-07 (1)**	**94.6%**
SPRING	(D, N)	**1.00000**	**0.33324**	**4.54 (7)**	**100%**
		1.00000	**0.33333**	**4.54 (7)**	**100%**

For discrete-TRUSS problem, the minimum-cluster innovized rules are given in Table VI. It can be seen that all b_j's have almost converged to their theoretical values and also, these rules are reported with more significance than in UAI-SIP, which reflects better search efficiency of the multi-objective formulation. The convergence analysis can not be performed since EAI has a single-objective formulation. However, the Pareto-fronts obtained from UAI-MIP are shown in Figures 5 and 6 for two test-cases.

TABLE VI
INNOVIZATION RESULTS ON DISCRETE-TRUSS PROBLEM (MULTI-OBJECTIVE ANALYSIS).

Rule	**Coefficient** b_j ϕ_1	ϕ_2	c**-Value** ($\mathcal{C}$)	**Significance**
(S, V)	0.99995	1.00000	400.550 (1)	91.0%
(x_1, x_2)	-0.99866	1.00000	2.028 (1)	91.0%
(x_1, V)	1.00000	-0.99988	0.1108 (1)	91.3%
(x_2, V)	1.00000	-0.99836	0.2215 (1)	89.5%

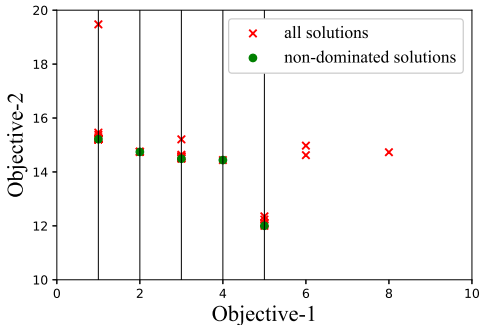

Fig. 5. Obtained front from UAI-MIP on TRUSS (x_1,x_2) rule

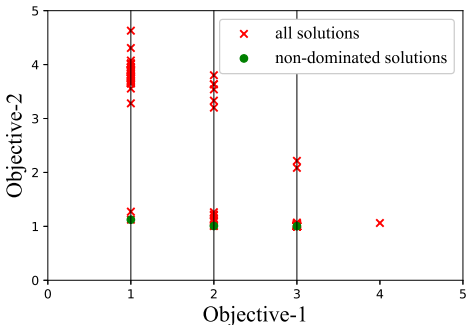

Fig. 6. Obtained front from UAI-MIP on BEAM (b,D) rule

It can be observed from Tables V and VI that minimum-cluster rules have an identical number of clusters as the theoretically-known relationships for these problems. However, we have proposed a trade-off evaluation procedure in Section III-C to automate the choice of one solution from several obtained solutions automatically. These best-trade-off solutions, which might be different from the presented min-cluster solutions, are given in Table VII for some cases. These solutions represent the best trade-off between achieving the minimum possible clusters and achieving the best accuracy.

It is known that the rules with lesser $\mathcal{C}$ are more physically explainable whereas the multi-objective formulation ensures that the solutions with more clusters have better representation, i.e., they can explain the input PO dataset better. The trade-off analysis in Table VII points out a few cases where the

TABLE VII
AUTOMATED TRADE-OFF ANALYSIS FOR SOME CASES OBTAINED FROM UAI-MIP. BEST TRADE-OFF SOLUTION IS HIGHLIGHTED.

Case	Point	Obj-1	Obj-2	$R(\mathbf{x}^{(i)}, \mathbf{x}^{(i+1)})$	$R(\mathbf{x}^{(i+1)}, \mathbf{x}^{(i)})$	Average
1. TRUSS DISCRETE ($\mathbf{x}_1$,V)	$\mathbf{x}^{(1)}$	1	15.964	7.784	-	7.784
	$\mathbf{x}^{(2)}$	**2**	**15.585**	**21.299**	**0.128**	**10.714**
	$\mathbf{x}^{(3)}$	4	15.308	0.384	0.047	0.216
	$\mathbf{x}^{(4)}$	5	7.639	11.238	2.599	6.919
	$\mathbf{x}^{(5)}$	7	7.114	-	0.089	0.089
2. TRUSS (x_1,x_2)	$\mathbf{x}^{(1)}$	**1**	**20.729**	**3.710**	-	**3.710**
	$\mathbf{x}^{(2)}$	2	19.704	3.857	0.269	2.063
	$\mathbf{x}^{(3)}$	3	18.718	4.560	0.259	2.409
	$\mathbf{x}^{(4)}$	4	17.884	3.202	0.219	1.710
	$\mathbf{x}^{(5)}$	5	16.696	1.031	0.312	0.672
	$\mathbf{x}^{(6)}$	7	9.318	-	0.969	0.969
3. BEAM (b,D)	$\mathbf{x}^{(1)}$	1	1.123	1.034	-	1.034
	$\mathbf{x}^{(2)}$	**2**	**1.007**	**30**	**0.966**	**15.483**
	$\mathbf{x}^{(3)}$	3	1.003	-	0.033	0.033
4. SPRING (D,N)	$\mathbf{x}^{(1)}$	**7**	**1.0017**	**2.077**	-	**2.077**
	$\mathbf{x}^{(2)}$	65	1.0001	0.633	0.481	0.557
	$\mathbf{x}^{(3)}$	66	1.0000	-	1.579	1.579

designer should consider a reasonable choice between it's physical significance and it's accuracy. In case-1, it is evident that the best-trade-off solution has the same number of clusters as was identified by SIP, while in case-3, the best-trade-off comes with $\mathcal{C} = 2$ which both SIP and EAI failed to produce. Moreover, the MIP procedure also provided the theoretical $\mathcal{C} = 1$ solution which is an additional advantage.

C. Discussions

It is clear that the UAI framework works well for both continuous and/or discrete datasets. In UAI-SIP results (Table III), there is slight deviation in (D,σ) rule, whereas considerable improvement can be seen in (C,D) rule taking the significance from 78.6% to 92.8% in EAI. In UAI-MIP results (Table V), our framework performed worse in (x_1,x_2) rule but there is an increase in the significance value from 86.5% to 91.5% which was further verified using manual innovization. This generates more confidence as even though the coefficients did not converge well to their theoretical value, the rule represents the PO solutions better.

In discrete variable problems, we have a clear advantage over EAID since our framework can evaluate the true significance of the innovized rule. In addition, it is also evident from Tables VI and VII that UAI-MIP can yield multiple solutions including both (i) the theoretical solution ($\mathcal{C} = 1$, which both SIP and EAI failed to give) and (ii) the best-trade-off solution which is explainable with reasonably good accuracy.

V. CONCLUSIONS AND FUTURE STUDIES

In this paper, a Unified Automated Innovization framework has been proposed with an ability to deal with both continuous and discrete variable problems to identify single- or multi-cluster rules. As an improved procedure, the new clustering procedure has replaced the old clustering-parameters (d and ϵ), with a new, more intuitive and better explainable cluster-threshold parameter $\mathcal{T}$. The paper has demonstrated that both versions of the proposed framework (UAI-SIP and UAI-MIP) can be applied in design tasks involving continuous and/or discrete variables. Also, the multi-objective approach with an automated trade-off analysis has been able to find a single preferred solution having the best weighted trade-off. The choice with the designer is whether to choose UAI-SIP and associated lower computational expense, or choose UAI-MIP and associated additional knowledge about the trade-off explored. This trade-off between the number of clusters and the accuracy of the rule is another dimension of interest to the designers. Although more computationally expensive, our results clearly shows advantages of using UAI-MIP approach.

Recent works have reported that innovization is a key to improve our knowledge and understanding of design optimization tasks by extracting hidden relationships among PO solutions. This paper proposed a robust framework to realize this goal of innovization. In future, this single- and multi-objective analysis with the improved clustering process can be extended to other innovization procedures like *temporal innovization*, *higher-* and *lower-level innovization*, or for *learning free-form rules* using Geometric Programming.

ACKNOWLEDGEMENT

Authors would like to acknowledge the support provided by the Government of India under SPARC project No. P-66.

REFERENCES

[1] H. Taboada and D. Coit, "Data mining techniques to facilitate the analysis of the Pareto-optimal set for multiple objective problems," in *Proceedings of the 2006 Industrial Engineering Research Conference (CD-ROM)*, 2006.

[2] S. Obayashi, S. Jeong, and K. Chiba, "Multi-objective design exploration for aerodynamic configurations," *AIAA Paper*, vol. 4666, p. 2005, 2005.

[3] D. J. Walker, R. M. Everson, and J. E. Fieldsend, "Visualisation and ordering of many-objective populations," in *2010 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, 2010, pp. 1–8.

[4] S. Bandaru and K. Deb, "Towards automating the discovery of certain innovative design principles through a clustering based optimization technique," *Engineering optimization*, vol. 43, no. 9, pp. 911–941, 2011.

[5] A. Gaur and K. Deb, "Effect of size and order of variables in rules for multi-objective repair-based innovization procedure," in *Proceedings of Congress on Evolutionary Computation (CEC-2017) Conference*. Piscatway, NJ: IEEE Press, 2017.

[6] K. Deb, "Unveiling innovative design principles by means of multiple conflicting objectives," *Engineering Optimization*, vol. 35, no. 5, pp. 445–470, 2003.

[7] K. Deb and A. Srinivasan, "Innovization: Innovating design principles through optimization." in *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2006)*, New York: ACM, 2006, pp. 1629–1636.

[8] S. Bandaru, A. H. C. Ng, and K. Deb, "Data mining methods for knowledge discovery in multi-objective optimization: Part a – survey," *Expert Systems With Applications*, vol. 70, pp. 139–159, 2017.

[9] ——, "Data mining methods for knowledge discovery in multi-objective optimization: Part b – new developments and applications," *Expert Systems With Applications*, vol. 70, pp. 119–138, 2017.

[10] S. Bandaru and K. Deb, "Automated innovization for simultaneous discovery of multiple rules in bi-objective problems," in *Proceedings of Sixth International Conference on Evolutionary Multi-Criterion Optimization (EMO-2011)*. Heidelberg: Springer, 2011, pp. 1–15.

[11] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II," *IEEE transactions on evolutionary computation*, vol. 6, no. 2, pp. 182–197, 2002.

[12] L. Rachmawati and D. Srinivasan, "Multiobjective evolutionary algorithm with controllable focus on the knees of the Pareto front," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 4, pp. 810–824, 2009.