



# Generalized higher-level automated innovization with application to inventory management



Sunith Bandaru<sup>a,\*</sup>, Tehseen Aslam<sup>a</sup>, Amos H.C. Ng<sup>a</sup>, Kalyanmoy Deb<sup>b</sup>

<sup>a</sup> Virtual Systems Research Centre, University of Skövde, Skövde 54128, Sweden

<sup>b</sup> Department of Electrical and Computer Engineering, Michigan State University, East Lansing, 428 S. Shaw Lane, 2120 EB, MI 48824, USA

## ARTICLE INFO

### Article history:

Received 31 October 2013

Accepted 7 November 2014

Available online 20 November 2014

### Keywords:

Automated innovization

Higher-level innovization

Genetic programming

Inventory management

Knowledge discovery

## ABSTRACT

This paper generalizes the automated innovization framework using genetic programming in the context of higher-level innovization. Automated innovization is an unsupervised machine learning technique that can automatically extract significant mathematical relationships from Pareto-optimal solution sets. These resulting relationships describe the conditions for Pareto-optimality for the multi-objective problem under consideration and can be used by scientists and practitioners as thumb rules to understand the problem better and to innovate new problem solving techniques; hence the name innovization (*innovation through optimization*). Higher-level innovization involves performing automated innovization on multiple Pareto-optimal solution sets obtained by varying one or more problem parameters. The automated innovization framework was recently updated using genetic programming. We extend this generalization to perform higher-level automated innovization and demonstrate the methodology on a standard two-bar bi-objective truss design problem. The procedure is then applied to a classic case of inventory management with multi-objective optimization performed at both system and process levels. The applicability of automated innovization to this area should motivate its use in other avenues of operational research.

© 2014 Elsevier B.V. All rights reserved.

## 1. Introduction

Solution analysis is an important part of post-optimality studies. In the context of optimization formulations involving multiple conflicting objectives, the presence of more than one solution, collectively referred to as Pareto-optimal solutions, necessitates the use of specialized tools for analyzing multiple data points. Patterns discovered in the Pareto-optimal solution sets can throw light on hidden problem structure and reveal interesting design features which in turn can serve as thumb rules in future. The search for such patterns takes multi-objective post-optimality studies into the realm of data-mining and machine learning. Over the years many techniques have been proposed for the analysis of Pareto-optimal solutions. They range from visual data-mining methods like parallel-coordinate plots (Inselberg, 2009), scatter plots, heatmap visualization (Pryke, Mostaghim, & Nazemi, 2007) and self-organizing maps (Obayashi & Sasaki, 2003) to non-visual methods like decision tree learning (Sugimura, Obayashi, & Jeong, 2010), association rule-mining (Sugimura, Obayashi, & Jeong, 2009), dimensionality reduction (Oyama, Nonomura, & Fujii, 2010)

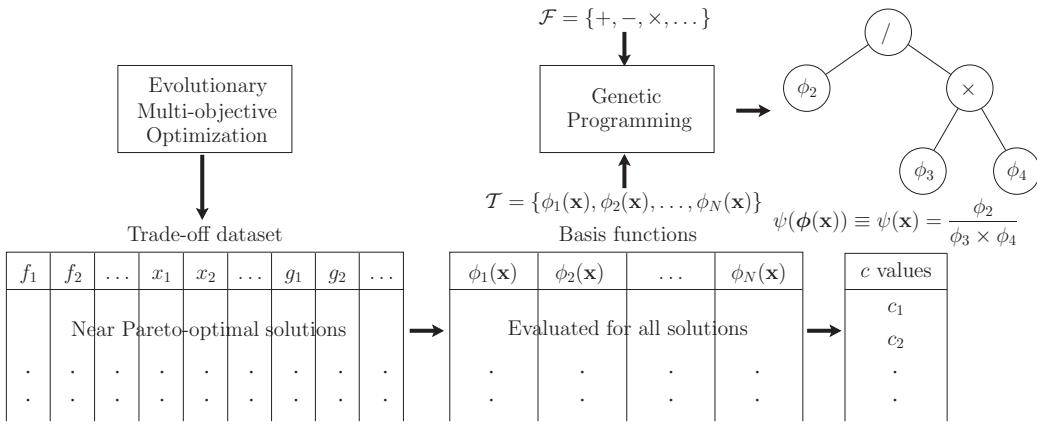
and automated innovization (Bandaru, 2013). A survey of these methods can be found in Bandaru (2013) and Jeong and Shimoyama (2011).

While most techniques proposed in the literature rely on visual assessment of data, thus leading to qualitative interpretation and mostly subjective deductions, non-visual methods result in crisp or explicit knowledge. Of particular interest to this paper is automated innovization (Bandaru & Deb, 2011b) where the knowledge is in the form of mathematical relationships between the variables and/or objectives of the optimization problem. On the other hand, decision trees and association rule-mining deduce knowledge in the form of 'if-then' conditions, which may also be desirable in some cases (Ng, Deb, & Dudas, 2009).

In the following sections, we give an overview of the generalized automated innovization framework developed in Bandaru and Deb (2013a). It uses a dimensionally-aware genetic programming architecture to generate and evolve physically valid mathematical expressions that describe various relationships between variable and/or objective function values in the Pareto-optimal dataset. In Section 2 we introduce the concept of higher-level innovization and discuss the merits of such a study, following up with an illustrative example in Section 3. The genetic programming architecture is extended to higher-level innovization in Section 2.1. For application to the field of manufacturing system engineering, we describe a standard supply chain model in Section 4 and give its multi-objective problem

\* Corresponding author. Tel.: +46500448506.

E-mail addresses: [sunith.bandaru@his.se](mailto:sunith.bandaru@his.se), [sunith.bandaru@gmail.com](mailto:sunith.bandaru@gmail.com)  
(S. Bandaru), [tehseen.aslam@his.se](mailto:tehseen.aslam@his.se) (T. Aslam), [amos.ng@his.se](mailto:amos.ng@his.se) (A. H.C. Ng), [kdeb@egr.msu.edu](mailto:kdeb@egr.msu.edu) (K. Deb).



**Fig. 1.** Evaluation of binary trees in automated innovation.

formulation. To illustrate the broad range of optimization problems where innovation can be performed, we also introduce a manufacturing process formulation in Section 5 which can, in practice, be a sub-system (manufacturing process) of the supply chain. The results of higher-level automated innovation are discussed in Sections 4.3 and 5.1 for both problems, respectively.

### 1.1. Automated innovation

Originally, innovation was proposed to be semi-visual technique (Deb & Srinivasan, 2006). Starting with a well-optimized dataset containing Pareto-optimal or near-Pareto-optimal solutions of a multi-objective problem, the user chooses different combinations of variables, constraint functions and objective functions (together called ‘problem entities’ in this paper) and plots them against each other, visually looking for correlations in the dataset. If found, a regression analysis is performed on the correlated entities. The resulting regression function then describes the mathematical relationship between the entities which causes a feasible point to be Pareto-optimal or near-Pareto-optimal. The procedure as such has been applied to a number of design problems (Deb et al., 2009; Deb & Jain, 2003; Deb & Sindhy, 2008). The process however, is cumbersome and comes with all the shortcomings of a typical manual data analysis approach.

Automated innovation is an unsupervised machine learning algorithm that uses a grid-based clustering algorithm to automatically detect the existence of correlations (Bandaru & Deb, 2011b). It can also process several different entity combinations at a time so that all relationships hidden in the dataset can be discovered simultaneously (Bandaru & Deb, 2011a). Theoretically, correlations involving more than three entities can also be discovered, a task that is otherwise difficult for a human. Initial studies in automated innovation use a restricted mathematical structure for the relationships. Recently however, the structure was generalized using parse trees for representing possible relationships (Bandaru & Deb, 2013a). Here, we present an overview of the same.

### 1.2. Genetic programming based automated innovation

In automated innovation, the relationships are composed of basis functions. The variables ( $\mathbf{x}$ ), constraint ( $\mathbf{g}(\mathbf{x})$ ) and objective functions ( $\mathbf{f}(\mathbf{x})$ ) are the default basis functions. Additional basis functions may also be defined as combinations of these default functions. For a given set of  $m$  trade-off solutions, each of the  $N$  basis functions ( $\phi(\mathbf{x})$ 's) can be evaluated as shown in Fig. 1. In genetic programming (GP) mathematical expressions are usually represented using parse trees. By choosing the basis functions from the terminal set  $\mathcal{T} = \{\phi_1, \phi_2, \dots, \phi_N\}$  and mathematical operators from the function

set  $\mathcal{F} = \{+, -, \times, /, \dots\}$ , generic mathematical expressions can be generated in the form of binary trees. Given a binary tree, its equivalent mathematical expression can be obtained by traversing the tree in a predefined manner. Inorder tree traversal is chosen for this purpose. It is defined as follows:

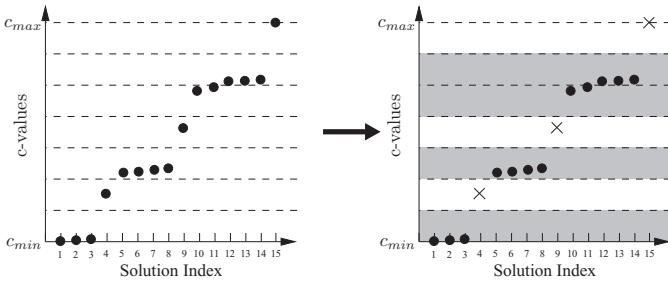
```
inorder(node)
  if node == null then return
  inorder(node.left)
  visit(node)
  inorder(node.right)
```

Fig. 1 shows an example of a binary tree and its equivalent mathematical expression  $\psi(\phi(\mathbf{x}))$  or simply  $\psi(\mathbf{x})$ . When evaluated for each of the  $m$  trade-off solutions, it gives a numeric value, say  $c$ . The distribution of the  $m$   $c$ -values thus calculated determines whether  $\psi(\mathbf{x})$  is a significant relationship hidden in the trade-off dataset under consideration.

**Definition.** An expression  $\psi(\mathbf{x})$  is said to be applicable over the trade-off dataset with a significance  $S$ , if  $S$  percent of the  $m$  trade-off solutions, evaluate approximately to the same  $c$  value.

The reason for defining  $\psi(\mathbf{x})$  as above is that the terms in any arbitrary relationship can always be rearranged to have a constant on the right hand side, i.e.  $\psi(\mathbf{x}) = c$ . However, numerical computation methods prevent the exact same  $c$  value for different solutions and therefore, the requirement on  $\psi(\mathbf{x})$  is relaxed to have approximately equal values on the right hand side.

Integrating GP with automated innovation allows a randomly generated population of binary trees to be evolved through selection, crossover and mutation operators and eventually converge to one or more significant relationships. As mentioned above, the evaluation of the fitness of binary trees requires assessing their corresponding  $c$ -value distributions. For this purpose, the  $m$   $c$ -values obtained for each binary tree are grouped using the grid-based clustering algorithm into  $C$  clusters. Grid-based clustering first divides the range of  $c$ -values  $[c_{\min}, c_{\max}]$  into  $d$  divisions, a parameter of the algorithm. A clustering criterion defines the minimum number of  $c$ -values required in a division for it to become a *sub-cluster*. Thereafter, adjacent sub-clusters are combined to form clusters. Unlike most clustering algorithms, grid-based clustering allows some data points to remain as outliers. The trade-off solutions corresponding to the outlying  $c$ -values are labeled as *unclustered* and their number is denoted by  $U$ . Fig. 2 shows  $m = 15$   $c$ -values clustered using  $d = 7$  divisions. The criterion used is  $P_t \geq \lfloor \frac{m}{d} \rfloor$ , where  $P_t$  is the number of  $c$ -values in division  $t$ . Divisions 1, 3, 5 and 6 form sub-clusters according to this criterion. Merging adjacent sub-clusters gives three clusters (shown in gray in Fig. 2), namely, {1, 2, 3}, {5, 6, 7, 8} and {10, 11, 12, 13, 14}. Points 4,



**Fig. 2.** Grid-based clustering.

9 and 15 remain unclustered, i.e.  $\mathcal{U} = 3$ . Though,  $d$  is usually a user-specified parameter, in automated innovation the task of setting it is overcome by considering it as a variable along with the GP tree. As the number of clusters cannot be more than the number of data points, we have  $1 \leq d \leq m$ .

The objective function  $O$  in automated innovation (Bandaru & Deb, 2011a, 2011b) is the weighted summation of the number of clusters and the percentage coefficient of variation,  $c_v$ , within those clusters.

$$\text{Minimize } O = C + \sum_{k=1}^C c_v^{(k)} \times 100 \text{ (in percent)},$$

$$c_v^{(k)} = \frac{\sigma_c^{(k)}}{\mu_c^{(k)}} \quad \forall c \in k\text{th cluster} \quad (1)$$

Within each cluster  $k$ , the coefficient of variation  $c_v$  of the  $c$  values is a function of the GP expression  $\psi(\mathbf{x})$ . The minimization of the number of clusters is always desirable, while the minimization of  $c_v$  causes the  $c$ -values to be as close as possible, thus giving a narrow distribution of  $c$ -values, thereby making them *approximately* equal. The constraint  $\mathcal{U} = 0$  is imposed to improve the accuracy of the obtained relationships (Bandaru & Deb, 2011b). It forces the number of divisions to increase to a level so that unclustered points become individual clusters. The significance  $S$  of a binary tree is defined as the percentage of  $c$ -values that remain clustered even with a stricter clustering criterion of  $P_t \geq \lfloor \frac{m}{d} \rfloor + \epsilon$ , where  $\epsilon$  is a small integer value (Bandaru & Deb, 2011b).

In Bandaru and Deb (2013a), the authors developed a dimensionally-aware GP. Its purpose is to generate and maintain not only mathematically, but also physically valid trees. The relationships generated by automated innovation are expected to give a deeper understanding of a physical problem. It is therefore important that they are meaningful to the end user. To this end, constraints are imposed in the algorithm for ensuring dimensional consistency in the generated expressions. The dimensions of each basis function are provided *a priori* to the algorithm. When evaluating a binary tree from the population, the exponents  $E$  of all fundamental dimensions (mass, length, time, etc.) are monitored. If *incommensurable* basis functions (basis functions with different dimensions) are added or subtracted, the binary tree (or population member) is assigned a large constraint violation  $E_{\max}$ , so that its structure is deemphasized in the following generations. For discussion on the logic behind setting of  $E_{\max}$  value in the commensurability constraint, refer to Bandaru and Deb (2013a).

The GP-based automated innovation problem can thus be formulated as (Bandaru, 2013),

$$\begin{aligned} \text{Minimize } & TS \times O \\ \text{Subject to } & 1 \leq d \leq m, \\ & \mathcal{U} = 0, \\ & S \geq S_{\text{reqd}}, \\ & |E| \leq E_{\max} \quad \forall \text{ fundamental dimensions}, \\ \text{Variables: } & \psi(\mathbf{x}) \text{ is a GP expression and } d \text{ is an integer.} \end{aligned} \quad (2)$$

Within the GP framework, the tree size  $TS$  is multiplied to the original objective function  $O$  in Eq. (1) to promote smaller trees and hence easily interpretable relationships. The constraint on the significance  $S$  simply says that only relationships above a threshold significance  $S_{\text{reqd}}$  are obtained.

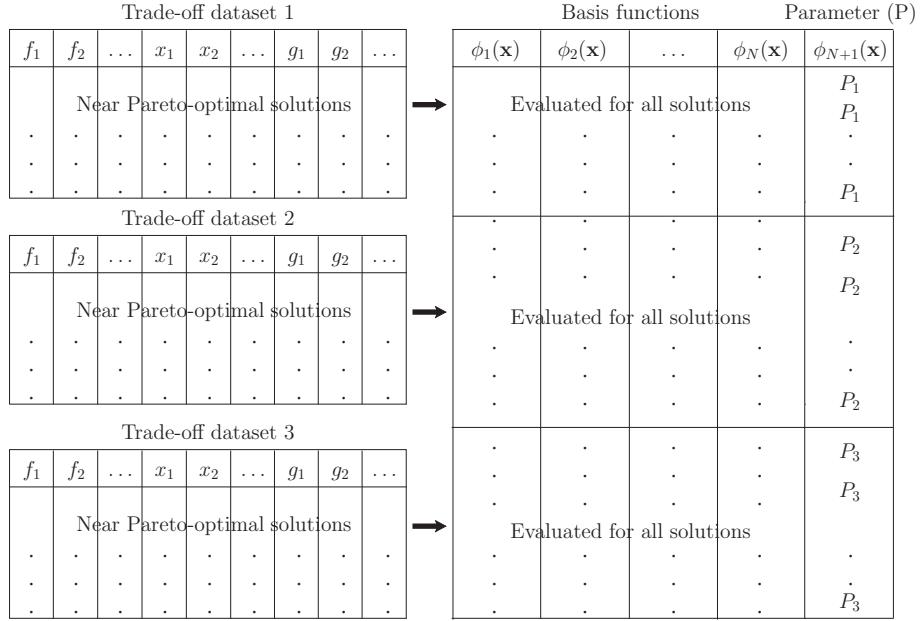
## 2. Higher-level innovation

Higher-level innovation extends the concept of innovation to multiple Pareto-optimal datasets. Multi-objective optimization problems usually involve problem parameters that are not changed during the optimization process, but are expected to vary in the real world. Each change in such parameters necessitates a fresh optimization run. Generally, this leads to a shift of the Pareto-optimal front. In higher-level innovation the task is to identify relationships that are applicable over all such fronts. More interesting are relationships which involve the parameter being varied, because they explain design/system/process behavior to changes that are external to the problem itself. The Pareto-optimal front may also change due to the addition/deletion/modification of one or more constraints or by the inclusion/fixing of a variable (Bandaru & Deb, 2013b). In any case, a higher-level innovation study will reveal solution properties that are inherent to the underlying physics of design/system/process rather than to a particular setting. In the following section we consider higher-level automated innovation when a single problem parameter  $P$  is varied across multiple optimization runs.

### 2.1. Genetic programming based higher-level automated innovation

In order to obtain relationships varying with the parameter  $P$ , it is necessary to include it as the  $(N+1)$ th basis function as shown in Fig. 3. The parameter value is repeated for all solutions in the trade-off dataset for all the datasets. Fig. 3 shows this for three datasets. The presence of a constant-valued basis function may however lead to some redundant/artificial relationships. For example, if  $\psi_1(\mathbf{x})$  is a relationship which has high significance in all datasets, then  $\psi_2(\mathbf{x}) = \phi_{(N+1)}\psi_1(\mathbf{x})$  will also be trivially significant, because it is equivalent to multiplying the  $c$ -values of  $\psi_1(\mathbf{x})$  with  $P_1$  for the first dataset, with  $P_2$  for the second dataset and so on. If however, the presence of  $\phi_{(N+1)}$  in a relationship negates the effect of the change in the parameter  $P$  across the input datasets, then it is a potential higher-level relationship. We show the difference between a valid and an artificial relationship later in Section 3. The point to note here is that a relationship involving  $\phi_{(N+1)}$  can be a higher-level relationship only when its  $c$ -values from all input datasets are approximately equal. Higher-level automated innovation uses a metric called *Cluster Overlap Index (COI)* to differentiate between possible higher-level relationships and the artificial ones discussed above. *COI* of a candidate relationship is defined as the number of adjacent (with respect to sorted  $c$ -values) trade-off dataset pairs for which clusters of one dataset, together completely enclose the largest cluster of the other. A candidate relationship with  $COI = D - 1$ , where  $D$  is the number of input trade-off fronts, indicates that the clusters (and hence the  $c$ -values) in all input datasets are close enough for that relationship to be a likely candidate for a higher-level relationship.

In the following discussion, for simplicity of notation, let the datasets be numbered from 1 to  $D$  in increasing order of the cluster average  $c$ -value of the largest cluster,  $\mu_L$ . For assigning fitness to candidate relationships (that satisfy  $COI = D - 1$ ) on the basis of closeness of clusters, the original weighted objective function  $O$  in Eq. (1) is *magnified* for each dataset by the greater ratio between  $\mu_L$ 's from all subsequent adjacent dataset pairs. In mathematical terms, for the  $p$ th dataset, the magnified weighted objective



**Fig. 3.** Higher-level innovation with a parameter  $P$  varied to obtain three trade-off datasets from three different optimization runs.

function is,

$$\prod_{q=p}^{D-1} \frac{\mu_L^{(q+1)}}{\mu_L^{(q)}} O^{(p)} \quad \text{or, when simplified, } \mu_L^{(D)} \frac{O^{(p)}}{\mu_L^{(p)}}. \quad (3)$$

This magnification of the objective function causes potential higher-level relationships to improve in accuracy by bringing closer the largest clusters from different datasets. The term  $O^{(p)}$  is different for each dataset and can be inferred from Eq. (1) to be,

$$O^{(p)} = C^{(p)} + \sum_{k=1}^{C^{(p)}} c_v^{(k)} \times 100 \text{ (in percent)},$$

$$c_v^{(k)} = \frac{\sigma_c^{(k)}}{\mu_c^{(k)}} \quad \forall c \in k\text{th cluster} \quad (4)$$

Grid-based clustering for the  $p$ th dataset uses the parameter  $d^{(p)}$ . Minimizing  $O^{(p)}$  for all datasets simultaneously calls for a multi-objective study possibly requiring a decision making step to select a single relationship. However, this defeats the whole purpose of automated innovation, where the goal is to simultaneously obtain all relationships with good significance in all input datasets. This requires a compromise between the above mentioned objectives. Noting that  $O^{(p)}$ 's for all datasets are of the same order of magnitude, the magnified objectives can be added together (Bandaru & Deb, 2013b) to form a new objective function for higher-level relationships,

$$\text{Minimize } F = \mu_L^{(D)} \sum_{p=1}^D \frac{O^{(p)}}{\mu_L^{(p)}}. \quad (5)$$

The calculation of  $COI$  and the magnification of the objective function as shown above are required only for binary trees that have the parameter ( $\phi_{(N+1)}$ ) in any of the terminal nodes. Other binary trees are evaluated by simply summing the objective functions in Eq. (4) over all datasets, for the same reason as discussed above. The objective function for higher-level automated innovation is thus split into two parts, one that is suited for binary trees involving  $\phi_{(N+1)}$  and the other for those that do not contain this basis function. Hence, the following optimization formulation will be used in this paper for

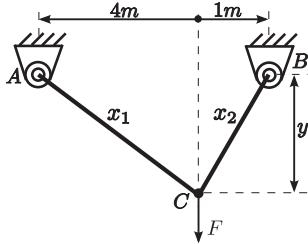
higher-level automated innovation,

$$\begin{aligned} \text{Minimize } & TS \times \left( aF + (1-a) \sum_{p=1}^D O^{(p)} \right) \\ \text{Subject to } & 1 \leq d^{(p)} \leq m^{(p)}, \quad \forall p = \{1, 2, \dots, D\}, \\ & \sum_p \mathcal{U}^{(p)} = 0, \\ & S_{\text{eff}} = \frac{\sum_p S^{(p)}}{D} \geq S_{\text{reqd}}, \\ & |E| \leq E_{\text{max}} \quad \forall \text{ fundamental dimensions}, \\ & a(COI - D + 1) = 0, \\ \text{Variables: } & \psi(\mathbf{x}) \text{ is a GP expression and } d^{(p)}\text{'s are integers.} \end{aligned} \quad (6)$$

The presence ( $a = 1$ ) or absence ( $a = 0$ ) of the parameter in the binary tree is used as a 'switch' in the objective function and the  $COI$  constraint. The last constraint imposes  $COI = D - 1$  for binary trees with  $a = 1$ . It is trivially satisfied when  $a = 0$ . As mentioned above, each dataset uses a different number of divisions ( $d^{(p)}$ ) since the number of trade-off solutions ( $m^{(p)}$ ) in them may differ. Grid-based clustering is separately performed on each dataset and the number of unclustered points,  $\mathcal{U}^{(p)}$ , from each are added to form a single constraint  $\sum_p \mathcal{U}^{(p)} = 0$ , as it is same as using  $\mathcal{U}^{(p)} = 0 \forall p$ . The effective significance ( $S_{\text{eff}}$ ) is defined as the average of significance values ( $S^{(p)}$ ) over all datasets. As in the case of automated innovation, the user specifies the required threshold significance  $S_{\text{reqd}}$ . Multiple higher-level design principles are obtained using the niching strategy proposed in Bandaru and Deb (2013a), which prevents competition between binary trees with different set of terminal nodes, thus allowing them to coexist in the population.

The optimization formulation in Eq. (6) is also valid when multiple parameters are varied across the optimization runs. The only change required is that  $a = 1$  when *any* of the varied parameters is present in the binary tree. However, practically speaking, a parameter is varied to isolate and study its effect on the Pareto-optimal solutions. Therefore, it is more prudent to vary one parameter at a time over multiple optimization runs. Interpreting relationships involving multiple parameters may also pose difficulties. All problems used in this paper involve the variation of only one of the parameters.

To summarize the higher-level automated innovation process, it starts with a multi-objective optimization problem which is solved using a numerical optimization method to generate a trade-off



**Fig. 4.** Standard two-bar truss configuration.

dataset. One of the external parameters is then changed and the problem is re-solved to generate more trade-off fronts as shown in Fig. 3. The combined dataset is used to extract higher-level relationships by solving the single-objective optimization problem shown in Eq. (6). The higher-level formulation in Eq. (6) is the result of combining various aspects of innovation from the authors' previous works. Readers are encouraged to refer to the following papers for more details on the topics mentioned against them.

1. Automated innovation (Bandaru & Deb, 2011a, 2011b): Grid-based clustering, coefficient of variation ( $c_v$ ), scaling of number of clusters  $C$  and  $c_v$ , constraint on number of unclustered solutions ( $\mathcal{U}$ ) and niching for multiple relationships.
2. Higher-level automated innovation (Bandaru & Deb, 2013b):  $COL$ , magnification of objective function, objective switching ( $a$ ) and effective significance ( $S_{\text{eff}}$ ).
3. Dimensionally-aware GP (Bandaru & Deb, 2013a): Binary tree representation of relationships, tree size ( $TS$ ) and exponents of fundamental dimensions ( $E$ ).

Next we consider a standard engineering design problem to theoretically explain the concepts of innovation and higher-level innovation and to illustrate the above procedure for extracting higher-level relationships. We also verify the proposed higher-level automated innovation technique by comparing the obtained results to known higher-level relationships for this problem.

### 3. Illustrative example: Truss design problem

The two-bar truss has the configuration shown in Fig. 4. The design requires that the total volume  $V$  (and hence the cost) of the truss structure be minimized along with the minimization of the maximum stress  $S$  induced in either of the bars. The truss is acted on by a vertical downward load of  $F = 100$  kilonewton. Geometrical constraints restrict the cross-sectional areas  $x_1$  and  $x_2$  of the bars and the height  $y$ . The induced stress should not exceed the elastic strength,  $10^5$  kilopascal, of the material used, which gives rise to a third constraint. The optimization formulation is given by,

$$\begin{aligned} \text{Minimize } & f_1(\mathbf{x}) = V = x_1\sqrt{16+y^2} + x_2\sqrt{1+y^2}, \\ \text{Minimize } & f_2(\mathbf{x}) = S = \max(\sigma_{AC}, \sigma_{BC}), \\ \text{Subject to } & \max(\sigma_{AC}, \sigma_{BC}) \leq 10^5 \text{ kilopascal}, \\ & 0 \leq x_1, x_2 \leq 0.01 \text{ square meter}, \\ & 1 \leq y \leq 3 \text{ meter}, \\ \text{Variables: } & x_1, x_2 \text{ and } y \text{ are continuous.} \end{aligned} \quad (7)$$

The Pareto-optimal curve for this problem can be derived analytically using the identical resource allocation strategy. Increasing the cross-sectional area of one bar reduces the stress induced in it and so the second objective takes the other bar into account at some point. But since both the objectives are equally important, a balance can be obtained only when the stresses in both the bars are equal. Thus,

$$\sigma_{AC} = \sigma_{BC} \Rightarrow \frac{20\sqrt{16+y^2}}{yx_1} = \frac{80\sqrt{1+y^2}}{yx_2}. \quad (8)$$

**Table 1**

Relationships or design principles (DP) obtained using GP-based automated innovation in Bandaru and Deb (2013a).

Notation	$\psi(\mathbf{x})$	Significance ( $S$ ) (in percent)	Fundamental dimensions		
			Mass	Length	Time
DP1	$y$	86.60	0.0	1.0	0.0
DP2	$SV$	87.00	1.0	2.0	-2.0
DP3	$Sx_1$	85.00	1.0	1.0	-2.0
DP4	$SVy$	87.00	1.0	3.0	-2.0
DP5	$(Vy)/x_2$	86.20	0.0	2.0	0.0
DP6	$(Vy)/x_1$	88.20	0.0	2.0	0.0
DP7	$V/x_1$	86.40	0.0	1.0	0.0
DP8	$V/(Sx_1x_2)$	87.20	-1.0	0.0	2.0
DP9	$V^2/(x_1x_2)$	87.40	0.0	2.0	0.0
DP10	$y/(Sx_1)$	88.00	-1.0	0.0	2.0
DP11	$x_2/x_1$	83.80	0.0	0.0	0.0
DP12	$(SVx_2y)/x_1$	88.00	1.0	3.0	-2.0
DP13	$V/x_2$	86.80	0.0	1.0	0.0
DP14	$(SV^2y)/x_1$	87.20	1.0	4.0	-2.0
DP15	$(x_2y)/x_1$	86.40	0.0	1.0	0.0
DP16	$x_2/(Sx_1^2)$	86.40	-1.0	-1.0	2.0
DP17	$V^2/(x_1x_2y)$	91.40	0.0	1.0	0.0
DP18	$(SV^2)/x_2$	87.20	1.0	3.0	-2.0
DP19	$Sx_2y$	87.00	1.0	2.0	-2.0
DP20	$(x_2y)/(Sx_1^2)$	86.80	-1.0	0.0	2.0

Similarly,

$$x_1\sqrt{16+y^2} = x_2\sqrt{1+y^2}. \quad (9)$$

Solving Eqs. (8) and (9) gives the following relationships applicable on the Pareto-optimal front,

$$y = 2, \quad x_2 = 2x_1, \quad V = 4\sqrt{5}x_1 = 2\sqrt{5}x_2, \quad SV = 400. \quad (10)$$

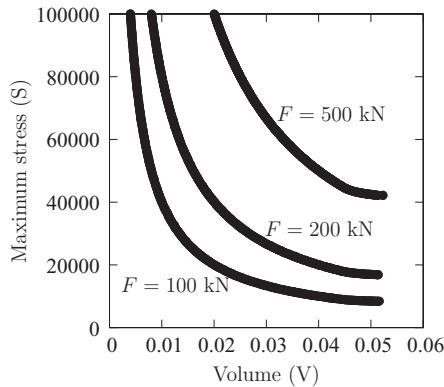
These relationships hold on the Pareto-optimal curve irrespective of the material and loading. They are thus the *design principles* of the two-bar truss structure. Designers can remember them as thumb rules when optimizing similar structures in future and easily handcraft an optimal solution. Of course the solution has to be checked for feasibility before actual implementation because the derivation of the design principles did not involve the use of constraints.

Solving Eq. (7) using NSGA-II (Deb, Agarwal, Pratap, & Meyarivan, 2002) gives  $m = 500$  non-dominated solutions at the end of 500 generations. For automated innovation  $\mathcal{F} = \{+, -, \times, \%, \wedge\}$  (where  $\%$  represents protected division and  $\wedge$  represents the power function) and  $\mathcal{T} = \{V, S, x_1, x_2, y, \mathcal{R}\}$  (where  $\mathcal{R}$  represents ephemeral constants) have been used in Bandaru and Deb (2013a) to obtain many relationships as shown in Table 1. All the relationships were shown to be combinations of the analytically obtained relationships in Eq. (10). For example,  $V = 4\sqrt{5}x_1$  and  $V = 2\sqrt{5}x_2$  are seen as DP7 and DP13, respectively. Multiplying them gives  $V^2 = 40x_1x_2$ , which is seen as DP9. The relationships do not have a significance of 100 percent, because part of the Pareto-optimal curve lies on a constraint boundary where the relationships no longer hold. This could not have been predicted analytically using Eq. (10). The last three columns show the exponents of fundamental dimensions of mass ( $M$ ), length ( $L$ ) and time ( $T$ ) calculated by the algorithm using the user-input dimensions of the terminal set, i.e.,  $V : M^0L^3T^0$ ,  $S : ML^{-1}T^{-2}$ ,  $x_1 : M^0L^2T^0$ ,  $x_2 : M^0L^2T^0$ ,  $y : M^0LT^0$  and  $\mathcal{R} : M^0L^0T^0$ .

### 3.1. Higher-level innovation in truss design

The loading parameter  $F$  was kept constant at 100 kilonewton in the above analysis. However, in practice it may vary and a fresh optimization will be required to determine the new trade-off front. The simple mathematical formulation of the truss design allows us to rework Eq. (8) for a general load  $F$  as follows:

$$\sigma_{AC} = \sigma_{BC} \Rightarrow \frac{F\sqrt{16+y^2}}{5yx_1} = \frac{4F\sqrt{1+y^2}}{5yx_2}. \quad (11)$$



**Fig. 5.** Pareto-optimal fronts of truss design problem for different loading conditions.

Solving Eqs. (11) and (9) gives the following higher-level relationships,

$$y = 2, \quad x_2 = 2x_1, \quad V = 4\sqrt{5}x_1 = 2\sqrt{5}x_2, \quad SV = 4.0F. \quad (12)$$

The first three relationships are independent of the term  $F$  and hence are applicable as such for any  $F$ . However, the fourth relationship is an important type of higher-level relationship because it varies with the parameter  $F$ . In other words, though  $SV$  changes with  $F$ , the relationship  $\frac{SV}{F}$  always has the same value of 4.0, because the change in  $SV$  negates the change in  $F$ . Note that  $SV$  in itself is also a valid higher-level relationship because it remains constant on the Pareto-optimal front for any given  $F$ . On the contrary,  $SVF$  is an artificial relationship, i.e. it remains constant for a given  $F$  for trivial reasons.

For higher-level automated innovation we use  $F = 200$  kilonewton and  $F = 500$  kilonewton to generate two more datasets using NSGA-II with a population size of 500. The Pareto-optimal fronts from all three datasets, containing 500 solutions each, are shown in Fig. 5. The optimization formulation proposed in (6) is now used to perform higher-level innovation automatically using these trade-off datasets as input. In this case, we have  $m^{(1)} = m^{(2)} = m^{(3)} = 500$ . We introduce  $F$  in the terminal set, i.e.  $\mathcal{T} = \{V, S, x_1, x_2, y, F, \mathcal{R}\}$ . Other parameters are as follows:

1. Population size = 500,
2. Number of generation = 100,
3. Discrete simulated binary crossover for all  $d^{(p)}$  with  $p_c = 0.9$  and  $\eta_c = 10$ ,
4. Discrete polynomial mutation for all  $d^{(p)}$  with  $p_m = 0.05$  and  $\eta_m = 50$ ,
5. Tree crossover probability = 0.9, mutation probability = 0.2,

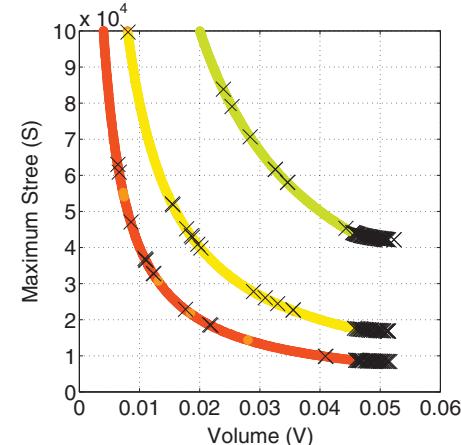
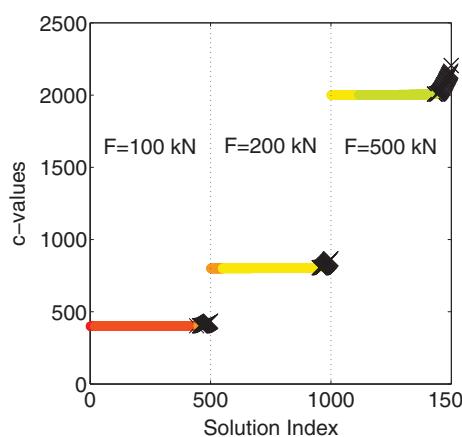
**Table 2**  
Higher-level design principles (HDP) obtained for truss design problem.

Notation	$\psi(\mathbf{x})$	$S_{\text{eff}}$ (in percent)	Notation	$\psi(\mathbf{x})$	$S_{\text{eff}}$ (in percent)
HDP1	$SV + Sx_1y$	88.13	HDP13	$SVy$	87.53
HDP2	$SV + Sx_2y$	86.73	HDP14	$V^2y/(x_1x_2)$	87.80
HDP3	$y$	84.87	HDP15	$x_2y/x_1$	86.00
HDP4	$Sx_2y/F$	85.67	HDP16	$Sx_1y/F$	84.40
HDP5	$V/x_2$	84.67	HDP17	$S(x_1 + x_2)y$	86.87
HDP6	$V/x_1$	86.53	HDP18	$Sx_1$	80.93
HDP7	$SV$	89.07	HDP19	$S(x_1 + x_2)$	80.47
HDP8	$SVx_2/x_1$	86.13	HDP20	$SVx_2y/x_1$	87.20
HDP9	$Vy/x_1$	87.33	HDP21	$V^2/(x_1x_2)$	87.73
HDP10	$Vy/x_2$	87.53	HDP22	$SV^2/(Fx_2)$	87.67
HDP11	$SV/F$	88.47	HDP23	$y/(Sx_1)$	86.00
HDP12	$Sx_2y$	85.87	HDP24	$x_2/x_1$	83.93

6. Maximum tree depth = 10, Maximum tree length = 10,
7. Threshold significance  $S_{\text{reqd}} = 80$  percent,  $\epsilon = 3$ .

The fifth and sixth parameters are specific to GP. More information about them can be found in Bandaru and Deb (2013a). Table 2 shows all relationships obtained through the proposed approach. The dimensions have been omitted for conciseness. All relationships seen in this table can be obtained by combining different higher-level relationships obtained above analytically. Note how through the use of the COI constraint the artificial relationship  $SVF$  has been avoided. Here we also observe the commensurability constraint coming into play in HDP1, HDP2, HDP17 and HDP19, where only expressions with the same units are added. Finally, it should be pointed out that most relationships can be further simplified using other relationships. For example, since  $y$  is constant according to HDP3, it can be dropped from other relationships, thus reducing the number of unique relationships. Automatically processing and simplifying algebraic expressions is however not a simple task. Moreover, simplification may affect the significance of a relationship, sometimes taking it below the threshold  $S_{\text{reqd}}$  (seen later in Section 5.1). Therefore, we will not pursue it in this paper.

Automated innovation uses *cluster plots* to visualize the obtained relationships. A cluster plot is simply a graph showing the distribution of sorted  $c$ -values against the solution index. The clusters are represented with different colors or gray levels and the unclustered points are denoted using 'x'. When these are mapped back onto the Pareto-optimal front, the user is able to visually perceive which relationships apply to what region of the front. Higher-level automated innovation uses similar cluster plots, with the exception that the datasets are shown together. However,  $c$ -values are sorted within each dataset separately. Fig. 6 shows the cluster plot for HDP7. Even



**Fig. 6.** Cluster plot for HDP7.

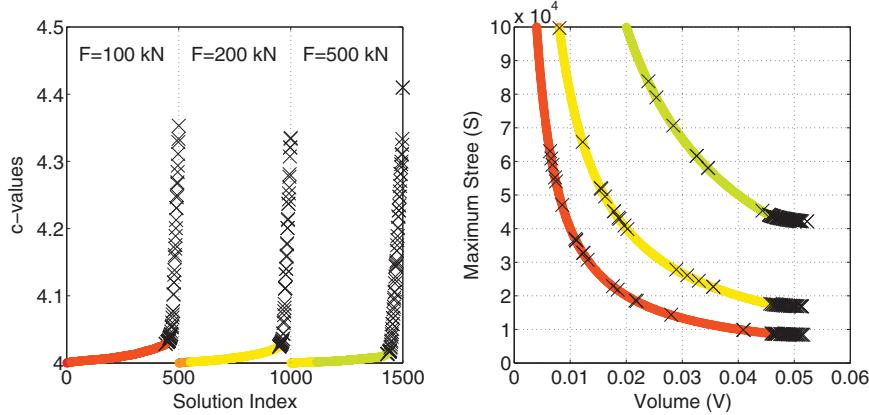


Fig. 7. Cluster plot for HDP12.

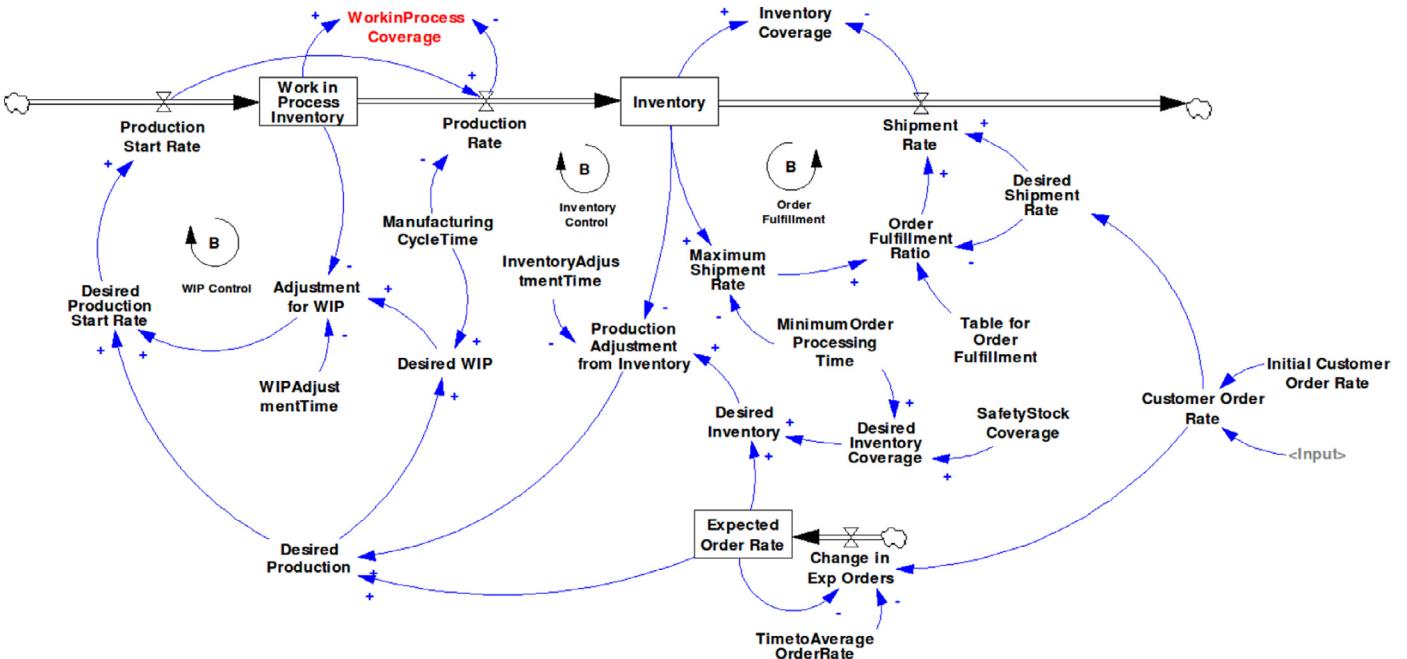


Fig. 8. Inventory management model used in this paper (available from Sterman, 2001).

though the value of  $SV$  changes between datasets, it remains constant within them for most solutions, thus it is a higher-level relationship. Fig. 7 shows the cluster plot for HDP12, where  $F$  negates the change in values of  $SV$  between datasets. Any relationship in which the presence of the changing parameter causes  $c$ -values across all datasets to be approximately equal is also a higher-level relationship. In both figures, it is observed that the unclustered solutions in all three datasets lie on a specific region of the Pareto-optimal front. Here this happens because one of the constraints is active in this region. For more discussion specific to the truss-design problem, readers are referred to Bandaru and Deb (2011b, 2013b).

In the following sections, we describe two applications relevant to a manufacturing company where higher-level innovation may be useful. The first application looks at the broader case of inventory management of the products manufactured by the company. The second application is to a metal cutting process which is one of the operations required for the manufacturing of the product. These applications have been chosen to exemplify the use of automated innovation in a wide range of problem areas. In fact, any design/system/process that can be posed as a multi-objective problem can be a suitable case study for automated innovation and its

extensions, namely higher and lower-level innovation (Bandaru & Deb, 2013b). However, in this paper we will limit ourselves to discussions on higher-level innovation only.

#### 4. Application I: Inventory management problem

The inventory management problem depicts the internal production and inventory process of a manufacturing company with the aim to investigate how a single manufacturing company manages its capacity to balance orders, production and inventory so as to meet customer demand. It is a subset of the generic stock management problem which finds its application in several domains like capital investment, equipment and human resources (Sterman, 2000) and at various levels of aggregation.

Specifically, we adopt a classical inventory management model developed and presented in Sterman (2000) and shown in Fig. 8. In this model the manufacturing company implements a *make-to-stock* production strategy, in which the orders are produced for storage in the inventory according to a forecast rather than actual customer orders. In other words, the company tries to maintain a sufficient stock of finished products in the inventory so that incoming customer

orders may be fulfilled. However, there exists a delay in filling the inventory as manufacturing of a product takes time. In the inventory management model the supply line, which contains all the units that have been ordered by the decision maker but not yet received in the inventory, is represented by the work-in-process (*WIP*) inventory. Thus, *WIP* inventory contains all the unfinished products that have started to be manufactured in the production line but are not yet finished. As Fig. 8 shows, the *WIP* inventory is affected through the production start rate and the production rate. The main decision point of the model is to define a sufficient production start rate that will in time, replace the expected shipments of products from the inventory, as well as keep a sufficient supply line and inventory to provide a good customer service level (Sterman, 2006). The model also does not have a backlog of unfilled orders, and all the orders not immediately fulfilled are lost as customers seek alternate suppliers.

As mentioned before, this inventory management model has been adopted from Sterman (2000). However an additional variable has been added to the model, namely the work-in-process coverage (*WIPC*), which defines the amount of time the *WIP* inventory could supply the inventory with products at the current production rate given its *WIP* level. The *WIPC* at time  $t$  is given by:

$$WIPC_t = \frac{WIPInv_t}{PR_t}, \quad (13)$$

where *WIPInv* <sub>$t$</sub>  stands for the amount of unfinished products still in production at time  $t$ , and *PR* <sub>$t$</sub>  is the current production rate or rate at which the products leave *WIPInv*. For further details about the model, readers are referred to Sterman (2000).

#### 4.1. Multi-objective optimization of inventory management model

The aim of the stock management problem is to optimally regulate the stocks to meet some system objective. The above discussed inventory management model, which encompasses this issue, aims to provide a good customer service level by meeting customer demand which in turn can be achieved by maintaining sufficient amount of inventory and *WIP* levels. Here it could be argued that customer demand can always be met if the finished goods inventory and its supply line, i.e. *WIP* inventory, are high enough. However, keeping high inventories, both of finished and unfinished goods, generates significant holding costs for the manufacturing company. Thus, the dilemma for the decision maker is to find optimal trade-offs between supply line and inventory coverage and customer demand. One way of dealing with this dilemma is to perform a multi-objective optimization study. The objectives are to minimize *WIPC* and inventory coverage (*INVC*) while maximizing the shipment rate (*SR*) of customer orders. The inventory coverage denotes the time for which inventory (*INV*) is able to supply the customers with products based on the current shipment rate, and is given in units of weeks. Its value at a time  $t$  is,

$$INVC_t = \frac{INV_t}{SR_t}. \quad (14)$$

The third objective denotes the shipment rate of the system. It is defined through the desired shipment rate (*DSR*) and the order fulfilment ratio (*OFR*) and is given in units of products/week. The order fulfilment ratio is a function of the ratio of maximum shipment rate (*MSR*) to the desired shipment rate. Thus, shipment rate at a time  $t$  is,

$$SR_t = DSR_t \times OFR_t = DSR_t \times f(MSR/DSR), \quad (15)$$

where the values for  $f$  are specified by the Table for Order Fulfillment. Shipments fall below desired shipments when the feasible shipment rate, i.e. the maximum shipment rate (*MSR*), is lower than the desired shipment rate, indicating that some products are unavailable in the inventory. See Fig. 18 later. Sterman (2000) uses a constant demand signal (*D*) of 10,000 products/week over a period of  $T = 100$  weeks. Thus, *D* becomes a parameter (external factor) for the optimization

formulation. In this study, we consider two scenarios, one with the original demand and the second with 50 percent higher demand, i.e.  $D = 15,000$  products/week. The aim of optimization in both scenarios is to generate a well-optimized and diverse Pareto-optimal front. Here, it should also be pointed out that all time units, both endogenous and exogenous variables, in the model are given in weeks. The optimization formulation is given as:

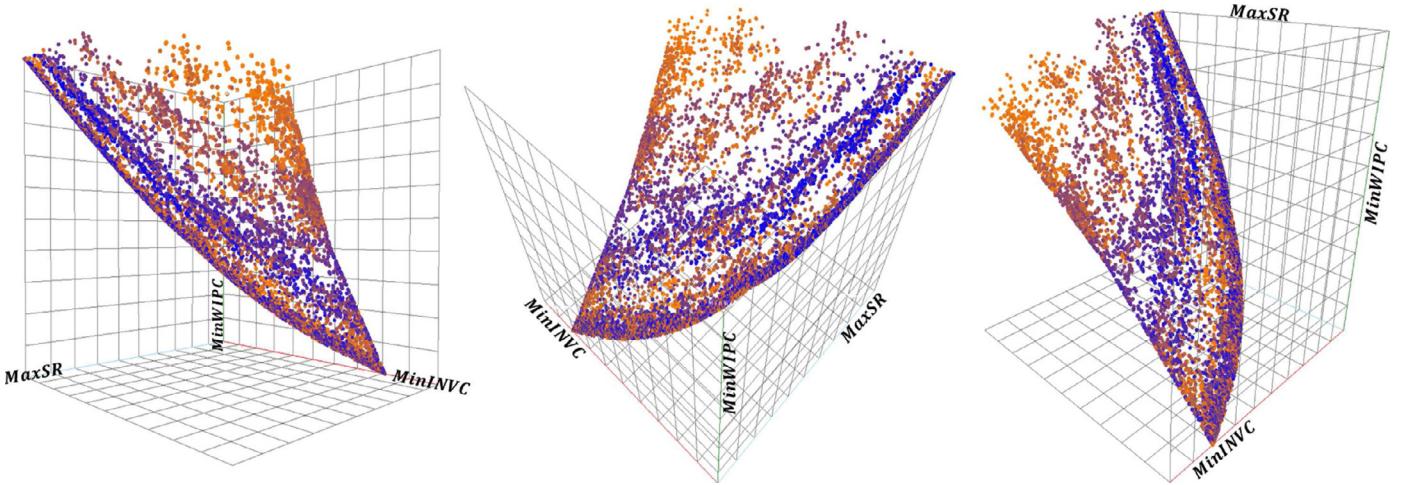
$$\begin{aligned} \text{Minimize } & f_1(\mathbf{x}) = \mu_{WIPC} = \sum_{t=0}^T \frac{WIPC}{T}, \\ \text{Minimize } & f_2(\mathbf{x}) = \mu_{INVC} = \sum_{t=0}^T \frac{INVC}{T}, \\ \text{Maximize } & f_3(\mathbf{x}) = \mu_{SR} = \sum_{t=0}^T \frac{SR}{T}, \\ \text{Subject to } & 0.25 \leq WIPAT, MOPT, SSC \leq 10.0, \\ & 5.0 \leq MCT, IAT, TAOR \leq 15.0, \\ \text{Variables: } & WIPAT, MOPT, SSC, MCT, IAT, TAOR \text{ are continuous.} \end{aligned} \quad (16)$$

The objective functions represent the mean values of the *WIP* coverage, inventory coverage and the system's shipment rate respectively over the period of simulation  $T$ . The system dynamics approach is used to calculate *WIPC*, *INVC*, *SR* and other system quantities at various time-steps during the simulation period. The Vensim® model of the system can be downloaded from Sterman (2001). The variables used in the optimization are described below:

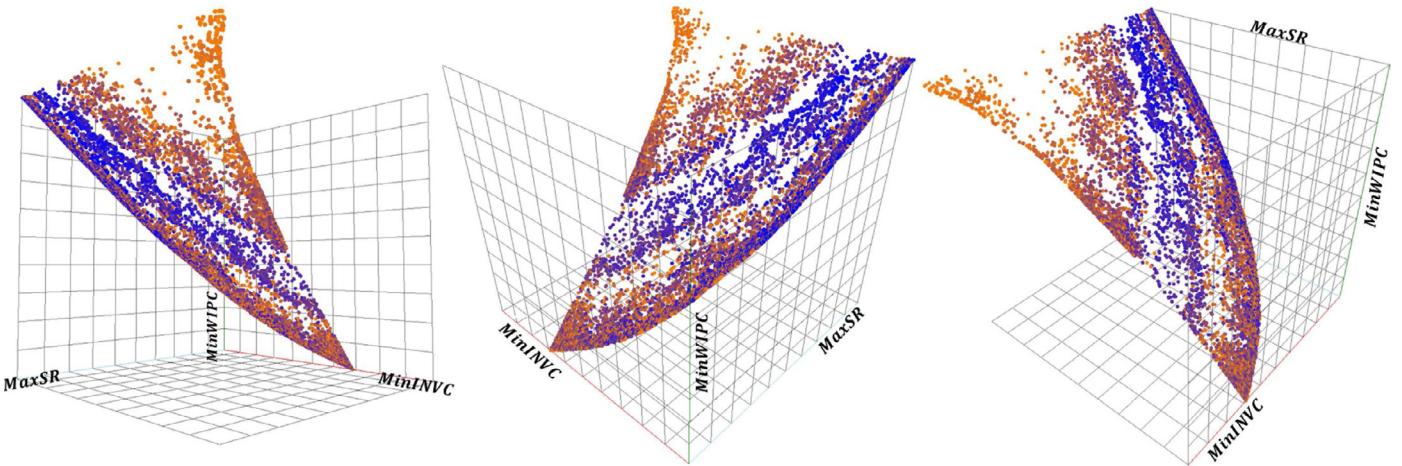
1. *WIPAT*: WIP Adjustment Time represents the time required to adjust *WIPInv* to the desired level, and is given in units of weeks.
2. *MOPT*: Minimum Order Processing Time denotes the minimum time required by the company to process and ship a customer order, and is given in units of weeks.
3. *SSC*: Safety Stock Coverage represents the time over which the company would like to maintain a safety stock, excess to the order processing time, in order to meet any variations in customer demand or in the supply of finished product. It is given in units of weeks.
4. *MCT*: Manufacturing Cycle Time represents the average delay time of the production process from the start till the completion of a product, and is given in units of weeks.
5. *IAT*: Inventory Adjustment Time represents the time over which the manufacturing company seeks to bring the inventory with the finished products in balance with the desired inventory level. It is also given in units of weeks.
6. *TAOR*: Time to Average Order Rate denotes the time over which the demand forecast is adjusted to actual customer order, and is given in units of weeks.

#### 4.2. Visual analysis of Pareto-optimal fronts

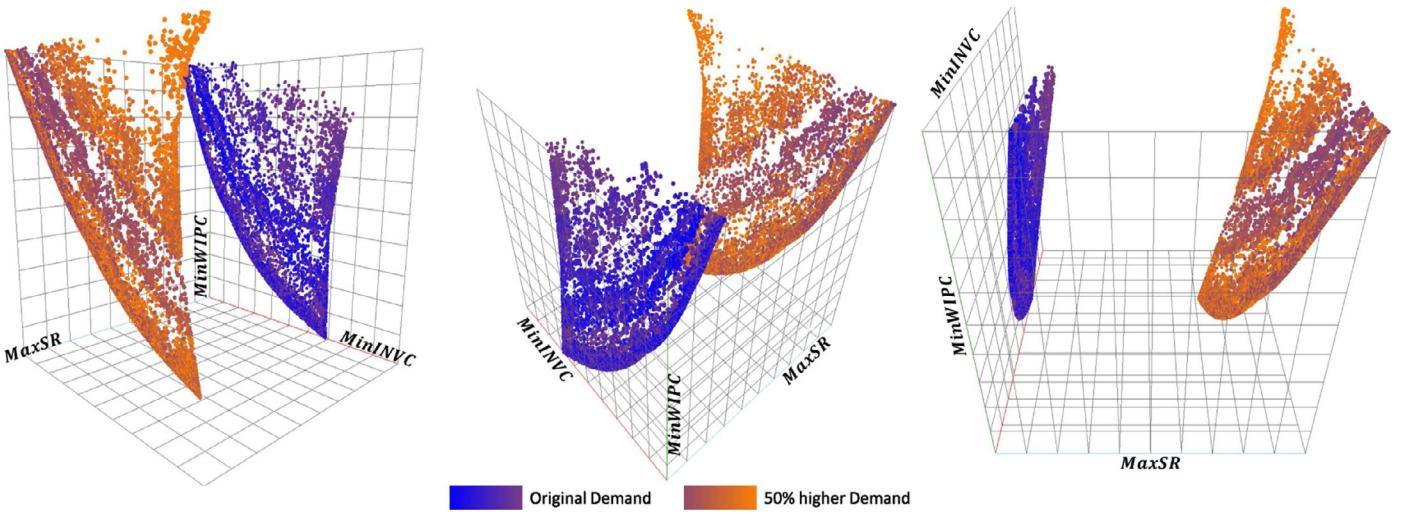
The optimization runs for both scenarios ( $D = 10,000$  and  $D = 15,000$  products/week) start with an initial population of 100 solutions generated through the uniform Latin hypercube approach. Each scenario is optimized using NSGA-II for 60,000 function evaluations and the non-dominated solutions from different generations are archived. Experimentation with a few parameter combinations yielded well-distributed fronts in both scenarios. However, crossover parameters of  $p_c = 0.9$ ,  $\eta_c = 10$  and mutation parameters of  $p_m = 0.05$ ,  $\eta_m = 50$  generate the highest number of non-dominated points. Sterman's original demand level of  $D = 10,000$  leads to 11,425 non-dominated solutions, as shown in Fig. 9 while 11,472 solutions are obtained with 50 percent higher demand, as shown in Fig. 10. The similarity in the shapes of the Pareto-optimal fronts is a typical feature observed in higher-level innovation studies (Bandaru & Deb, 2013b), also seen in Fig. 5. The reason for this similarity is the fact that variation of external parameters (*F* for truss design and *D* here) does not change the underlying physics of the problem. The mechanism that is responsible for causing optimality in solutions remains the same. However, a shift of the solutions in decision and/or objective



**Fig. 9.** Different perspectives of the Pareto-optimal set obtained for  $D = 10,000$ .



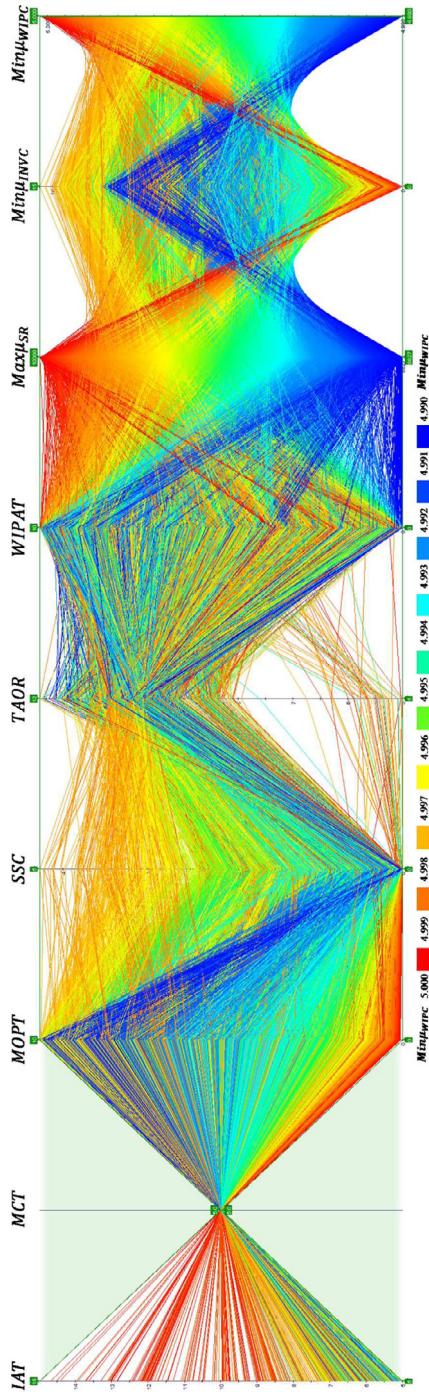
**Fig. 10.** Different perspectives of the Pareto-optimal set obtained for  $D = 15,000$ .



**Fig. 11.** Similarities in the Pareto-optimal sets of both scenarios.

space may be observed as shown in [Fig. 11](#) for the current problem. It is clear from the figure that the Pareto-optimal front of the higher demand scenario contains solutions operating at a higher range of shipment rates. From the supply chain perspective, this happens because the system tends to generate a higher WIP inventory under high demand conditions and more importantly, because there is no

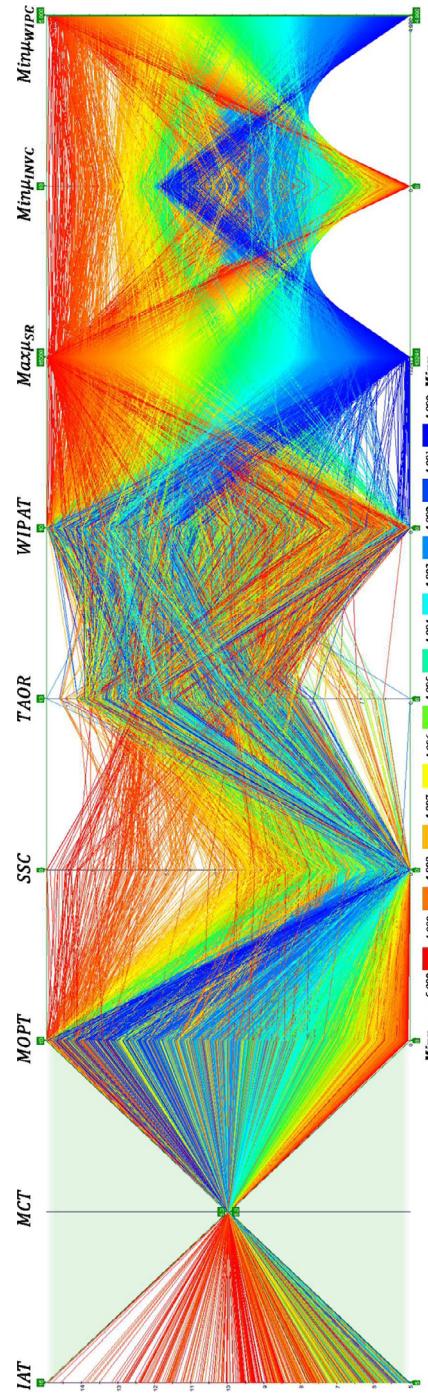
capacity constraint on the production start rate. Thus whatever demand rate is set downstream, the system should start manufacturing it. The system as it is holds excessive production capacity, which is an additional reason as to why its behavior is almost identical despite the change in customer demand. However, at very high demands, even without capacity constraints, the production rate (and hence the



**Fig. 12.** Parallel coordinates plot for solutions of original demand scenario.

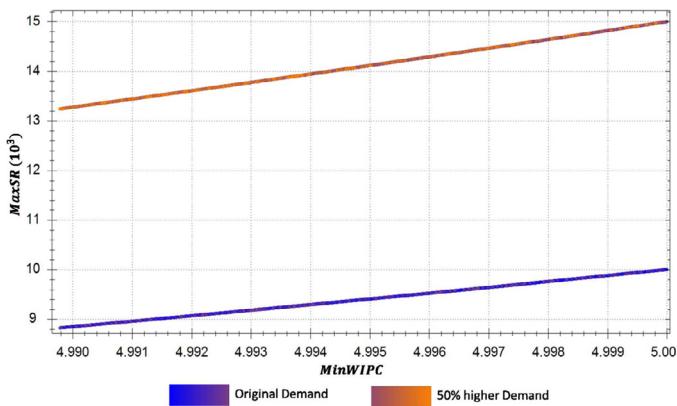
shipment rate) will be limited due to the time required for adjusting the WIP inventory level to the desired WIP.

Another interesting characteristic of the two Pareto-optimal fronts is that they are flat, as clearly seen in Fig. 14 when viewing the fronts alongside an 'edge'. This indicates that two of the optimization objectives, namely  $\mu_{WIPC}$  and  $\mu_{SR}$ , which form the two axes in Fig. 14 are linearly correlated making one of them redundant. The positive correlation can also be seen through the parallel coordinate plots in Figs. 12 and 13, where these two objectives show similar heatmap patterns. However, establishing linear interdependence through these plots is difficult. Fig. 14 also shows that the higher demand scenario has a greater slope indicating that for the same increase in WIPC, a greater demand requires a greater increase in shipment rate.



**Fig. 13.** Parallel coordinates plot for solutions of higher demand scenario.

The range of variation in the variables and objectives values on the Pareto-optimal fronts of the two scenarios are shown in Table 3. Firstly, we observe that  $\mu_{WIPC}$  has the same narrow range of [4.9898, 5.0000] weeks in both cases, indicating that WIPC is not directly affected by the demand  $D$ . Secondly, Table 3 (and Figs. 12 and 13) shows that  $MCT$  takes the same value of 5.0000 for all solutions in both cases, which also happens to be the lower bound of the variable. This is quite logical since any optimally operating system will have the minimum possible manufacturing cycle time (unless it conflicts with some objective or violates a constraint, neither of which is true for the model under consideration). A few representative solutions, namely the extreme solutions and the solution closest to the



**Fig. 14.** Linear correlation is observed between two of the objectives with the greater demand case having a higher slope.

**Table 3**

Ranges of variable and objective values for the original ( $D = 10,000$ ) and higher ( $D = 15,000$ ) demand scenarios.

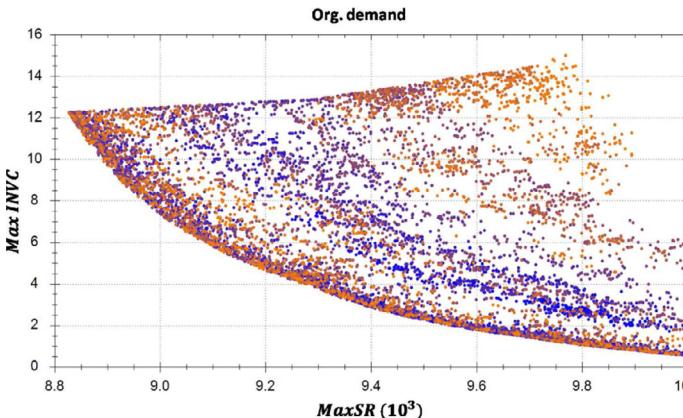
	Original demand	Higher demand
WIPAT	[0.2500, 10.0000]	[0.2500, 10.0000]
MOPA	[0.2500, 10.0000]	[0.2500, 9.9997]
SSC	[0.2500, 4.5275]	[0.2500, 7.7033]
MCT	[5.0000, 5.0000]	[5.0000, 5.0000]
IAT	[5.0000, 14.9798]	[5.0000, 14.9664]
TAOR	[5.2556, 11.8767]	[5.7463, 13.1676]
$\mu_{WIPC}$	[4.9898, 5.0000]	[4.9898, 5.0000]
$\mu_{INVC}$	[0.5000, 14.9949]	[0.5000, 17.6344]
$\mu_{SR}$	[88, 27.18, 10, 000.00]	[13, 240.86, 15, 000.00]

centroid of the extreme solutions, are shown in **Tables 4** and **5** for both demand cases.

Observing the plane of the two Pareto-optimal fronts as in **Fig. 15**, we see that both fronts have two distinct edges with solutions lying in between them. While the lower edge represents the trade-off front between  $\mu_{SR}$  and  $\mu_{INVC}$ , the second edge which is found to be a constraint boundary, where the search space has been limited by the upper limit of  $MOPA$ . In other words, all solutions on the this edge have  $MOPA = 10$  weeks.

#### 4.3. Higher-level innovation results

Visual analysis of the solutions can take us only so far in deciphering the inter-relationships between variables and/or objectives. In this section, we apply the higher-level automated innovation approach developed in **Section 2.1** and illustrated in **Section 3** for the



**Fig. 15.** The planes of the Pareto-optimal fronts in both scenarios.

**Table 4**

Some representative solutions for the original demand ( $D = 10,000$ ) scenario.

	Min. $\mu_{SR}$	Max. $\mu_{INVC}$	Max. $\mu_{WIPC}$	Centroid
WIPAT	0.2500	5.1126	0.3244	8.5735
MOPA	9.9994	9.9755	0.2500	6.9576
SSC	0.2500	4.4846	0.2500	1.9394
MCT	5.0000	5.0000	5.0000	5.0000
IAT	5.0000	8.5330	5.4509	5.6124
TAOR	10.0083	10.1868	10.0135	10.4190
$\mu_{WIPC}$	4.9898	4.9980	5.0000	4.9960
$\mu_{INVC}$	12.2544	14.9949	0.5000	9.6048
$\mu_{SR}$	8827.18	9768.36	10, 000.00	9531.66

**Table 5**

Some representative solutions for the higher demand ( $D = 15,000$ ) scenario.

	Min. $\mu_{SR}$	Max. $\mu_{INVC}$	Max. $\mu_{WIPC}$	Centroid
WIPAT	0.2500	6.7624	5.1250	7.2090
MOPA	9.9997	9.9010	0.2500	6.7099
SSC	0.2502	7.7033	0.2500	2.1359
MCT	5.0000	5.0000	5.0000	5.0000
IAT	5.0000	12.4092	8.8829	6.0767
TAOR	11.2730	11.1474	9.9305	10.8366
$\mu_{WIPC}$	4.9898	4.9999	5.0000	4.9967
$\mu_{INVC}$	12.2548	17.6344	0.5000	9.4514
$\mu_{SR}$	13, 240.86	14, 984.41	15, 000.00	14, 409.35

truss design problem. The terminal set is given by

$$\mathcal{T} = \{\mu_{WIPC}, \mu_{INVC}, \mu_{SR}, WIPAT, MOPA, SSC, MCT, IAT, TAOR, D\}$$

and a threshold significance of  $S_{reqd} = 70$  percent is used to extract the higher-level principles shown in **Table 6** from the trade-off datasets of the two scenarios. The other parameters remain the same as for the truss design problem. The fundamental dimension of time  $T$  is given in weeks, while a new dimension  $P$  is introduced to represent number of products. Thus shipment rate will have the dimensions  $SR : M^0 L^0 T^{-1} P$ ,  $WIPAT : M^0 L^0 T P^0$  and so on. The relationship visually observed in **Figs. 12** and **13** is seen as HDP1. The other principles can be categorized into three groups based on their influence on forecasting uncertainty, behavior in the objective space and relationship with the customer demand  $D$ .

#### 4.3.1. Forecasting uncertainty

HDP2 which has the second highest significance value and can be simplified as  $TAOR \propto \mu_{INVC}^{1.5}$ , states that for both demand cases, an increase in the value of  $TAOR$  requires an increase in  $INVC$  for Pareto-optimal performance of the system. This is quite intuitive from the forecasting point of view because a higher  $TAOR$  value, which works as the smoothing factor in the forecasting approach utilized in the inventory management model, means that the customer demand is

**Table 6**

Higher-level design principles obtained for inventory management problem.

Notation	$\psi(\mathbf{x})$	$S_{\text{eff}}$ (in percent)
HDP1	$MCT$	100.00
HDP2	$TAOR^2/\mu_{INVC}^3$	81.36
HDP3	$\mu_{SR}/\mu_{INVC}^3$	80.64
HDP4	$\mu_{WIPC}/\mu_{INVC}^3$	80.78
HDP5	$SSC \times TAOR/\mu_{INVC}^3$	77.76
HDP6	$SSC/\mu_{INVC}^3$	76.55
HDP7	$SSC/((TAOR - \mu_{WIPC}) \times \mu_{INVC}^2)$	76.21
HDP8	$D/((TAOR - \mu_{WIPC}) \times TAOR \times \mu_{SR})$	75.51
HDP9	$\mu_{SR}/((TAOR - \mu_{WIPC}) \times TAOR \times \mu_{INVC})$	75.46
HDP10	$1/((TAOR - \mu_{WIPC}) \times TAOR)$	75.18
HDP11	$D/((TAOR - \mu_{WIPC}) \times TAOR \times \mu_{INVC})$	74.99
HDP12	$\mu_{SR}/(SSC \times \mu_{INVC}^2)$	72.22
HDP13	$\mu_{WIPC}/(SSC \times \mu_{INVC}^2)$	72.02
HDP14	$\mu_{SR}/(TAOR \times \mu_{INVC}^2)$	71.62

averaged over a greater period of time, causing the forecast to be less accurate. This uncertainty therefore requires a greater  $INVC$ . However, what is not intuitive is the fact that this increase is less in comparison to the increase in  $TAOR$  value since  $\mu_{INVC}$  has a higher exponent.

According to HDP2, when the forecast uncertainty increases due to a higher  $TAOR$ ,  $INVC$  also increases. In HDP5  $\mu_{INVC}$  has an exponent of 3 and therefore the increase in the denominator is not entirely compensated by the increase in  $TAOR$ . As a result,  $SSC$  should also increase. As Fig. 8 shows,  $SSC$  does not have a direct influence on the forecast, i.e. expected order rate, however HDP5 implies that for

Pareto-optimal operation of the system  $TAOR$  and  $SSC$  should display a similar behavior. Thus for an increase in forecast uncertainty, HDP5 requires a cubic increase of  $INVC$  and to compensate the latter 77.76 percent (effectively) of the Pareto-optimal solutions in both scenarios require an increase of  $SSC$ . The equivalency of  $SSC$  and  $TAOR$  on the Pareto-optimal front is also confirmed through HDP14 which is obtained by replacing  $SSC$  in HDP12 with  $TAOR$ . Figs. 16 and 17 show the cluster plots of HDP12 and HDP14 respectively. Note that even though the two relationships cluster the datasets differently, the clusters together cover roughly the same region on both the Pareto-optimal fronts.

HDP6 also describes a direct relationship between  $SSC$  and  $INVC$  which can also be obtained by dropping  $TAOR$  from HDP5 along with a small decrease in the significance value. From a model perspective, HDP6 indicates that an increase of  $INVC$  requires a cubic increase of  $SSC$  for the resulting solution to remain on the Pareto-optimal front. The relation between the two entities in HDP6 is also independently apparent since  $SSC$  directly affects the desired inventory coverage variable as the system tries to minimize the gap between  $INVC$  and desired inventory coverage.

#### 4.3.2. Behavior in the objective space

HDP3 and HDP4 say that two of the objectives ( $\mu_{SR}$  and  $\mu_{WIPC}$ ) have a cubic relationship with the third objective  $\mu_{INVC}$ . Since the significance of the two rules is almost identical and also as seen from Figs. 19 and 20 both rules involve approximately the same clusters on the Pareto-optimal front, it is possible to combine the two rules

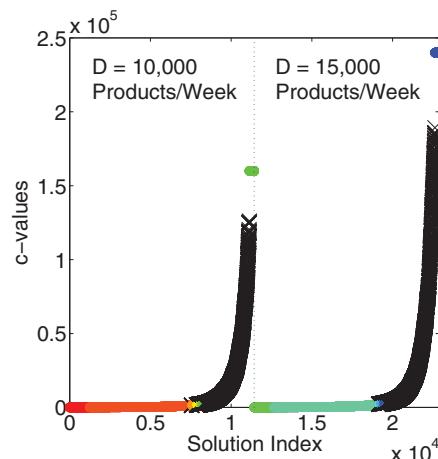


Fig. 16. Cluster plot for HDP12.

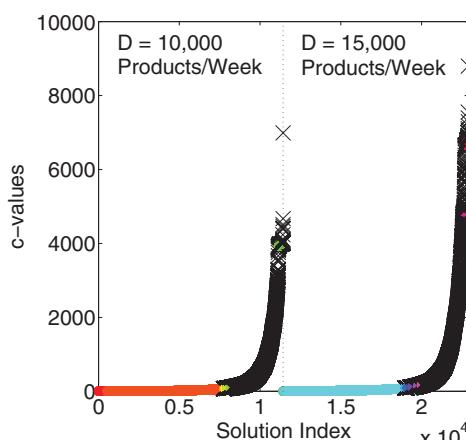
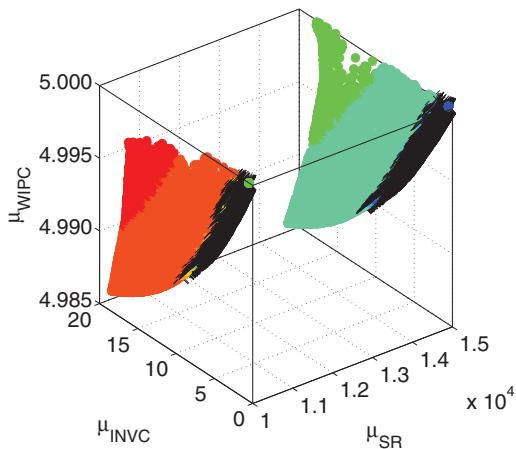
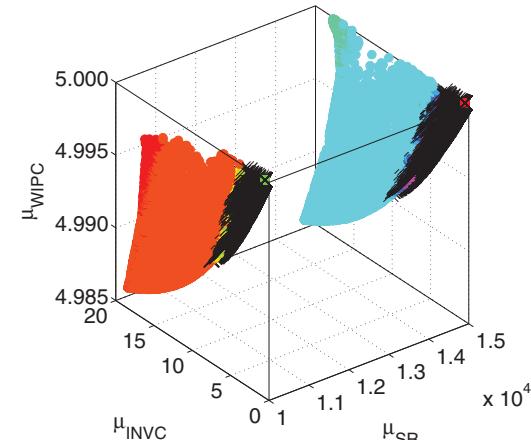


Fig. 17. Cluster plot for HDP14.



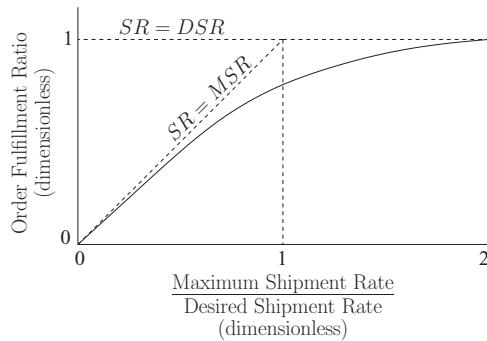


Fig. 18. Order fulfillment function for multi-product inventory (Sterman, 2000).

to obtain  $\mu_{SR} \propto \mu_{WIPC}$ , which is verified in Fig. 14. HDP3 by itself is expected as the inventory management model implements a make-to-stock production strategy which means that the customer orders are being fulfilled through the inventory, in turn meaning that SR and INVC should increase together. However, it is interesting to note that this relationship is cubic with respect to INVC for about 80 percent of the Pareto-optimal solutions. There are two possible explanations here; (i) the increase in production rate, which is the rate at which the inventory is filled, is larger than the increase in SR resulting in an accumulation of products in the inventory, thereby requires less than proportional increase in INVC, (ii) the order fulfillment function in

the model is restricting products in the inventory from being shipped. This happens because the Table for Order Fulfillment mimics a multi-product inventory and when the inventory is not holding the right product, 100 percent of the customer orders will not be fulfilled as shown in Fig. 18 (Sterman, 2000), and the inventory grows.

HDP4 displays a relation between WIPC and INVC, which also is quite evident as WIP inventory directly feeds the inventory with products. However, the only possible explanation for the cubic relationship in HDP4 is the behavior of the order fulfillment function as discussed above. HDP12 is a slight variation of HDP3 where SSC compensates (with about 8 percent decrease in significance) the change in the exponent of  $\mu_{INVC}$  from cubic in HDP3 to squared in HDP12. The reason behind this compensation can be explained with regard to the discussion on HDP6 presented above. The same applies to HDP13 which is again a similar variation of HDP4.

#### 4.3.3. Customer demand

Higher-level innovation principles involving the customer demand  $D$  are especially important as they represent relationships applicable on the Pareto-optimal front that change with the varying parameter, which in this case is  $D$ . Firstly, HDP10 can further be simplified to  $\mu_{WIPC} = \frac{TAOR^2 - 1}{TAOR}$  where  $c$  is a constant. For positive values of TAOR,  $\mu_{WIPC}$  increases with TAOR which is true for uniform demand patterns as implemented here for  $D = 10,000$  products/week and  $D = 15,000$  products/week. Simplifying HDP8 (whose cluster plot is shown in Fig. 21) using HDP10 we get  $\mu_{SR} \propto D$ , that can be observed

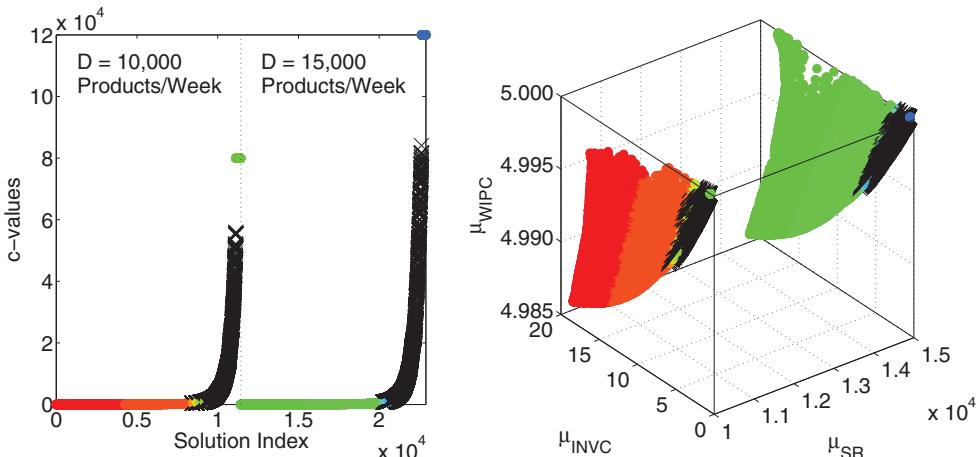


Fig. 19. Cluster plot for HDP3.

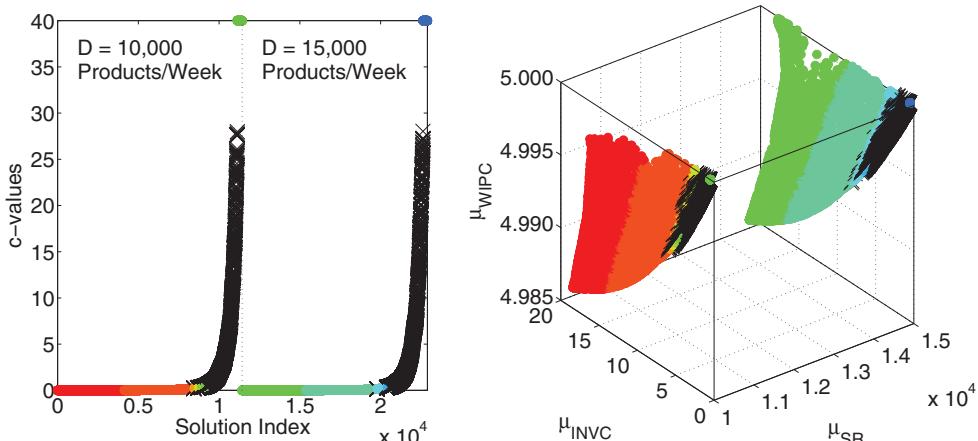


Fig. 20. Cluster plot for HDP4.

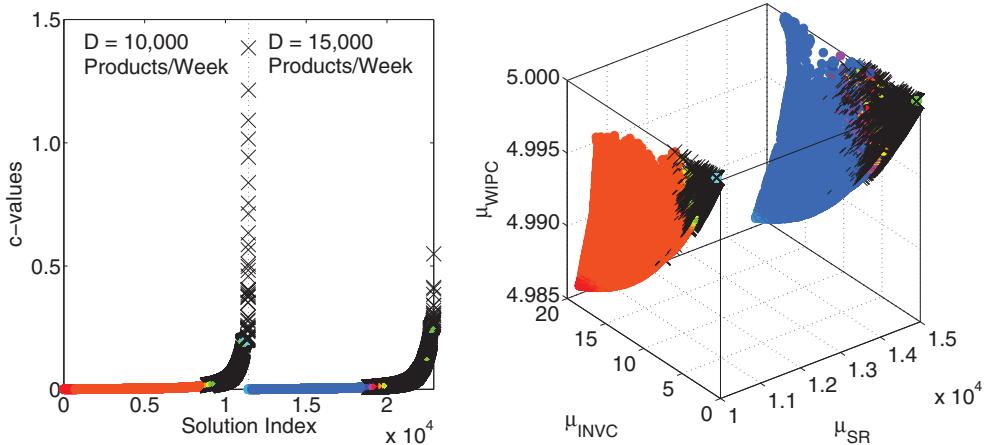


Fig. 21. Cluster plot for HDP8.

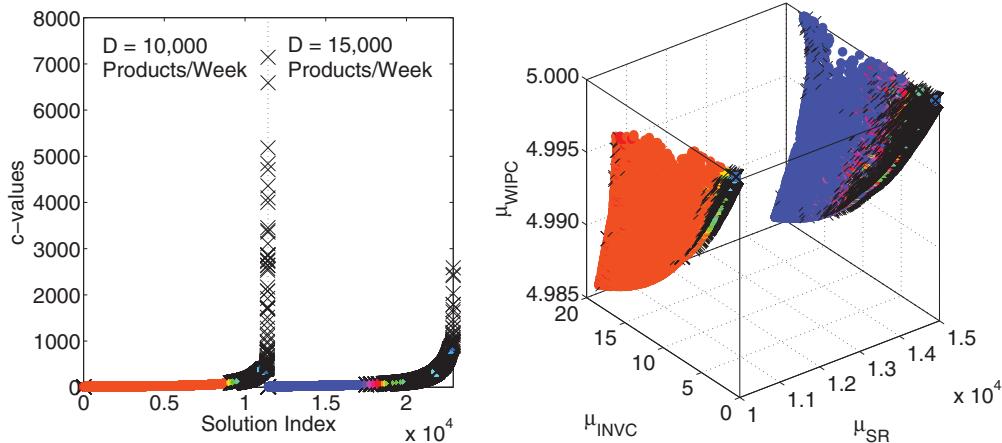


Fig. 22. Cluster plot for HDP11.

and verified in Fig. 11, and is an expected behavior of the inventory management model when the system is not capacity constrained. Another way of verifying  $\mu_{SR} \propto D$  is to calculate the percentage increase in  $\mu_{SR}$  between the two demand cases considered here. Noting from Table 3 the  $\mu_{SR}$  range to be [8827.18, 10,000.00] for the original demand and [13,240.86, 15,000.00] for the higher demand, this percentage can be calculated to be 50 percent. This means that 50 percent increase in demand is proportional to an equivalent increase in SR. Similarly, combining HDP10 and HDP11 (whose cluster plot is shown in Fig. 22) we get  $\mu_{INVC} \propto D$ , which accounts for 17.9 percent increase in the range of  $\mu_{INVC}$  between the two demand scenarios.

Before concluding this section, it should again be pointed out that despite the successful implementation of GP-based higher-level automated innovation to the inventory management model and logical interpretations of the obtained design principles, the relationships are valid only as long as the underlying mechanism of the system remains unchanged. The design principles therefore need not apply to any generic supply chain model.

## 5. Application II: Metal cutting problem

The work-in-process or WIP is a critical component of the supply chain. The manufacturing cycle times and costs play an important role among other factors in maintaining adequate WIP levels to feed the inventory in time for meeting customer demands. The optimal

performance of the machining and assembly processes thus becomes essential for a smooth running of the supply line. In this section, we consider a metal cutting problem as an example of a manufacturing process in the supply chain. The raw material will go through several such processes to become a finished product, which is eventually held in the inventory for shipping.

The metal cutting problem (Sardiñas, Santana, & Brindis, 2006) requires a steel bar to be machined using a carbide tool of nose radius  $r_n = 0.8$  millimeter on a lathe with  $P^{\max} = 10$  kilowatts rated motor to remove 2,19,912 cubic millimeter of material. A maximum cutting force of  $F_c^{\max} = 5000$  Newton is allowed. The motor has a transmission efficiency  $\eta = 75$  percent. The operation time  $T_p$  (a component of the total manufacturing cycle time) and the percentage used tool life  $\xi$  (an indicator of the total manufacturing cost) are to be minimized by optimizing the cutting speed ( $v$ ), the feed rate ( $f$ ) and the depth of cut ( $a$ ) while maintaining a surface roughness of  $R^{\max} = 50$  micrometer. The problem is formulated as,

$$\begin{aligned}
 \text{Minimize } & f_1(\mathbf{x}) = T_p(\mathbf{x}) \\
 \text{Minimize } & f_2(\mathbf{x}) = \xi(\mathbf{x}) \\
 \text{Subject to } & P(\mathbf{x}) \leq \eta P^{\max}, \\
 & F_c(\mathbf{x}) \leq F_c^{\max}, \\
 & R(\mathbf{x}) \leq R^{\max}, \\
 & 250 \leq v \leq 400 \text{ meter per minute}, \\
 & 0.15 \leq f \leq 0.55 \text{ millimeters per revolution}, \\
 & 0.5 \leq a \leq 6 \text{ millimeters}, \\
 \text{Variables: } & v, f \text{ and } a \text{ are continuous.}
 \end{aligned} \tag{17}$$

**Table 7**

Design principles obtained for metal-cutting problem in [Bandaru and Deb \(2013a\)](#).

Notation	$\psi(\mathbf{x})$	Significance ( $S$ ) (in percent)	Fundamental dimensions			
			Mass	Length	Time	Life
DP1	$v/(f^2\xi)$	72.70	0.0	-1.0	-1.0	-1.0
DP2	$av/f$	74.60	0.0	1.0	-1.0	0.0
DP3	$v/(f^2T_p\xi)$	73.40	0.0	-1.0	-2.0	-1.0
DP4	$f$	72.90	0.0	1.0	0.0	0.0
DP5	$a/(fT_p)$	72.90	0.0	0.0	-1.0	0.0
DP6	$a^{5.5}f\xi/T_p$	77.50	0.0	6.5	-1.0	1.0
DP7	$aT_pv/f$	74.20	0.0	1.0	0.0	0.0
DP8	$a^{5.5}T_p\xi$	82.60	0.0	5.5	1.0	1.0
DP9	$aT_pv$	74.10	0.0	2.0	0.0	0.0
DP10	$a^2T_p\xi/v$	74.40	0.0	1.0	2.0	1.0
DP11	$a^2\xi/v$	76.00	0.0	1.0	1.0	1.0
DP12	$a^{5.5}f\xi$	76.80	0.0	6.5	0.0	1.0

where

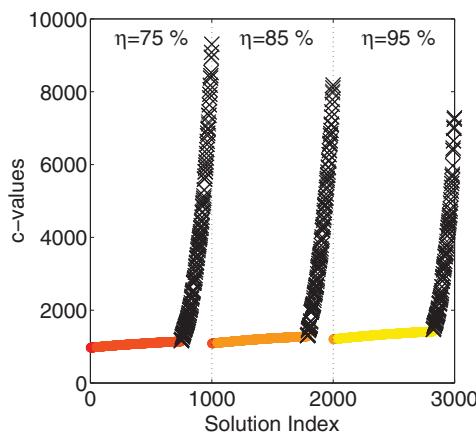
$$T_p(\mathbf{x}) = 0.15 + 219912 \left( \frac{1 + \frac{0.20}{T(\mathbf{x})}}{MRR(\mathbf{x})} \right) + 0.05,$$

$$\xi(\mathbf{x}) = \frac{219912}{MRR(\mathbf{x})T(\mathbf{x})} \times 100,$$

$$T(\mathbf{x}) = \frac{5.48 \times 10^9}{v^{3.46} f^{0.696} a^{0.460}}, \quad MRR(\mathbf{x}) = 1000vfa,$$

$$P(\mathbf{x}) = \frac{vF_c(\mathbf{x})}{60,000}, \quad F_c(\mathbf{x}) = \frac{6.56 \times 10^3 f^{0.917} a^{1.10}}{\nu^{0.286}}, \quad R(\mathbf{x}) = \frac{125f^2}{r_n}.$$

Optimization using NSGA-II with a population of 1000 results in 1000 trade-off solutions. The GP-based automated innovation approach has been used in [Bandaru and Deb \(2013a\)](#) with  $S_{reqd} = 70$  percent to obtain the design principles shown in [Table 7](#). Once again, the flexibility of dimensionally-aware GP to introduce new fundamental dimensions for empirical metrics, allows the use of 'Life' ( $L_i$ ) as a unit for the tool life  $\xi$  expressed as a percentage of the total tool life. The terminal set used is  $\mathcal{T} = \{T_p, \xi, v, f, a, \mathcal{R}\}$ , having the dimensions  $T_p : M^0 L^0 T^0 L_i^0, \xi : M^0 L^0 T^0 L_i^0, v : M^0 L^0 T^{-1} L_i^0, f : M^0 L^0 T^0 L_i^0, a : M^0 L^0 T^0 L_i^0, \mathcal{R} : M^0 L^0 T^0 L_i^0$ . The resultant dimensions automatically calculated by the algorithm for all principles are shown in the last four columns of [Table 7](#). Some of the obtained relationships were shown to corroborate the results of a previous study ([Deb & Datta, 2012](#)) while others were completely new yet significant. For example, DP4 represents the fact that for more than 70 percent of the trade-off solutions, the feed rate is at its upper limit, which is an indication that the optimization tries to reduce operation time for majority of the solutions, even if that means increase in tool wear and reduction of tool life.

**Table 8**

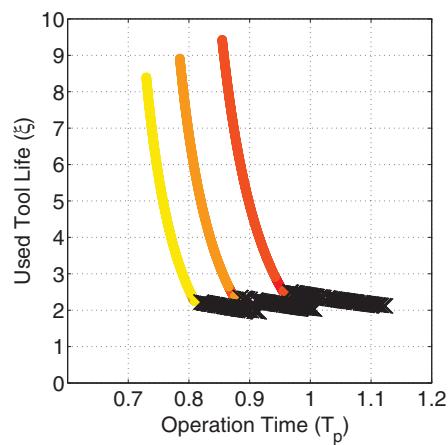
Higher-level design principles obtained for metal cutting problem.

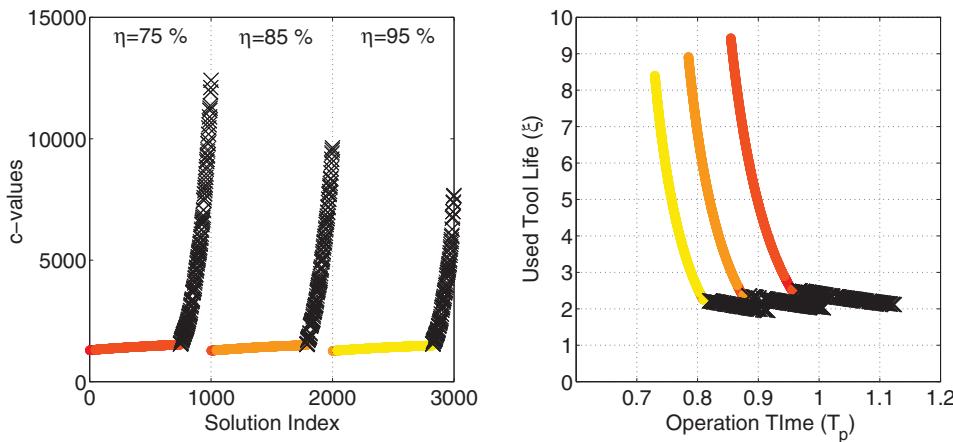
Notation	$\psi(\mathbf{x})$	$S_{eff}$ (in percent)	Notation	$\psi(\mathbf{x})$	$S_{eff}$ (in percent)
HDP1	$f$	76.43	HDP8	$a/(fT_p)$	76.50
HDP2	$a/f$	76.73	HDP9	$av/(fT_p\eta)$	79.67
HDP3	$av/(fT_p)$	79.73	HDP10	$a^2/(fT_p\xi)$	70.97
HDP4	$a/T_p^2$	72.37	HDP11	$vT_p/f^2$	80.07
HDP5	$a/(\eta f)$	76.43	HDP12	$av/(f\eta)$	78.50
HDP6	$T_p/f$	76.50	HDP13	$aT_p/(f\eta)$	76.10
HDP7	$av/f$	78.57	HDP14	$a^2/(f\xi)$	73.30

From DP2 we observe that the depth of cut,  $a$ , and cutting speed,  $v$ , are inversely proportional, which is quite intuitive from a machining science perspective. A higher depth of cut means more material to be removed which requires a higher cutting load. In order to reduce tool wear at high loads, it is a rule of thumb followed by machine operators to reduce the speed of the cutting tool and hence the inverse relationship. However, this slows down the machining process and increases total operation time, which is also realized through DP5, where  $a$  and  $T_p$  are directly proportional. The increase in tool wear with cutting speed is also supported by DP1.

### 5.1. Higher-level innovation results

The motor efficiency  $\eta$  is an example of an external parameter that is kept constant during optimization and here we vary it for our higher-level innovation study. Two additional NSGA-II trade-off fronts are generated with  $\eta = 85$  and 95 percent both having 1000 solutions each ([Bandaru & Deb, 2013b](#)). The dimensionless basis function  $\eta$  is added to the terminal set used above. Again, using the three datasets and the same parameters as in the truss design problem, with the exception that  $S_{reqd} = 70$  percent,  $\epsilon = 10$  and  $m^{(1)} = m^{(2)} = m^{(3)} = 1000$ , the higher-level design principles obtained are shown in [Table 8](#). Comparing the higher-level principles to those in [Table 8](#), we observe that  $HDP1 \equiv DP4$ ,  $HDP7 \equiv DP2$  and  $HDP8 \equiv DP5$ . Even though many relationships were discovered in the  $\eta = 75$  percent dataset, a higher-level innovation is required to identify those which remain significant with changing  $\eta$ . For example, HDP1 says that the feed rate still remains constant for most solutions in all three datasets. Also, notice that in HDP5, HDP9 and HDP12 the changing value of  $\eta$  negates the change in the values of HDP2, HDP3 and HDP7 respectively across datasets; though in this problem the change is too small to be seen on a cluster plot. For example, the cluster plots for HDP7 and HDP12 are shown in [Figs. 23](#) and [24](#) respectively. The cluster average  $c$ -values in the three datasets for HDP7 are 1068.06, 1197.16 and 1323.82 while for HDP12 they are

**Fig. 23.** Cluster plot for HDP7.



**Fig. 24.** Cluster plot for HDP12.

1424.08, 1408.42 and 1393.49 (respective ratios being 0.75, 0.85 and 0.95).

It is generally true that increase in performance of a sub-component improves the performance of the system as a whole. Here, we see the same with respect to motor transmission efficiency. As  $\eta$  increases across datasets, the optimization allows greater depth of cut values according to HDP5. Higher  $\eta$  also means greater cutting speeds (and lower operation times as seen in HDP11) since more power can be transmitted by the motor. However, the simultaneous increase in depth of cut and cutting speeds contradict each other as explained above and also as shown by HDP7 and HDP12.

In the truss design and inventory management problems, we were able to combine two or more relationships to obtain other significant relationships. However, in this problem, the significance drops on doing so. For example, though combining HDP1 and HDP2 implies that  $a$  is constant, this relationship is found to be less than 70 percent significant. Similar is the case for HDP1 and HDP6. Thus care should be taken when implementing an automatic simplification algorithm for reducing the innovization principles. Of course, the significance of the resultant relationship can always be used as an indicator.

## 6. Conclusions

The main contribution of this paper is in extending the dimensionally-aware genetic programming framework to higher-level automated innovization. As opposed to usual innovization, higher-level innovization considers multiple trade-off datasets generated by changing an external problem parameter. The task is to identify relationships that are common to all the datasets at a given threshold significance. Like in automated innovization, the task is formulated as an optimization problem that minimizes the number of clusters, the variance within clusters and the distance between clusters from different datasets, using a weighted objective function. A niching technique (niched-tournament selection) is used to prevent competition between dissimilar relationships, i.e. relationships that use different set of basis functions. This allows obtaining multiple relationships in a single algorithmic run. The dimensional-awareness of the approach ensures that only physically meaningful relationships are obtained and not relationships that are mere artifacts in the dataset. The developed procedure is first applied on a standard truss design problem to illustrate its efficacy and introduce cluster plots.

The other important contribution of this paper is the application of the above higher-level automated innovization approach to a well-known inventory management problem. Firstly, a three-objective system dynamics simulation model of the problem is created and optimized for a constant customer demand signal to obtain a trade-off dataset. Secondly, the demand signal is increased by 50 percent and another trade-off set is generated. The datasets are analyzed visu-

ally in the objective space and through parallel coordinate plots to reveal many characteristic features. Next, the two datasets are provided as input to the higher-level automated innovization algorithm to gain deeper insights into the underlying problem structure. The importance of the forecast parameter of the model and its equivalency to safety stock coverage stand out in some of the obtained rules. While visual analysis revealed correlations in the objective space, higher-level principles described the same correlations mathematically. Principles involving the customer demand (varying parameter) were also obtained. Most relationships could be explained from the theoretical perspective of inventory management systems. Higher-level innovization is also applied to a manufacturing process within the above system in order to demonstrate its usefulness irrespective of the problem domain.

While present approaches for automated innovization and its extensions (higher and lower-level) are satisfactory in generating significant relationships, much work needs to be done for automatic simplification of the obtained principles, owing to some of the difficulties presented in this paper. Understanding and interpreting the relationships can be a difficult task, even for an expert of the problem domain, if they are in a mathematically unsimplified form. Moreover, not all relationships can be explained from a purely theoretical viewpoint.

## References

- Bandaru, S. (2013). *Automated innovization: Knowledge discovery through multi-objective optimization* (Ph.D. thesis). Kanpur: Indian Institute of Technology.
- Bandaru, S., & Deb, K. (2011a). Automated innovization for simultaneous discovery of multiple rules in bi-objective problems. In *Proceedings of the 6th international conference on evolutionary multi-criterion optimization* (pp. 1–15). Springer.
- Bandaru, S., & Deb, K. (2011b). Towards automating the discovery of certain innovative design principles through a clustering-based optimization technique. *Engineering Optimization*, 43(9), 911–941.
- Bandaru, S., & Deb, K. (2013a). A dimensionally-aware genetic programming architecture for automated innovization. In *Proceedings of the 7th international conference on evolutionary multi-criterion optimization (EMO'13)* (pp. 513–527). Berlin, Heidelberg: Springer.
- Bandaru, S., & Deb, K. (2013b). Higher and lower-level knowledge discovery from Pareto-optimal sets. *Journal of Global Optimization*, 57(2), 281–298. doi: 10.1007/s10898-012-0026x
- Deb, K., Agarwal, S., Pratap, A., & Meyarivan, T. (2002). A fast and elitist multi-objective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2), 182–197.
- Deb, K., & Datta, R. (2012). Hybrid evolutionary multi-objective optimization and analysis of machining operations. *Engineering Optimization*, 44(6), 685–706.
- Deb, K., Gupta, S., Daum, D., Branke, J., Mall, A., & Padmanabhan, D. (2009). Reliability-based optimization using evolutionary algorithms. *IEEE Transactions on Evolutionary Computation*, 13(5), 1054–1074.
- Deb, K., & Jain, S. (2003). Multi-speed gearbox design using multi-objective evolutionary algorithms. *Journal of Mechanical Design*, 125(3), 609–619.
- Deb, K., & Sindhya, K. (2008). Deciphering innovative principles for optimal electric brushless D.C. permanent magnet motor design. In *Proceedings of the world congress on computational intelligence (WCCI'08)* (pp. 2283–2290). Piscataway, NY: IEEE Press.

- Deb, K., & Srinivasan, A. (2006). Innovation: Innovating design principles through optimization. In *Proceedings of the 8th annual conference on genetic and evolutionary computation (GECCO'06)*, (pp. 1629–1636). New York: ACM.
- Inselberg, A. (2009). *Parallel coordinates: Visual multi-dimensional geometry and its applications*. New York: Springer.
- Jeong, S., & Shimoyama, K. (2011). Review of data mining for multi-disciplinary design optimization. *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, 225(5), 469–479.
- Ng, A., Deb, K., & Dudas, C. (2009). Simulation-based innovation for production systems improvement: An industrial case study. In *The 3rd international Swedish production symposium* (pp. 278–286). Gothenburg, Sweden: The Swedish Production Academy.
- Obayashi, S., & Sasaki, D. (2003). Visualization and data mining of pareto solutions using self-organizing map. In *Evolutionary multi-criterion optimization* (pp. 796–809). Berlin, Heidelberg: Springer.
- Oyama, A., Nonomura, T., & Fujii, K. (2010). Data mining of pareto-optimal transonic airfoil shapes using proper orthogonal decomposition. *Journal of Aircraft*, 47(5), 1756–1762.
- Pryke, A., Mostaghim, S., & Nazemi, A. (2007). Heatmap visualization of population based multi objective algorithms. In *Evolutionary multi-criterion optimization* (pp. 361–375). Berlin, Heidelberg: Springer.
- Sardinas, R., Santana, M., & Brindis, E. (2006). Genetic algorithm-based multi-objective optimization of cutting parameters in turning processes. *Engineering Applications of Artificial Intelligence*, 19(2), 127–133.
- Sterman, J. D. (2000). *Business dynamics: Systems thinking and modeling for a complex world*. Irwin/McGraw-Hill Boston: McGraw-Hill.
- Sterman, J. D. (2001). Sterman models. Available from: [www.mhhe.com/business/opscl/sterman/models.mhtml](http://www.mhhe.com/business/opscl/sterman/models.mhtml).
- Sterman, J. D. (2006). Operational and behavioral causes of supply chain instability. In *The bullwhip effect in supply chains: A review of methods, components and cases* (pp. 17–56). New York: Palgrave Macmillan.
- Sugimura, K., Obayashi, S., & Jeong, S. (2009). A new design method based on cooperative data mining from multi-objective design space. *Journal of Computational Science and Technology*, 3(1), 287–302.
- Sugimura, K., Obayashi, S., & Jeong, S. (2010). Multi-objective optimization and design rule mining for an aerodynamically efficient and stable centrifugal impeller with a vaned diffuser. *Engineering Optimization*, 42(3), 271–293.