<http://www.diva-portal.org>

Postprint

This is the accepted version of a paper published in *Expert systems with applications*. This paper has been peer-reviewed but does not include the final publisher proof-corrections or journal pagination.

Citation for the original published paper (version of record):

Bandaru, S., Ng, A H., Deb, K. (2017)

Data mining methods for knowledge discovery in multi-objective optimization: Part A - Survey.

Expert systems with applications, 70: 139-159

<https://doi.org/10.1016/j.eswa.2016.10.015>

Access to the published version may require subscription.

N.B. When citing this work, cite the original published paper.

This is an article distributed under the terms of Creative Commons Attribution-Non-Commercial No Derivatives license: <https://creativecommons.org/licenses/by-nc-nd/4.0/legalcode>

Permanent link to this version:

<http://urn.kb.se/resolve?urn=urn:nbn:se:his:diva-13267>

Data Mining Methods for Knowledge Discovery in Multi-Objective Optimization: Part A - Survey

Sunith Bandaru^{a,*}, Amos H. C. Ng^a, Kalyanmoy Deb^b

^a*School of Engineering Science, University of Skövde, Skövde 541 28, Sweden*

^b*Department of Electrical and Computer Engineering, Michigan State University,
East Lansing, 428 S. Shaw Lane, 2120 EB, MI 48824, USA*

Abstract

Real-world optimization problems typically involve multiple objectives to be optimized simultaneously under multiple constraints and with respect to several variables. While multi-objective optimization itself can be a challenging task, equally difficult is the ability to make sense of the obtained solutions. In this two-part paper, we deal with data mining methods that can be applied to extract knowledge about multi-objective optimization problems from the solutions generated during optimization. This knowledge is expected to provide deeper insights about the problem to the decision maker, in addition to assisting the optimization process in future design iterations through an expert system. The current paper surveys several existing data mining methods and classifies them by methodology and type of knowledge discovered. Most of these methods come from the domain of exploratory data analysis and can be applied to any multivariate data. We specifically look at methods that can generate explicit knowledge in a machine-usable form. A framework for *knowledge-driven optimization* is proposed, which involves both *online* and *offline* elements of knowledge discovery. One of the conclusions of this survey is that while there are a number of data mining methods that can deal with data involving continuous variables, only a few ad hoc methods exist that can provide explicit knowledge when the variables involved are of a discrete nature. Part B of this paper proposes new techniques that can be used with such datasets and applies them to discrete variable multi-objective problems related to production systems.

Keywords: Data mining, multi-objective optimization, descriptive statistics, visual data mining, machine learning, knowledge-driven optimization

1. Introduction

In optimization problems involving multiple objectives, when the minimization or maximization of one of the objectives conflicts with the simultaneous minimization or maximization of any of the other objectives, a trade-off exists in the solution space

*Corresponding author

Email addresses: sunith.bandaru@his.se (Sunith Bandaru), amos.ng@his.se (Amos H. C. Ng), kdeb@egr.msu.edu (Kalyanmoy Deb)

Preprint submitted to *Expert Systems with Applications*

October 25, 2016

and, hence, no one solution can optimize all the objectives. Rather, multiple solutions are possible, each of which is better than all the others in at least one of the objectives. Thus, only a partial order exists among the solutions. The manifold containing these solutions is termed as the Pareto-optimal front and solutions on it are referred to as Pareto-optimal solutions (Miettinen, 1999). Before the advent of multi-objective optimization algorithms, the usual approach to solving nonlinear multi-objective optimization problems was to define a *scalarizing function*. A scalarizing function combines all the objectives to form a single function that can be optimized using single-objective numerical optimization techniques. The resultant solution represents a compromise between all the objectives. The most common type of scalarization is the weighted sum function, in which each objective is multiplied with a weight factor and then added together. Such scalarization requires some form of prior knowledge about the expected solution and, hence, the associated methods are referred to as *a priori* techniques. Other *a priori* approaches, such as transforming all but one of the objectives into constraints and ordering the objectives by relative importance (lexicographic ordering), were also popular (Miettinen, 1999). The drawbacks of such ad hoc methods were quickly noticed by many (Deb, 2001; Fleming et al., 2005), which led to research into the development of population-based metaheuristics that utilize the concept of Pareto-dominance and niching to drive candidate solutions towards the Pareto-optimal front. Evolutionary algorithms, mainly genetic algorithms and evolutionary strategy, were already popular for single objective optimization and this trend continued with multi-objective evolutionary algorithms. Over the past 30 years, many other selection and variation mechanisms have been developed, some more successful than others (Coello Coello, 1999).

1.1. Data Mining and Multi-Objective Optimization

The availability of multiple trade-off solutions opened up many research areas in the domain of multi-objective optimization. Primarily, it expanded the scope of multi-criteria decision-making (MCDM) from *a priori* techniques to include *a posteriori* and interactive methods. *A posteriori* techniques consider preferences specified by the decision maker *after* a set of Pareto-optimal solutions are obtained, whereas interactive techniques bring the decision maker into the search process (Shin & Ravindran, 1991). More recently, with the increased accessibility of computing power and the demonstrated potential of data mining methods, interest has risen in the analysis of Pareto-optimal solutions to gain important knowledge regarding the design or system being optimized. The Pareto-optimal front is an $m-1$ dimensional slice in the m dimensional objective space (provided none of the objectives are redundant). Unarguably, therefore, solutions that lie on the Pareto-optimal front are special and may possess certain properties that make the design or system to operate in an optimal manner (Deb & Srinivasan, 2006). Knowledge of such properties will help a user to better understand the optimal behavior in relation to the physics of the problem. Additionally, the user can gain insights into how a completely new optimally performing solution can be constructed simply by complying with those properties. Through the use of an expert system, such knowledge can also be used to computationally aid future optimization scenarios of a similar nature. Data mining methods accomplish this task of extracting useful knowledge from multivariate data and therefore can be applied to the dataset of Pareto-optimal solutions obtained through optimization. Data mining can also be applied to the entire set of feasible solutions in order to understand the structure of the objective space and its relation to the decision

space. When a preferred set of solutions is made available by a decision maker, data mining can also yield knowledge specific to the region of interest. Much of the research at this intersection of data mining and multi-objective optimization has been focused around the representation of the extracted knowledge, which can be implicit or explicit.

1.2. *Implicit vs. Explicit Knowledge Representation*

An appropriate knowledge representation is an essential part of all knowledge discovery tasks. Most representations can be categorized into two broad types, implicit and explicit (Nonaka & Takeuchi, 1995). An implicit representation is one that does not have a formal notation and, hence, the associated knowledge cannot be articulated or transferred unambiguously (Meyer & Sugiyama, 2007). The interpretation of such knowledge can also be subjective and may require the user to have specific experience. Nevertheless, methods that use an implicit representation are often the primary choice for understanding and summarizing the data at hand. For example, graphical methods convey knowledge in an implicit form, but are still used to gain a quick sense of the data. On the other hand, explicit representation uses a well-defined mathematical notation that enables the extracted knowledge to be expressed completely and concisely (Faucher et al., 2008). For the same reason, explicit knowledge can be stored efficiently in an expert system and shared easily. A distinctive advantage of an explicit representation over the implicit kind is that the knowledge can be processed automatically within a computer program. For example, explicit knowledge coming from two different expert systems can be combined programmatically (Hatzilygeroudis & Prentzas, 2004), whereas human intervention would be required to carry out such an operation on implicit knowledge. Thus, explicit representation is more practical in the context of online knowledge-driven optimization discussed later in Section 4.1.

Data mining is a vast area of research and there is an abundance of methods and techniques that can generate implicit and explicit knowledge (Witten et al., 2011). In this paper, we look at conventional visualization, statistical and machine learning techniques that can be applied to data generated during multi-objective optimization. Most of these methods have been developed for generic data analysis tasks and several variants exist for specific applications (Liao et al., 2012). However, here they are discussed in terms of their suitability for extracting knowledge from solutions generated through multi-objective optimization. In the next section, we highlight certain characteristics of multi-objective optimization datasets that warrant this parallel field of study. In Section 3, we classify and describe each method, while highlighting key differences and shortcomings. Finally, in Section 4, we discuss the current challenges and future research directions concerning knowledge discovery in multi-objective optimization.

2. Characteristics of Multi-Objective Optimization Datasets

For the sake of completeness, we begin with the standard form of a multi-objective optimization (MOO) problem and lay down some basic notations. A MOO problem is given by,

$$\begin{array}{ll} \text{Minimize} & \mathbf{F}(\mathbf{x}) = \{f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_m(\mathbf{x})\} \\ \text{Subject to} & \mathbf{x} \in S \end{array} \quad (1)$$

Figure 1: Mapping of solutions from the two-dimensional decision space to the two-dimensional objective space. The dominance relations between the four arbitrary solutions are $\mathbf{a} \prec \mathbf{b}$, $\mathbf{c} \prec \mathbf{b}$, $\mathbf{d} \prec \mathbf{a}$, $\mathbf{d} \prec \mathbf{b}$, $\mathbf{d} \prec \mathbf{c}$ and $\mathbf{a} \parallel \mathbf{c}$.

where $f_i : \mathbb{R}^n \rightarrow \mathbb{R}$ are m (≥ 2) conflicting objectives that have to be simultaneously minimized and the variable vector $\mathbf{x} = [x_1, x_2, \dots, x_n]$ belongs to the non-empty feasible region $S \subset \mathbb{R}^n$. The feasible region is formed by the constraints of the problem (which also include the bounds on the variables). A variable vector $\mathbf{x}_1$ is said to *dominate* $\mathbf{x}_2$ and is denoted as $\mathbf{x}_1 \prec \mathbf{x}_2$ if and only if the following conditions are satisfied:

1. $f_i(\mathbf{x}_1) \leq f_i(\mathbf{x}_2) \forall i \in \{1, 2, \dots, m\}$,
2. and $\exists j \in \{1, 2, \dots, m\}$ such that $f_j(\mathbf{x}_1) < f_j(\mathbf{x}_2)$.

If only the first of these conditions is satisfied, then $\mathbf{x}_1$ is said to *weakly dominate* $\mathbf{x}_2$ and is denoted as $\mathbf{x}_1 \preceq \mathbf{x}_2$. If neither $\mathbf{x}_1 \preceq \mathbf{x}_2$ nor $\mathbf{x}_2 \preceq \mathbf{x}_1$, then $\mathbf{x}_1$ and $\mathbf{x}_2$ are said to be non-dominated with respect to each other and denoted as $\mathbf{x}_1 \parallel \mathbf{x}_2$. A vector $\mathbf{x}^* \in S$ is said to be Pareto-optimal, if there does not exist any $\mathbf{x} \in S$ such that $\mathbf{x} \prec \mathbf{x}^*$. The set of all such $\mathbf{x}^*$ (which are non-dominated with respect to each other) is referred to as the Pareto-optimal set. The projection of the Pareto-optimal set in the objective space, $\mathbf{F}(\mathbf{x}^*) \forall \mathbf{x}^*$, is called the Pareto-optimal front.

The n dimensional space formed by the variables is called the *decision space*, while the m dimensional space formed by the objectives is called the *objective space*. As an example, Figure 1 shows the mapping of several solutions for a problem with $n = 2$ variables and $m = 2$ conflicting objectives that are to be minimized. According to the aforementioned definition of dominance, $\mathbf{a} \prec \mathbf{b}$, $\mathbf{c} \prec \mathbf{b}$, $\mathbf{d} \prec \mathbf{a}$, $\mathbf{d} \prec \mathbf{b}$, $\mathbf{d} \prec \mathbf{c}$ and $\mathbf{a} \parallel \mathbf{c}$. The curve $\mathbf{ef}$ represents the Pareto-optimal front. All points on this front are Pareto-optimal solutions because no solution $\mathbf{x} \in S$ exists that dominates any part of the front.

Depending on the selection strategy, most population-based multi-objective optimizers can be classified into (i) Pareto-dominance-based, (ii) decomposition-based, (iii) indicator-based. Pareto-dominance based methods work by dividing the population into non-dominated levels (also known as ranks or fronts) and selecting high rank solutions in successive generations. Niching is usually employed to preserve diversity among the solutions. Examples of such methods include MOGA, NSGA, NPGA, PAES, NSGA-II and

SPEA2 (Deb, 2001; Talbi, 2009). Decomposition-based methods divide the original problem into a set of sub-problems which are solved simultaneously in a collaborative manner. Each sub-problem is usually an aggregation of the objectives with uniformly distributed weight vectors. Popular methods include MOGLS (Jaszkiewicz, 2002), MOEA/D (Zhang & Li, 2007) and the more recent NSGA-III (Deb & Jain, 2014). Finally, indicator-based methods work by establishing a complete order among the solutions, using a single scalar metric like hypervolume, ϵ and $R2$. HypE (Bader & Zitzler, 2011), IBEA (Zitzler & Künzli, 2004) and SMS-EMOA (Beume et al., 2007) are popular in this category. When many objectives are involved (typically $m > 10$), the proportion of non-dominated solutions in the initial population grows exponentially with m , rendering Pareto-dominance-based methods ineffective, due to the absence of selection pressure. This is known as the curse of dimensionality. The other two selection strategies are more popular in these scenarios, along with other non-direct methods, such as modification of Pareto-dominance and objective reduction.

Multi-objective optimizers can also be classified on the basis of the variation mechanism. Typically, these mechanisms are advocated as nature or bio-inspired metaheuristics. Some popular categories are evolutionary (genetic algorithms, evolutionary strategy, differential evolution), swarm-based methods (particle swarm optimization, cuckoo search) and colony-based algorithms (ant colony, bee colony). A survey can be found in Boussaïd et al. (2013). Most of these metaheuristic algorithms can be adapted to solve MOO problems, using one of the selection strategies discussed above (Jones et al., 2002).

Most multi-objective optimizers start with a population of randomly generated solutions or individuals, which undergo variation and selection iteratively, until a specified number of generations or performance target or function evaluations is reached. All solutions generated and evaluated in the process may hold vital knowledge about the problem. In this paper, we refer to these solutions as the *MOO dataset*. Each entry of this dataset is a row of decision vector components x_i and the corresponding objective values $f_i(\mathbf{x})$, i.e. $x_1 \ x_2 \ \dots \ x_n \ f_1 \ f_2 \ \dots \ f_m$. Sometimes, problem parameters¹, constraint function values, and other auxiliary information, such as ranks of the solutions or their distance from a reference point, etc., may also be included. The MOO dataset can be divided into feasible and infeasible sets, as shown in Figure 2, the former can be further sub-divided into different ranks or by generations. Data mining methods should be able to use all or parts of this MOO dataset to discover knowledge. For example, knowledge derived from the Pareto-optimal solutions can be useful in understanding the optimal behavior of the system or process being optimized. Similarly, the use of data mining methods on the infeasible set can help an expert identify constraints that are difficult to satisfy or just redundant. When applied to feasible solutions, data mining can reveal the overall structure of the objective space, i.e., associate regions in the objective space to corresponding regions in the decision space. If *a posteriori* decision-making is involved, then the set of solutions preferred by a domain expert can be provided to data mining methods to determine how they differ from the rest of the solutions. Any discovered knowledge can also be used with optimization to guide the search towards interesting

¹In optimization literature, the terms ‘variables’ and ‘parameters’ are used interchangeably. However, in this paper, problem parameters refer to quantities of the problem that remain unaltered during an optimization run, but can be varied by the user between runs. These are, in turn, different from algorithmic parameters, by which we refer to the parameters of the optimizer.

Figure 2: A typical MOO dataset consists of n variable values, m objective function values, problem parameters and other auxiliary values. The dataset can be divided into feasible and infeasible solutions which can further be grouped by generation numbers. Feasible solutions can also be grouped by their ranks.

regions of the objective space. The concept of such knowledge-driven optimization is discussed in Section 4.1.

Intuitively, the application of data mining to MOO datasets may seem straightforward and akin to any other knowledge extraction procedure. However, the task comes with caveats which dictate some desirable properties for the data mining methods:

1. **Presence of two separate spaces:** Solutions from MOO are usually visualized in the objective space, whereas knowledge extraction is usually carried out in the decision space, because only the decision variables can be directly manipulated by an optimization algorithm. However, clusters of solutions in the objective space may not correspond to clusters in the decision space. Therefore, ignoring one of the spaces or applying the same methods irrespective of the spaces are not the right approaches to handling MOO datasets.
2. **Involvement of a decision maker:** Practical MOO problems end with the selection of one or more solutions from the Pareto-optimal front. Decision makers are usually concerned with the objective space (Miettinen, 1999). When a set of preferred solutions is provided, the task of data mining is not just to discover knowledge pertaining to the preferred set, but also to specifically identify properties that distinguish the preferred set from the rest of the solutions. Thus, the methods should support some kind of supervised learning, when such supervision is available. Also, for a decision maker, clusters of solutions in the objective space are more important than those in the decision space (Bandaru, 2013). Therefore, the data mining methods should be able to use neighborhood information from the objective space while representing knowledge in the decision space.
3. **Representation of knowledge:** Many data mining methods represent knowledge in an implicit form, even in the presence of supervision. For example, when a preferred set is available, classification algorithms, such as support vector machines and neural networks, can only generate black-box models that mimic the decision maker’s preferences on new solutions. However, they cannot explain the

Table 1: Desirable forms of explicit knowledge for different data-types.

Variable type	Explicit form	Examples
Continuous	Analytical relationships	$x_1 + x_2^2 = 4$
	Statistical measures	Mean, Pearson’s r , etc.
Discrete and Ordinal	Decision rules	if $x_1 > 4 \wedge x_2 < 3$ then $x_3 \neq 2$
	Statistical measures	Median, Spearman’s ρ , etc.
Nominal	Patterns/Association rules	$\langle x_1 = \text{Male}, x_2 = \text{Buy} \rangle$
	Statistical measures	Mode, Cramér’s V , etc.

preferences in terms of the features of the dataset. Such knowledge, while useful for understanding the data at hand, cannot be easily applied to a new problem. Moreover, as discussed above, it is difficult to store, retrieve and transfer such implicit knowledge. Therefore, it is desirable to have the data mining process generate knowledge in an explicit form.

4. **Presence of different variable types:** Optimization problems involve three types of variables, (i) continuous, (ii) discrete and ordinal², and (iii) nominal. Data mining methods should preferably be able to handle all data-types. The form of the derived knowledge will depend on the type of the variables involved. For explicit knowledge discovery, Table 1 shows some desired forms for different variable types. *Analytical relationships* represent interdependence between two or more continuous variables. However, they are not suitable for discrete and ordinal variables, when the values between different possible options are not defined. *Decision rules*, i.e., conditional statements combined with logical operators, are more suitable for discrete and ordinal data-types. Nominal variables are coded using arbitrary numbers in optimization algorithms. The notion of distance and neighborhood does not exist for such variables. Therefore, neither analytical relationships nor decision rules can capture knowledge associated with them. *Patterns*, i.e., a repetitive sequence of values, are a more pragmatic approach to knowledge representation for nominal variables. Often, patterns can be expressed as *association rules*, which, like decision rules, take an ‘if-then’ form. Some statistical measures can also yield explicit knowledge about MOO datasets. These are discussed in Section 3.1.
5. **Presence of problem parameters:** MOO problems typically involve many problem parameters that are not altered during optimization. However, in practice, the user may want to perturb these parameters to understand how they affect the Pareto-optimal solutions. The inclusion of problem parameters in the MOO dataset (as shown in Figure 2) can reveal *higher-level* knowledge about the problem, such as sensitivity of the Pareto-optimal solutions. We discuss this aspect further in Section 3.3.2.

3. Knowledge Discovery from MOO Datasets

In this section, we classify different knowledge discovery methods based on the type of knowledge extracted from the MOO dataset and the form in which it is obtained.

²In optimization problems, there is a small distinction between discrete and ordinal variables, which is addressed in Section 4. With respect to knowledge representation, they can be treated alike.

Table 2: Commonly used descriptive statistics: Measures of Central Tendency (CT), Variability (V), Distribution Shape (DS) and Correlation/Association (C/A).

Type	Measures	Formula	Remarks
CT	Mean	$\bar{y} = \sum y_i / N$	for non-skewed $\mathbf{y}$
	Median	$\tilde{y} = Q_2 = 50 \text{ \%ile}$	for skewed or ordinal $\mathbf{y}$
	Mode	Most frequent y_i	for nominal $\mathbf{y}$
V	Standard deviation	$s_y = \sqrt{\frac{1}{N} \sum (y_i - \bar{y})^2}$	for continuous or ordinal $\mathbf{y}$ Variance, $\sigma_y = s_y^2$
	Range	$[\min(\mathbf{y}), \max(\mathbf{y})]$	for continuous or ordinal $\mathbf{y}$
	Quartiles	$Q_1 = 25 \text{ \%ile}, Q_3 = 75 \text{ \%ile}$	Interquartile range = $Q_3 - Q_1$
DS	Skewness	$g_y = \frac{\sum (y_i - \bar{y})^3}{N s_y^3}$	for continuous or ordinal $\mathbf{y}$
	Kurtosis	$\kappa_y = \frac{\sum (y_i - \bar{y})^4}{N s_y^4} - 3$	
C/A	Pearson r	$r_{yz} = \frac{\sum y_i z_i - N \bar{y} \bar{z}}{(N-1) s_y s_z}$	for non-skewed continuous $\mathbf{y}, \mathbf{z}$
	Spearman ρ	$\rho_{yz} = 1 - \frac{6 \sum (y_i - z_i)^2}{N(N^2 - 1)}$	for skewed or ordinal $\mathbf{y}, \mathbf{z}$
	Kendall τ	$\tau_{a_{yz}} = \frac{(N_c - N_d)}{N(N-1)/2}$	N_c = No. of concordant pairs, i.e. $y_i < y_j \leftrightarrow z_i < z_j$ or $y_i > y_j \leftrightarrow z_i > z_j$
	Goodman & Kruskal γ	$\gamma_{yz} = \frac{(N_c - N_d)}{(N_c + N_d)}$	N_d = No. of discordant pairs, i.e. $y_i < y_j \leftrightarrow z_i > z_j$ or $y_i > y_j \leftrightarrow z_i < z_j$
	Cramér V	$V_{yz} = \sqrt{\frac{\chi^2 / N}{\min(n_y, n_z) - 1}}$	for nominal $\mathbf{y}, \mathbf{z}$ with n_y and n_z
	Contingency Coefficient	$C_{yz} = \sqrt{\frac{\chi^2 / N}{1 + \chi^2 / N}}$	levels respectively. χ^2 = chi-squared statistic

As mentioned previously, knowledge can either be implicit or explicit. In Section 3.1, we begin with descriptive statistics which are used to obtain basic quantitative information about the data in the form of numbers (hence, explicit form). All visual data mining methods lead to implicit knowledge. These are discussed in Section 3.2. Due to the abundance of these methods in literature, they are further classified as graphical, clustering-based and manifold learning methods. Machine learning methods can generate both implicit and explicit knowledge and these are discussed in Section 3.3. We deal with supervised and unsupervised methods separately.

3.1. Descriptive Statistics

Descriptive statistics simply refers to numbers that summarize data. They can be of four types, (i) measures of central tendency, (ii) measures of variability or dispersion, (iii) measures of distribution shape, and (iv) measures of correlation/association. Together, they give the user a quick and quantitative feel of the data. Almost all empirical studies in the field of numerical optimization and data mining use these measures for the very reason that they convey information in a concise manner. Naturally, their importance in capturing knowledge cannot be ignored. In Table 2, we enumerate some of the most common measures, with some pointers to their appropriate usage. Since these measures can be applied to both variable (x) and objective (f) values, we avoid any presumptions by using y and z to denote statistical random variables.

Measures of central tendency and variability are univariate descriptors. The former is a score that summarizes the location of a distribution, whereas the latter specifies how the data points differ from this score. Both measures are quite commonly used and therefore need no introduction. The shapes of the distributions can be described using skewness and kurtosis. The former quantifies the asymmetry of the distribution, with respect to the mean, and the latter measures the degree of concentration of the data points at the mean of the distribution (also called *peakedness* of the distribution). Measures of correlation and association are bivariate descriptors. They are normalized scores that describe how two random variables are related to each other. This is characterized by the *type* and the *strength* of the relationship between them. Most correlation measures can only quantify a linear or monotonic relationship (both positive and negative types) between two random variables. The absolute magnitude of the measure indicates the strength of the relationship. We discuss some common correlation/association measures in the next two paragraphs.

Different correlation measures have been proposed for different variable types. The Pearson r (Bennett & Fisher, 1995) is the most popular correlation measure for continuous random variables. Several variants of the Pearson's r exist, such as weighted correlation coefficient and Pearson's correlation distance. For ordinal variables, the term 'rank correlation' is often used. Two popular measures for rank correlation are Spearman's ρ and Kendall's τ (Kendall, 1948). The formula for Spearman's ρ (shown in Table 2) can be derived from the Pearson r under the assumption of no rank ties, i.e., $y_i \neq y_j \forall i \neq j$ and $z_i \neq z_j \forall i \neq j$. Even with a few tied ranks, the Spearman ρ provides a good approximation of correlation. Unlike Pearson's r and Spearman's ρ which use a variance based approach, Kendall's τ (Kendall, 1948) uses a probability based approach. It is proportional to the difference between the number of pairs of observations that are positively (concordant, N_c) and negatively (discordant, N_d) correlated. When not accounting for rank ties, the total number of possible pairs is $N(N-1)/2$, as seen in the denominator of Kendall τ_a in Table 2. Kendall's τ_b (Agresti, 2010) adjusts this measure for tied ranks by using $\sqrt{(N_c + N_d + T_y)(N_c + N_d + T_z)}$ in the denominator instead of $N(N-1)/2$. Here, T_y and T_z represent the number of pairs of tied ranks for random variables $\mathbf{y}$ and $\mathbf{z}$ respectively. Another variant, Kendall τ_c (Stuart, 1953), considers the case when $\mathbf{y}$ and $\mathbf{z}$ have an unequal number of levels, represented as $n_y \neq n_z$. It uses the denominator $N^2(\min(n_y, n_z) - 1)/(2 \min(n_y, n_z))$. A third rank correlation measure, known as the Goodman & Kruskal's γ (Goodman & Kruskal, 1954), is used when the number of ties is very small and can be ignored altogether. Here, the denominator $(N_c + N_d)$ is used, as shown in Table 2. Note that any rank correlation measure can be applied to continuous variables, by replacing the values with their ranks. All the above correlation/association measures fall between -1 and $+1$, corresponding to perfect negative and perfect positive correlation respectively.

The association between nominal variables $\mathbf{y}$ and $\mathbf{z}$ with n_y and n_z levels, respectively, can be measured using Cramér's V (Cramér, 1999), shown in Table 2. This measure is based on the Pearson's chi-squared statistic given by

$$\chi^2 = \sum_{p=1}^{n_y} \sum_{q=1}^{n_z} \frac{(N_{pq} - \hat{N}_{pq})^2}{\hat{N}_{pq}}.$$

Here, N_{pq} is the observed frequency of samples for which both $y = p$ and $z = q$, and $\hat{N}_{pq}$

is the expected frequency of the same, i.e., $\hat{N}_{pq} = N_{p*}N_{*q}/N$, where N_{p*} is the number of samples with $y = p$ and N_{*q} is the number of samples with $z = q$. When at least one of the variables is dichotomous, i.e., either $n_y = 2$ or $n_z = 2$, Cramér's V reduces to the ϕ coefficient, which is a popular measure of association for two binary variables. Given by $\phi = \chi/\sqrt{N}$, this measure is equivalent to the Pearson r . Often, the association is expressed as $\phi^2 = \chi^2/N$ instead of ϕ . A variation of the ϕ coefficient, suggested by Karl Pearson, is the contingency coefficient C , also shown in Table 2. A comparative discussion on these and other measures like Tschuprow's T and Guttman's λ can be found in Goodman & Kruskal (1954). Since there is no natural order among nominal variables values, all measures of association are conventionally measured between 0 and 1, corresponding to no association and complete association respectively.

Nonlinear correlation measures, such as the coefficient of determination (R^2), require a predefined model to which the samples are fit and, therefore, are not used to summarize data. They are more commonly used in regression analysis as a measure of goodness of fit. For a linear model, $R^2 = r^2$. When the model consists of more than one dependent variable, R^2 is called the multiple correlation coefficient.

3.2. Visual Data Mining Methods

Visual data mining methods rely on the visual assessment of MOO datasets by the user, in order to gain implicit knowledge. Much research in visual data mining concerns depicting multidimensional data in a human-perceivable manner. Such multidimensional multivariate (MDMV) techniques can visually reveal correlations. In some cases, this implicit knowledge can be converted to an explicit form through further processing. The following graphical methods are customarily used for a preliminary analysis of generic multivariate datasets:

1. Scatter plots: 2D and 3D plots of data points (solutions) for different variable combinations (often shown as a matrix of plots) can reveal correlations visually.
2. Pie charts and bar plots: Used in a variety of ways, such as, the proportion of solutions of different ranks, proportion of variable values in different intervals, etc.
3. Histograms: Distributions of values over a set of solutions for individual variables.
4. Box plots (Tukey, 1977): Graphical representation of the median and the quartiles of variable values. Bag plots (Rousseeuw et al., 1999) are bivariate versions of box plots in which the quartiles of two different variables form the two adjacent sides of the box.
5. Violin (Hintze & Nelson, 1998) and bean plots (Kampstra, 2008): Similar to box plots, but instead of using a rectangular box of constant width to represent the interquartile range, they use a symmetrical shape of varying width proportional to the density of points at different variable values. The shape resembles that of a violin for bimodal distribution of values.
6. Spider/radar/star/polar plots: The range of each variable is normalized and represented by a line joining the center to each vertex of a regular polygon. The number of vertices is equal to the number of variables. Solutions are represented by polygons (within this regular polygon) formed by connecting corresponding variable values. Naturally, at least three variables are required to be able to use these plots.
7. Parallel coordinate plots (Inselberg, 1985): The range of each variable is normalized (say between $[0, 1]$) and represented by a vertical line. All these vertical lines are

drawn next to each other with equal spacing. Solutions can now be represented by polylines formed by joining the corresponding variable values between adjacent vertical lines. Parallel coordinate plots are often used to visualize the solutions in many-objective optimization.

8. Biplots (Gabriel, 1971): Multivariate analogues to scatter plots where the axes are the first two (in case of 2D biplots) principal directions of the data obtained using Principal Component Analysis. By expressing the variables as a linear combination of the principal directions, they are shown as position vectors on the biplot. Similarly, the transformed solutions are shown as points. Biplots have been used with MOO datasets in Lewandowski & Granat (1991).
9. Conditioning plots/coplots (Cleveland, 1993): The original set of solutions is divided into subsets based on the values of a chosen variable called the *conditional*. Next, each subset is shown as a scatter plot by choosing 2 (in case of 2D coplots) or 3 (for 3D coplots) variables from the remaining variables. These plots are useful for studying the affect of conditional variables on the variables chosen for the scatter plots.
10. Glyph plots: Solutions are represented using shapes/icons whose features are determined by the variable values. For example, in Chernoff faces (Chernoff, 1973), different features that define the shape of a human face are controlled by variable values. Thus, different solutions result in different looking faces. Other types of glyph plots are, Andrew's curves (Andrews, 1972), star or circle glyphs (Siegel et al., 1972) and stick figures (Grinstein et al., 1989). *Harmonious houses* (Korhonen, 1991) inspired by Chernoff faces have been used for MOO datasets.
11. Mosaic and spine plots (Friendly, 2002): A typical bar plot can show different variable values for a single solution as bars placed next to each other. In spine plots, the bars are stacked on top of each other, so that multiple solutions can be shown next to each other. Mosaic plots are a variation of spine plots where the lengths of all stacked bars are normalized to fit in a rectangular region. They are generally used for categorical variables (ordinal and nominal).
12. Treemaps (Shneiderman, 1992): Variation of mosaic plot that is suitable for hierarchical data. In MOO datasets, hierarchy can be established using the generation number or solution ranks. All solutions at the same level of hierarchy are shown as a mosaic plot. In treemaps, mosaic plots of a group of lower-level hierarchies are embedded within the mosaic plot of a higher-level hierarchy.
13. Dimensional stacking (LeBlanc et al., 1990): Starting with a rectangular region, two variables from the dataset are chosen to represent the two dimensions. Both dimensions are discretized to divide the rectangular region into smaller rectangles. Again, two new variables are chosen from the dataset to represent the two dimensions of the smaller rectangles, which are again discretized. The process is repeated until all the variables are assigned. Thus, each cell of the resultant grid represents a bin. Colors are used to show the number of points in each bin.
14. Radial Coordinate Visualization, RadViz (Hoffman et al., 1997): A circular coordinate system (called barycentric system) is formed by creating uniformly spaced vertices on a circle. The number of vertices is equal to the number of variables. Solutions are mapped within the circle by calculating the equilibrium position of an imaginary particle connected by springs to all the vertices. The stiffness of

each spring is equal to the normalized value of the corresponding variable for the solution being mapped. See (Walker et al., 2012) for an example that uses MOO datasets.

Note that even though we use variables for describing the above methods, they can be similarly applied on the objectives. Detailed descriptions of these and other such generic methods and their variants can be found in survey articles related to visual data mining, such as (Keim, 2002; de Oliveira & Levkowitz, 2003; Hoffman & Grinstein, 2002; Chan, 2006). Methods and procedures that have been developed specifically for visual data mining of MOO datasets are discussed in the following sections. Due to their abundance, we further group them as (i) graphical methods, (ii) clustering-based methods, and (iii) manifold learning methods.

3.2.1. Graphical Visualization Methods

As mentioned previously, the objective space of a MOO problem is considered to be more important to the decision maker than the decision space. Therefore, a class of visual data mining methods that specifically deal with the visualization of the objective space has emerged, particularly from the domain of MCDM:

1. Distance and distribution charts (Ang et al., 2002): Distance charts are line plots that show the distances of different non-dominated solutions in the objective space from their nearest point on the Pareto-optimal front. On the other hand, distribution charts show their distance from the nearest solution within the same non-dominated set. Figure 3 illustrates how these charts can be interpreted.
2. Value paths (Geoffrion et al., 1972): They are similar to parallel coordinate plots, except that the vertical lines are used to represent the solutions and the polylines to represent individual objectives. Figure 4 illustrates value paths for five objectives over 10 solutions. In Thiele et al. (2009) and Klamroth & Miettinen (2008), value paths have been used for decision making.
3. Star coordinate system (Mañas, 1982): It is similar to spider plots, except that the center of the regular polygon represents the *ideal point* and the polygon itself represents the *nadir point*. An ideal point refers to a hypothetical point constructed using the best objective values from all non-dominated solutions in the MOO dataset. A nadir point is its exact opposite, meaning that it is constructed from the worst objective values of the non-dominated solutions. These points are shown schematically in Figure 1. In the star coordinate system, solutions that have smaller polygonal areas are better as they are closer to the ideal point. Figure 5 shows an example with six objectives.
4. Petal diagrams (Angehrn, 1991): They are similar to star coordinate systems, except that the objective values are represented by the radii of equi-angle sectors. A limitation of petal diagrams is that only a single solution can be shown per diagram. Figure 6 shows the petal diagrams for the solutions shown in Figure 5.
5. Pareto race (Korhonen & Wallenius, 1988): It allows the decision maker to traverse through the solutions via keyboard controls and visualize changes in the objective values using dynamic bar graphs which change the length of the bars depending on the position of the on-screen pointer.
6. Interactive decision maps (Lotov et al., 2004): These are animated two-dimensional slices of the objective space showing approximations of the Edgeworth-Pareto hull

Figure 3: The distance chart shows that solutions 4 and 9 are further away from the Pareto-optimal front and the distribution chart shows that solutions 2 and 5 are isolated.

Figure 4: Value paths for five objectives. The performance of the solutions on individual objectives is clearly seen. For example, solution 4 has the highest value for f_4 , solution 2 has the lowest value for f_3 , etc.

(see Figure 7) instead of the Pareto-optimal solutions. The slices are obtained by fixing the values of all objectives except those being visualized. The fixed values can be varied by the decision maker using sliders, as shown in Figure 7. Application studies involving the use of interactive decision maps can be found in Lotov et al. (2004); Efremov et al. (2009).

7. Pareto shells (Walker et al., 2012): The ranks obtained by non-dominated sorting are referred to as *shells* here. Solutions in different shells are arranged in columns, as shown in Figure 8, where the numbers and colors represent solution indices and average solution ranks. The latter is obtained by averaging ranks over all objectives. A directed graph is defined with the solutions as nodes and directed edges representing dominance relations (to solutions in the immediate next rank only).
8. Level Diagrams (Blasco et al., 2008): A level diagram is essentially a scatter plot of solutions showing one of the objectives versus the distance of the solutions from the ideal point. For visualizing m objectives, m level diagrams are required. Level diagrams can also be used for the decision variables in a similar manner. Figure 9 shows how the non-dominated solutions of a two-objective problem look on level diagrams.
9. Two-stage mapping (Koppen & Yoshida, 2007): This approach attempts to find a mapping from the m -objective space to a two-dimensional space such that the dominance relations and distances between the solutions are preserved as much as possible. The first stage maps only the non-dominated solutions on to the circumference of a quarter circle, whose radius is the average norm over all non-dominated solutions. The order of solutions along the circumference is optimized to minimize errors in mapping. In the second stage, each dominated solution is mapped to a position inside the quarter circle, again ensuring that the dominance relations are preserved as much as possible. Figure 10 shows a schematic explaining the final state of two-stage mapping process.

Figure 5: Representation of two solutions of a six-objective space in the star coordinate system. The thicker polygon is a better solution than the other in all objectives except f_6 .

Figure 6: Petal diagrams for the two solutions shown in Figure 5. Smaller petal sizes mean better objective values. The included angles for all petals are equal.

10. Hyperspace diagonal counting (Agrawal et al., 2004): In this approach, each objective is first discretized into several bins. The m objectives are then divided into two subsets, each of which is reduced to a single dimension by combining the bins of all corresponding objectives. Each bin combination in both the subsets is assigned an index value by diagonal counting. These indices form the x and y -axes of a three-dimensional plot. The count of solutions in the bin combinations at different (x, y) positions is plotted on the z -axis as a vertical bar. This method does not attempt to preserve the dominance relations. It is useful for assessing the distribution of points in the objective space. Figure 11 shows an example of bin combinations with four objectives.
11. Heatmaps (Pryke et al., 2007): Inspired from biological microarray data analysis, the heatmap visualization uses a color map on the ranges of variables and objectives, as shown in Figure 12. Hierarchical clustering is used to find the order in which solutions, variables and objectives appear in the heatmap. In Nazemi et al. (2008), solutions are sorted into ascending order of the first objective. The use of *seriated* heatmaps was proposed in Walker et al. (2013, 2012) where the solutions are ordered by a similarity measure that takes the per-objective ranks of the solutions into account. Columns corresponding to the objectives are also ordered in a similar manner. However, the cosine similarity is used for the variables.

The first use of feasible dominated solutions for knowledge discovery is seen in (Chichakly & Eppstein, 2013). The heatmaps of individual variables are visualized in the objective space, as shown in Figure 13 for variable t of a welded beam design optimization problem. Starting from different solutions on the non-dominated front, the variable of interest is perturbed incrementally (while keeping the other variables fixed, or *ceteris paribus*) to obtain a trace of objective values. These so called ‘*cp* lines’ are shown in white in Figure 13. The figure shows that while higher values of t are better in a Pareto sense, decreasing t even slightly for the expensive designs (Cost > \$25) gives a greater reduction in cost for only a minor increase in deflection.

Figure 7: Interactive decision map for a five objective problem. The first two objectives are shown on the axes and the third objective as gray contour lines. The last two objectives are set by the decision maker using the sliders. Taken from Lotov et al. (2004).

Figure 8: Visualization of MOO dataset as Pareto shells. As an example, solution 11 dominates solutions 5, 40, 20 and 25. Taken from Walker et al. (2012).

Figure 9: Generation of level diagrams illustrated for MOO dataset with two objectives.

12. **Prosection Method** (Tusar & Filipic, 2015): This is a visualization method for the projection of a section of four-dimensional objective vectors that preserves dominance relations, shape and distribution features for some of the Pareto-optimal solutions. The so called *prosection* is defined by a prosection plane $f_i f_j$, an angle φ about a given origin and a section width d . Essentially, this means that all solutions in the plane $f_i f_j$, within width d of a line going through the origin and oriented at φ from f_i , will be mapped to form a single dimension. The prosection method reduces the number of objectives by one and hence is most suitable for $m \leq 4$ objectives. Figure 14 shows how the prosection method combines two objectives into one by projection followed by rotation.

Detailed surveys of visualization methods in MCDM can be found in (Miettinen, 2003; Korhonen & Wallenius, 2008; Lotov & Miettinen, 2008; Miettinen, 2014). An often desired requirement for MCDM visualization is that the dominance relations between the solutions are preserved (Tusar & Filipic, 2015; Walker et al., 2013). However, as shown in Koppen & Yoshida (2007), such mapping does not generally exist and only Pareto shells

Figure 10: The two-stage mapping process for a MOO dataset with nine non-dominated solutions and three dominated solutions. The mapping attempts to capture most of the dominance relations, which are shown here by lines connecting the solutions.

Figure 11: Hyperspace diagonal counting for four objectives. All objectives are discretized into five bins. The combined bins of f_1 and f_2 and those of f_3 and f_4 are chosen as the axes for visualization. The vertical bars represent the number of solutions in each bin combination (bin count).

and the prosection method are able to achieve partial preservation (Tušar, 2014). Also, none of the methods, except level diagrams and heatmaps, can be extended to visualize the decision space when the number of variables is large.

Experimental comparisons on the effectiveness of various visualization methods in MCDM can be found in Walker et al. (2012); Gettinger et al. (2013); Taieb-Maimon Meirav et al. (2013); Walker et al. (2013); Tušar (2014).

3.2.2. Clustering-based Visualization Methods

Clustering methods are popular for finding hidden structures in multivariate datasets. Although, technically speaking, clustering is an unsupervised learning task, the clusters themselves are represented graphically (or less commonly, through a cluster-membership matrix (Xu & Wunsch, 2005)), which makes the knowledge implicit. Only through further processing, which makes use of these clusters, can explicit knowledge be obtained in terms of the features (or variables for MOO datasets). Clustering methods can either be partitional or hierarchical. The latter generates a nested series of cluster memberships with a varying number of clusters, whereas the former generates only one based on a prespecified or predetermined number of clusters (Jain et al., 1999).

1. *K*-means clustering (MacQueen, 1967), which is a popular partitional approach, was used in (Taboada & Coit, 2006, 2007) to cluster Pareto-optimal solutions in the normalized objective space to simplify the task of decision-making. The number of clusters is determined using silhouette plots (Rousseeuw, 1987). One representative solution from each cluster is chosen (usually, the one closest to the centroid) and presented to the decision maker for further analysis. If the decision maker can also provide preference orderings for the objectives, then a similar clustering procedure is used after filtering the solutions (Taboada et al., 2007; Taboada & Coit, 2008). The clear advantage either way is that there are fewer solutions to

Figure 12: An *unseriated* heatmap of variable and objective values. Taken from (Pryke et al., 2007).

Figure 13: Heatmap and *cp* lines of a variable of interest. Taken from (Chichakly & Eppstein, 2013).

Figure 14: Basic workings of the prosection method. Once a prosection is defined, solutions are projected and then rotated to form a single dimension.

be analyzed manually. Figure 15 shows the clusters obtained for a reliability optimization problem (redundancy allocation) presented in (Taboada & Coit, 2006). Once an interesting representative solution is identified, the solutions in the corresponding cluster are normalized and clustered again to generate a second set of representative solutions, as shown in Figure 16.

2. In Morse (1980), the author compares partitional and hierarchical clustering methods in the objective space and recommend the latter for the decision-making process because it does not require the number of clusters to be prespecified by the decision maker. With hierarchical clustering, the number of clusters can be chosen by visualizing the cluster memberships in the form of a *dendrogram*. A dendrogram offers different levels of clustering as shown in Figure 17 allowing one to clearly see the arrangement of clusters in the data.
3. In Jeong et al. (2003, 2005a), clustering is performed in the 90 dimensional decision space of a ‘turbine blade cooling passage’ shape optimization problem, involving the minimization of heat transferred to the blade. The obtained clusters are visualized by projecting the solutions onto the plane of the first two principal directions. After filtering the clusters based on the objective values, the chosen clusters can

Figure 15: Clustering of Pareto-optimal solutions in the objective space using *K*-means clustering. Taken from Taboada & Coit (2007).

Figure 16: *K*-means clustering of solutions from cluster 4 of Figure 15. Taken from Taboada & Coit (2007).

Figure 17: The structure of the Pareto-optimal front is visualized in the form of biclusters and dendrogram. Taken from Ulrich et al. (2008).

be clustered again. Even though the application involved a single objective, the procedure can be adopted for MOO.

4. The clustering of solutions in the decision space can also be combined with the clustering of variables, a process known as biclustering (Cheng & Church, 2000). In Ulrich et al. (2008), biclustering is performed on the Pareto-optimal solutions of a network processor design problem with binary decision variables. The biclusters are visualized as shown in the left panel of Figure 17. While informative in itself, this representation does not reveal how the subsets of variables are linked to each other. To this end, starting from the largest bicluster, the solutions are split recursively into groups until each group contains only one solution. The resultant hierarchy is visualized as a dendrogram, as shown in the right panel of Figure 17, which clearly shows strongly related subsets of decision variables.
5. A procedure for obtaining clusters that are compact and well-separated in both the objective and the decision spaces has been proposed in Ulrich (2012). This clustering is formulated as a biobjective problem of maximizing cluster goodness (a cluster validity index combining intercluster and intracluster distances) in both the spaces. Several solution representations and validity indices are tested. Application

to a truss bridge problem reveals that the approach finds clusters of bridges that are both “similar looking” (similarity in decision space) and also closer in the objective space.

All clustering methods discussed above generate hard clusters, i.e. each solution can belong to exactly one cluster. However, MOO datasets may also consist of overlapping clusters of solutions obtained from multiple runs of optimization. Fuzzy clustering methods such as fuzzy c -means and possibilistic c -means allow solutions to belong to multiple clusters with a certain degree of membership (Xu & Wunsch, 2005). To the best of our knowledge, fuzzy clustering methods have not been used in the literature on MOO datasets and should be explored in the future.

A desirable feature of clustering algorithms is the ability to detect arbitrarily shaped clusters. Most clustering algorithms that purely rely on distance measures can only detect globular clusters. On the other hand, kernel-based and density-based clustering methods can find arbitrarily shaped clusters in higher dimensions. See Xu & Wunsch (2005) for a comprehensive survey of clustering methods. High-dimensional decision and objective spaces may also cause conventional clustering methods to be ineffective. This problem can often be addressed by performing dimensionality reduction prior to clustering. Most techniques discussed in the following section can be used for this purpose.

3.2.3. *Manifold Learning*

In this section, we discuss linear and nonlinear dimensionality reduction methods that have been applied to MOO datasets. It is worth reiterating here that a mapping that fully preserves dominance relations from a higher-dimensional space to a lower-dimensional space does not exist (Koppen & Yoshida, 2007). Like clustering, manifold learning methods are closely related to the domain of machine learning. However, since low dimensional mappings only yield implicit knowledge in a visual form, they are classified here. This is not uncommon in the literature concerning visual data mining (de Oliveira & Levkowitz, 2003; Jeong & Shimoyama, 2011).

1. The simplest dimensionality reduction technique is Principal Component Analysis (PCA) (Friedman et al., 2001), which finds a linear orthogonal projection for the data that captures variance maximally. Its use in biplots has been mentioned above. The method is equally useful in both objective and decision spaces. It was first used in MCDM for the graphical representation of solutions (Korhonen et al., 1980; Mareschal & Brans, 1988). When a linear utility function is provided by the decision maker, a preference preserving PCA can be formulated, as shown in (Vetschera, 1992). More recently, in Masafumi et al. (2010), a nurse scheduling optimization problem is solved using PCA-based visualization.
2. A closely related technique called Linear Discriminant Analysis (LDA) (Friedman et al., 2001), also aims at finding a low-dimensional representation of the data. However, instead of maximizing the variance like PCA, LDA chooses the linear orthogonal projection that maximizes the difference between two or more class specified in the data. Note that this is different from clustering, where such class labels are not specified. Once the set of orthogonal directions are found, the first two directions can be used to generate biplots in a manner similar to that in PCA. Colors can be used to differentiate between the classes. To our knowledge, LDA has not been used in the literature to visualize MOO datasets.

3. Proper orthogonal decomposition has been used in Oyama et al. (2010a,b) to extract design knowledge from the Pareto-optimal solutions of an airfoil shape optimization problem. Like PCA, it extracts dominant features in the data by decomposing it into a set of optimal orthogonal base vectors of decreasing importance. Also, like PCA, it works best for datasets containing linearly correlated variables, transforming them into datasets of lower dimensionality.
4. Multidimensional scaling (MDS) (Kruskal & Wish, 1978; Borg & Groenen, 2005) refers to a group of linear mapping methods that retain the pairwise distance relations between the solutions as much as possible (Van Der Maaten et al., 2009), by minimizing the cost function given by

$$C = \sum_{i \neq j} \left(d_{ij}^{(h)} - d_{ij}^{(l)} \right)^2, \quad (2)$$

where d is a distance measure and its superscripts h and l represent the distance in higher and lower dimensional spaces, respectively. Usually, a Euclidean distance metric is used. MDS has been used in Walker et al. (2013) with a *dominance distance* metric for d that takes into account the degree of dominance between solutions. In effect, non-dominated solutions which dominate the same solutions are considered to be closer. MDS has also been used to visualize clustered non-dominated solutions during the optimization process (Kurasova et al., 2013). The procedure was later extended for interactive MOO in Filatovas et al. (2015). Chemical engineering applications involving the use of MDS to understand the Pareto-optimal solutions in the objective and decision spaces can be found in Žilinskas et al. (2006, 2015).

5. Sammon mapping (Sammon, 1969) is a nonlinear version of MDS that does a better job of retaining the local structure of the data (Van Der Maaten et al., 2009), by using a normalized cost function given by

$$C = \frac{1}{\sum_{i,j} d_{ij}^{(h)}} \sum_{i \neq j} \frac{\left(d_{ij}^{(h)} - d_{ij}^{(l)} \right)^2}{d_{ij}^{(h)}}. \quad (3)$$

It has been used in Valdes & Barton (2007) to obtain three dimensional mappings for higher dimensional objective spaces in a virtual reality environment. In Pohlheim (2006), it has been used for the visualization of the decision space during optimization. Neuroscale (Lowe & Tipping, 1997) is a variant of Sammon mapping that uses radial basis function to carry out the mapping. Use cases for both methods can be seen in Walker et al. (2013); Tusar & Filipic (2015).

6. Isomaps (Tenenbaum, 2000), short for isometric mappings, are another variant of Sammon mapping where the geodesic distances along an assumed manifold are used instead of Euclidean distances. The assumed manifold is approximated by constructing a neighborhood graph and the geodesic distance between two points is given by the shortest path between them in the graph. A classic example used to show its effectiveness is the Swiss Roll dataset (Van Der Maaten et al., 2009). For such nonlinear manifolds, isomaps are found to be better than PCA and Sammon mapping. In (Kudo & Yoshikawa, 2012), they have been used to map the solutions

Figure 18: Self-organizing map of the objective function values and typical wing planform shapes. Taken from Obayashi & Sasaki (2003).

Figure 19: Generative topographic map of the design space generated using DOE samples. Taken from Holden & Keane (2004).

from the decision space, considering their distances in the objective space. The application concerns the conceptual design of a hybrid rocket engine.

7. Locally linear embedding (Roweis, 2000) is similar in principle to isomaps, in that it uses a graph representation of the data points. The difference is that it only attempts to preserve local data structure. All points are represented as a linear combination of their nearest neighbors. The approach has been used in Mukerjee & Dabbeeru (2009) to identify manifolds embedded in high-dimensional decision spaces and deduce the number of intrinsic dimensions.
8. Self-organizing maps (SOMs) (Kohonen, 1990) can also provide a graphical and qualitative way of extracting knowledge. A SOM allows the projection of information embedded in the multidimensional objective and decision spaces onto a two-dimensional map. SOMs preserve the topology of the higher-dimensional space, meaning that neighboring points in the input space are mapped to neighboring units in the SOM. Thus, it can serve as a cluster analysis tool for high-dimensional data, when combined with clustering algorithms, such as hierarchical clustering, to reveal clusters of similar design solutions (Vesanto & Alhoniemi, 2000). Figure 18 shows its application in a practical design of a supersonic aircraft wing and wing-fuselage design that indicates the role of certain variables in making design improvements. Clusters of similar wing shapes are obtained by projecting the design objectives on a two-dimensional SOM and clustering the nodes using the SOM-Ward distance measure.

Several practical applications of SOM-based data mining of solutions from optimization, ranging from multidisciplinary wing shape optimization to robust aerofoil design, can be found in Chiba et al. (2005); Jeong et al. (2005b); Kumano et al. (2006); Parashar et al. (2008). The studies indicate that such design knowledge can be used to produce better designs. For example, in Chiba et al. (2007), the design variables that had the greatest impact on wing design were found using SOMs. In Doncieux & Hamdaoui (2011), which involves the design of a flapping

wing aircraft, design variables that significantly affected the velocity of the aircraft were identified through SOMs.

Multi-objective design exploration (MODE) (Obayashi et al., 2005, 2007) uses a combination of kriging (Simpson et al., 2001) and self-organizing maps to visualize the structure of the decision variables of non-dominated solutions. This approach has been used to study the optimal design space of aerodynamic configurations and centrifugal impellers (Obayashi et al., 2005; Sugimura et al., 2007).

9. Generative topographic maps (GTM) (Bishop et al., 1998) are similar to SOMs in principle, but instead of discretizing the input space like SOMs, they formulate a density model over the data. While SOMs provide the mapping from high-dimensional space to two dimensions directly, GTMs use radial basis functions to provide an intermediate latent variable model through which the visualization is achieved. GTMs have been used to visualize the solution space of aircraft designs in Holden & Keane (2004), in order to perform solution screening and optimization. The design points are generated using the design of experiments. Figure 19 shows the GTM of a 14 variable decision space.

Some of the manifold learning methods discussed above have been applied to MOO datasets obtained from standard test problems in Walker et al. (2013) and Tušar (2014). In addition to comparing them with graphical visualization methods, these studies also propose new approaches which are discussed in the previous sections. Walker et al. (2013) propose seriated heatmaps and a similarity measure for solutions called the dominance distance. Tušar (2014) proposes the prosecution method, and also compares visualization methods based on various desired properties ranging from preservation of dominance relation, front shape and distribution to robustness, simplicity and scalability.

Data visualization is now a research field in itself. Furthermore, visual data mining is increasingly becoming a part and package of modern data visualization tools which incorporate clustering, PCA and other dimensionality reduction techniques, as discussed above. Animations, such as grand tours (Asimov, 1985) and state-of-the-art interactive multidimensional visualization technologies, like virtual reality, can also aid visual data mining (Nagel et al., 2001; Hoffman & Grinstein, 2002; Valdes & Barton, 2007) immensely. Nevertheless, it is to be borne in mind that understanding a visual representation of knowledge often requires user’s expertise (Valdés et al., 2012), which may lead to a subjective interpretation of results, making such implicit knowledge rather difficult to be used and transferred within the context of multi-objective optimization.

3.3. Machine Learning

Non-visual data mining techniques usually incorporate some form of machine learning to extract knowledge in an explicit form. Machine learning tasks are mostly classified as either being supervised or unsupervised. Both approaches have been used for knowledge discovery from MOO datasets, as described in the following sections.

3.3.1. Supervised Learning

Supervised learning primarily involves training a computer program to distinguish between two or more classes of data-points, a task referred to as classification, using instances with known class labels. Over the years, many classification algorithms have been

proposed in the literature. However, as discussed in Section 2, any classification algorithm used for knowledge discovery should preferably generate knowledge in an explicit form. Support vector machines (SVMs) and neural networks are two popular classification methods that cannot directly be used for knowledge discovery, since they only produce a black-box model³ for prediction. In other words, given a new data-point, they can only predict its class, but do not say anything about how its features affect the prediction, at least not in a human-perceivable way. In both methods, prediction is achieved through a set of weights that transform the features of the data, thus obfuscating the prediction process. The term ‘learning’ here refers to the optimization of these weights, with respect to some measure calculated over the set of training instances. In SVMs, the criterion is to maximize the separation between the classes involved, while in neural networks, it is to minimize prediction error.

Unlike the classifiers mentioned above, decision trees represent knowledge in an explicit form called *decision rules*. They take the form of constructs involving conditional statements that are combined using logical operators. An example of a decision rule is,

$$\text{if } (x_1 < v_1) \wedge (x_2 > v_2) \wedge \dots \text{ then (Class 3).} \quad (4)$$

Such rules are easy to understand and interpret. Decision tree learning algorithms, such as the popular ID3 (Quinlan, 1986) and C4.5 (Quinlan, 2014) algorithms, work by recursively dividing the training dataset using one feature at a time to generate a tree, as shown in Figure 20. At each node, the feature and its value are chosen such that the division maximizes the dissimilarity between the two resulting groups. As a result, the most sensitive features appear close to the root of the tree and the least sensitive features appear at the leaves. Each branch of the tree represents a conditional statement, and the paths connecting the root to the leaves represent different decision rules. Decision trees can be of two types, classification trees and regression trees, depending on whether the class variable is discrete or continuous. Both have been used with MOO datasets. A typical problem with decision tree learning is its tendency to overfit the training data leading to a high generalization error. Pruning methods that truncate the decision trees, based on certain thresholds, are used to counteract this issue to some extent (Quinlan, 1987).

Supervised learning can also refer to regression, in which case the class label is replaced by a continuous feature. Statistical regression methods, such as polynomial regression and Gaussian regression (kriging), use explicit mathematical models (Simpson et al., 2001). Black-box models, like neural networks, radial basis function networks and SVMs, can also be used for regression, in which case the knowledge is, of course, captured implicitly (Knowles & Nakayama, 2008).

All supervised learning tasks require that an output feature is associated with each training instance. For classification, a class label is expected and for regression, a continuous feature is required. Since MOO datasets do not naturally come with a unique output feature, it is up to the user to assign one before using any of the supervised learning techniques. In MOO, there are four main ways of doing this:

1. using ranks obtained by non-dominated sorting of solutions,

³A black-box model is one that obfuscates the process of transformation of its inputs to its outputs.

Figure 20: Decision trees divide the Pareto-optimal dataset to maximize dissimilarity between the two resulting groups. The most sensitive variables appear close to the root of the tree while less sensitive variables occur at the leaves.

2. using one of the objective functions,
3. using preference information provided by the decision maker, and
4. using clustering methods.

The first approach aims at making a crisp distinction between ‘high’ rank and ‘low’ rank solutions. Although black-box models are uninterpretable to humans, the knowledge captured by them can still be used inside an optimization algorithm. For example, SVMs have been used in Chun-Wei Seah et al. (2012) as surrogate models for predicting the ranks of new solutions generated during multi-objective optimization. The training data consists of archived solutions whose ranks are determined by non-dominated sorting. These ranks function as class labels for supervised learning. New solutions that have a predicted rank of one are exactly evaluated and added to the archive, while the rest of the solutions are discarded, thus saving function evaluations. Different ranks can also be combined to form a single class, as shown in Loshchilov et al. (2010), where all dominated solutions are assigned to one class which is learned through a *one-class* variant of SVM. Ranks can also be used for explicit knowledge discovery, as shown in Ng et al. (2011), where classification trees are employed to extract decision rules that distinguish between dominated (rank > 1) and non-dominated solutions (rank $= 1$). The problem of class imbalance arises in this case because the number of non-dominated solutions is usually much lower than the number of dominated solutions. In Ng et al. (2011), class imbalance is handled by oversampling the minority class, i.e. the non-dominated set. One of the most popular techniques for dealing with class imbalance is SMOTE, short for Synthetic Minority Over-sampling Technique (Chawla et al., 2002), where instead of replicating samples, new synthetic samples are added to the minority class. In combination with under-sampling of the majority class, this method has been shown to improve the classification accuracy of C4.5 algorithm. Another common technique to account for class imbalance is to associate a higher misclassification cost with the minority class, effectively pushing the decision boundary away from minority samples. This approach is known to be effective even in the presence of ten classes (Abuomar et al., 2015).

The second and the third methods are more practical from a decision maker’s point of view. In Sugimura et al. (2010), the authors use the second approach to extract decision rules from the feasible solutions for the design of a centrifugal impeller. The decision maker chooses one of the objective functions as the class variable. Due to the continuous nature of objective function values, regression tree learning is used. As seen in Figure 20,

the range of the chosen objective function f is discretized into different levels that serve as classes for decision tree learning. x_i represent the decision variables - features on which the splits are made. Using a similar procedure, different *rulesets* were generated for different objectives of a real-world, flexible machining line optimization problem in Dudas et al. (2011). In addition to decision trees, the authors also used ensembles (voting from multiple decision trees) to increase the predictive performance. However, in this case, no rules are generated, only a ranking for the importance of variables.

When no decision maker is involved, objective function values can still be used to build surrogate(s), in which case the type of knowledge depends on the regression model used. As in the case of SVMs, this knowledge is also only useful to the optimization algorithm.

The third method helps the decision maker to understand how certain preferred solutions differ from others. In Dudas et al. (2014), this preference is expressed by defining a region of interest in the objective space, whereas in Dudas et al. (2015) the same is achieved through the specification of a reference point. In both cases, the normalized distance of the feasible solutions from the preferred region/reference point is used as a continuous feature for regression tree learning. Effectively, this eliminates the problem of class imbalance and transforms a classification problem into a regression problem. Rules describing solutions that are close to the preferred region/reference point reflect the decision maker's choice(s) in terms of decision variables. The advantage of these methods lies in the simplistic way in which preference is articulated.

The fourth method, proposed in Part B of this paper, involves clustering the solutions in the objective space and then using the cluster membership of each solution as its class label for decision tree learning in the decision space. Such an approach has not been used previously, to the best of our knowledge.

3.3.2. Unsupervised Learning

Unsupervised learning techniques do not require labeled data and are therefore more suited for use with raw MOO datasets. Following are methods from the literature that utilize unsupervised learning to extract knowledge in an explicit form from MOO datasets.

1. Rough set theory (Sugimura et al., 2007, 2010; Greco et al., 2008) and association rule mining (Sugimura et al., 2009) have been used on datasets obtained from the multi-objective design optimization of a centrifugal impeller. Both techniques work by first discretizing both the variable and the objective values into different levels, as shown in Table 3. In rough set theory, each solution then becomes a sequence of levels. Rule induction methods can now be used to extract *association rules* of the form,

$$\text{if } (x_1 \in \text{Level 1}) \wedge (x_2 \in \text{Level 2}) \wedge \dots \text{ then } (y \in \text{Level 2}) \quad (5)$$

For a rule 'if A then C ' (denoted as $A \rightarrow C$), where A is the *antecedent* and C is the *consequent*, the *support* and *confidence* are defined as

$$\text{support}(A \rightarrow C) = \frac{N_{A \cup C}}{N}, \quad \text{confidence}(A \rightarrow C) = \frac{N_{A \cup C}}{N_A}. \quad (6)$$

Table 3: Design variables and objectives are divided into different levels. Rule induction methods can extract all association rules that satisfy minimum *support* and *confidence* values.

Solution index	Condition attributes				Decision attribute
	x_1	x_2	x_3	...	y
1	Level 1	Level 2	Level 5	...	Level 2
2	Level 5	Level 4	Level 1	...	Level 1
3	Level 3	Level 4	Level 3	...	Level 5
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$

While support represents the relative frequency of the rule in the dataset, confidence represents its accuracy. The accuracy can be interpreted as the probability of C given A , i.e., $P(C|A)$. Though both decision rules and association rules take the same ‘if-then’ form, there is an important difference between them. All decision rules obtained from a decision tree use the same feature in C that was selected as the class variable, whereas different association rules can have different features in C . For example, in Table 3, the decision attribute y can be a variable or an objective, or any of the other columns included in the MOO dataset. Given a threshold support value, association rule mining can extract rules meeting a specified confidence level.

2. Automated *innovization* (Bandaru, 2013) is a recent unsupervised learning algorithm that can extract knowledge from MOO datasets in the form of *analytical relationships* between the variables and objective functions. The term *innovization*, short for *innovation through optimization*, was coined by Deb in Deb & Srinivasan (2006) to refer to the manual approach of looking at scatter plots of different variable combinations and performing regression with appropriate models on the correlated parts of the dataset. The procedure was specifically developed to analyze Pareto-optimal solutions, since the obtained relationships can then act as *design principles* for directly creating optimal solutions in an innovative way, without the use of optimization. In a series of papers since 2010 (Bandaru & Deb, 2010, 2011a,b, 2013a; Deb et al., 2014), the authors automated the innovization process using grid-based clustering and genetic programming. While grid-based clustering replaces the human task of identifying correlations (Bandaru & Deb, 2010, 2011b), genetic programming eliminates the need to specify a regression model (Bandaru & Deb, 2013a). Relationships are encoded as parse trees using a terminal set $\mathcal{T}$ consisting of basis functions ϕ_i (usually simply the variables and the objective functions), and a function set $\mathcal{F}$ consisting of mathematical operators. Randomly initialized parse trees are evolved using genetic programming to minimize the variance of the relationship in parts of the data where the corresponding basis functions are correlated. To identify such subsets of data, each candidate relationship $\psi(\mathbf{x})$ is evaluated for all Pareto-optimal solutions to obtain a set of c -values, as shown in Figure 21, which are then clustered using grid-based clustering. The advantage of using grid-based clustering is that the number of clusters does not need to be prespecified.

Automated innovization also incorporates niching (Bandaru & Deb, 2011a), which enables the processing of several variable combinations at a time, so that all relationships hidden in the dataset can be discovered simultaneously. Applications

Figure 21: Automated innovization in a nutshell. Taken from Bandaru et al. (2015).

to multi-objective noise-barrier optimization, extrusion process optimization and friction-stir welding process optimization can be found in Deb et al. (2014).

3. Higher-level innovization (Bandaru, 2013) is an extension to the concept of innovization discussed above. Most multi-objective optimization problems inherently involve certain design, system or process parameters that remain fixed during optimization. However, a decision maker might want to vary them for future optimization tasks. Higher-level automated innovization (Bandaru & Deb, 2013b) uses multiple Pareto-optimal datasets obtained by varying such parameters to extract analytical relationships not only between different basis functions, but also between the basis functions and the problem parameters. This results in higher-level knowledge that pertains to the underlying physics of the design/system/process, rather than to a particular problem setting. Proof-of-principle results were first presented in Bandaru & Deb (2013b). Applications to friction-stir welding process optimization and inventory management problems can be found in Bandaru et al. (2011) and (Bandaru et al., 2015) respectively.
4. For nominal variables, an alternate form of representation is *patterns*. A pattern is simply a sequence of values for a subset of variables that appears frequently in the dataset. Patterns hidden in a dataset can be extracted using methods such as biclustering (Cheng & Church, 2000) and sequential pattern mining (Agrawal & Srikant, 1995). To our knowledge, no current study makes use of these methods for knowledge discovery from MOO datasets. Sequential pattern mining is used and extended in Part B of this paper.

3.4. Summary of Methods

A graphical summary of the methods discussed in the previous section is presented in Figure 22. This summary is only representative of the current published literature that discusses knowledge discovery from MOO datasets. It may be naive to call it exhaustive, since other methods probably exist that have the potential to be used with MOO datasets. As mentioned before, our classification is motivated by the type of knowledge representation, which is also shown in the figure. Descriptive statistics represent univariate and bivariate knowledge in the form of numbers and are, hence, classified as explicit.

All graphical methods naturally come under visual data mining. Though clustering and manifold learning fall into the domain of unsupervised learning, the knowledge obtained from them is represented visually and, hence, they are also grouped as visual data mining methods. On the other hand, though black-box models obtained from some supervised learning methods also represent knowledge implicitly, the model itself can still be used by an automated algorithm. Moreover, some studies exist that discuss methods for extracting explicit rules from trained SVMs (Diederich, 2008; Barakat & Bradley, 2010) and neural networks (Andrews et al., 1995; Tsukimoto, 2000). Therefore, we have grouped them as ‘Implicit/Explicit’ in Figure 22. The remaining methods grouped under machine learning techniques express knowledge as either decision rules, association rules, patterns or analytical relationships, which are all explicit forms.

As a side note, the term ‘exploratory data analysis’ (Tukey, 1977) is often used to collectively describe data visualization, descriptive statistics and dimensionality reduction.

Despite the abundance of data mining methods, a few challenges remain to be addressed. We discuss them in the next section.

4. Challenges and Future Research Directions

Most methods described in this paper are generic data analysis and data mining techniques that have simply been applied to MOO datasets. As such, they do not distinguish between the two distinct spaces (i.e. objective space and decision space) of MOO datasets. For example, visual data mining methods either ignore one of the spaces or deal with them separately. This makes the involvement of a decision maker difficult, because (s)he is usually interested in the objective space, while knowledge is often extracted with respect to the decision variables. The distance-based regression tree learning approaches, proposed in Dudas et al. (2014) and (Dudas et al., 2015), are the only methods that come close to achieving this. The shortage of such *interactive data mining* methods is the biggest hurdle in the analysis of MOO datasets. In Part B of this paper, we partially address this issue, using a more natural preference articulation approach that involves brushing over solutions with a mouse interface. However, even this method is only effective for two and three dimensional objective spaces.

Discreteness in MOO datasets also presents some new challenges to most of the methods discussed in this paper. In order to elaborate on these challenges, we first enumerate the different ways in which discreteness can occur in optimization problems:

1. Inherently discrete or integers: Variables that are by their very nature, integers. Examples include number of gear teeth, number of cross-members, etc.
2. Practically discrete: Variables that are forced to only take certain real values, due to practical considerations. For example, though the diameter of a gear shaft is theoretically a continuous variable, only shafts of certain standard diameters are manufactured and readily available. The optimization problem should therefore only consider these discrete options for the corresponding variable.
3. Categorical: Variables for which the numerical value is of no significance, but only a programmatic convenience. They can be further divided as,

Figure 22: A graphical summary of the methods discussed in this paper.

- (a) Ordinal: Variables that represent position on a scale or order. For example, variables that can take ‘Low’, ‘Medium’ or ‘High’ as options are usually encoded with numerical values 1, 2 and 3, respectively. The values themselves are of no importance, as long as the order between them is maintained.
- (b) Nominal: Variables that represent unordered options. For example, a variable that changes between ‘Machine 1’, ‘Machine 2’ and ‘Machine 3’, or a variable that represents different grocery items. In statistics, the term ‘dichotomous’ is used for variables that can only take two nominal options. Examples are ‘True’ and ‘False’ or ‘Male’ and ‘Female’. Again, the numerical encoding of a nominal variable is irrelevant, but programmatically useful.

The following difficulties are observed when dealing with MOO datasets containing any of the variables mentioned above:

1. With visual data mining methods, discreteness may lead to apparent but non-

existent correlations between the variables. Such artifacts can occur due to overplotting, visualization bias (scaling), or low number of discrete choices. When present, they make the subjectiveness associated with visual interpretation of data even more prominent.

2. Most distance measures used in both visual and non-visual methods are not applicable to ordinal and nominal variables. For example, the distance between variable options ‘True’ and ‘False’, ‘Male’ and ‘Female’ or ‘Machine 1’ and ‘Machine 3’ is usually not quantifiable. Similarly, the distance between ordinal variable options ‘Low’ and ‘High’ will depend on the numerical values assigned to them. Although many distance measures for categorical data have been proposed (McCane & Albert, 2008; Boriah et al., 2008), there is no clear consensus on their efficacy.
3. Data mining methods that use clustering may also result in superficial clusters. Consider, for example, a simple one-variable discrete dataset with 12 solutions, $\{1, 1, 1, 2, 2, 2, 2, 3, 3, 10, 10, 10\}$. Most distance-based clustering algorithms will partition it into four clusters given by, $\{\{1, 1, 1\}, \{2, 2, 2, 2\}, \{3, 3\}, \{10, 10, 10\}\}$, each with zero intra-cluster variance. However, this partitioning does not capture any meaningful knowledge, because each cluster simply contains one of the options for the variable. A more useful partitioning is $\{\{1, 1, 1, 2, 2, 2, 2, 3, 3\}, \{10, 10, 10\}\}$ which tells the user that more data points have a lower value for the variable.
4. Decision tree learning generates rules of the form shown in Equation (4). However, since nominal variables do not have any ordering among their options, an expression such as $x_1 < v_1$ has little meaning. Association rules are a suitable form of representation for nominal variables. However, since association rule mining is unsupervised, it is difficult for a decision maker to become involved in the knowledge discovery process.
5. Automated innovization is also not directly applicable to discrete datasets for two reasons. Firstly, since it uses clustering to identify correlations and, as discussed above, this can lead to superficial clusters. Secondly, ordinal and nominal variables are not usually expected in mathematical relationships, because they do not have a specific numerical value.

Note that none of the methods discussed in this paper specifically address problems associated with discreteness. Some of them are explored in Part B of this paper.

4.1. Knowledge-Driven Optimization (KDO)

Real-world multi-objective optimization often involves several iterations of problem formulation. A user (or decision maker) may wish to modify the original optimization problem on the basis of the knowledge gained through a post-optimal analysis of the obtained solutions (see Figure 23). Explicit knowledge is preferable, but since most decisions are taken by humans, implicit knowledge obtained from visual data mining methods can also be used effectively, although subjectiveness cannot be ruled out. In the context of innovization, such an approach was first demonstrated in Deb & Datta (2012). Taking the multi-objective optimization of a metal-cutting process as the case study, the authors first obtain a set of near Pareto-optimal solutions. Regression analysis is carried out on the solutions to obtain analytical relationships between the variables. These relationships are then imposed as constraints on the original optimization problem and a local search is performed on the previously obtained solutions to generate a better

approximation of the Pareto front. Similar procedures were adopted in Ng et al. (2012, 2013). Given a region of preference (Ng et al., 2012) or a reference point (Ng et al., 2013) by the decision maker, the distance-based regression tree approach (described above) is used to extract decision rules. The validity of the rules is verified by adding them as constraints and resolving the optimization problem, which then only generates solutions that the decision maker would prefer. All three of these studies involve *offline knowledge-driven optimization* (offline KDO). Since the knowledge is extracted post-optimally, the optimization algorithm remains well separated from the data mining procedures and, hence, can function independently. Thus, any multi-objective optimization algorithm can be used without changes. Sometimes, post-optimal knowledge can also be used to modify the parameters or the initial population of the optimization algorithm. Eiben et al. (1999) refer to this type of parameter setting as *parameter tuning*. It is a common practice in many real-world MOO tasks to use the non-dominated solutions from previous runs as seeds to initialize new runs. More sophisticated systems may involve the use of codified knowledge to build a *knowledge base*, which in turn can drive an expert system for automating modifications to the MOO problems and algorithms.

Another important research direction is the extraction of knowledge during optimization to facilitate faster convergence to the entire Pareto-optimal front or to a preferred region on it. This *online knowledge-driven optimization* (online KDO) process is shown schematically in Figure 23, alongside the one described above. As discussed above, when a decision maker is involved, interactive data mining methods should be used to focus the search towards the preferred region(s). In the absence of such preference, data mining methods can utilize the search history of past solutions to generate new promising solutions. Explicit representation of knowledge is especially important here, since it has to be stored in computer memory and processed programmatically. Significant changes to the optimization algorithm may also be required, in order to incorporate the knowledge into the search process. Three aspects of this integration which can greatly influence the performance of the procedure are (i) when knowledge should be extracted, (ii) which solutions should be used, and (iii) what knowledge should be imposed. Possible issues that can arise due to a misguided search are premature convergence to local Pareto fronts and search stagnation. As far as implicit knowledge is concerned, the only viable way of using it online is through black-box models for meta-modeling. A related avenue for future research is the conversion of implicit knowledge obtained from visual data mining methods to explicit forms. As stated before, this has already been achieved for trained SVMs and neural networks. As in the case of offline KDO, knowledge obtained during runtime can also be used to modify algorithmic parameters. The difference is that here these modifications are made on the fly. Eiben *et al.* (Eiben et al., 1999) call it *parameter control*. Rules obtained through machine learning have been used in the past to adaptively control crossover (Sebag & Schoenauer, 1994) and mutation (Sebag et al., 1997a), and even to prevent optimization algorithms from repeating past search pitfalls (Sebag et al., 1997b).

Online KDO is also closely associated with the domain of data stream mining (Gaber et al., 2005). Modern computing hardware, parallel processing and cloud computing enable evolutionary algorithms to generate new solutions at a high rate. Any data mining technique can get easily overwhelmed by the vast amount of new data created in each generation. New preference articulation methods will have to be developed to filter out a major part of the data, while retaining the solutions that matter. For example, if

Figure 23: A generic framework for knowledge-driven optimization.

a reference point is provided by the decision maker, the incoming data stream could be filtered to obtain solutions that are closest to the reference point. When a new stream of solutions arrives, these closest solutions are updated and the data mining algorithm can be rerun. In the absence of preference information, concepts such as load shedding, aggregation and sliding windows can be borrowed from data stream mining. Gaber et al. (2005) provide a good overview of these techniques. An application of data stream mining in smart grids can be found in Dahal et al. (2015).

Online KDO has received renewed attention recently. The Learnable Evolution Model (LEM) (Michalski, 2000) uses AQ learning (also proposed by Michalski) to deduce *attributional rules*, an enhanced form of decision rules, to differentiate between high-performing and low-performing solutions in the context of single-objective optimization. The algorithm was later extended in Jourdan et al. (2005) for multi-objective problems where different definitions of high-performing and low-performing were evaluated. A local PCA approach is used in Zhou et al. (2005) to detect regularity in the decision space and a probability model is built to sample new promising solutions. The model is built using the Estimation of Distribution Algorithm (EDA) (Larrañaga & Lozano, 2002) at alternate generations of NSGA-II. In Saxena et al. (2013), linear and nonlinear dimensionality reduction methods are used to identify redundant objectives. Here, the knowledge extraction process takes place only in the objective space. Online objective reduction has been shown to be effective in many-objective optimization problems (Deb & Saxena, 2006; Saxena & Deb, 2007; Brockhoff & Zitzler, 2009) with redundant objectives. Decision rules generated on the basis of preference information from the decision maker are used in Greco et al. (2008) to constrain the objective and variable values to a region of interest. A logical preference model is built using Dominance-based Rough Set Approach and utilized for interactive multi-objective optimization. All these studies show promise

in the idea, but several performance issues are yet to be tackled, a few of which are mentioned above. Online KDO also includes the wide variety of meta-modeling methods available in the literature (Knowles & Nakayama, 2008). However, the purpose of meta-modeling is usually not knowledge extraction, but to reduce the number of expensive function evaluations by using approximations of the objective functions. These approximations can be said to hold knowledge about the fitness landscape in an implicit form and are updated at regular intervals during optimization.

The generic framework for KDO proposed above has, in essence, encompassed the learning cycle of interactive multi-objective optimization (IMO) described in Belton et al. (2008). The cycle of IMO depicts what information is being exchanged between the decision maker and the optimizer (or model) to facilitate the learning cycle. Within this cycle, on one side, the decision maker learns from the solutions explored by the model (optimizer), while, on the other side, the preference model within the optimizer learns the preferences of the decision maker. Thus, the output of the model learning is an explicit preference model of the decision maker who provided the information, which may then be used to guide the search for more preferred solution(s). Despite the similarity between the IMO and the KDO framework, it has to be noted that IMO is only targeted to support the learning through interactions between the decision maker and the optimizer. On the other hand, the KDO framework aims at guiding the optimization by making use of knowledge extracted using both interactive and non-interactive data mining methods reviewed in this paper. This is why a comprehensive survey of available data mining methods for handling MOO datasets, and identification of new research directions to address their current limitations, are so important for realizing the KDO framework. The authors hope that the present paper serves these purposes.

5. Conclusions

Multi-objective optimization problems are usually solved using population-based evolutionary algorithms. These methods generate and evaluate a large number of solutions. In this survey paper, we have reviewed several data mining methods that are being used to extract knowledge from such MOO datasets. Noting that this knowledge can be either implicit or explicit, we classified available methods according to the type and form of knowledge that they generate. Three groups of methods were identified, namely, (i) descriptive statistics, (ii) visual data mining, and (iii) machine learning. Descriptive statistics are simple univariate and bivariate measures that summarize the location, spread, distribution and correlation of the variables involved. Visual data mining methods span a wide range from simple graphical approaches to manifold learning methods. We discussed both generic and specific data visualization methods. Sophisticated methods that extract knowledge in various explicit forms, such as decision rules, association rules, patterns and relationships were discussed as part of machine learning techniques. We observed that there are a number of visual data mining methods but relatively fewer machine learning methods that have been developed specifically to handle MOO datasets. Self-organizing maps seem to be the popular choice of implicit representation. For explicit representation, descriptive statistics, decision trees and association rules are more common, due to the ease of availability of corresponding implementations.

We identified a few research areas that are yet to be explored, one of which is the effective handling of discrete optimization datasets, and also highlighted the need for in-

teractive data mining that gives equal importance to both objective and decision spaces. We discussed various aspects of online knowledge-driven optimization including its resemblance to data stream mining. Considering the ever-growing interest in the application of multi-objective optimization algorithms to real-world problems, knowledge-driven optimization is likely to emerge as an important research topic in the coming years.

Acknowledgments

The first author acknowledges the financial support received from KK-stiftelsen (Knowledge Foundation, Stockholm, Sweden) for the ProSpekt 2013 project KDISCO.

References

- Abuomar, O., Nouranian, S., King, R., Ricks, T., & Lacy, T. (2015). Comprehensive mechanical property classification of vapor-grown carbon nanofiber/vinyl ester nanocomposites using support vector machines. *Computational Materials Science*, 99, 316–325.
- Agrawal, G., Lewis, K., Chugh, K., Huang, C.-H., Parashar, S., & Bloebaum, C. (2004). Intuitive visualization of Pareto frontier for multiobjective optimization in n-dimensional performance space. In *Proceedings of the 10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference* (pp. AIAA 2004-4434). Reston, Virginia: AIAA.
- Agrawal, R., & Srikant, R. (1995). Mining sequential patterns. In *Proceedings of the 11th International Conference on Data Engineering* (pp. 3–14). IEEE.
- Agresti, A. (2010). *Analysis of ordinal categorical data*. Wiley.
- Andrews, D. F. (1972). Plots of high-dimensional data. *Biometrics*, 28, 125–136.
- Andrews, R., Diederich, J., & Tickle, A. B. (1995). Survey and critique of techniques for extracting rules from trained artificial neural networks. *Knowledge-based Systems*, 8, 373–389.
- Ang, K., Chong, G., & Li, Y. (2002). Visualization techniques for analyzing non-dominate set comparison. In *Proceedings of the 4th Asia-Pacific Conference on Simulated Evolution and Learning, SEAL 2002* (pp. 36–40). Nanyang Technological University, School of Electrical & Electronic Engineering.
- Angehrn, A. A. (1991). *Supporting Multicriteria Decision Making: New Perspectives and New Systems*. Technical Report INSEAD, European Institute of Business Administration.
- Asimov, D. (1985). The grand tour: A tool for viewing multidimensional data. *SIAM Journal on Scientific and Statistical Computing*, 6, 128–143.
- Bader, J., & Zitzler, E. (2011). HypE: An algorithm for fast hypervolume-based many-objective optimization. *Evolutionary Computation*, 19, 45–76.
- Bandaru, S. (2013). *Automated Innovization: Knowledge Discovery through Multi-Objective Optimization*. Ph.D. thesis Indian Institute of Technology Kanpur.
- Bandaru, S., Aslam, T., Ng, A. H. C., & Deb, K. (2015). Generalized higher-level automated innovization with application to inventory management. *European Journal of Operational Research*, 243, 480–496.
- Bandaru, S., & Deb, K. (2010). Automated discovery of vital knowledge from Pareto-optimal solutions: First results from engineering design. In *2010 IEEE Congress on Evolutionary Computation, CEC* (pp. 18–23). IEEE.
- 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, EMO 2011* (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, 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 2013* (pp. 513–527). Springer.
- Bandaru, S., & Deb, K. (2013b). Higher and lower-level knowledge discovery from Pareto-optimal sets. *Journal of Global Optimization*, 57, 281–298.
- Bandaru, S., Tutum, C. C., Deb, K., & Hattel, J. H. (2011). Higher-level innovization: A case study from friction stir welding process optimization. In *2011 IEEE Congress on Evolutionary Computation, CEC* (pp. 2782–2789). IEEE.

- Barakat, N., & Bradley, A. P. (2010). Rule extraction from support vector machines: A review. *Neuro-computing*, 74, 178–190.
- Belton, V., Branke, J., Eskelinen, P., Greco, S., Molina, J., Ruiz, F., & Slowinski, R. (2008). Interactive multiobjective optimization from a learning perspective. In *Multiobjective Optimization* (pp. 405–433). Springer.
- Bennett, J., & Fisher, R. A. (1995). *Statistical methods, experimental design, and scientific inference*. Oxford University Press.
- Beume, N., Naujoks, B., & Emmerich, M. (2007). SMS-EMOA: Multiobjective selection based on dominated hypervolume. *European Journal of Operational Research*, 181, 1653–1669.
- Bishop, C., Svensén, M., & Williams, C. (1998). GTM: The generative topographic mapping. *Neural computation*, 10, 215–234.
- Blasco, X., Herrero, J., Sanchis, J., & Martínez, M. (2008). A new graphical visualization of n-dimensional Pareto front for decision-making in multiobjective optimization. *Information Sciences*, 178, 3908–3924.
- Borg, I., & Groenen, P. J. (2005). *Modern multidimensional scaling: Theory and applications*. Springer Science & Business Media.
- Borah, S., Chandola, V., & Kumar, V. (2008). Similarity measures for categorical data: A comparative evaluation. In *Proceedings of the 2008 SIAM International Conference on Data Mining* (pp. 243–254). SIAM.
- Boussaïd, I., Lepagnot, J., & Siarry, P. (2013). A survey on optimization metaheuristics. *Information Sciences*, 237, 82–117.
- Brockhoff, D., & Zitzler, E. (2009). Objective reduction in evolutionary multiobjective optimization: Theory and applications. *Evolutionary Computation*, 17, 135–166.
- Chan, W. W.-Y. (2006). A survey on multivariate data visualization. *Science And Technology*, 8, 1–29.
- Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 16, 321–357.
- Cheng, Y., & Church, G. (2000). Biclustering of expression data. In *Proceedings of the 8th International Conference on Intelligent Systems for Molecular Biology* (pp. 93–103). AAAI.
- Chernoff, H. (1973). The use of faces to represent points in k-dimensional space graphically. *Journal of the American Statistical Association*, 68, 361–368.
- Chiba, K., Obayashi, S., Nakahashi, K., & Morino, H. (2005). High-fidelity multidisciplinary design optimization of aerostructural wing shape for regional jet. In *Proceedings of the 23rd AIAA Applied Aerodynamics Conference* (pp. 621–635). AIAA.
- Chiba, K., Oyama, A., Obayashi, S., Nakahashi, K., & Morino, H. (2007). Multidisciplinary design optimization and data mining for transonic regional-jet wing. *Journal of Aircraft*, 44, 1100–1112.
- Chichakly, K. J., & Eppstein, M. J. (2013). Discovering design principles from dominated solutions. *IEEE Access*, 1, 275–289.
- Chun-Wei Seah, Ong, Y.-S., Tsang, I. W., Siwei Jiang, Seah, C.-W., Ong, Y.-S., Tsang, I. W., & Jiang, S. (2012). Pareto rank learning in multi-objective evolutionary algorithms. In *2012 IEEE Congress on Evolutionary Computation, CEC* (pp. 1–8). IEEE.
- Cleveland, W. S. (1993). *Visualizing data*. Hobart Press.
- Coello Coello, C. A. (1999). A comprehensive survey of evolutionary-based multiobjective optimization techniques. *Knowledge and Information Systems*, 1, 269–308.
- Cramér, H. (1999). *Mathematical methods of statistics*. Princeton University Press.
- Dahal, N., Abuomar, O., King, R., & Madani, V. (2015). Event stream processing for improved situational awareness in the smart grid. *Expert Systems with Applications*, 42, 6853–6863.
- Deb, K. (2001). *Multi-objective optimization using evolutionary algorithms*. Wiley.
- Deb, K., Bandaru, S., Greiner, D., Gaspar-Cunha, A., & Tutum, C. C. (2014). An integrated approach to automated innovation for discovering useful design principles: Case studies from engineering. *Applied Soft Computing*, 15, 42–56.
- Deb, K., & Datta, R. (2012). Hybrid evolutionary multi-objective optimization and analysis of machining operations. *Engineering Optimization*, 44, 685–706.
- Deb, K., & Jain, H. (2014). An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, Part I: Solving problems with box constraints. *IEEE Transactions on Evolutionary Computation*, 18, 577–601.
- Deb, K., & Saxena, D. (2006). Searching for Pareto-optimal solutions through dimensionality reduction for certain large-dimensional multi-objective optimization problems. In *2006 IEEE Congress on Evolutionary Computation, CEC* (pp. 3352–3360). IEEE.
- Deb, K., & Srinivasan, A. (2006). Innovation: Innovating design principles through optimization. In

- Proceedings of the 8th Annual Conference on Genetic and Evolutionary Computation, GECCO 2006* (pp. 1629–1636). ACM.
- Diederich, J. (2008). *Rule extraction from support vector machines* volume 80. Springer Science & Business Media.
- Doncieux, S., & Hamdaoui, M. (2011). Evolutionary algorithms to analyse and design a controller for a flapping wings aircraft. In *New Horizons in Evolutionary Robotics* (pp. 67–83). Springer.
- Dudas, C., Frantzen, M., & Ng, A. H. C. (2011). A synergy of multi-objective optimization and data mining for the analysis of a flexible flow shop. *Robotics and Computer-Integrated Manufacturing*, 27, 687–695.
- Dudas, C., Ng, A. H. C., & Boström, H. (2015). Post-analysis of multi-objective optimization solutions using decision trees. *Intelligent Data Analysis*, 19, 259–278.
- Dudas, C., Ng, A. H. C., Pehrsson, L., & Boström, H. (2014). Integration of data mining and multi-objective optimisation for decision support in production systems development. *International Journal of Computer Integrated Manufacturing*, 27, 824–839.
- Efremov, R., Insua, D. R., & Lotov, A. (2009). A framework for participatory decision support using Pareto frontier visualization, goal identification and arbitration. *European Journal of Operational Research*, 199, 459–467.
- Eiben, A. E., Hinterding, R., & Michalewicz, Z. (1999). Parameter control in evolutionary algorithms. *IEEE Transactions on Evolutionary Computation*, 3, 124–141.
- Faucher, J.-B. P., Everett, A. M., & Lawson, R. (2008). Reconstituting knowledge management. *Journal of knowledge management*, 12, 3–16.
- Filatovas, E., Podkopaev, D., & Kurasova, O. (2015). A visualization technique for accessing solution pool in interactive methods of multiobjective optimization. *International Journal of Computers, Communications & Control*, 10, 508–519.
- Fleming, P. J., Purshouse, R. C., & Lygoe, R. J. (2005). Many-objective optimization: An engineering design perspective. In *Proceedings of the 3rd international Conference on Evolutionary Multi-criterion Optimization, EMO 2005* (pp. 14–32). Springer.
- Friedman, J., Hastie, T., & Tibshirani, R. (2001). *The Elements of Statistical Learning*. Springer.
- Friendly, M. (2002). A brief history of the mosaic display. *Journal of Computational and Graphical Statistics*, 11, 89–107.
- Gaber, M. M., Zaslavsky, A., & Krishnaswamy, S. (2005). Mining data streams: A review. *ACM Sigmod Record*, 34, 18–26.
- Gabriel, K. R. (1971). The biplot-graphical display of matrices with applications to principal components analysis. *Biometrika*, 58, 453–467.
- Geoffrion, A. M., Dyer, J. S., & Feinberg, A. (1972). An interactive approach for multi-criterion optimization, with an application to the operation of an academic department. *Management Science*, 19, 357–368.
- Gettinger, J., Kiesling, E., Stummer, C., & Vetschera, R. (2013). A comparison of representations for discrete multi-criteria decision problems. *Decision Support Systems*, 54, 976–985.
- Goodman, L. A., & Kruskal, W. H. (1954). Measures of association for cross classifications. *Journal of the American Statistical Association*, 49, 732–764.
- Greco, S., Matarazzo, B., & Slowinski, R. (2008). Dominance-based rough set approach to interactive multiobjective optimization. In *Multiobjective Optimization* (pp. 121–155). Springer.
- Grinstein, G., Pickett, R., & Williams, M. G. (1989). Exvis: An exploratory visualization environment. In *Proceedings of Graphics Interface '89* (pp. 254–261).
- Hatzilygeroudis, I., & Prentzas, J. (2004). Integrating (rules, neural networks) and cases for knowledge representation and reasoning in expert systems. *Expert Systems with Applications*, 27, 63–75.
- Hintze, J. L., & Nelson, R. D. (1998). Violin plots: A box plot-density trace synergism. *American Statistician*, 52, 181–184.
- Hoffman, P., Grinstein, G., Marx, K., Grosse, I., & Stanley, E. (1997). DNA visual and analytic data mining. In *Proceedings of the 8th Conference on Visualization, Visualization '97* (pp. 437–441). IEEE.
- Hoffman, P. E., & Grinstein, G. G. (2002). A survey of visualizations for high-dimensional data mining. In *Information Visualization in Data Mining and Knowledge Discovery* (pp. 47–82). Morgan Kaufmann.
- Holden, C. M. E., & Keane, A. J. (2004). Visualization methodologies in aircraft design. In *Proceedings of the 10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference* (pp. 1–13). AIAA.
- Inselberg, A. (1985). The plane with parallel coordinates. *The Visual Computer*, 1, 69–91.
- Jain, A. K., Murty, M. N., & Flynn, P. J. (1999). Data clustering: A review. *ACM computing surveys (CSUR)*, 31, 264–323.

- Jaszkiewicz, A. (2002). On the performance of multiple-objective genetic local search on the 0/1 knapsack problem – A comparative experiment. *IEEE Transactions on Evolutionary Computation*, 6, 402–412.
- Jeong, M. J., Dennis, B. H., & Yoshimura, S. (2003). Multidimensional solution clustering and its application to the coolant passage optimization of a turbine blade.
- Jeong, M. J., Dennis, B. H., & Yoshimura, S. (2005a). Multidimensional clustering interpretation and its application to optimization of coolant passages of a turbine blade. *Journal of Mechanical Design*, 127, 215–221.
- Jeong, S., Chiba, K., & Obayashi, S. (2005b). Data mining for aerodynamic design space. *Journal of aerospace computing, information, and communication*, 2, 452–469.
- 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, 469–479.
- Jones, D. F., Mirrazavi, S. K., & Tamiz, M. (2002). Multi-objective meta-heuristics: An overview of the current state-of-the-art. *European journal of operational research*, 137, 1–9.
- Jourdan, L., Corne, D., Savic, D., & Walters, G. (2005). Preliminary investigation of the ‘learnable evolution model’ for faster/better multiobjective water systems design. In *Proceedings of the 3rd International Conference on Evolutionary Multi-criterion Optimization, EMO 2005* (pp. 841–855). Springer.
- Kampstra, P. (2008). Beanplot: A boxplot alternative for visual comparison of distributions. *Journal of Statistical Software*, 28, 1–9.
- Keim, D. (2002). Information visualization and visual data mining. *IEEE Transactions on Visualization and Computer Graphics*, 8, 1–8.
- Kendall, M. G. (1948). *Rank correlation methods*. Griffin.
- Klamroth, K., & Miettinen, K. (2008). Integrating approximation and interactive decision making in multicriteria optimization. *Operations Research*, 56, 222–234.
- Knowles, J., & Nakayama, H. (2008). Meta-modeling in multiobjective optimization. In *Multiobjective Optimization* (pp. 245–284). Springer.
- Kohonen, T. (1990). The self-organizing map. *Proceedings of the IEEE*, 78, 1464–1480.
- Koppen, M., & Yoshida, K. (2007). Visualization of Pareto-sets in evolutionary multi-objective optimization. In *Proceedings of the 7th International Conference on Hybrid Intelligent Systems, HIS 2007* (pp. 156–161). IEEE.
- Korhonen, P. (1991). Using harmonious houses for visual pairwise comparison of multiple criteria alternatives. *Decision Support Systems*, 7, 47–54.
- Korhonen, P., & Wallenius, J. (1988). A Pareto race. *Naval Research Logistics*, 35, 615–623.
- Korhonen, P., & Wallenius, J. (2008). Visualization in the multiple objective decision-making framework. In *Multiobjective Optimization* (pp. 195–212). Springer.
- Korhonen, P., Wallenius, J., & Zionts, S. (1980). A bargaining model for solving the multiple criteria problem. In *Multiple Criteria Decision Making Theory and Application* (pp. 178–188). Springer.
- Kruskal, J. B., & Wish, M. (1978). *Multidimensional scaling* volume 11. Sage Publications.
- Kudo, F., & Yoshikawa, T. (2012). Knowledge extraction in multi-objective optimization problem based on visualization of Pareto solutions. In *2012 IEEE Congress on Evolutionary Computation, CEC* (pp. 1–6). IEEE.
- Kumano, T., Jeong, S., Obayashi, S., Ito, Y., Hatanaka, K., & Morino, H. (2006). Multidisciplinary design optimization of wing shape for a small jet aircraft using kriging model. In *Proceedings of the 44th AIAA Aerospace Sciences Meeting and Exhibit* (pp. AIAA 2006–932).
- Kurasova, O., Petkus, T., & Filatovas, E. (2013). Visualization of Pareto front points when solving multi-objective optimization problems. *Information Technology and Control*, 42, 353–361.
- Larrañaga, P., & Lozano, J. A. (2002). *Estimation of distribution algorithms: A new tool for evolutionary computation* volume 2. Springer.
- LeBlanc, J., Ward, M., & Wittels, N. (1990). Exploring N-dimensional databases. In *Proceedings of the 1st IEEE Conference on Visualization, Visualization '90* (pp. 230–237). IEEE.
- Lewandowski, A., & Granat, J. (1991). Dynamic BILOT as the interaction interface for aspiration based decision support systems. *Lecture Notes in Economics and Mathematical Systems*, 356, 229–241.
- Liao, S.-H., Chu, P.-H., & Hsiao, P.-Y. (2012). Data mining techniques and applications – A decade review from 2000 to 2011. *Expert Systems with Applications*, 39, 11303–11311.
- Loshchilov, I., Schoenauer, M., & Sebag, M. (2010). A mono surrogate for multiobjective optimization. In *Proceedings of the 12th Annual Conference on Genetic and Evolutionary Computation, GECCO 2010* (pp. 471–478).
- Lotov, A. V., Bushenkov, V. A., & Kamenev, G. K. (2004). *Interactive decision maps: Approximation*

- and visualization of Pareto frontier. Springer.
- Lotov, A. V., & Miettinen, K. (2008). Visualizing the Pareto frontier. In *Multiobjective Optimization* (pp. 213–243). Springer.
- Lowe, D., & Tipping, M. E. (1997). NeuroScale: Novel topographic feature extraction using RBF networks. In *Advances in Neural Information Processing Systems 9* (pp. 543–549).
- MacQueen, J. B. (1967). Some methods for classification and analysis of multivariate observations. In *Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability* (pp. 281–297).
- Mañas, M. (1982). Graphical methods of multicriterial optimization. *Zeitschrift für Angewandte Mathematik und Mechanik*, 62, 375–377.
- Mareschal, B., & Brans, J.-P. (1988). Geometrical representations for MCDA. *European Journal of Operational Research*, 34, 69–77.
- Masafumi, Y., Tomohiro, Y., & Takeshi, F. (2010). Study on effect of MOGA with interactive island model using visualization. In *2010 IEEE Congress on Evolutionary Computation, CEC* (pp. 1–6). IEEE.
- McCane, B., & Albert, M. (2008). Distance functions for categorical and mixed variables. *Pattern Recognition Letters*, 29, 986–993.
- Meyer, B., & Sugiyama, K. (2007). The concept of knowledge in KM: A dimensional model. *Journal of Knowledge Management*, 11, 17–35.
- Michalski, R. S. (2000). Learnable evolution model: Evolutionary processes guided by machine learning. *Machine Learning*, 38, 9–40.
- Miettinen, K. (1999). *Nonlinear multiobjective optimization*. Kluwer Academic Publishers.
- Miettinen, K. (2003). Graphical illustration of Pareto optimal solutions. In *Multi-Objective Programming and Goal Programming* (pp. 197–202). Springer.
- Miettinen, K. (2014). Survey of methods to visualize alternatives in multiple criteria decision making problems. *OR Spectrum*, 36, 3–37.
- Morse, J. (1980). Reducing the size of the nondominated set: Pruning by clustering. *Computers & Operations Research*, 7, 55–66.
- Mukerjee, A., & Dabbeeru, M. (2009). The birth of symbols in design. In *ASME 2009 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference* (pp. 817–827). San Diego, CA, USA: ASME.
- Nagel, H., Granum, E., & Musaeus, P. (2001). Methods for visual mining of data in virtual reality. In *International Workshop on Visual Data Mining at ECML/PKDD 2001* (pp. 13–28).
- Nazemi, A., Chan, A. H., & Yao, X. (2008). Selecting representative parameters of rainfall-runoff models using multi-objective calibration results and a fuzzy clustering algorithm. In *Proceedings of the BHS 10th National Hydrology Symposium* (pp. 13–20).
- Ng, A. H. C., Dudas, C., Boström, H., & Deb, K. (2013). Interleaving innovization with evolutionary multi-objective optimization in production system simulation for faster convergence. In *Proceedings of the 7th International Conference on Learning and Intelligent Optimization, LION 7* (pp. 1–18). Springer.
- Ng, A. H. C., Dudas, C., Nießen, J., & Deb, K. (2011). Simulation-based innovization using data mining for production systems analysis. In *Multi-objective Evolutionary Optimisation for Product Design and Manufacturing* (pp. 401–429). Springer.
- Ng, A. H. C., Dudas, C., Pehrsson, L., & Deb, K. (2012). Knowledge discovery in production simulation by interleaving multi-objective optimization and data mining. In *Proceedings of the 5th Swedish Production Symposium* (pp. 461–471).
- Nonaka, I., & Takeuchi, H. (1995). *The knowledge-creating company: How Japanese companies create the dynamics of innovation*. Oxford University Press.
- Obayashi, S., Jeong, S., & Chiba, K. (2005). Multi-objective design exploration for aerodynamic configurations. In *Proceedings of the 35th AIAA Fluids Dynamics Conference and Exhibit* (pp. AIAA 2005–4666).
- Obayashi, S., Jeong, S., Chiba, K., & Morino, H. (2007). Multi-objective design exploration and its application to regional-jet wing design. *Transactions of the Japan Society for Aeronautical and Space Sciences*, 50, 1–8.
- Obayashi, S., & Sasaki, D. (2003). Visualization and data mining of Pareto solutions using self-organizing map. In *Proceedings of the 2nd International Conference on Evolutionary Multi-Criterion Optimization, EMO 2003* (pp. 796–809). Springer.
- de Oliveira, M., & Levkowitz, H. (2003). From visual data exploration to visual data mining: A survey. *IEEE Transactions on Visualization and Computer Graphics*, 9, 378–394.
- Oyama, A., Nonomura, T., & Fujii, K. (2010a). Data mining of Pareto-optimal transonic airfoil shapes

- using proper orthogonal decomposition. *Journal of Aircraft*, 47, 1756–1762.
- Oyama, A., Verburg, P., Nonomura, T., T. Hoeijmakers, M., & Fujii, K. (2010b). Flow field data mining of Pareto-optimal airfoils using proper orthogonal decomposition. In *Proceedings of the 48th AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition* (pp. AIAA 2010–1140). AIAA.
- Parashar, S., Pediroda, V., & Poloni, C. (2008). Self organizing maps (SOM) for design selection in robust multi-objective design of aerofoil. In *Proceedings of the 46th AIAA Aerospace Sciences Meeting and Exhibit* (pp. 2008–2914).
- Pohlheim, H. (2006). Multidimensional scaling for evolutionary algorithms – Visualization of the path through search space and solution space using sammon mapping. *Artificial Life*, 12, 203–209.
- Pryke, A., Mostaghim, S., & Nazemi, A. (2007). Heatmap visualization of population based multi objective algorithms. In *Proceedings of the 4th International Conference on Evolutionary Multi-Criterion Optimization, EMO 2007* (pp. 361–375). Springer.
- Quinlan, J. R. (1986). Induction of decision trees. *Machine Learning*, 1, 81–106.
- Quinlan, J. R. (1987). Simplifying decision trees. *International Journal of Man-machine Studies*, 27, 221–234.
- Quinlan, J. R. (2014). *C4.5: Programs for machine learning*. Elsevier.
- Rousseeuw, P. J. (1987). Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20, 53–65.
- Rousseeuw, P. J., Ruts, I., & Tukey, J. W. (1999). The bagplot: A bivariate boxplot. *The American Statistician*, 53, 382–387.
- Roweis, S. T. (2000). Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290, 2323–2326.
- Sammon, J. (1969). A nonlinear mapping for data structure analysis. *IEEE Transactions on Computers*, C-18, 401–409.
- Saxena, D. K., & Deb, K. (2007). Non-linear dimensionality reduction procedures for certain large-dimensional multi-objective optimization problems: Employing correntropy and a novel maximum variance unfolding. In *Proceedings of the 4th International Conference on Evolutionary Multi-Criterion Optimization, EMO 2007* (pp. 772–787). Springer.
- Saxena, D. K., Duro, J. A., Tiwari, A., Deb, K., & Zhang, Q. (2013). Objective reduction in many-objective optimization: Linear and nonlinear algorithms. *IEEE Transactions on Evolutionary Computation*, 17, 77–99.
- Sebag, M., & Schoenauer, M. (1994). Controlling crossover through inductive learning. In *Parallel Problem Solving from Nature - PPSN III* (pp. 209–218).
- Sebag, M., Schoenauer, M., & Ravisé, C. (1997a). Inductive learning of mutation step-size in evolutionary parameter optimization. In *Proceedings of the 6th Annual Conference on Evolutionary Programming* (pp. 247–261).
- Sebag, M., Schoenauer, M., & Ravise, C. (1997b). Toward civilized evolution: Developing inhibitions. In *Proceedings of the 7th International Conference on Genetic Algorithms* (pp. 291–298).
- Shin, W. S., & Ravindran, A. (1991). Interactive multiple objective optimization: Survey icontinuous case. *Computers & Operations Research*, 18, 97–114.
- Shneiderman, B. (1992). Tree visualization with tree-maps: 2-d space-filling approach. *ACM Transactions on Graphics*, 11, 92–99.
- Siegel, J. H., Farrell, E. J., Goldwyn, R. M., & Friedman, H. P. (1972). The surgical implications of physiologic patterns in myocardial infarction shock. *Surgery*, 72, 126–141.
- Simpson, T. W., Poplinski, J. D., Koch, P. N., & Allen, J. K. (2001). Metamodels for computer-based engineering design: Survey and recommendations. *Engineering with Computers*, 17, 129–150.
- Stuart, A. (1953). The estimation and comparison of strengths of association in contingency tables. *Biometrika*, 40, 105–110.
- Sugimura, K., Jeong, S., Obayashi, S., & Kimura, T. (2009). Kriging-model-based multi-objective robust optimization and trade-off-rule mining using association rule with aspiration vector. In *2009 IEEE Congress on Evolutionary Computation, CEC* (pp. 522–529). IEEE.
- Sugimura, K., Obayashi, S., & Jeong, S. (2007). Multi-objective design exploration of a centrifugal impeller accompanied with a vaned diffuser. In *Proceedings of the 5th Joint ASME/JSME Fluid Engineering Conference* (pp. 939–946). ASME.
- 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, 271–293.
- Taboada, H. A., Baheranwala, F., Coit, D. W., & Wattanapongsakorn, N. (2007). Practical solutions

- for multi-objective optimization: An application to system reliability design problems. *Reliability Engineering & System Safety*, 92, 314–322.
- Taboada, H. A., & Coit, D. W. (2006). 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* (pp. 43–48). Orlando, FL.
- Taboada, H. A., & Coit, D. W. (2007). Data clustering of solutions for multiple objective system reliability optimization problems. *Quality Technology & Quantitative Management*, 4, 191–210.
- Taboada, H. A., & Coit, D. W. (2008). Multi-objective scheduling problems: Determination of pruned Pareto sets. *IIE Transactions*, 40, 552–564.
- Taieb-Maimon Meirav, Limonad, L., Amid, D., Boaz, D., & Anaby-Tavor, A. (2013). Evaluating multi-variate visualizations as multi-objective decision aids. In *Human-Computer Interaction, INTERACT 2013* (pp. 419–436). Springer.
- Talbi, E.-G. (2009). *Metaheuristics: From design to implementation*. Wiley.
- Tenenbaum, J. B. (2000). A global geometric framework for nonlinear dimensionality reduction. *Science*, 290, 2319–2323.
- Thiele, L., Miettinen, K., Korhonen, P. J., & Molina, J. (2009). A preference-based evolutionary algorithm for multi-objective optimization. *Evolutionary computation*, 17, 411–436.
- Tsukimoto, H. (2000). Extracting rules from trained neural networks. *IEEE Transactions on Neural Networks*, 11, 377–389.
- Tukey, J. W. (1977). *Exploratory Data Analysis*. Pearson.
- Tusar, T., & Filipic, B. (2015). Visualization of Pareto front approximations in evolutionary multiobjective optimization: A critical review and the prosecution method. *IEEE Transactions on Evolutionary Computation*, 19, 225–245.
- Tušar, T. (2014). *Visualizing Solution Sets in Multiobjective Optimization*. Ph.D. thesis Jožef Stefan International Postgraduate School.
- Ulrich, T. (2012). Pareto-set analysis: Biobjective clustering in decision and objective spaces. *Journal of Multi-Criteria Decision Analysis*, 20, 217–234.
- Ulrich, T., Brockhoff, D., & Zitzler, E. (2008). Pattern identification in Pareto-set approximations. In *Proceedings of the 10th Annual Conference on Genetic and Evolutionary Computation, GECCO 2008* (pp. 737–744). New York, USA: ACM.
- Valdes, J. J., & Barton, A. J. (2007). Visualizing high dimensional objective spaces for multi-objective optimization: A virtual reality approach. In *2007 IEEE Congress on Evolutionary Computation, CEC* (pp. 4199–4206). IEEE.
- Valdés, J. J., Romero, E., & Barton, A. J. (2012). Data and knowledge visualization with virtual reality spaces, neural networks and rough sets: Application to cancer and geophysical prospecting data. *Expert Systems with Applications*, 39, 13193–13201.
- Van Der Maaten, L. J. P., Postma, E. O., & Van Den Herik, H. J. (2009). *Dimensionality Reduction: A Comparative Review*. Technical Report Tilburg University.
- Vesanto, J., & Alhoniemi, E. (2000). Clustering of the self-organizing map. *IEEE transactions on Neural Networks*, 11, 586–600.
- Vetschera, R. (1992). A preference-preserving projection technique for MCDM. *European Journal of Operational Research*, 61, 195–203.
- Žilinskas, A., Fraga, E. S., Beck, J., & Varoneckas, A. (2015). Visualization of multi-objective decisions for the optimal design of a pressure swing adsorption system. *Chemometrics and Intelligent Laboratory Systems*, 142, 151–158.
- Žilinskas, A., Fraga, E. S., & Mackut, A. (2006). Data analysis and visualisation for robust multi-criteria process optimisation. *Computers & Chemical Engineering*, 30, 1061–1071.
- Walker, D. J., Everson, R., & Fieldsend, J. E. (2013). Visualizing mutually nondominating solution sets in many-objective optimization. *IEEE Transactions on Evolutionary Computation*, 17, 165–184.
- Walker, D. J., Fieldsend, J. E., & Everson, R. M. (2012). Visualising many-objective populations. In *Proceedings of the 14th Annual Conference Companion on Genetic and Evolutionary Computation, GECCO 2012* (pp. 451–458). ACM.
- Witten, I. H., Frank, E., & Hall, M. A. (2011). *Data mining: Practical machine learning tools and techniques*. (3rd ed.). Elsevier.
- Xu, R., & Wunsch, D. (2005). Survey of clustering algorithms. *IEEE Transactions on Neural Networks*, 16, 645–678.
- Zhang, Q., & Li, H. (2007). MOEA/D: A multiobjective evolutionary algorithm based on decomposition. *IEEE Transactions on Evolutionary Computation*, 11, 712–731.
- Zhou, A., Zhang, Q., Tsang, E., Jin, Y., & Okabe, T. (2005). A model-based evolutionary algorithm

- for bi-objective optimization. In *2005 IEEE Congress on Evolutionary Computation, CEC* (pp. 2568–2575). IEEE.
- Zitzler, E., & Künzli, S. (2004). Indicator-based selection in multiobjective search. In *Parallel Problem Solving from Nature - PPSN VIII* (pp. 832–842). Springer.