European Journal of Operational Research 221 (2012) 469–479

Contents lists available at SciVerse ScienceDirect

European Journal of Operational Research

journal homepage: www.elsevier.com/locate/ejor

Invited Review

Synergies between operations research and data mining: The emerging use of multi-objective approaches

David Corne [a], Clarisse Dhaenens [b,c], Laetitia Jourdan [b,c,*]

[a] Heriot-Watt University, Scotland, United Kingdom
[b] Université Lille 1, Laboratoire d'Informatique Fondamentale de Lille, UMR CNRS 8022, Cité Scientifique, Bâtiment M3, 59655 Villeneuve d'Ascq cedex, France
[c] INRIA Lille-Nord Europe, Parc Scientifique de la Haute Borne, 40 avenue Halley, 59650 Villeneuve d'Ascq, France

ARTICLE INFO

Article history:
Received 23 September 2011
Accepted 23 March 2012
Available online 30 March 2012

Keywords:
State-of-the-art
Operations research
Knowledge-based systems
Knowledge discovery
Multi-objective optimization

ABSTRACT

Operations research and data mining already have a long-established common history. Indeed, with the growing size of databases and the amount of data available, data mining has become crucial in modern science and industry. Data mining problems raise interesting challenges for several research domains, and in particular for operations research, as very large search spaces of solutions need to be explored. Hence, many operations research methods have been proposed to deal with such challenging problems. But the relationships between these two domains are not limited to these natural applications of operations research approaches. The counterpart is also important to consider, since data mining approaches have also been applied to improve operations research techniques. The aim of this article is to highlight the interplay between these two research disciplines. A particular emphasis will be placed on the emerging theme of applying multi-objective approaches in this context.

1. Introduction

Data mining (DM) has recently seen an explosion of interest in many fields of applications, owing to the increasing amount of data available, and the growing understanding that deeper analyzes are far more valuable than simple summary statistics. Data mining is an inductive (not deductive) process. Its aim is to infer knowledge that is generalized from the data in the database. This process is generally not supported by classical DataBase Management systems. Data mining problems raise interesting challenges for several research domains, such as statistics, information theory, databases, machine learning, data visualization, and also for operations research (OR), since very large search spaces of solutions need to be explored. Hence, for several years, numerous research efforts using operational research methods to solve data mining problems have been reported, and several reviews of such approaches have been published [68,91,92]. However, the synergy between operations research (OR) and data mining (DM) is not a one-way street; as described by Meisel and Mattfeld, three kinds of synergies may be achieved [82]: 1/OR can contribute to the efficiency of DM techniques, 2/DM can increase the number of problems in which OR can be applied by means of a less rigorous model building process, 3/finally, increased system performance can result from complementary uses of these two research domains.

In this article we will use a simpler categorization of the synergies between DM and OR, which emphasizes two types of interaction, in terms of how OR and DM can contribute to each other. Hence, the first point of view (similar to the first of Meisel and Mattfeld's synergies) is to analyze how OR can contribute to the efficiency of DM techniques. The second point of view looks at how DM can contribute to OR methods. In our view, the second synergy of Meisel and Mattfeld, concerned with using DM techniques to better capture the structure of the underlying system, may be merged into our second type of DM/OR interaction, since it yields the same overall result of enhancing OR via deployment of DM.

Our first point of interest is to analyze how OR can be useful in the challenges faced by applications of DM. In other words, how OR approaches can contribute in helping DM difficult problems. We will see in this review that there are several answers, using several approaches, which all tend to center on using OR to deal with one or other NP-hard optimization problem that arises in a DM task. In particular, metaheuristics have been widely used in this context, and several books dedicated to metaheuristics and data mining have been published [26,35]. Meanwhile, multi-objective metaheuristic approaches are increasingly also being proposed in this context [61,59]. Thus, this article will pay a particular attention to this multi-objective aspect and methods that have been proposed for that. Therefore, the notion of quality criterion related to the objective function, for example, will be discussed.

* Corresponding author at: Université Lille 1, Laboratoire d'Informatique Fondamentale de Lille, UMR CNRS 8022, Cité Scientifique, Bâtiment M3, 59655 Villeneuve d'Ascq cedex, France. Tel.: +33 0 3 59 57 78 81.
E-mail addresses: david.corne@gmail.com (D. Corne), clarisse.dhaenens@lifl.fr (C. Dhaenens), laetitia.jourdan@inria.fr (L. Jourdan).

0377-2217/$ - see front matter © 2012 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.ejor.2012.03.039

The second fundamental question, when synergies between OR and DM are under analysis, is to understand how DM techniques can help OR methods. Even though this thread of research is less studied, significant such work is emerging [64]. The objectives of such a synergy may be for example, to either improve the quality of results obtained by OR approaches, or to speed up the execution of algorithms.

The aim of this review is to provide interesting pointers to how OR and DM can enrich each other. The remainder is organized as follows: the second section is designed to present to the OR community a short introduction to 'knowledge discovery', in order to help define the scope of this very general term and to make this article be self-content. It will describe the main data mining tasks and the principal and classical algorithms in this field. Section 3 will then deal with the first question: how operations research can help data mining. Section 4 is dedicated to the other side of the coin: how data mining may be useful for operations research techniques. In both Sections 3 and 4, a particular emphasis will be given to multi-objective models and methods. Section 5 will conclude the review and will suggest some interesting research perspectives for both communities.

2. Knowledge discovery: a brief introduction

'Knowledge discovery and data mining' (KDD) is a phrase that describes a large area of research concerned with discovering and exploiting the considerable amount of potentially useful knowledge that is often 'hidden' in databases. Such knowledge is regarded as hidden, since standard statistical techniques simply fail to find it, and the discovery of interesting rules or associations tends to be infeasible with exact algorithms owing to the size of the database. Data mining is at the heart of the KDD process (see Fig. 1). It allows us to extract useful information from large data sets or databases in reasonable time, usually by employing approximate algorithms. This discipline lays at the intersection of statistics, machine learning, data management and databases, pattern recognition, artificial intelligence, and similar domains. The reader interested in knowledge discovery in general, or datamining in particular, can refer to some of the following books [46,67,80]. In this section we focus on data mining tasks and algorithms (see Fig. 2). Each of these tasks are briefly described in this section, presenting their goals, and the algorithms most commonly associated with them.

2.1. (Supervised) classification

The aim of this task is to build a model that predicts the value of one variable from the known values of other variables. In classification, the variable being predicted, called the 'class', is categorical, and the task becomes regression when the predicted variable is numerical. Several approaches have been proposed. We will expose here briefly some of them, but to have a more general view of these methods, the reader may, for example, refer to [73].

2.1.1. K-nearest-neighbor

K-nearest-neighbor (K-NN) classification is one of the most fundamental and simple classification methods [20]. It is suitable for a classification study when there is little or no prior knowledge about the distribution of the data. The N-nearest-neighbor algorithm relies on the distances between examples in the feature space: an object is assigned to the most common class shared by its K nearest neighbors. It can be useful to weight the contribution of the neighbors, so that nearer neighbors contribute more to the average than more distant ones. If $K = 1$, then the object is simply assigned to the class of its nearest neighbor. The neighborhood is defined by the distance metric used, which is commonly the Euclidean distance.

2.1.2. Decision trees

Decision trees (or classification trees) are very popular for classification, since they are simple to understand and to interpret. A decision tree is built through a process known as *binary recursive partitioning*. This process recursively splits (or 'partitions') the data into groups. At each stage, the splitting is realized in a way that maximizes a score function for the split. The score function is chosen so that it favors the degree to which each individual group contains datapoints that are all of the same class. The main difference between different decision tree construction algorithms is the score function that is used to guide the splitting process. For example, *Information gain* is used by the ID3, C4.5 [101] and C5.0 tree generation algorithms, and is based on the concept of entropy used in information theory. *Gini impurity* is used by the CART algorithm [13] and measures how often a randomly chosen datapoint from the group would be incorrectly labeled if it was randomly labeled according to the distribution of labels within the group.

2.1.3. Naive Bayes

The Naive-Bayes classifier uses a probabilistic approach based on applying Bayes' theorem with strong (Naive) independence assumptions. To assign the class to a sample, it computes the conditional probabilities of different classes given the values of the features, and predicts the class with the highest conditional probability. Naive-Bayes is simple and can be applied to multi-class classification problems, but it assumes independence between variables, which is typically untrue (i.e. it is a naive assumption). In spite of its simplified assumptions, Naive-Bayes classifiers often work very well in many complex real-world situations.

2.1.4. Neural networks

Artificial neural networks (ANN) are widely used for classification and are a promising alternative to various conventional classification methods [124]. An artificial neural network is essentially a

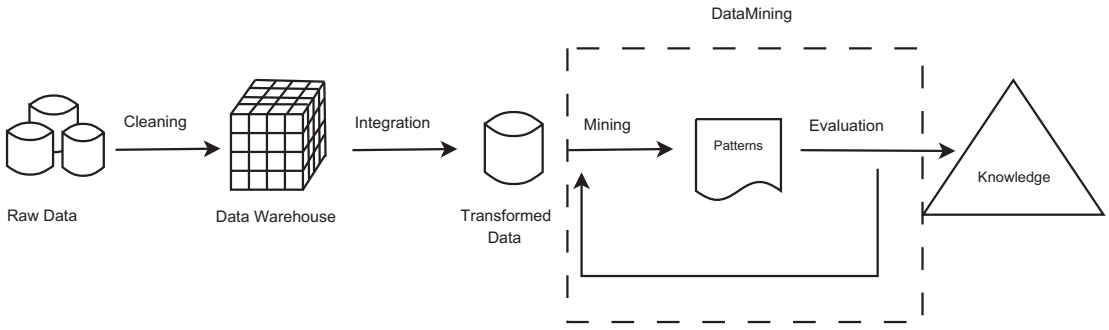

Fig. 1. An overview of the KDD process.

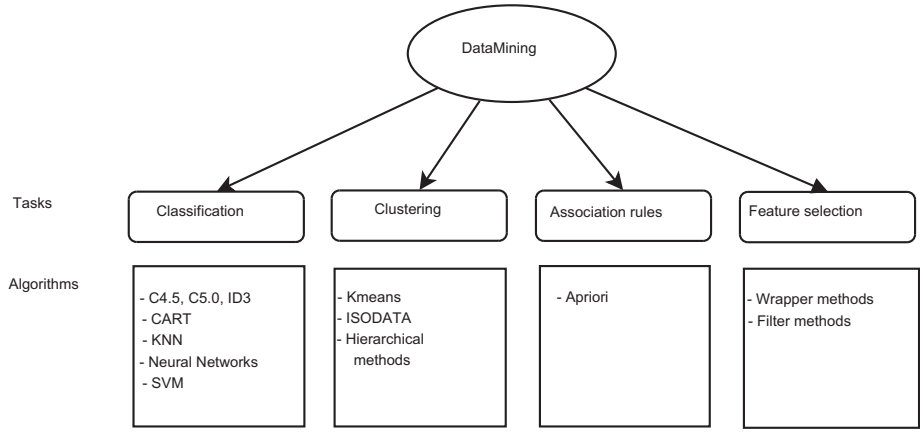

Fig. 2. An overview of tasks and main algorithms in data mining.

directed network of simple processors (called neurons) arranged in layers, with at least three layers. The first layer has input neurons, which send data via weighted links to the second layer of neurons; the nodes in the second layer perform a simple computation over the weighted input values that they receive, and then send the results onto the third layer, again via weighted links. The third layer, usually, consists of output neurons. The nodes in this output layer will, again, perform a simple computation on their inputs, and the result of this computation becomes the output of the network. More complex systems will have more layers.

This previous description corresponds to the feed-forward MultiLayer Perceptrons (MLPs), which is the most widely used models for addressing regression and classification problems. A data sample is presented to the MLP by presenting it as a vector of input values fed to the input later, and the resulting output of the MLP provides the class, or predicted numerical value, in the case of regression.

2.1.5. Support vector machines

Support vector machines (SVMs), derived from statistical learning theory, classify points by assigning them to one of two disjoint half spaces [19]. These half spaces are either in the original input space of the problem (for linear classifiers) or in a higher dimensional feature space (for non-linear classifiers). The assignment is realized by a kernel function. A specific property of SVMs is that they simultaneously minimize the empirical classification error and maximize the geometric margin. Hence they are also known as maximum margin classifiers. SVMs produce generally very good results in term of class-prediction accuracy, especially on small datasets.

2.2. Unsupervised classification or clustering

The second DM tasks exposed here deals with the clustering in which no variable has to be predicted but similarity between objects has to be identified. Hence, the cluster analysis task aims at decomposing or partitioning a (usually multivariate) data set into groups so that points in a group are similar to each other, and are as different as possible from points of other groups. Most known methods are explained below but more information about clustering can be found in [119].

2.2.1. Partition based methods

In partition based clustering, the objective is to partition a data set into K disjoint sets of points such that points of a set are as homogeneous as possible. The homogeneity is computed thanks to a score function that is often based on a similarity notion and the objective is to minimize the dissimilarity between each point and the centroid of the cluster it is assigned to. The centroid of a set could be an actual data point, or a 'position' in feature space. The most well-known algorithm of this category is K-means [79]. In its most basic version, this algorithm begins by randomly picking K cluster centers, and then labeling each point according to its nearest cluster center. Now that each point is labeled, a new cluster center can be calculated according to the centroid of each labeled cluster. The algorithm then iterates until there is no longer any movement in the cluster centers. There are several variants of the K-means algorithm, for example, the cluster centers can be objects from the dataset, or splitting and/or merging of clusters may be allowed within the algorithm.

2.2.2. Hierarchical methods

Two distinct types of hierarchical methods are identifiable: the agglomerative ones, which start with singleton clusters and gradually merge them, and the divisive ones, which begin with all the data in a single cluster, and gradually divide into smaller clusters. The agglomerative methods are the most used. Hierarchical methods of cluster analysis permit a convenient graphical display in which the entire sequence of merging (or splitting) of clusters is shown. Because of its tree-like structure, the resulting display is called a dendrogram.

Agglomerative methods. Agglomerative methods are based on measures of distance between clusters. They iteratively merge the two clusters that are the nearest to each other, to reduce the number of clusters. Usually the starting point of the process consists of each point in its own cluster. Then, the merging is realized until just one cluster containing all the data points is obtained.

Different distance measures between clusters have been proposed, and lead to different algorithms: for example, the single-link hierarchical clustering algorithm merges the two clusters with the smallest minimum pairwise distance. Another variant is the complete-link hierarchical clustering algorithm, where the two clusters with the smallest maximum pairwise distance, are merged.

Divisive methods. Divisive methods begin with a single cluster which contains all the data points and splits it into subsets. The process of splitting is iterated as far as necessary and will end with singleton clusters.

2.2.3. Biclustering

Biclustering, co-clustering, or two-mode clustering, allow simultaneous clustering of the rows and columns of a matrix [85,74]. This subspace clustering approach allows the algorithm to treat attributes and objects interchangeably, and to find relationships between elements regarding these two directions. Different types

of biclusters may be required: either biclusters with constant values, or constant values on lines or columns, or biclusters with coherent additive or multiplicative values. These approaches have been widely used in bioinformatics and many algorithms, mainly based on statistics, have been proposed. The complexity of biclustering problems depends on the exact formulation of the problem, but most of them are $\mathcal{NP}$-complete.

2.3. Rule mining

The third DM task discussed here deals with the problem of discovering association rules. It was first formulated in [1] and was called the market-basket problem. The initial problem was the following: given a set of items and a large collection of sales records, which consist of a transaction date and the items purchased in that transaction, the task is to find significant relationships between the items contained in different transactions. Since this first application, many other problems have been studied with association rules that may be defined in a more general way. Let us consider a database composed of transactions (records) described according to several, maybe many, attributes (columns). Association rules provide a very simple (but useful) way to present correlations or other relationships among attributes expressed in the form $A \Rightarrow C$ where A is the antecedent part (condition) and C the consequent part (prediction). A and C are sets of attributes that are disjoint. The best-known algorithm to mine association rules is *A-priori*, proposed by Agrawal and Srikant [2]. This two-phase algorithm first finds all 'frequent item sets' (sets of items that often occur together within transactions) that have at least a given minimum level of *confidence*. This is done via efficient search exploiting the downward closure property of *support*. Support and Confidence are widely used metrics that will be defined in Section 3.3. A lot of improvements upon the initial method, as well as efficient implementations (including parallel implementations) have been proposed to be able to deal with very large databases [3,10,120,123].

2.4. Feature selection

This last tasks deals with the feature selection that may be applied before one of the three previously presented tasks. Indeed, an important problem in data mining is the huge size of datasets and the presence of too many attributes. The selection of some attributes to retain, and consequently others to remove, is frequently necessary to reduce the computational cost, to simplify a model, or to have an accurate discrimination, and usually all three of these. Indeed, including all attributes can often lead to a worse classification model. For example, some attributes could be redundant or unrelated to the class variable. The objective of feature selection is to find a subset of p' relevant variables where $p' \ll p$, but where this subset retains all or most of the information necessary for discriminating between classes (see [45]). Therefore, the main goal of feature selection in supervised learning is to find a feature subset that produces higher classification accuracy. In unsupervised learning, feature selection aims to find a good subset of features that forms high quality clusters for a given number of clusters.

3. Exploiting operations research for data mining

As previously indicated, data mining (DM) tasks lead to challenging combinatorial optimization problems. Hence, operations research (OR) can give interesting contributions to the field as exposed in [82,68,91,92] and some specific surveys focus on evolutionary algorithms for data mining [37,38]. This part is dedicated to the description of OR approaches that have been used in DM and a peculiar interest is given to multi-objective approaches.

Let us remark that one of the difficulties in turning a DM task into an optimization problem is to define the criterion to optimize. The choice of the optimization criterion, which measures the quality of the knowledge extracted in a candidate rule (or other model) is very important, and the quality of the results of the approach depend of it. Indeed, developing a very efficient method that does not use the right criterion will lead to obtaining "the right answer to the wrong question"! The optimization criterion can be either specific to the DM task or dependent on the application, and several choices exist which explains that multi-objective approaches have been proposed to deal with all these possible criteria. Once this is done, another question occurs about the complexity of the generated problems and the methods that can be used.

Hence, in this section, for the previously described DM tasks, modeling aspects as well as their solutions are discussed. An illustrative application is also given.

3.1. Supervised classification

Supervised classification deals with the task of inferring a function from supervised training data in order to classify unknown data.

3.1.1. Modelling approaches

One of the peculiarities of supervised classification is the opportunity to compute exactly the performance of a classifier on training data. Nevertheless, several alternative performance indicators may be used such as: maximization of the minimum deviation (MMD), minimization of the sum of deviations (MSD), the number of misclassified instances, and so on. In an OR context, these indicators may be considered as optimization criteria.

Mathematical programming techniques have been used for various data classification problems. A recent work of Xu et al. [117] reports many references on this subject. They indicate that linear or non-linear classifiers can be designed via a series of algebraic equations and the relationships between training samples and classifiers can be expressed through logical constraints. In linear programming the two most commonly used criteria are MMD and MSD. In mixed integer programming (MIP), an additional binary variable is added to indicate whether the sample is correctly classified or not. Hence the number of misclassifications may be used to evaluate the quality of the model.

It is well known that learning algorithms that are based only on error minimization do not guarantee models with good generalization performance. In order to take into account other criteria, such that generalization performance may be improved, some multi-objective models have been proposed. In a two-group classification context, for example, a bi-objective margin maximization model has been used [17]. For this problem, a goal programming model based around the concepts of non-standard preference functions and penalty function modeling has also been proposed [62] as well as an extension for the multi-group classification problem [4].

3.1.2. Solution methods

Operations research approaches used to solve classification problems are manifold. Exact optimization methods have often been proposed (LP, MILP), but clearly have difficulty with larger problems, while heuristic approaches are increasingly common, to deal with larger size problems or multi-objective ones.

For example, in the case of multi-class classification, a MILP model has been proposed. It uses Hyper-boxes to capture the discrete regions of the training samples. Special constraints are introduced to avoid overlapping of boxes that belong to different classes. An iterative solution algorithm is then proposed to improve the training and prediction accuracy by allowing multiple boxes for each class [114,117].

In the case of bi-objective optimization for a two-group classification task via margin maximization, the SVM approach has been used, and an interesting set of Pareto optimal solutions is obtained using several β values for the SVM model [17].

Decision trees have the advantage of being easily understandable. However, this technique has drawbacks (correlated variables are not considered, explanatory tree structures are not stable if multiple samples are taken from the same population, and so on). To overcome these drawbacks, Sorensen et al. propose a genetic algorithm that finds a set of diverse classification trees, all having good explanatory power [109]. Then, from this set of trees, the data analyst is able to choose the tree that fulfils his requirements. Specific operators are proposed to deal with such binary trees. On their side, Fieldsend et al. proposed a particle swarm optimization (PSO) algorithm to train near optimal decision trees. They successfully apply this approach both in a single objective formulation (minimizing misclassification cost), and a multi-objective formulation (trading off misclassification rates across classes) using a Pareto dominance function that allows them to find a set of decision trees of best compromise [33].

As artificial neural networks (ANN) have been widely used for classification, some optimization methods have been proposed to train ANNs. In particular multi-objective approaches have been used to design models which aim to maximize accuracy while minimizing structural complexity. Caballero et al. recently proposed a memetic Pareto evolutionary approach based on accuracy and sensitivity measures. This approach introduces an improved *Rprop* (resilient backpropagation) algorithm as a local search within NSGA2 [25], and showed interesting results on classical benchmarks [32].

3.1.3. Example of multi-objective application

Classification occurs in many application domains. We may cite, among others, industry, medecine, credit scoring, quality control, medical diagnosis, and so on. In the context of malicious network traffic analysis, the objective is to identify intrusions. Therefore, classifiers have been proposed to discriminate information about the network traffic. Ostaszewski et al. proposed to address this problem using a multi-objective approach with specificity and sensitivity criteria [94]. They proposed a multi-objective version of Gene Expression Programming (GEP) that will identify the Pareto front in the search space. In addition, a crowding distance mechanism is used to provide a good spread of individuals on the front during the evolutionary process. The proposed approach is validated using data provided by a network traffic simulator. Results obtained show high classification capabilities.

3.2. Unsupervised classification: clustering

3.2.1. Model

Clustering deals with the problem of grouping similar objects into homogeneous groups. Unlike classification, clustering does not deal with known categories and the notion of similarity has to be defined with care. This notion is closely linked to the notion of distance, which may be expressed using several functions, depending on the nature of the data (Euclidean distance, Manhattan distance, Hamming distance, and so on). Moreover, in order to assess the quality of clusters found, several measures have been proposed. The quality measure should indicate whether clusters have a high similarity within a cluster and low similarity between clusters. We may cite: the *DaviesBouldin index* [22], the *Dun index* [28], and more recently the *cluster diameter* [103], the *overall deviation*, the *connectivity* [49], among others. We note that the efficiency of a clustering approach may also be validated using some classification datasets in order to evaluate how well the resulting clusters (obtained without using class information), discriminate between the "real" classes.

As the quality of clusters depends on two complementary measures, several multi-objective models have been proposed. For example, Handl and Knowles proposed, in a bi-objective scheme, the *overall deviation* to measure the overall intra-cluster re-partition of the data and the *connectivity* to evaluate the degree to which neighboring objects have been placed in the same cluster [50].

Another important aspect when dealing with unsupervised clustering is to determine the number of clusters. Often this number is considered as a parameter that may vary (such as in the *k-means* like approaches). But new attempts tend to extract this number automatically.

3.2.2. Solution methods

There is a large number of approaches to solve the clustering problem, including optimization based methods that involve mathematical programming models for developing efficient and meaningful clustering schemes. Exact and heuristic algorithms for these models have been proposed. For example, Saglam et al. proposed, in a customer segmentation context, to formulate the clustering problem as a mixed-integer programming problem with the objective of minimizing the maximum cluster diameter among all clusters [103]. They develop a heuristic approach that improves computational expense dramatically without compromising optimality in most cases.

As these problems are mostly $\mathcal{NP}$-hard, many heuristics and meta-heuristics have been proposed. As indicated in [52], evolutionary algorithms (EAs) have been widely used for clustering. A good survey on using evolutionary algorithms for clustering can be found in [55]. When dedicated to the single-objective case, the most commonly employed objectives are based on cluster compactness, as these measures provide smooth incremental guidance in all parts of the search space. Several encodings have been proposed, from straightforward ones, with a gene for each data item and its allele value specifying the cluster to which the data item should belong, to more complex ones [30]. In addition, in order to reduce the size of the search space, indirect encodings have been used, which store information about how to build the clusters rather than the clusters themselves. Interest in multi-objective clustering using meta-heuristics, and in particular EAs, has recently grown [72], and a recent book is dedicated to this subject [81]. In this domain, an interesting work is that of Handl and Knowles that proposes MOCK, a multiobjective clustering with automatic *k*-determination [52]. This algorithm has the advantage of keeping the number of clusters dynamic which is an important point in a unsupervised context.

3.2.3. Example of multi-objective application

Clustering is encountered in many fields. In bioinformatics, for example, clustering and bi-clustering approaches have been proposed to deal with microarray data. DNA microarray experiments are of great interest and importance for biologists, thanks to their ability to simultaneously measure the expression and interactions of thousands of genes. Much data is generated and need to be analyzed. As it is often observed that a subset of genes are coexpressed under a subset of conditions, but behave almost independently under other conditions, biclustering offers the possibility to identify these correlations. A recent example is the work of Mitra and Banka, who proposed a new quantitative measure to evaluate the quality of the biclusters and a multi-objective evolutionary biclustering framework which incorporates local search strategies [86]. Their experiments show the promise of such an approach in dealing with microarray data. A more recent example of a similar work is [105].

3.2.4. Challenges/post-optimization

Once clusters have been found, in this unsupervised context, it stays difficult to assess the quality of the partition. In their recent

work, Fernandez et al. suggest ways to handle multiple criteria and associated preferences in cluster analysis [31]. Their method (based on comparing cluster centers and an average net flow score betweem clusters) aims at building a ranking of the set of clusters.

3.3. Rule mining

3.3.1. Model

The task of discovering effective association rules may be seen as a combinatorial optimization problem, as rules are combinations of attributes. As the number of attributes may be very large (perhaps thousands), efficient methods are direly needed. We note that a specific case of rule mining deals with classification rules where the consequent is the same for every rule. This may be seen as a straightforward classification task; however, the models and methods used for this are close to those used more generally in rule mining, hence it will be considered in this section.

In order to solve the association rules discovery problem as a combinatorial optimization problem, the optimization criterion or criteria must first be defined. A lot of measures exist for estimating the quality of association rules. The most widely used measures are the *support* and the *confidence* (or accuracy). The support of a rule represents the proportion of transactions that verify it in the database and allows us to estimate the generality of the rule $\left(Sup(A \Rightarrow C) = P(A \cup C) = \frac{|A \cup C|}{|DB|}\right)$. The confidence measure estimates the validity of the rule by measuring the proportion of transactions in the database that verify the prediction among those which verify the condition $\left(Conf(A \Rightarrow C) = P(C|A) = \frac{Sup(A \cup C)}{Sup(A)}\right)$. It is expected that a high value of confidence expresses a strong association rule. However, there are several ways in which this value may be high, but the rule is nevertheless not a useful one. This may happen, for example, when the antecedent disconfirms the conclusion by reducing the probability of this conclusion. Therefore, measures such as the *lift ratio* measure has been introduced. It compares the ratio of the observed support to that expected if A and C were independent $\left(Lift(A \Rightarrow C) = \frac{Sup(A \cup C)}{Sup(A) \times Sup(C)}\right)$. Many other quality measures have been proposed to evaluate association rules. For an overview, readers can refer to Freitas [36], Tan et al. [112], Hilderman et al. [53] or Geng et al. [41].

Let us remark that Bayesian confirmation measures have also been proposed to asses the quality of rules. These measures allow to indicate in what degree an antecedent confirms a conclusion. In their analysis, Greco et al. study the monotonicity of several of these measures and show that only two of them have this property, which may be useful in some context [43]. However, as Olafsson indicates "relevant good measures and factors for identifying good association rules need more exploration" [92]. Indeed, for example, selecting frequent rules (with a high support) is not always interesting as it may lead to the production of trivial rules not valuable for the decision making.

Given the importance of and variety of quality measures in this context, the idea of multi-criteria models provides a promising alternative, since it avoids selecting a single quality measure, and instead enables the use of multiple distinct measures. Therefore, in her work, Szczech analyzes relationships between interestingness measures and enclosure relationships between the sets of non-dominated rules in different evaluation spaces [110] in order to propose a multicriteria evaluation space in which the set of non-dominated rules will contain all optimal rules with respect to any attractiveness measure with the monotonicity property. Another interesting and useful idea in this context has been proposed by Khabzaoui et al. [69,70], which attempt to analyze experimentally the correlation between quality measures of association rules and eleven measures (the most commonly used measures) introduced by the statistics, probability, information theory and data mining communities. This work leads to the proposition of a multi-objective model to deal with this data mining task. On their side, Brzezinska et al. study the replacement of the confidence measure by other confirmation measures in a Pareto context where both support and a confirmation measure are optimized [14]. This approach has been compared to the joint optimization of support and anti-support, and experiments show that optimizing this couple of objectives allows to obtain solutions optimizing support and confirmation measure. This offers interesting perspectives for multi-objective optimization.

3.3.2. Solution methods

Considering the association rules task as an optimization problem, several different potential approaches emerge that can be used to address it. Exact methods may be used in order to solve small instances, or heuristics, and particularly metaheuristics, are available to find approximate solutions on large instances, which are much more common.

The *A-priori* algorithm (see part Section 2.3) can be considered as an exact approach, since it enumerates all the rules respecting a minimal level of support and confidence. However, even though this method has received great attention, its main drawback is that it is limited to the search for a specific kind of rule. This may be interesting in the supermarket transaction and similar applications context, but not for many other applications.

Hence, most methods developed to deal with large-scale applications use heuristics or metaheuristics. In his book, Freitas gives fundamental insights on developing an EA for rule mining [35], and many approaches have been proposed since; in particular, multi-obejctive methods are of growing popularity. For example, Ishida et al. explore a Greedy Randomized Adaptive Search Procedure (GRASP) with path-relinking to create rules and to build the Pareto Front covering the tradeoff betwen sensitivity and specificity criteria [57]. Another multi-objective approach has been proposed to deal with classification rules (and in particular partial classification), and NSGA2 [25] has been compared to an algorithm able to enumerate the *cc-optimal* (coverage-confidence optimal) subset of rules [23].

3.3.3. Example of multi-objective application

The typical and original application of rule mining deals with supermarket basket analysis, where the objective consists in determining if certain groups of items are consistently purchased together. Following this initial application, many other fields have used this approach to extract models from data. This has been the case in DNA microarray data analysis; for example, Khabzaoui et al. proposed to analyze microarray data with an association rule based technique, in order to determine associations between differently regulated genes. The association rule problem is then modeled as a multi-objective combinatorial optimization problem and solved using an evolutionary algorithm based on genetic algorithms. Therefore, specific mechanisms (mutation and crossover operators, elitism, and so on) are designed for this task [69]. In order to improve the quality of the rules obtained, cooperative approaches are proposed [65,70].

3.3.4. Challenges for OR: integrating domain knowledge

Association rules represent a very general model that allows the combination of different types of attributes, different types of rules, and also the enrichment of rules using domain knowledge. In their work, for example, Kuene and Weistroffer proposed to incorporate decision makers prior domain knowledge into the data mining process using multicriteria analysis [75].

In the context of classification rules, Slowinski et al. propose an interesting approach, the *dominance-based rough set approach* (DBRSA), that models the preferences of a decision maker with easily understandable decision rules of the type *"if antecedent then decision"* [42,107]. This approach allows a multicriteria classification

[9]. DBRSA rules can express the antecedents using the more general preference relations $\geqslant$ and $\leqslant$ [77]. This approach manages to combine optimization and decision making and offers good opportunities to develop useful domain models. This complementarity is also very useful to deal with the large number of rules often generated with multi-objective approaches.

3.4. Feature selection

3.4.1. Model

The general task of feature selection can be formulated as an optimization problem. Binary values of the variable x_i are used in order to indicate the appearance ($x_i = 1$) or the absence ($x_i = 0$) of the feature f_i in the globally optimal feature set. Then, the problem is formulated as $max_{x=(x_1,\ldots,x_n)\in\{0;1\}^n} F(x)$. The task is therefore to find F, and the main difficulty is that this problem is highly task and domain dependent. E.g. the problem is of a different nature depending on whether we are considering supervised learning or unsupervised learning.

Feature selection for classification can be classified in three classes depending on how the selection process is combined with the classifier: the wrapper approach, the filter approach and the embedded approach. The wrapper approach model uses learning algorithms during the feature selection process and assesses the selected features by the learning algorithm's performance by using, for example, accuracy, sensitivity or specificity. The filter approach model considers statistical characteristics of a data set directly without involving any learning algorithm. Such filter models have been proposed with many different statistical feature selection measures, such as the correlation feature-selection (*CFS*) measure, the minimal-redundancy-maximal-relevance (*mRMR*) measure, the discriminant function and the Mahalanobis distance. In the embedded approach model, the learning algorithm uses its own embedded feature selection algorithm (either explicit or implicit); a good example is machine learning tools like decision trees. Molina et al. provide a comprehensive classification of measures and associated approaches [87].

3.4.2. Solution methods

Different approaches have been investigated depending of the size of the data set, the kind of approach (filter or wrapper) and the chosen measure of quality. For example, on small datasets, mixed Integer Programming has been used [56] and branch and bound algorithms have been investigated. In [88], B&B (Branch & Bound) provides an optimal search algorithm that uses a given threshold β (specified by the user). Branches are pruned below any node at which the evaluation is lower than β. However, this approach can only be used with a single quality objective. ABB (Automatic Branch & Bound) [78] is a variant of B&B in which the threshold is automatically set. An hybrid approach QBB (Quick Branch & Bound) [21] has been designed by using both LVF, a classical random approach for feature selection, and ABB.

Metaheuristics constitute a good choice to explore large datasets. Several articles propose approaches based on local search or population based algorithms [111] and a comparison between classical machine learning approaches and metaheuristics is proposed in [122].

3.4.3. Example of application

Optimization approaches to feature selection has been used in several domains. In particular, in the last decade, microarray datasets consisting of thousands of genes have provided a common platform for gene selection studies. Feature selection in the gene microarrray context aims at identifying a (small) subset of informative genes in order to obtain high predictive accuracy for classification.

In [39], Garcia et al. try to select genes from microarray data to discover genes involved in cancer. The authors propose to model the problem as a multiobjective optimization problem with three objectives: minimize the number of selected genes, maximize sensitivity, and maximize specificity. The sensitivity and specificity are computed by building a classification model from the reduced dataset using support vector machines. The optimization problem is solved using a multi-objective metaheuristic which attempts to simultaneously optimize these three objectives. The results suggest that the proposed approach is highly appropriate for solving the gene selection task, outperforming the compared techniques for all the datasets.

4. Using knowledge discovery to help operations research

The performance of operations research algorithms can be improved by integration with data mining techniques for different aims. Such approaches can help:

- to speed up the search process,
- to improve the quality of the obtained results,
- to tune the algorithm,
- and so forth.

These different aims will be used in the following to organize the presentation of the existing approaches. In some cases, such metaheuristics that use data mining (also called machine learning) to improve their behavior are called "learning metaheuristics".

Let us remark in addition that, this integration of data mining and operations research could be also categorized on the basis of three criteria:

- *when* **the knowledge is discovered**: the discovery can be realized before the search (an "off-line"/"a priori" strategy) or dynamically during the search (an "online" strategy),
- *why* **the hybridization is realized**: to improve the efficiency of the search by reducing the computation time, to improve the quality of the obtained solutions by guiding the search towards promising areas,
- *where* **the cooperation is realized**: to create an evaluation function, to define the parameters of the search method, to determine an initial solution, to design operators for metaheuristics, and so on.

4.1. Increasing the quality of the results of OR algorithms

The performance of metaheuristics can be improved by using data mining to incorporate knowledge and to allow a metaheuristic to tune its parameters [7] in an a priori manner or dynamically in an online manner [8,15,44]. For example, in [24], the authors use a technique of Reinforcement Learning, the Q-Learning Algorithm, for the constructive phase of the GRASP metaheuristic and to generate the initial population of a genetic algorithm. The proposed methods are applied to the symmetrical traveling salesman problem.

When the problem to be solved is itself a data mining task, such as to find optimal clusterings of data, some authors use a data mining technique in co-operation with the search algorithm [29,51,116].

In a real environment, the performance of optimization methods is directly linked to the quality of the data that is used to provide estimates of the quality of the optimized solution. It is then useful and important if the data can be augmented or modeled in a way that makes it suitably representative to supply better quality estimates. Some authors do this by using data mining techniques. For example, a regression tree approach is used and positively tested in order to estimate lead times in make-to-order manufacturing [95].

4.2. Speeding up OR algorithms

The first way to speed up OR algorithms is to reduce the cost of the computation of objective function when it is very expensive to compute. This kind of approach is frequent for design problems [58]. A good way is to use surrogate models in place of the full fitness evaluation function. The surrogate model is usually learned from online data (candidate solutions and their evaluated fitnesses) and then the surrogate model replaces the full evaluation model part of the time. This allows very large numbers of approximate evaluations to be done in a very short time, hopefully guiding the research in the right direction [76,93]. Usually the approximation is learned *a priori* by using a data mining algorithm. Commonly used data mining algorithms for this purpose are based on Artificial Neural Networks and Support Vector Machines. For example in [40], the fitness function is learned by a neural network. Decision trees have also been used for this purpose, with success, for example, in optimization of water distribution systems [63,115], and general function optimization [106].

The simplest variant of this approach, sometimes used to speed-up evolutionary algorithms, is called fitness inheritance. Fitness inheritance is proposed for tackling the cost of evaluation functions in some applications: instead of evaluating each individual, a certain percentage of the individuals are evaluated indirectly by interpolating between the fitness of their parents [27]. The Bayesian optimization algorithm (BOA) uses a more sophisticated variation of this principle [97,98].

Another possibility is to estimate fitness on the basis of distance between solutions, and to compute a guess of the objective function value by using the nearest neighbor algorithm [34] or using representatives that are selected via the use of clustering algorithms [60,71,121].

The second way to speed-up a search algorithm is to find interesting and useful properties of the candidate solutions that can be exploited to reduce the size of the search space. An example of this would be to learn good schemata (i.e. descriptions of subspaces or subsets of solutions). A good schema is one such that the solutions it represents tend to be better quality than those outside it. This has been explored for binary representation [47,48], and can be explored for any oher kind of representation (LEM [83,84]). In this context, DM-GRASP, an hybridization of GRASP with datamining, has been proposed. In DM-GRASP, after executing a significant number of GRASP iterations, the data mining process, based on rule mining, extracts patterns from an elite set of solutions which will guide the following iterations [102,104]. This technique has been used successfully for different problems: set packing problem [102], pmedian problem [99]. A new version with path-relinking hybrization has also been proposed [5]. We can notice that as multiobjective interest is growing a multiobjective DM-GRASP has been recently proposed [66]. In this approach, the search process is divided into two phases. In the first one, the DM-GRASP is applied to obtain an initial set of high quality solutions dispersed along the Pareto front. Then, the search efforts are intensified on the promising regions around these solutions through the second phase. The greedy randomized path-relinking with local search or reproduction operators are applied to improve the quality and to guide the search to explore the non-discovered regions in the search space.

As previously indicated, the majority of these approaches are focused on design problems in industry which have often many objectives and recently some attention has been given to how to integrate datamining to search interesting properties of multiobjective solutions. For example, in LEMMO [63,115], the authors proposed an adaption of the LEM algorithm for multiobjective optimization and its application for the design of water systems. They used the C4.5 rule induction algorithm as the machine learning component and NSGA2 [25] as the multi-objective evolutionary shell. LEMMO, which mainly differs from its originating algorithm as the decision tree is generated for each objective in turn, can operate both on integer and real variable representations. After every given number of evaluations of the expensive simulation function the C4.5 component is triggered, induction rules are learned and these are then used to modify the child population and hopefully speed up the search. Rules are compound *if then* statements generated for each leaf of the decision tree by making a conjunction of all the tests (i.e. nodes) encountered on the path from root (i.e. first node) to the leaf.

4.3. Using DM to select the OR algorithm based on instance properties

Data mining considerations can also help to find exploitable properties of problem instances and, based on these properties, aid the design of more effective search algorithms. In the SAT community, several studies have been conducted to study algorithm performance empirically, particularly focusing on identifying features that correlate with the empirical hardness of problem instances with data mining algorithms [89,90]. Another example, in the context of capacitated location routing problem, Barreto et al. [6] investigated several hierarchical and non-hierarchical clustering techniques to aid heuristic search. In a comprehensive analysis, they were able to provide guidelines for the appropriate choice of heuristic to use based on information about the instance. Along these lines, there is a growing body of research which is sometimes called 'per-instance tuning' which is generally concerned with using data mining to learn models (such as decision trees or neural networks) that can predict a good way to configure the search algorithm, based on input properties of the problem instance [54,96,16,118,100].

For example, Smith-Miles et al. [108] examined 75,000 instances of a single-machine earliness/tardiness problem. For each instance they compared the earliest-due-date heuristic (EDD) and the shortest-processing time heuristic (SPT). Various learning methods are then applied, which build a model able to predict the best choice of heuristic (either EDD or SPT) based on features of the instance (such as mean due date, or range of processing times), with quite promising results.

5. Conclusion

As we have aimed to demonstrate in this review, data mining and operations research already share a common history. The success of the interaction between them also motivates the further exploration of other connected domains, such as statistical learning, to find other opportunities for fruitful combinations [11]. Indeed, as we have seen in this survey, OR can help the data mining process and DM can help OR. We expect that the synergy between these two domains will continue to blossom, especially in the light of the surge of interest in both communities for using multi-objective approaches. Challenging questions still arise, and we point out two such questions here.

First, how can we integrate more domain knowledge while solving a problem? In an optimization context, for example, DM can help provide insight about how the best way to solve the problem might depend on instances of the problem, or it may help us learn during optimization, so that we can speed up the process by learning rules that help us avoid areas of the solution space where the solution quality is unpromising. Such integration of knowledge can be extremely helpful on larger and more complex problems. In a similar manner, to solve a data mining task, it may be useful to integrate knowledge about the domain. A good opportunity for this is given by multi-objective approaches, which enable us to combine objective functions specific to the data mining task with separate criteria

that exploit domain knowledge. In both OR and DM there are many ways knowledge can be integrated, but the research community as a whole lacks an overall theory or agreed set of guidelines that could make the process less *ad hoc*, and could help identify further possibilities and mechanisms for this integration.

Then the second question arises. How can we integrate preferences from an expert? This question is very important in the context of multi-objective approaches, which have the advantage of producing good compromise solutions, but have the drawback of usually producing too many solutions. This is a classical question when dealing with multi-objective approaches (even in fields other than DM - see for example [12,113,18] for the integration of User preferences in multi-objective optimization algorithms) and some works have been proposed to address it in the DM context, such as the proposition of dominance-based rough set approach, for example [107].

References

[1] R. Agrawal, T. Imielinski, A.N. Swami, Mining association rules between sets of items in large databases, in: Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data, ACM Press, Washington, DC, 1993, pp. 207–216.

[2] R. Agrawal, R. Srikant, Fast algorithms for mining association rules in large databases, in: VLDB '94: Proceedings of the 20th International Conference on Very Large Data Bases, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1994, pp. 487–499.

[3] S. Bai, X. Dai, An efficiency apriori algorithm: p_{matrix} algorithm, in: Proceedings of the First International Symposium on Data, Privacy, and E-Commerce, IEEE Computer Society, Washington, DC, USA, 2007, pp. 101–103.

[4] H. Bal, H.H. Orkcu, A new mathematical programming approach to multi-group classification problems, Computers and Operations Research 38 (1) (2011) 105–111.

[5] H. Barbalho, I. Rosseti, S. L. Martins, A. Plastino, A hybrid data mining grasp with path-relinking, Computers & Operations Research, ISSN 0305-0548, http://dx.doi.org/10.1016/j.cor.2012.02.022.

[6] S. Barreto, C. Ferreira, J. Paixo, B.S. Santos, Using clustering analysis in a capacitated location-routing problem, European Journal of Operational Research 179 (3) (2007) 968–977.

[7] M. Birattari, Tuning Metaheuristics: A Machine Learning Perspective, SCI, vol. 197, Springer, 2009.

[8] M. Birattari, Z. Yuan, P. Balaprakash, T. Stuetzle, F-race and iterated f-race: an overview, in: Experimental Methods for the Analysis of Optimization Algorithms, Springer, 2010, pp. 311–336.

[9] J. Blaszczynski, S. Greco, R. Slowinski, Multi-criteria classification – a new scheme for application of dominance-based decision rules, European Journal of Operational Research 181 (3) (2007) 1030–1044.

[10] C. Borgelt, Efficient implementations of apriori and eclat, in: Proc. 1st IEEE ICDM Workshop on Frequent Item Set Mining Implementations (FIMI 2003, Melbourne, FL), CEUR Workshop Proceedings 90, 2003, p. 90.

[11] R.I. Bot, N. Lorenz, Optimization problems in statistical learning: duality and optimality conditions, European Journal of Operational Research 213 (2) (2011) 395–404.

[12] J. Branke, K. Deb, Integrating user preferences into evolutionary multi-objective optimization, in: Y. Jin (Ed.), Knowledge Incorporation in Evolutionary Computation. Studies in Fuzziness and Soft Computing, vol. 167, Springer, Berlin/Heidelberg, 2005, pp. 461–477.

[13] L. Breiman, J.H. Friedman, R.A. Olshen, C.J. Stone, Classification and Regression Trees, Wadsworth International, Belmont, CA, 1984.

[14] I. Brzezinska, S. Greco, R. Slowinski, Mining pareto-optimal rules with respect to support and confirmation or support and anti-support, Engineering Applications of Artificial Intelligence 20 (5) (2007) 587–600.

[15] J.M. Cadenas, M.C. Garrido, E. Muñoz, Using machine learning in a cooperative hybrid parallel strategy of metaheuristics, Information Sciences 179 (19) (2009) 3255–3267.

[16] T. Carchrae, J.C. Beck, Applying machine learning to low-knowledge control of optimization algorithms, Computational Intelligence 21 (4) (2005) 372–387.

[17] E. Carrizosa, B. Martin-Barragan, Two-group classification via a biobjective margin maximization model, European Journal of Operational Research 173 (3) (2006) 746–761.

[18] Z.-H. Chen, Z.-Q. Zhuang, F.-H. Huang, J.-S. Lee, User-preference-oriented multi-objective optimization algorithm, in: 2010 International Computer Symposium (ICS), December 2010, pp. 1045–1049.

[19] C. Cortes, V. Vapnik, Support-vector networks, Machine Learning 20 (1995) 273–297.

[20] T. Cover, P. Hart, Nearest neighbor pattern classification, IEEE Transactions on Information Theory 13 (1) (1967) 21–27.

[21] M. Dash, H. Liu, Hybrid search of feature subsets, in: H.-Y. Lee, H. Motoda (Eds.), PRICAI'98: Topics in Artificial Intelligence, LNCS, vol. 1531, Springer, 1998, pp. 238–249.

[22] D.L. Davies, D.W. Bouldin, A cluster separation measure, IEEE Transactions on Pattern Analysis and Machine Intelligence PAMI-1 (1979) 224–227.

[23] B. de la Iglesia, G. Richards, M. Philpott, V. Rayward-Smith, The application and effectiveness of a multi-objective metaheuristic algorithm for partial classification, European Journal of Operational Research 169 (3) (2006) 898–917.

[24] F. de Lima, J. de Melo, A. Neto, Using q-learning algorithm for initialization of the grasp metaheuristic and genetic algorithm, in: International Joint Conference on Neural Networks, 2007 (IJCNN 2007), 2007, pp. 1243–1248.

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

[26] M. del Jess, J. Gmez, J. Puerta, Evolutionary and metaheuristics based data mining, Soft Computing – A Fusion of Foundations, Methodologies and Applications 13 (2009) 209–212.

[27] E.I. Ducheyne, B.D. Baets, R.R.D. Wulf, Fitness inheritance in multiple objective evolutionary algorithms: a test bench and real-world evaluation, Applied Soft Computing 8 (1) (2008) 337–349.

[28] J.C. Dunn, Well-separated clusters and optimal fuzzy partitions, Journal of Cybernetics 4 (1) (1974) 95–104.

[29] E. Falkenauer, A new representation and operators for genetic algorithms applied to grouping problems, Evolutionary Computation 2 (2) (1994) 123–144.

[30] E. Falkenauer, Genetic Algorithms and Grouping Problems, John Wiley & Sons, Inc., 1998.

[31] E. Fernandez, J. Navarro, S. Bernal, Handling multicriteria preferences in cluster analysis, European Journal of Operational Research 202 (3) (2010) 819–827.

[32] J. Fernandez Caballero, F. Martinez, C. Hervas, P. Gutierrez, Sensitivity versus accuracy in multiclass problems using memetic pareto evolutionary neural networks, IEEE Transactions on Neural Networks 21 (5) (2010) 750–770.

[33] J.E. Fieldsend, R.M. Everson, Multiobjective supervised learning, in: J. Knowles, D. Corne, K. Deb, D.R. Chair (Eds.), Multiobjective Problem Solving from Nature, Natural Computing Series, Springer, 2008, pp. 155–176.

[34] L. Fonseca, H. Barbosa, A. Lemonge, A similarity-based surrogate model for expensive evolutionary optimization with fixed budget of simulations, in: IEEE Congress on Evolutionary Computation, 2009 (CEC '09), May 2009, pp. 867–874.

[35] A. Freitas, Data Mining and Knowledge Discovery with Evolutionary Algorithms, Natural Computing Series, Springer, 2002.

[36] A.A. Freitas, On rule interestingness measures, Knowledge-Based Systems 12 (1999) 309–315.

[37] A.A. Freitas, A review of evolutionary algorithms for data mining, in: O. Maimon, L. Rokach (Eds.), Soft Computing for Knowledge Discovery and Data Mining, Springer, 2008, pp. 79–111.

[38] A.A. Freitas, A review of evolutionary algorithms for data mining, in: O. Maimon, L. Rokach (Eds.), Data Mining and Knowledge Discovery Handbook, Springer, 2010, pp. 371–400.

[39] J. García-Nieto, E. Alba, L. Jourdan, E. Talbi, Sensitivity and specificity based multiobjective approach for feature selection: Application to cancer diagnosis, Information Processing Letters 109 (2009) 887–896.

[40] A. Gaspar-Cunha, A. Vieira, A hybrid multi-objective evolutionary algorithm using an inverse neural network, in: LNCS (Ed.), Hybrid Metaheuristic, 2004, pp. 25–30.

[41] L. Geng, H.J. Hamilton, Interestingness measures for data mining: a survey, ACM Computing Surveys 38 (3) (2006).

[42] S. Greco, B. Matarazzo, R. Slowinski, Rough sets theory for multicriteria decision analysis, European Journal of Operational Research 129 (1) (2001) 1–47.

[43] S. Greco, Z. Pawlak, R. Slowinski, Can bayesian confirmation measures be useful for rough set decision rules?, Engineering Applications of Artificial Intelligence 17 (4) (2004) 345–361

[44] H. Guo, W.H. Hsu, A machine learning approach to algorithm selection for $p|np$ -hard optimization problems: a case study on the mpe problem, Annals OR 156 (1) (2007) 61–82.

[45] I. Guyon, A. Elisseeff, An introduction to variable and feature selection, Journal of Machine Learning Research 3 (2003) 1157–1182.

[46] J. Han, M. Kamber, J. Pei, Data Mining: Concepts and Techniques, The Morgan Kaufmann Series in Data Management Systems, Elsevier Science, 2011.

[47] H. Handa, T. Horiuchi, O. Katai, M. Baba, A novel hybrid framework of coevolutionary ga and machine learning, International Journal of Computational Intelligence and Applications. (2002).

[48] H. Handa, T. Horiuchi, O. Katai, T. Kaneko, T. Konishi, M. Baba, Fusion of coevolutionary GA and machine learning techniques through effective schema extraction, in: Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2001), Morgan Kaufmann, 2001. p. 764.

[49] J. Handl, J. Knowles, Evolutionary multiobjective clustering, in: Parallel Problem Solving from Nature – PPSN VIII, Lecture Notes in Computer Science, vol. 3242, Springer, 2004, pp. 1081–1091.

[50] J. Handl, J. Knowles, Exploiting the trade-off the benefits of multiple objectives in data clustering, in: Evolutionary Multi-Criterion Optimization, Lecture Notes in Computer Science, vol. 3410, Springer, 2005, pp. 547–560.

[51] J. Handl, J. Knowles, Improvements to the scalability of multiobjective clustering, in: IEEE Congress on Evolutionary Computation, 2005, pp. 2372–2379.

[52] J. Handl, J. Knowles, An evolutionary approach to multiobjective clustering, IEEE Transactions on Evolutionary Computation 11 (1) (2007) 56–76.
[53] R. Hilderman, H. Hamilton, Knowledge discovery and interestingness measures: a survey, Tech. rep., Department of Computer Science, University of Regina, 1999.
[54] E. Horvitz, Y. Ruan, C.P. Gomes, H.A. Kautz, B. Selman, D.M. Chickering, A bayesian approach to tackling hard computational problems, in: UAI '01: Proceedings of the 17th Conference in Uncertainty in Artificial Intelligence, Morgan Kaufmann, 2001, pp. 235–244.
[55] E.R. Hruschka, R.J.G.B. Campello, A.A. Freitas, A.C.P.L.F. de Carvalho, A survey of evolutionary algorithms for clustering, IEEE Transactions on Systems, Man, and Cybernetics, Part C 39 (2) (2009) 133–155.
[56] F.J. Iannarilli, P. Rubin, Feature selection for multiclass discrimination via mixed-integer linear programming, IEEE Transactions on Pattern Analysis and Machine Intelligence 25 (6) (2003) 779–783.
[57] C. Ishida, A. Pozo, E. Goldbarg, M. Goldbarg, Multiobjective optimization and rule learning: subselection algorithm or meta-heuristic algorithm?, in: Innovative Applications in Data Mining, Studies in Computational Intelligence, vol 169, Springer, 2009, pp. 47–70.
[58] Y. Jin, A comprehensive survey of fitness approximation in evolutionary computation, Soft Computing Journal 9 (1) (2005) 3–12.
[59] Y. Jin, Multi-Objective Machine Learning, Springer, 2006.
[60] Y. Jin, B. Sendhoff, Reducing fitness evaluations using clustering techniques and neural networks ensembles, in: Genetic and Evolutionary Computation Conference, LNCS, vol. 3102, Springer, 2004, pp. 688–699.
[61] Y. Jin, B. Sendhoff, Pareto-based multiobjective machine learning: An overview and case studies, Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on 38 (3) (2008) 397–415.
[62] D. Jones, A. Collins, C. Hand, A classification model based on goal programming with non-standard preference functions with application to the prediction of cinema-going behaviour, European Journal of Operational Research 177 (1) (2007) 515–524.
[63] L. Jourdan, D. Corne, D.A. Savic, G.A. Walters, Preliminary investigation of the learnable evolution model for faster/better multiobjective water systems design, in: EMO, 2005, pp. 841–855.
[64] L. Jourdan, C. Dhaenens, E.-G. Talbi, Using datamining techniques to help metaheuristics: a short survey, in: LNCS (Ed.), Hybrid Metaheuristics, vol. 4030, Springer, 2006, pp. 57–69.
[65] L. Jourdan, M. Khabzaoui, C. Dhaenens, E.-G. Talbi, Handbook of Bioinspired Algorithms and Applications, CRC Press, 2005. Chapter: A Hybrid Evolutionary Algorithm for Knowledge Discovery in Microarray Experiments, pp. 491–508.
[66] A. Kafafy, A. Bounekkar, S. Bonnevay, A hybrid evolutionary metaheuristics (hemh) applied on 0/1 multiobjective knapsack problems, in: GECCO, 2011, pp. 497–504.
[67] M. Kantardzic, Data Mining: Concepts, Models, Methods, and Algorithms, John Wiley & Sons, 2011.
[68] B. Karasozen, A. Rubinov, G.-W. Weber, Optimization in data mining, European Journal of Operational Research 173 (3) (2006) 701–704.
[69] M. Khabzaoui, C. Dhaenens, E.-G. Talbi, A multicriteria genetic algorithm to analyze microarray data, in: Congress on Evolutionary Computation, 2004 (CEC2004), vol. 2, 2004, pp. 1874–1881.
[70] M. Khabzaoui, C. Dhaenens, E.-G. Talbi, Combining evolutionary algorithms and exact approaches for multi-objective knowledge discovery, RAIRO Operations Research 42 (2008) 69–83.
[71] H.-S. Kim, S.-B. Cho, An efficient genetic algorithms with less fitness evaluation by clustering, in: Proceedings of IEEE Congress on Evolutionary Computation, IEEE, 2001, pp. 887–894.
[72] E. Korkmaz, Multi-objective genetic algorithms for grouping problems, Applied Intelligence 33 (2010) 179–192.
[73] S.B. Kotsiantisr, Supervised machine learning: a review of classification techniques, Informatica 31 (2007) 249–268.
[74] H.-P. Kriegel, P. Kröger, A. Zimek, Clustering high-dimensional data: a survey on subspace clustering, pattern-based clustering, and correlation clustering, ACM Transactions on Knowledge Discovery from Data 3 (2009) 1:1–1:58.
[75] K.N. Kunene, H.R. Weistroffer, An approach for predicting and describing patient outcome using multicriteria decision analysis and decision rules, European Journal of Operational Research 185 (3) (2008) 984–997.
[76] D. Lim, Y. Jin, Y.-S. Ong, B. Sendhoff, Generalizing surrogate-assisted evolutionary computation, IEEE Transactions on Evolutionary Computation 14 (3) (2010) 329–355.
[77] J.J. Liou, L. Yen, G.-H. Tzeng, Using decision rules to achieve mass customization of airline services, European Journal of Operational Research 205 (3) (2010) 680–686.
[78] H. Liu, H. Motoda, M. Dash, A monotonic measure for optimal feature selection, in: Proceedings of European Conference on Machine Learning, 1998, pp. 101–106.
[79] J.B. MacQueen, Some methods for classification and analysis of multivariate observations, in: Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability, 1967, pp. 281–297.
[80] O. Maimon, L. Rokach, Data Mining and Knowledge Discovery Handbook, The Kluwer International Series in Engineering and Computer Science, Springer, 2005.
[81] U. Maulik, S. Bandyopadhyay, A. Mukhopadhyay, Multiobjective Genetic Algorithms for Clustering: Applications in Data Mining and Bioinformatics, Springer, 2011.
[82] S. Meisel, D. Mattfeld, Synergies of operations research and data mining, European Journal of Operational Research 206 (1) (2010) 1–10.
[83] R. Michalski, Learnable evolution model: evolutionary processes guided by machine learning, Machine Learning 38 (1–2) (2000) 9–40.
[84] R. Michalski, G. Cervon, K. Kaufman, Speeding up evolution through learning: LEM, in: Intelligent Information Systems 2000, 2000, pp. 243–256.
[85] B. Mirkin, Mathematical Classification and Clustering, Kluwer Academic Publishers, 1996.
[86] S. Mitra, H. Banka, Multi-objective evolutionary biclustering of gene expression data, Pattern Recognition (2006) 2464–2477.
[87] L.C. Molina, L. Belanche, A. Nebot, Feature selection algorithms: a survey and experimental evaluation, in: Proceedings of the 2002 IEEE International Conference on Data Mining (ICDM '02), IEEE Computer Society, Washington, DC, USA, 2002, p. 306.
[88] P. Narendra, K. Fukunaga, A branch and bound algorithm for feature subset selection, IEEE Transactions on Computers C-26 (9) (1977) 917–922.
[89] M. Nikoli, F. Mari, P. Janii, Simple algorithm portfolio for sat, Artificial Intelligence Review (2011), accepted for publication.
[90] E. Nudelman, K. Leyton-Brown, H. Hoos, A. Devkar, Y. Shoham, Understanding random sat: beyond the clauses-to-variables ratio, in: Principles and Practice of Constraint Programming CP 2004, in: M. Wallace (Ed.), Lecture Notes in Computer Science, vol. 3258, Springer, Berlin/Heidelberg, 2004, pp. 438–452.
[91] S. Olafsson, Introduction to operations research and data mining, Computers and Operations Research 33 (11) (2006) 3067–3069.
[92] S. Olafsson, X. Li, S. Wu, Operations research and data mining, European Journal of Operational Research 187 (3) (2008) 1429–1448.
[93] Y.S. Ong, A. Keane, P. Nair, Surrogate-assisted coevolutionary search, in: 9th International Conference on Neural Information Processing, Special Session on Trends in Global Optimization, Singapore, 2002, pp. 2195–2199.
[94] M. Ostaszewski, P. Bouvry, F. Seredynski, Multiobjective classification with mogep: an application in the network traffic domain, in: Proceedings of the 11th Annual conference on Genetic and evolutionary computation (GECCO '09), ACM, 2009, pp. 635–642.
[95] A. Ozturk, S. Kayaligil, N.E. Ozdemirel, Manufacturing lead time estimation using data mining, European Journal of Operational Research 173 (2) (2006) 683–700.
[96] D.J. Patterson, H.A. Kautz, Auto-walksat: a self-tuning implementation of walksat, Electronic Notes in Discrete Mathematics 9 (2001) 360–368.
[97] M. Pelikan, D.E. Goldberg, E. Cantú-Paz, BOA: the Bayesian optimization algorithm, Proceedings of the Genetic and Evolutionary Computation Conference GECCO-99, vol. 1, Morgan Kaufmann Publishers, 1999, pp. 525–532.
[98] M. Pelikan, K. Sastry, Fitness inheritance in the bayesian optimization algorithm, in: GECCO-04, Lecture Notes in Computer Science, vol. 3103, 2004, pp. 48–59.
[99] A. Plastino, R. Fuchshuber, S.d.L. Martins, A.A. Freitas, S. Salhi, A hybrid data mining metaheuristic for the p-median problem, Statistical Analysis and Data Mining 4 (3) (2011) 313–335.
[100] M. Preuss, Adaptability of algorithms for real-valued optimization, in: EvoWorkshops, Lecture Notes in Computer Science, vol. 5484, Springer, 2009, pp. 665–674.
[101] J.R. Quinlan, C4.5: Programs for Machine Learning, Morgan Kaufmann Publishers Inc., 1993.
[102] M. Ribeiro, A. Plastino, S. Martins, Hybridization of grasp metaheuristic with data mining techniques, Journal of Mathematical Modelling and Algorithms 5 (2006) 23–41, http://dx.doi.org/10.1007/s10852-005-9030-1.
[103] B. Saglam, F.S. Salman, S. Sayln, M. Turkay, A mixed-integer programming approach to the clustering problem with an application in customer segmentation, European Journal of Operational Research 173 (3) (2006) 866–879.
[104] L.F. Santos, S.L. Martins, A. Plastino, Applications of the dm-grasp heuristic: a survey, International Transactions in Operational Research 15 (4) (2008) 387–416.
[105] K. Seridi, L. Jourdan, E.-G. Talbi, Multi-objective evolutionary algorithm for bi-clustering in microarrays data, in: CEC 2011, IEEE, 2011, pp. 2593–2599.
[106] G. Sheri, D. Corne, Learning-assisted evolutionary search for scalable function optimization: LEM(ID3), in: IEEE Congress on Evolutionary Computation, IEEE, 2010, pp. 1–8.
[107] R. Slowinski, S. Greco, B. Matarazzo, Rough sets in decision making, in: R.A. Meyers (Ed.), Encyclopedia of Complexity and Systems Science, Springer, New York, 2009, pp. 7753–7786.
[108] K. Smith-Miles, R.J.W. James, J.W. Giffin, Y. Tu, A knowledge discovery approach to understanding relationships between scheduling problem structure and heuristic performance, in: LION 3, LNCS, vol. 5851, Springer, 2009, pp. 89–103.
[109] K. Sorensen, G.K. Janssens, Data mining with genetic algorithms on binary trees, European Journal of Operational Research 151 (2) (2003) 253–264.
[110] I. Szczech, Transactions on Rough Sets X, Springer-Verlag, Berlin, Heidelberg, 2009. Chapter: Multicriteria Attractiveness Evaluation of Decision and Association Rules, pp. 197–274.
[111] E.-G. Talbi, L. Jourdan, J. Garcia-Nieto, E. Alba, Comparison of population based metaheuristics for feature selection: application to microarray data classification, in: AICCSA'08, 2008, pp. 45–52.
[112] P.N. Tan, V. Kumar, Selecting the right interestingness measure for association patterns, in: Proceedings of the Eight ACM SIGKDD Conference, 2002, pp. 32–41.

[113] L. Thiele, K. Miettinen, P.J. Korhonen, J. Molina, A preference-based evolutionary algorithm for multi-objective optimization, Evolutionary Computation 17 (3) (2009) 411–436.
[114] F. Uney, M. Turkay, A mixed-integer programming approach to multi-class data classification problem, European Journal of Operational Research 173 (3) (2006) 910–920.
[115] L. Vermeulen-Jourdan, D. Corne, D.A. Savic, G.A. Walters, Hybridising rule induction and multi-objective evolutionary search for optimising water distribution systems, in: HIS, 2004, pp. 434–439.
[116] L. Vermeulen-Jourdan, C. Dhaenens, E.-G. Talbi, Clustering nominal and numerical data: a new distance concept for a hybrid genetic algorithm, in: EvoCOP 2004, LNCS, vol. 3004, Springer Verlag, 2004, pp. 220–229.
[117] G. Xu, L.G. Papageorgiou, A mixed integer optimisation model for data classification, Computers and Industrial Engineering 56 (4) (2009) 1205–1215.
[118] L. Xu, F. Hutter, H.H. Hoos, K. Leyton-Brown, Satzilla: portfolio-based algorithm selection for sat, Journal of Artificial Intelligence Research (JAIR) 32 (2008) 565–606.
[119] R. Xu, D.I. Wunsch, Survey of clustering algorithms, IEEE Transactions on Neural Networks 16 (3) (2005) 645–678.
[120] Y. Ye, C.-C. Chiang, A parallel apriori algorithm for frequent itemsets mining, in: Proceedings of the Fourth International Conference on Software Engineering Research, Management and Applications, IEEE Computer Society, Washington, DC, USA, 2006, pp. 87–94.
[121] S.-H. Yoo, S.-B. Cho, Partially evaluated genetic algorithm based on fuzzy c-means algorithm, in: LNCS (Ed.), PPSN, 2004, pp. 440–449.
[122] S.C. Yusta, Different metaheuristic strategies to solve the feature selection problem, Pattern Recognition Letters 30 (5) (2009) 525–534.
[123] M.J. Zaki, Parallel sequence mining on shared-memory machines, Journal of Parallel and Distributed Computing (2001) 401–426.
[124] G.P. Zhang, Neural networks for classification: a survey, IEEE Transactions on Systems, Man and Cybernetics, Part C (Applications and Reviews) 30 (4) (2000) 451–462.