# Scalable preference disaggregation: A multiple criteria sorting approach based on the MapReduce framework

**Xiaoxin Mao[1]Jiapeng Liu[2]Xiuwu Liao[3]**

**Abstract.** In the era of big data, analyzing and extracting knowledge from large-scale data sets to support better decision making constitutes an interesting but challenging task. The sizes of such data sets exceed the processing capabilities of existing multiple criteria sorting (MCS) methods in terms of the required time and memory consumption. In this paper, we propose a scalable decision-making approach based on the MapReduce framework, in order to address the MCS problem with large sets of alternatives and massive preference information in a parallel manner. This approach employs the additive piecewise-linear value function as the preference model and adopts the disaggregation-aggregation paradigm to infer a preference model from assignment examples on reference alternatives. To facilitate the parallel implementation of the sorting procedure, a new convex optimization model is proposed to estimate the parameters of the preference model. Then, a new parallel algorithm is developed to solve this optimization model and the MapReduce framework is utilized to deal with the set of reference alternatives and associated preference information in a parallel manner in order to accelerate the computation. Furthermore, the performance of the proposed approach is investigated on a real-word data set and a series of generated data sets consisting of up to 400,000 alternatives. The results show that our approach provides an efficient tool for addressing the MCS problem with a large set of alternatives and massive preference information.

## 1 Introduction

The purpose of multiple criteria decision analysis (MCDA) is to assist a decision maker (DM) in choosing, ranking, or sorting a finite set of decision alternatives according to multiple evaluation criteria [54]. The choice problem consists in selecting a subset of alternatives to be judged as the most satisfactory of the whole set. The ranking problem consists in establishing a preference pre-order (either partial or complete) on the whole set. Finally, the sorting problem concerns the ordinal classification of alternatives, and consists in assigning all alternatives to predefined and preferentially ordered categories [28]. Greco et al. [22, 23] provided a comprehensive review of various MCDA approaches for handling different decision scenarios, and reported new trends and recent developments from the last several years.

In this paper, we focus on the MCS problem. The MCS problem is of major practical interest in many domains, such as global investing risk assessment [26], MBA program ranking [6], country risk evaluation [55], regional competitiveness analysis [21], recommender systems [46], project management [14], water contamination risk assessment [43], climate classification [44], environmental impact assessment [1], industrial innovation [52], customer satisfaction analysis [41], credit rating [18, 4], ABC inventory classification [42], and research unit evaluation [31]. The MCS problem has attracted significant interest over the past decade, and various MCS approaches have been proposed in the literature. According to the employed preference model, which reflects the value system of the DM, MCS approaches in the literature can be categorized into four types: (1) methods inspired by outranking relations, such as the ELECTRI Tri-B methods [7, 56, 13, 20], ELECTRI Tri-C methods [2, 3, 33], ELECTRI-based methods [34, 53, 29] and PROMETHEE-based methods [49, 8, 29]; (2) methods motivated by value functions, such as the UTADIS method and its variants [16, 35, 28, 11, 40] and the MHDIS method [16, 57]; (3) methods based on the weighted Euclidean distance, such as [9, 10]; and (4) rule induction-oriented methods, such as the DRSA method and its extensions [24, 25, 30, 32].

Like the choice and ranking problems, the MCS problem requires the DM to provide preference information in either a direct or indirect manner. In the direct case, the DM is guided by an analyst to specify precise values for or constraints on the parameters of the preference model, such as weights, trade-offs, and category profiles. Such an elicitation process is usually organized through interactive structured communication sessions between the DM and the analyst, and thus the success of this approach strongly depends on the willingness of the DM to actively participate in this process, as well as the ability of the analyst to guide the interactive process in order to address the DM's cognitive limitations [17]. On the other hand, in the indirect case the DM provides holistic judgments in the form of assignment examples, from which compatible values for the parameters of the preference model are induced. Such assignment examples may either be provided by the DM on a set of real or hypothetical alternatives, or may come from observations of the DM's past decisions, both of which are referred to as reference alternatives [12]. MCS approaches based on indirect preference information are considered to be more user-friendly than approaches based direct preference information, because the former requires a simple cognitive interpretation for the DM in order to express preference information.

Indirect preference information is used in the disaggregation-aggregation paradigm, which aims to construct a preference model that is as consistent as possible with the assignment examples provided by the DM [12]. The inferred preference model is applied to new decision instances to arrive at an assignment recommendation. The best-known method in this paradigm is the UTADIS method

[1] School of Management, Xi'an Jiaotong University, Xi'an, 710049, Shaanxi, PR China, email: maoxiaoxin29@stu.xjtu.edu.cn
[2] School of Management, Xi'an Jiaotong University, Xi'an, 710049, Shaanxi, PR China, email: jiapengliu@mail.xjtu.edu.cn
[3] School of Management, Xi'an Jiaotong University, Xi'an, 710049, Shaanxi, PR China, email: liaoxiuwu@mail.xjtu.edu.cn

[16]. Based on a set of assignment examples, UTADIS aims to estimate an additive value function, as well as global value thresholds delimiting consecutive categories, with the minimum misclassification error. The marginal value function on each criterion is assumed to be piecewise linear, and each criterion range is divided into several subintervals. The marginal value of each cut-point is estimated using the linear programming technique, and the marginal value of each criterion performance can then be calculated using a simple linear interpolation. The inferred parameters that minimize the sum of the misclassification errors can be applied to assign a non-reference alternative by comparing its global value with the estimated value thresholds. The disaggregation-aggregation paradigm has also been implemented in outranking-based sorting methods [7, 31, 29], distance-based sorting methods [9, 10], and rule induction-oriented sorting methods [24, 30]. Indeed, the philosophy underlying the disaggregationaggregation paradigm is consistent with the posterior rationality postulated by March et al. [45], as well as the induction principle used in artificial intelligence and data mining.

Recent developments in information technology have resulted in an explosive growth in data gathered from various fields. Ninety percent of the data in the world has been created in the last two years, at a rate of 2.5 quintillion bytes on a daily basis. This is commonly referred to as big data [4]. Retrieving useful information and knowledge from big data can enable organizations to reach better decisions, including deepening customer engagement, optimizing operations, preventing threats and fraud, and capitalizing on new sources of revenue. An estimated 89 percent of enterprises believe that those who do not take advantage of big data analytics run the risk of losing a competitive edge in the market [5], and thus they are gearing up to leverage their information assets to gain a competitive advantage. MCS is also gaining importance in the era of big data, because it can help firms, organizations, and governments to reach better data-driven decisions. For example, in the field of finance a rating agency can adopt MCS approaches to evaluate the credit risks of millions of firms, and assign a credit rating to each firm. As another example, in the field of e-business firms can utilize big data analytics to analyze the preferences of a large population of consumers, and then apply MCS approaches to divide a large market into segments in order to tailor different marketing policies for targeted segments.

However, it is challenging for existing MCS methods based on indirect preference information and the disaggregation-aggregation paradigm to deal with decision-making problems that contain large sets of alternatives and massive preference information. In fact, most MCS methods were originally designed to deal with relatively small data sets, i.e., for problems with several dozen alternatives, and the development of computationally efficient algorithms that scale up efficiently with the number of alternatives is not at the core of such approaches [12]. More specifically, for MCS methods based on indirect preference information and the disaggregationaggregation paradigm, the construction of a preference model from a set of assignment examples usually involves linear programming (LP) and integer programming (IP) formulations. LP models are usually employed to minimize some predefined norms ($L_1$ or $L_\infty$) of real-valued error variables representing the violations of preference relations. IP formulations are used to minimize the number of disagreements between the recommendations of the estimated decision model and the actual evaluation of the reference alternatives by the DM [17, 19]. To the best of our knowledge, all LP and IP formulations in existing MCS methods require the data to fit into the main memory. This requirement stems from the fact that the constraint matrix must be loaded into the main memory of a computer in order to determine the optimal solution. Thus, this exceeds the processing capabilities of existing MCS methods in terms of the memory consumption and/or computational time when dealing with huge amounts of data. MCS methods that assume that data can fit into the main memory must to be revised or redesigned to adapt them to the new storage and time requirements.

In this paper, we propose a novel approach based on the MapReduce framework [47], in order to address the MCS problem with a large set of alternatives and massive preference information. MapReduce is a popular parallel computing paradigm, which is designed to process large-scale data sets. It divides the original set into subsets, which can more easily be addressed, and then combines intermediate solutions to obtain the final outcomes. Such a computing paradigm allows us to parallelize applications on a computer cluster in a highly scalable manner, and has been widely employed for social network analysis [38, 50], e-commerce [36, 37], industrial engineering [39], and so on.

In the proposed MapReduce-based MCS approach, the preference information refers to assignment examples on reference alternatives, which may come from past decision examples, such as historical credit ratings of firms during a certain period or past customer segmentation. We employ the additive piecewise-linear value function as the preference model and adopts the disaggregation-aggregation paradigm to infer a preference model from a large set of assignment examples. According to the principle of assignment consistency, we propose a convex optimization model to derive a preference model that can restore the preference information as consistently as possible. This optimization model uses the Sigmoid function to measure the degree of inconsistency when the assignment of any pair of reference alternatives $a^*, b^*$ that violates the consistency principle and the margin to the decision boundary when the assignment of $a^*, b^*$ accords the consistency principle, thus deriving a preference model that is both consistent and robust. In contrast to traditional MCS methods, this optimization model avoids using slack variables to specify the degree of inconsistency for any assignment example, which is especially suitable for addressing large-scale MCS problems. In order to solve the optimization model efficiently, we propose a new parallel algorithm based on the Zoutendijk's feasible direction method [51] and the MapReduce framework. The algorithm iteratively searches for the optimal solution by checking a direction that is both feasible and descent. During the iterative process, the MapReduce framework is used to accelerate the calculation of the value and the gradient of current solution in a parallel manner.

The contributions of the proposed approach are threefold, and described as follows. First, our work represents the first MCS approach implemented with the MapReduce framework to address large sets of alternatives and massive preference information. Although data-driven decision making has become a popular contemporary subject, no previous methods can deal efficiently with such an MCS problem. Our approach displays a high scalability and an ability to tackle the MCS problem with a large set of alternatives and massive preference information. Second, we propose a convex optimization model to derive a preference model, which answers to consistency and robustness concerns simultaneously. Differently from traditional techniques, this optimization model avoids using slack variables to specify the degree of inconsistency for any assignment example and thus contains a relatively small set of variables and linear constraints, which is especially suitable for addressing large-scale MCS prob-

[4] http://www-01.ibm.com/software/data/bigdata/what-is-big-data.html

[5] http://www.investopedia.com/articles/active-trading/040915/how-big-data-has-changed-finance.asp

lems. Finally, we propose a parallel algorithm to solve the developed optimization model efficiently. This algorithm does not need to load the whole data set into the main memory and thus has no specific requirements regarding the processing capabilities of working nodes. It proceeds by scanning the preference information sequentially, which is beneficial for the parallel implementation with the MapReduce framework to accelerate the computation. Moreover, it is robust to the splitting and combining operations in the MapReduce framework and the final results are irrelevant for the parallel implementation.

## 2 Brief introduction to MapReduce

The MapReduce framework was proposed by Google Inc. in 2004. It is a programming paradigm for processing massive amounts of data in a parallel manner, and allows large data sets to be tackled over a cluster of computers regardless of the underlying hardware. It is based on the philosophy that most computing tasks share a similar structure, i.e., the same computation is performed over a large number of instances and then intermediate results are aggregated in some way. As the name suggests, this programming paradigm is built upon the two functions Map and Reduce, which are inherited from the classical functional programming paradigm.

The cluster of computers is composed of a master working node and several slave working nodes. In the Map phase, the master node divides the input data set into independent sections and distributes these to slave nodes. Then, the slave nodes process smaller problems in a parallel manner and pass answers back to the master node. Finally, in the Reduce phase the master node receives the answers for all sub-problems and combines them in such a way that the output is formed. The users in this paradigm are only required to define what should be computed in the Map and Reduce functions, while the system automatically distributes the data processing over a highly distributed cluster of machines [15].

Both the Map and Reduce functions employ $< key, value >$ pairs as input and output. The Map phase takes each $< key, value >$ pair as input and generates a set of intermediate $< key, value >$ pairs as output. This could be presented as follows:

$$map(key1,\ value1) \rightarrow list(key2,\ value2).$$

Then, the master node merges and groups all the values associated with the same intermediate key as a list (known as the shuffle phase). The Reduce phase takes a key and its associated value list as input, and produces the final value. This could be presented as follows:

$$reduce(key2,\ list(value2)) \rightarrow list(key2,\ value3).$$

Figure 1 presents the flowchart of the MapReduce framework.

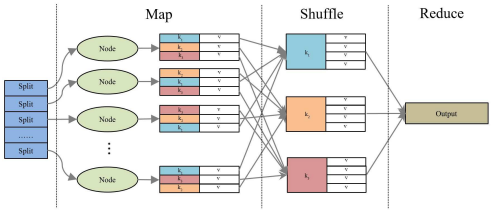


**Figure 1.** Flowchart of the MapReduce framework.

## 3 Proposed approach

### 3.1 Problem description

The aim of this study is to assign a finite set of $m$ alternatives $A = \{a_1, a_2, ..., a_m\}$ to $p$ predefined and preferentially ordered categories $\overline{C} = \{C_1, C_2, ..., C_p\}$, such that $C_{h+1}$ is preferred to $C_h$ (denoted by $C_{h+1} \succ C_h$), $h = 1, ..., p-1$. Such a classification decision is made based on historical preference information that is provided a priori, which consists of a set of assignment examples on a finite set of reference alternatives $A^R = \{a^*, b^*, ...\}$. An assignment example specifies a reference alternative $a^* \in A^R$ and its assignment $C(a^*) \in \overline{C}$. All the alternatives $a \in A \cup A^R$ are evaluated in terms of $n$ criteria $g_1, g_2, ..., g_n$. The performance of alternative $a \in A \cup A^R$ on criterion $g_j$, $j \in G = \{1, ..., n\}$, is denoted by $g_j(a)$. Without loss of generality, we assume that all criteria have an increasing direction of preference, i.e., the greater the value of $g_j(a)$, the stronger the preference is for the alternative $a$ on criterion $g_j$, for all $a \in A \cup A^R$ and $j \in G$.

We shall use an additive value function as the preference model:

$$U(a) = \sum_{j=1}^{n} u_j(g_j(a)), \quad a \in A \cup A^R,$$

where $U(a)$ is the value of alternative $a$, and $u_j(g_j(a))$, $j = 1, ..., n$, are marginal value functions for each criterion. Additive value functions are the most common preference model in MCDA due to their intuitive interpretation and relatively easy computation, despite the underlying assumption on the preferential independence of criteria and the compensation between criteria [27].

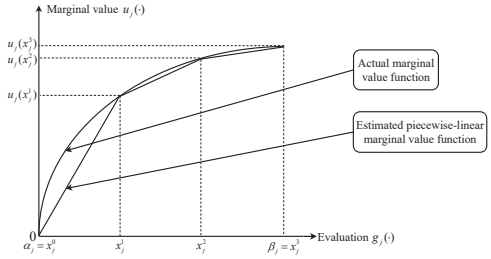


**Figure 2.** Piecewise-linear form of marginal value function.

In this paper, the marginal value function $u_j(g_j(a))$, $j = 1, ..., n$, is estimated by a piecewise-linear function (see Figure 2). Let $X_j = [\alpha_j, \beta_j]$ denote the evaluation scale of criterion $g_j$, such that $\alpha_j$ and $\beta_j$ are the worst and best evaluation, respectively. In defining the piecewise-linear form of the marginal value function $u_j(g_j(a))$, $X_j = [\alpha_j, \beta_j]$ is divided into $\gamma_j \geqslant 1$ equal sub-intervals $[x_j^0, x_j^1], [x_j^1, x_j^2], ..., [x_j^{\gamma_j - 1}, x_j^{\gamma_j}]$, where $x_j^k = \alpha_j + \frac{k}{\gamma_j}(\beta_j - \alpha_j)$, $k = 0, 1, ..., \gamma_j$. The marginal value of alternative $a$ on criterion $g_j$ is estimated by linear interpolation:

$$u_j(g_j(a)) = u_j(x_j^{k_j}) + \frac{g_j(a) - x_j^{k_j}}{x_j^{k_j+1} - x_j^{k_j}} \left( u_j(x_j^{k_j+1}) - u_j(x_j^{k_j}) \right), \quad \text{for } g_j(a) \in \left[ x_j^{k_j}, x_j^{k_j+1} \right]. \tag{1}$$

According to (1), the piecewise-linear value function $u_j(g_j(a))$ is completely defined by the marginal values at characteristic points, i.e., $u_j(x_j^0) = u_j(\alpha_j),\ u_j(x_j^1),\ ...,\ u_j(x_j^{\gamma_j}) = u_j(\beta_j)$. The use of piecewise-linear value functions is appropriate for wide application because any non-linear value function can be approximated given a sufficient number of characteristic points.

In particular, a new variable $\Delta u_j^t$ is introduced to represent the difference between marginal values at two consecutive characteristic points $x_j^t$ and $x_j^{t-1}$:

$$\Delta u_j^t = u_j(x_j^t) - u_j(x_j^{t-1}),\ \ t = 1, ..., \gamma_j.$$

Therefore, the marginal value $u_j(g_j(a))$ can be expressed as

$$u_j(g_j(a)) = \sum\nolimits_{t=1}^{k_j} \Delta u_j^t + \frac{g_j(a) - x_j^{k_j}}{x_j^{k_j+1} - x_j^{k_j}} \Delta u_j^{k_j+1},$$
$$\text{for } g_j(a) \in \left[x_j^{k_j}, x_j^{k_j+1}\right].$$

**Definition 1.** ([40]) The characteristic vector $\mathbf{V}(a) \in \mathbb{R}^{\gamma}$ of alternative $a$ is defined as a column vector whose entries are the coefficients of variables $\Delta u_j^t$ in marginal values $u_j(g_j(a))$ on each criterion $g_j$, $j = 1, ..., n$, where $\gamma = \sum\limits_{j=1}^{n} \gamma_j$:

$$\mathbf{V}(a) = \left( \underbrace{v_1^1, ..., v_1^{\gamma_1}}_{\text{criterion } g_1}, ..., \underbrace{v_j^1, ..., v_j^{\gamma_j}}_{\text{criterion } g_j}, ..., \underbrace{v_n^1, ..., v_n^{\gamma_n}}_{\text{criterion } g_n} \right)^{\mathrm{T}}$$
$$= \left( \underbrace{\underbrace{1, ..., 1}_{k_1}, \frac{g_1(a) - x_1^{k_1}}{x_1^{k_1+1} - x_1^{k_1}}, \underbrace{0, ..., 0}_{\gamma_1 - k_1 - 1}}_{\text{criterion } g_1}, ..., \right.$$
$$\underbrace{\underbrace{1, ..., 1}_{k_j}, \frac{g_j(a) - x_j^{k_j}}{x_j^{k_j+1} - x_j^{k_j}}, \underbrace{0, ..., 0}_{\gamma_j - k_j - 1}}_{\text{criterion } g_j}, ...,$$
$$\left. \underbrace{\underbrace{1, ..., 1}_{k_n}, \frac{g_n(a) - x_n^{k_n}}{x_n^{k_n+1} - x_n^{k_n}}, \underbrace{0, ..., 0}_{\gamma_n - k_n - 1}}_{\text{criterion } g_n} \right)^{\mathrm{T}}$$

With the definition of characteristic vector $\mathbf{V}(a)$, the value of alternative $a$ can be formulated as

$$U(a) = \mathbf{u}^{\mathrm{T}} \mathbf{V}(a),$$

where $\mathbf{u} = \left( \underbrace{\Delta u_1^1, ..., \Delta u_1^{\gamma_1}}_{\text{criterion } g_1}, ..., \underbrace{\Delta u_j^1, ..., \Delta u_j^{\gamma_j}}_{\text{criterion } g_j}, ..., \underbrace{\Delta u_n^1, ..., \Delta u_n^{\gamma_n}}_{\text{criterion } g_n} \right)^{\mathrm{T}}$.

To normalize the value of alternative $a$ such that $U(a) \in [0, 1]$, we can set

$$\begin{cases} \mathbf{u}^{\mathrm{T}} \mathbf{e} = 1, \\ \mathbf{u} \geqslant \mathbf{0}, \end{cases}$$

where $\mathbf{e}$ is a column vector whose entries are all equal to 1. In this case, the trade-off weights for each criterion $g_j$ can be retrieved as $w_j = u_j(\beta_j)$, $j = 1, ..., n$.

## 3.2 Model for estimating value function

The value of any alternative serves as an index used to decide its assignment. In this paper, let us consider the following consistency principle for assignment.

**Definition 2.** For any pair of alternatives $a$ and $b$, a given value function $U(\cdot)$ is said to be consistent with the assignment of $a$ and $b$ (denoted by $C(a)$ and $C(b)$, respectively, and $C(a), C(b) \in \overline{C}$), if and only if

$$U(a) \geqslant U(b) \Rightarrow C(a) \succsim C(b), \tag{2}$$
$$U(a) \leqslant U(b) \Rightarrow C(a) \precsim C(b), \tag{3}$$

where $\succsim$ and $\precsim$ means "as least as good as" and "as most as good as", respectively. Observe that (2) and (3) are equivalent to

$$C(a) \prec C(b) \Rightarrow \mathbf{u}^{\mathrm{T}} (\mathbf{V}(a) - \mathbf{V}(b)) < 0, \tag{4}$$
$$C(a) \succ C(b) \Rightarrow \mathbf{u}^{\mathrm{T}} (\mathbf{V}(a) - \mathbf{V}(b)) > 0, \tag{5}$$

since $U(a) = \mathbf{u}^{\mathrm{T}}\mathbf{V}(a)$ and $U(b) = \mathbf{u}^{\mathrm{T}}\mathbf{V}(b)$.

For many MCS problems, we cannot find a consistent value function given a set of reference alternatives because of some inconsistent assignment examples. In this case, traditional methods based on the disaggregation-aggregation paradigm introduce a set of slack variables, which would specify how much inconsistency there is for any pair of reference alternatives that violate the above consistency principle, and then minimize the sum of all the slack variables by solving a LP model to derive a value function that is as consistent with the preference information as possible. However, considering that a large-scale problem contains a huge number of reference alternatives, the number of linear constraints and that of terms in the objective of the LP model will be quite large, which exceeds the processing capabilities of most LP solvers. In this paper, differently from traditional methods, we propose a new model to estimate an additive value function as follows.

First, let us introduce a set of variables $t(a^*, b^*)$ for any pair of reference alternatives $a^*, b^* \in A^R$ such that $C(a^*) \neq C(b^*)$, which are defined as

$$t(a^*, b^*) = \begin{cases} 1, & \text{if } C(a^*) \succ C(b^*), \\ 0, & \text{if } C(a^*) \prec C(b^*). \end{cases}$$

According to the above consistency principle for assignment, we aim to find a vector $\mathbf{u}$ such that, for any pair of reference alternatives $a^*, b^* \in A^R$ with $C(a^*) \neq C(b^*)$, we have

$$\mathbf{u}^{\mathrm{T}} (\mathbf{V}(a^*) - \mathbf{V}(b^*)) \begin{cases} > 0, & \text{if } t\left(a^*, b^*\right) = 1, \\ < 0, & \text{if } t\left(a^*, b^*\right) = 0. \end{cases} \tag{6}$$

Then, instead of using slack variables, we can use the following Sigmoid function $y(\cdot)$ to measure how much inconsistency there is for any pair of reference alternatives $a^*, b^* \in A^R$ that violate the consistency principle

$$y\left(a^*, b^*\right) = \frac{1}{1 + e^{-\mathbf{u}^{\mathrm{T}}(\mathbf{V}(a^*) - \mathbf{V}(b^*))}}. \tag{7}$$

Note that the Sigmoid function $y(\cdot)$ is monotone, increasing, and bounded within the interval $(0, 1)$ and we have

- $0.5 < y\left(a^*, b^*\right) < 1$ when $\mathbf{u}^{\mathrm{T}} (\mathbf{V}(a^*) - \mathbf{V}(b^*)) > 0$,

- $y\left(a^*, b^*\right) = 0.5$ when $\mathbf{u}^{\mathrm{T}}\left(\mathbf{V}\left(a^*\right) - \mathbf{V}\left(b^*\right)\right) = 0$,
- $0 < y\left(a^*, b^*\right) < 0.5$ when $\mathbf{u}^{\mathrm{T}}\left(\mathbf{V}\left(a^*\right) - \mathbf{V}\left(b^*\right)\right) < 0$.

We hope to minimize the difference between $y\left(a^*, b^*\right)$ and $t\left(a^*, b^*\right)$ in order to derive $\mathbf{u}$ that is as consistent with the assignment of $a^*, b^*$ as possible. Specifically, for the case that $t\left(a^*, b^*\right) = 1$, if $\mathbf{u}^{\mathrm{T}}\left(\mathbf{V}\left(a^*\right) - \mathbf{V}\left(b^*\right)\right) < 0$, then the inconsistency occurs and the difference between $y\left(a^*, b^*\right)$ and $t\left(a^*, b^*\right)$ is greater than 0.5 because $0 < y\left(a^*, b^*\right) < 0.5$; if $\mathbf{u}^{\mathrm{T}}\left(\mathbf{V}\left(a^*\right) - \mathbf{V}\left(b^*\right)\right) > 0$, then there is no inconsistency and the difference between $y\left(a^*, b^*\right)$ and $t\left(a^*, b^*\right)$ is smaller than 0.5 because $0.5 < y\left(a^*, b^*\right) < 1$. The case that $t\left(a^*, b^*\right) = 1$ can be analyzed analogously. Minimizing the difference between $y\left(a^*, b^*\right)$ and $t\left(a^*, b^*\right)$ will not only minimize the degree of inconsistency when the assignment of $a^*, b^*$ violates the consistency principle, but also maximize the margin to the decision boundary when the assignment of $a^*, b^*$ accords the consistency principle. Therefore, we can consider the following non-linear optimization model to derive a value function that is as both consistent and robust as possible.

$$\max \prod_{a^*, b^* \in A^R \text{ with } C(a^*) \neq C(b^*)} \left[\frac{1}{1 + e^{-\mathbf{u}^{\mathrm{T}}\left(\mathbf{V}(a^*) - \mathbf{V}(b^*)\right)}}\right]^{t\left(a^*, b^*\right)} \cdot \left[1 - \frac{1}{1 + e^{-\mathbf{u}^{\mathrm{T}}\left(\mathbf{V}(a^*) - \mathbf{V}(b^*)\right)}}\right]^{1 - t\left(a^*, b^*\right)} \tag{8}$$

$$\text{s.t.} \quad \mathbf{u}^{\mathrm{T}}\mathbf{e} = 1, \ \mathbf{u} \geqslant \mathbf{0}. \tag{9}$$

Note that the objective (8) is in the multiplicative form, rather than the additive form, because the former avoids the compensation between different terms in the objective, which is especially useful when the number of pairs of reference alternatives is large. The objective (8) has been used in the logistic regression, a machine learning model for binary classification in the probabilistic framework [48].

### 3.3 Algorithm based on the MapReduce framework

In order to solve the non-linear optimization model (8)-(9) for a large-scale problem efficiently, we propose a new parallel algorithm based on the Zoutendijk's feasible direction method [51] and the MapReduce framework. By taking the logarithm of (8), the non-linear optimization problem (8)-(9) can be organized as follows

$$\max \sum_{a^*, b^* \in A^R \text{ with } C(a^*) \neq C(b^*)} \left[t\left(a^*, b^*\right) \ln \frac{1}{1 + e^{-\mathbf{u}^{\mathrm{T}}\left(\mathbf{V}(a^*) - \mathbf{V}(b^*)\right)}} + \left(1 - t\left(a^*, b^*\right)\right) \ln \frac{e^{-\mathbf{u}^{\mathrm{T}}\left(\mathbf{V}(a^*) - \mathbf{V}(b^*)\right)}}{1 + e^{-\mathbf{u}^{\mathrm{T}}\left(\mathbf{V}(a^*) - \mathbf{V}(b^*)\right)}}\right] \tag{10}$$

$$\text{s.t.} \ \mathbf{u}^{\mathrm{T}}\mathbf{e} = 1, \ \mathbf{u} \geqslant \mathbf{0}. \tag{11}$$

Then, the model (10)-(11) can be reformulated as follows

$$\min f\left(\mathbf{u}\right) = \sum_{a^*, b^* \in A^R \text{ with } C(a^*) \neq C(b^*)} \left[-t\left(a^*, b^*\right) \cdot \mathbf{u}^{\mathrm{T}}\left(\mathbf{V}\left(a^*\right) - \mathbf{V}\left(b^*\right)\right) + \ln\left(1 + e^{\mathbf{u}^{\mathrm{T}}\left(\mathbf{V}(a^*) - \mathbf{V}(b^*)\right)}\right)\right] \tag{12}$$

$$\text{s.t.} \ \mathbf{u}^{\mathrm{T}}\mathbf{e} = 1, \ \mathbf{u} \geqslant \mathbf{0}. \tag{13}$$

Note that the model (12)-(13) is a convex optimization problem because the objective (12) is convex [48].

**Definition 3.** [5]. Let $\hat{\mathbf{u}}$ be a feasible solution of the model (12)-(13). A vector $\mathbf{d} \in \mathbb{R}^{\gamma}$ is said to be a feasible direction at $\hat{\mathbf{u}}$, if there exists a positive number $\delta$ such that $\hat{\mathbf{u}} + \lambda\mathbf{d}$ is a feasible solution for any scalar $\lambda \in (0, \delta)$.

**Proposition 1.** Let $\hat{\mathbf{u}} = \left(\Delta\hat{u}_1^1, ..., \Delta\hat{u}_1^{\gamma_1}, ..., \Delta\hat{u}_n^1, ..., \Delta\hat{u}_n^{\gamma_n}\right)^{\mathrm{T}}$ be a feasible solution of the model (12)-(13), and

$$\Omega = \left\{(s, j) \mid \Delta\hat{u}_j^s = 0, \ s = 1, ..., \gamma_j, \ j = 1, ..., n\right\},$$

and

$$\bar{\Omega} = \left\{(s, j) \mid \Delta\hat{u}_j^s > 0, \ s = 1, ..., \gamma_j, \ j = 1, ..., n\right\}.$$

A vector $\mathbf{d} = \left(d_1^1, ..., d_1^{\gamma_1}, ..., d_n^1, ..., d_n^{\gamma_n}\right)^{\mathrm{T}}$ is a feasible direction at $\hat{\mathbf{u}}$, if and only if $d_j^s \geqslant 0, \ (s, j) \in \Omega$ and $\mathbf{d}^{\mathrm{T}}\mathbf{e} = 0$.

Proof. On the one hand, suppose that $\mathbf{d}$ is a feasible direction. According to the definition of feasible direction, there exists a positive scalar $\lambda$ for which $\hat{\mathbf{u}} + \lambda\mathbf{d}$ is a feasible solution, i.e., $(\hat{\mathbf{u}} + \lambda\mathbf{d})^{\mathrm{T}}\mathbf{e} = 1$ and $\hat{\mathbf{u}} + \lambda\mathbf{d} \geqslant \mathbf{0}$. Because $\hat{\mathbf{u}}$ is a feasible solution, we have $\hat{\mathbf{u}}^{\mathrm{T}}\mathbf{e} = 1$, and $\Delta\hat{u}_j^s = 0$ for $(s, j) \in \Omega$, and $\Delta\hat{u}_j^s > 0$ for $(s, j) \in \bar{\Omega}$. Therefore, we have $\mathbf{d}^{\mathrm{T}}\mathbf{e} = 0$ and $d_j^s \geqslant 0, \ (s, j) \in \Omega$. On the other hand, assume that $d_j^s \geqslant 0, \ (s, j) \in \Omega$ and $\mathbf{d}^{\mathrm{T}}\mathbf{e} = 0$. Because $\Delta\hat{u}_j^s > 0$ for $(s, j) \in \bar{\Omega}$, there exists a positive number $\delta$ such that $\Delta\hat{u}_j^s + \lambda d_j^s \geqslant 0$, $(s, j) \in \bar{\Omega}$ for any scalar $\lambda \in (0, \delta)$. Moreover, as $\Delta\hat{u}_j^s = 0$ for $(s, j) \in \Omega$, we have $\Delta\hat{u}_j^s + \lambda d_j^s = 0$ for $(s, j) \in \Omega$. Thus, $\hat{\mathbf{u}} + \lambda\mathbf{d} \geqslant \mathbf{0}$. Additionally, since $\mathbf{d}^{\mathrm{T}}\mathbf{e} = 0$ and $\hat{\mathbf{u}}^{\mathrm{T}}\mathbf{e} = 1$, it must be that $(\hat{\mathbf{u}} + \lambda\mathbf{d})^{\mathrm{T}}\mathbf{e} = 1$. Therefore, $\hat{\mathbf{u}} + \lambda\mathbf{d}$ is a feasible solution and $\mathbf{d}$ is a feasible direction. □

**Definition 4.** [51]. Let $\hat{\mathbf{u}}$ be a feasible solution of the model (12)-(13). A vector $\mathbf{d} \in \mathbb{R}^{\gamma}$ is said to be a descent direction at $\hat{\mathbf{u}}$, if there exists a positive number $\delta$ such that $f\left(\hat{\mathbf{u}} + \lambda\mathbf{d}\right) < f\left(\hat{\mathbf{u}}\right)$ for any scalar $\lambda \in (0, \delta)$.

The Zoutendijk's feasible direction method iteratively searches for a direction $\mathbf{d}$ that is both feasible and descent given a current feasible solution $\hat{\mathbf{u}}$. According to Definition 3.3 and Proposition 3.3, searching for such a direction $\mathbf{u}$ can be addressed by solving the following LP model

$$\min h\left(\mathbf{d}\right) = \nabla f(\hat{\mathbf{u}})^{\mathrm{T}}\mathbf{d} \tag{14}$$

$$\text{s.t.} \ d_j^s \geqslant 0, \ (s, j) \in \Omega, \tag{15}$$

$$\mathbf{d}^{\mathrm{T}}\mathbf{e} = 0, \tag{16}$$

$$\left|d_j^s\right| \leqslant 1, \ (s, j) \in \Omega \cup \bar{\Omega}, \tag{17}$$

where (17) guarantees to derive a bounded solution. Obviously, $\mathbf{d} = \mathbf{0}$ is a feasible solution of the LP model (14)-(17). Thus, $h\left(\mathbf{d}\right)$ at the optimum must be not greater than zero. If $h\left(\mathbf{d}\right) < 0$ at the optimum, then $\mathbf{d}$ is a direction that is both feasible and descent; otherwise, $\hat{\mathbf{u}}$ is the global optimal solution of the model (12)-(13) which is proved by the following proposition.

**Proposition 2.** Let $\hat{\mathbf{u}} = \left(\Delta\hat{u}_1^1, ..., \Delta\hat{u}_1^{\gamma_1}, ..., \Delta\hat{u}_n^1, ..., \Delta\hat{u}_n^{\gamma_n}\right)^{\mathrm{T}}$ be a feasible solution of the model (12)-(13), and

$$\Omega = \left\{(s, j) \mid \Delta\hat{u}_j^s = 0, \ s = 1, ..., \gamma_j, \ j = 1, ..., n\right\},$$

and

$$\bar{\Omega} = \left\{(s,j) \mid \Delta\hat{u}_j^s > 0,\ s = 1, ..., \gamma_j,\ j = 1, ..., n\right\}.$$

Then, $\hat{\mathbf{u}}$ is the global optimal solution of the model (12)-(13), if and only if, for the LP model (14)-(17), the objective $h(\mathbf{d})$ at the optimum is equal to zero.

Proof. For the model (12)-(13), according to the Karush–Kuhn–Tucker conditions [51], $\hat{\mathbf{u}}$ is a local optimal solution, if and only if there exist $\theta_j^s \geqslant 0$, $(s,j) \in \Omega$ and $\rho$ such that $\frac{\partial f(\hat{\mathbf{u}})}{\partial u_j^s} - \theta_j^s - \rho = 0,\ (s,j) \in \Omega$ and $\frac{\partial f(\hat{\mathbf{u}})}{\partial u_j^s} - \rho = 0,\ (s,j) \in \bar{\Omega}$. Let $\rho = \mu - \tau$ and $\mu, \tau \geqslant 0$. Then, we have

$$(-1, -1, 1)^{\mathrm{T}} \left(\theta_j^s, \mu, \tau\right) = -\frac{\partial f(\hat{\mathbf{u}})}{\partial u_j^s},\ \ (s,j) \in \Omega, \tag{18}$$

$$(0, -1, 1)^{\mathrm{T}} \left(\theta_j^s, \mu, \tau\right) = -\frac{\partial f(\hat{\mathbf{u}})}{\partial u_j^s},\ \ (s,j) \in \bar{\Omega}, \tag{19}$$

$$\left(\theta_j^s, \mu, \tau\right) \geqslant \mathbf{0}. \tag{20}$$

According to the Farkas' lemma [5], (18)-(20) is feasible if and only if $-d_j^s \leqslant 0,\ (s,j) \in \Omega$, $-\mathbf{d}^{\mathrm{T}}\mathbf{e} = 0$, $\mathbf{d}^{\mathrm{T}}\mathbf{e} = 0$ and $-\nabla f(\hat{\mathbf{u}})^{\mathrm{T}}\mathbf{d} > 0$ is infeasible. Then, $\nabla f(\hat{\mathbf{u}})^{\mathrm{T}}\mathbf{d} < 0$, $d_j^s \geqslant 0,\ (s,j) \in \Omega$ and $\mathbf{d}^{\mathrm{T}}\mathbf{e} = 0$ is infeasible. Thus, $\hat{\mathbf{u}}$ is a local optimal solution of the model (12)-(13), if and only if, for the LP model (14)-(17), the objective $h(\mathbf{d})$ at the optimum is equal to zero. Considering that the model (12)-(13) is a convex optimization problem, $\hat{\mathbf{u}}$ is the global optimal solution. □

With the current feasible solution $\hat{\mathbf{u}} = \left(\Delta\hat{u}_1^1, ..., \Delta\hat{u}_1^{\gamma_1}, ..., \Delta\hat{u}_n^1, ..., \Delta\hat{u}_n^{\gamma_n}\right)^{\mathrm{T}}$ and the obtained direction $\mathbf{d} = \left(d_1^1, ..., d_1^{\gamma_1}, ..., d_n^1, ..., d_n^{\gamma_n}\right)^{\mathrm{T}}$ that is both feasible and descent, the Zoutendijk's feasible direction method proceeds to the next iteration according to the following equation

$$\hat{\mathbf{u}}' = \hat{\mathbf{u}} + \lambda\mathbf{d}, \tag{21}$$

where $\hat{\mathbf{u}}'$ is the feasible solution for the next iteration. Let us consider to determine $\lambda$ for (21) which should guarantee that (1) $\hat{\mathbf{u}} + \lambda\mathbf{d}$ is feasible, and (2) the objective $f(\hat{\mathbf{u}} + \lambda\mathbf{d})$ of the model (12)-(13) should decrease as quickly as possible. This can be achieved by solving the following optimization problem

$$\min f(\hat{\mathbf{u}} + \lambda\mathbf{d}) \tag{22}$$

$$\text{s.t. } (\hat{\mathbf{u}} + \lambda\mathbf{d})^{\mathrm{T}}\mathbf{e} = 1, \tag{23}$$

$$\hat{\mathbf{u}} + \lambda\mathbf{d} \geqslant \mathbf{0}, \tag{24}$$

$$\lambda \geqslant 0. \tag{25}$$

**Proposition 3.** The model (22)-(25) is equivalent to the following optimization problem

$$\min f(\hat{\mathbf{u}} + \lambda\mathbf{d}) \tag{26}$$

$$\text{s.t. } 0 \leqslant \lambda \leqslant \lambda_{\max}, \tag{27}$$

where $\lambda_{\max} = \begin{cases} \min\left\{-\frac{d_j^s}{\Delta\hat{u}_j^s} \middle| d_j^s < 0,\ (s,j) \in \bar{\Omega}\right\}, & \mathbf{d} \ngeqslant \mathbf{0}, \\ \infty, & \mathbf{d} \geqslant \mathbf{0}. \end{cases}$

Proof. Since $\hat{\mathbf{u}}$ is a feasible solution and $\mathbf{d}$ is a feasible direction, we have $\hat{\mathbf{u}}^{\mathrm{T}}\mathbf{e} = 1$ and $\mathbf{d}^{\mathrm{T}}\mathbf{e} = 0$. Thus, Constraint (23) can be eliminated. Moreover, considering that $\hat{\mathbf{u}} \geqslant \mathbf{0}$ and $d_j^s \geqslant 0$ for $(s,j) \in \Omega$ and $\lambda \geqslant 0$, we have $\Delta\hat{u}_j^s + \lambda d_j^s \geqslant 0$ for $(s,j) \in \Omega$. Thus, Constraint (24) can be reduced to $\Delta\hat{u}_j^s + \lambda d_j^s \geqslant 0$ for $(s,j) \in \bar{\Omega}$. Then, we can derive the upper bound for $\lambda$ as $\lambda_{\max} = \begin{cases} \min\left\{-\frac{d_j^s}{\Delta\hat{u}_j^s} \middle| d_j^s < 0,\ (s,j) \in \bar{\Omega}\right\}, & \mathbf{d} \ngeqslant \mathbf{0}, \\ \infty, & \mathbf{d} \geqslant \mathbf{0}. \end{cases}$ Therefore, the model (22)-(25) can be transformed to the model (26)-(27) equivalently. □

Since the model (26)-(27) contains only one variable $\lambda$, we can use the golden section method [51] to address the optimization problem.

Based on the above analysis, we propose an algorithm to implement the Zoutendijk's feasible direction method based on the MapReduce framework for a large-scale problem. Considering that the number of reference alternatives in a large-scale problem is reasonably large, the objective (12) is composed of a huge number of terms (i.e., $-t\left(a^*, b^*\right)\mathbf{u}^{\mathrm{T}}\left(\mathbf{V}(a^*) - \mathbf{V}(b^*)\right) + \ln\left(1 + e^{\mathbf{u}^{\mathrm{T}}\left(\mathbf{V}(a^*) - \mathbf{V}(b^*)\right)}\right)$, $a^*, b^* \in A^R$). This inspires us to utilize the MapReduce framework to accelerate the computation of $\mathbf{u}$ and $\nabla f(\mathbf{u})$ for the Zoutendijk's feasible direction method. Algorithm 1 describes the whole process.

**Algorithm 1** The Zoutendijk's feasible direction method based on the MapReduce framework.

**Input:**

Initial feasible solution $\hat{\mathbf{u}} = (1/\gamma,\ ...,\ 1/\gamma)^{\mathrm{T}}$.

1: Determine $\Omega$ and $\bar{\Omega}$ according to the current feasible solution $\hat{\mathbf{u}}$.
2: Calculate $\nabla f(\hat{\mathbf{u}})$ using the MapReduce framework (see Algorithm (2)-(3)).
3: Solve the LP model (14)-(17) and obtain the optimal solution $\mathbf{d}^*$.
4: **if** $\nabla f(\hat{\mathbf{u}})^{\mathrm{T}}\mathbf{d}^* = 0$ **then**
5: Stop and $\hat{\mathbf{u}}$ is the global optimal solution.
6: **else**
7: Use Algorithm (4) to solve the model (26)-(27) and obtain the optimal solution $\lambda^*$.
8: Update $\hat{\mathbf{u}} \leftarrow \hat{\mathbf{u}} + \lambda^*\mathbf{d}^*$.
9: Go to step 1.
10: **end if**

**Output:**

The optimal solution $\hat{\mathbf{u}}$.

**Algorithm 2** Calculate $\nabla f(\hat{\mathbf{u}})$: Map phase.

**Input:**

$\langle key, value\rangle$ where $key$ is the index of subset and $value$ is the subset of pairs of reference alternatives $(a^*, b^*) \in A^R \times A^R$ such that $C(a^*) \neq C(b^*)$.

1: $\boldsymbol{\rho} \leftarrow \mathbf{0}$.
2: **for** any pair of reference alternatives $(a^*, b^*)$ in this subset **do**
3: $\boldsymbol{\rho} \leftarrow \boldsymbol{\rho} + \left[-t(a^*, b^*) + \frac{e^{\hat{\mathbf{u}}^{\mathrm{T}}\left(\mathbf{V}(a^*) - \mathbf{V}(b^*)\right)}}{1 + e^{\hat{\mathbf{u}}^{\mathrm{T}}\left(\mathbf{V}(a^*) - \mathbf{V}(b^*)\right)}}\right]\left(\mathbf{V}(a^*) - \mathbf{V}(b^*)\right)$.
4: **end for**

**Output:**

$\langle key = \hat{\mathbf{u}}, value = \boldsymbol{\rho}\rangle$.

With the optimal solution $\hat{\mathbf{u}}$, we can calculate global values $U(a)$ for each alternative $a \in A \cup A^R$. Then, in order to determine the assignment for each alternative $a \in A$, we can calculate the following consistency degree for quantifying the assignment $a \to C_s$, $s = 1, ..., p$, and choose $C_{s^*}$ with the maximum $S(a \to C_{s^*})$ as the

[3] Note that the implementation of calculation $f(\hat{\mathbf{u}} + \lambda_1\mathbf{d}^*)$ and $f(\hat{\mathbf{u}} + \lambda_2\mathbf{d}^*)$ based on the MapReduce framework is similar to Algorithm (2)-(3), and we do not present the algorithm here to save space.

**Algorithm 3** Calculate $\nabla f(\hat{\mathbf{u}})$: Reduce phase.

**Input:**

$\langle key = \hat{\mathbf{u}}, value = list(\boldsymbol{\rho})\rangle$.

1: $\boldsymbol{\delta} \leftarrow \mathbf{0}$.
2: **for** any $\boldsymbol{\rho}$ in $list(\boldsymbol{\rho})$ **do**
3: $\quad \boldsymbol{\delta} \leftarrow \boldsymbol{\delta} + \boldsymbol{\rho}$.
4: **end for**

**Output:**

$\langle key = \hat{\mathbf{u}}, value = \boldsymbol{\delta}\rangle$ where $\boldsymbol{\delta}$ is equal to $\nabla f(\hat{\mathbf{u}})$.

**Algorithm 4** The golden section method for the model (26)-(27).

**Input:**

$\lambda_{\max}$ and stopping tolerance $\varepsilon$.

1: $a \leftarrow 0$, $b \leftarrow \lambda_{\max}$, $\lambda_1 \leftarrow a + 0.382(b-a)$, $\lambda_2 \leftarrow a + 0.618(b-a)$.
2: **if** $b - a < \varepsilon$ **then**
3: $\quad$ Stop and $\lambda^* \leftarrow (a+b)/2$ is the optimal solution.
4: **else**
5: $\quad$ Calculate $f(\hat{\mathbf{u}} + \lambda_1 \mathbf{d}^*)$ and $f(\hat{\mathbf{u}} + \lambda_2 \mathbf{d}^*)$ based on the MapReduce framework [3].
6: $\quad$ **if** $f(\hat{\mathbf{u}} + \lambda_1 \mathbf{d}^*) > f(\hat{\mathbf{u}} + \lambda_2 \mathbf{d}^*)$ **then**
7: $\quad\quad a \leftarrow \lambda_1$, $\lambda_1 \leftarrow \lambda_2$, $\lambda_2 \leftarrow a + 0.618(b-a)$ and go to step 2.
8: $\quad$ **else**
9: $\quad\quad b \leftarrow \lambda_2$, $\lambda_2 \leftarrow \lambda_1$, $\lambda_1 \leftarrow a + 0.382(b-a)$ and go to step 2.
10: $\quad$ **end if**
11: **end if**

**Output:**

The optimal solution $\lambda^*$.

final assignment of alternative $a$:

$$S(a \to C_s) = \frac{1}{\left|A_1^R \cup ... \cup A_{s-1}^R \cup A_{s+1}^R \cup ... \cup A_p^R\right|} \cdot \left(\left|\left\{a^* \in A_1^R \cup ... \cup A_{s-1}^R | U(a) > U(a^*)\right\}\right| + \left|\left\{a^* \in A_{s+1}^R \cup ... \cup A_p^R | U(a) < U(a^*)\right\}\right|\right)$$

## REFERENCES


[1] Francesca Abastante, Marta Bottero, Salvatore Greco, and Isabella Lami, 'Addressing the location of undesirable facilities through the dominance-based rough set approach', *Journal of Multi-Criteria Decision Analysis*, **21**(1-2), 3–23, (2014).

[2] J Almeida-Dias, José Rui Figueira, and Bernard Roy, 'Electre Tri-C: A multiple criteria sorting method based on characteristic reference actions', *European Journal of Operational Research*, **204**(3), 565–580, (2010).

[3] J Almeida-Dias, José Rui Figueira, and Bernard Roy, 'A multiple criteria sorting method where each category is characterized by several reference actions: The Electre Tri-nC method', *European Journal of Operational Research*, **217**(3), 567–579, (2012).

[4] Silvia Angilella and Sebastiano Mazzù, 'The financing of innovative SMEs: A multicriteria credit rating model', *European Journal of Operational Research*, **244**(2), 540–554, (2015).

[5] Dimitris Bertsimas and John N Tsitsiklis, *Introduction to linear optimization*, volume 6, Athena Scientific Belmont, MA, 1997.

[6] Fu-Ling Cai, Xiuwu Liao, and Kan-Liang Wang, 'An interactive sorting approach based on the assignment examples of multiple decision makers with different priorities', *Annals of Operations Research*, **197**(1), 87–108, (2012).

[7] Olivier Cailloux, Patrick Meyer, and Vincent Mousseau, 'Eliciting ELECTRE TRI category limits for a group of decision makers', *European Journal of Operational Research*, **223**(1), 133–140, (2012).

[8] Ana Carolina Scanavachi Moreira Campos, Bertrand Mareschal, and Adiel Teixeira de Almeida, 'Fuzzy FlowSort: An integration of the FlowSort method and Fuzzy Set Theory for decision making on the basis of inaccurate quantitative data', *Information Sciences*, **293**, 115–124, (2015).

[9] Ye Chen, Keith W Hipel, and D Marc Kilgour, 'Multiple-criteria sorting using case-based distance models with an application in water resources management', *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, **37**(5), 680–691, (2007).

[10] Ye Chen, Kevin W Li, D Marc Kilgour, and Keith W Hipel, 'A case-based distance model for multiple criteria ABC analysis', *Computers & Operations Research*, **35**(3), 776–796, (2008).

[11] Salvatore Corrente, Michael Doumpos, Salvatore Greco, Roman Słowiński, and Constantin Zopounidis, 'Multiple criteria hierarchy process for sorting problems based on ordinal regression with additive value functions', *Annals of Operations Research*, **251**(1-2), 117–139, (2017).

[12] Salvatore Corrente, Salvatore Greco, Miłosz Kadziński, and Roman Słowiński, 'Robust ordinal regression in preference learning and ranking', *Machine Learning*, **93**(2-3), 381–422, (2013).

[13] Salvatore Corrente, Salvatore Greco, and Roman Słowiński, 'Multiple criteria hierarchy process for ELECTRE Tri methods', *European Journal of Operational Research*, **252**(1), 191–203, (2016).

[14] Caroline Maria de Miranda Mota and Adiel Teixeira de Almeida, 'A multicriteria decision model for assigning priority classes to activities in project management', *Annals of Operations Research*, **199**(1), 361–372, (2012).

[15] Sara Del Río, Victoria López, José Manuel Benítez, and Francisco Herrera, 'On the use of MapReduce for imbalanced big data using random forest', *Information Sciences*, **285**, 112–137, (2014).

[16] Michael Doumpos and Constantin Zopounidis, *Multicriteria decision aid classification methods*, Springer US, 2002.

[17] Michael Doumpos and Constantin Zopounidis, 'Disaggregation analysis and statistical learning: an integrated framework for multicriteria decision support', in *Handbook of Multicriteria Analysis*, 215–240, Springer, (2010).

[18] Michael Doumpos and Constantin Zopounidis, 'A multicriteria outranking modeling approach for credit rating', *Decision Sciences*, **42**(3), 721–742, (2011).

[19] Michael Doumpos and Constantin Zopounidis, 'Computational intelligence techniques for multicriteria decision aiding: An overview', *Multicriteria Decision Aid and Artificial Intelligence*, 1–23, (2013).

[20] Eduardo Fernández, José Rui Figueira, Jorge Navarro, and Bernard Roy, 'ELECTRE TRI-nB: A new multiple criteria ordinal classification method', *European Journal of Operational Research*, **263**(1), 214–224, (2017).

[21] Eduardo Fernandez, Jorge Navarro, Alfonso Duarte, and Guillermo Ibarra, 'Core: A decision support system for regional competitiveness analysis based on multi-criteria sorting', *Decision Support Systems*, **54**(3), 1417–1426, (2013).

[22] Salvatore Greco, Matthias Ehrgott, and José Rui Figueira, *Trends in multiple criteria decision analysis*, Springer US, 2010.

[23] Salvatore Greco, Matthias Ehrgott, and José Rui Figueira, *Multiple Criteria Decision Analysis: State of the Art Surveys*, Springer, 2016.

[24] Salvatore Greco, Benedetto Matarazzo, and Roman Slowinski, 'Rough sets theory for multicriteria decision analysis', *European Journal of Operational Research*, **129**(1), 1–47, (2001).

[25] Salvatore Greco, Benedetto Matarazzo, and Roman Słowiński, 'Dominance-based rough set approach to decision under uncertainty and time preference', *Annals of Operations Research*, **176**(1), 41–75, (2010).

[26] Salvatore Greco, Benedetto Matarazzo, Roman Slowinski, and Stelios Zanakis, 'Global investing risk: a case study of knowledge assessment via rough sets', *Annals of Operations Research*, **185**(1), 105–138, (2011).

[27] Salvatore Greco, Vincent Mousseau, and Roman Słowiński, 'Ordinal regression revisited: multiple criteria ranking using a set of additive value functions', *European Journal of Operational Research*, **191**(2), 416–436, (2008).

[28] Salvatore Greco, Vincent Mousseau, and Roman Słowiński, 'Multiple criteria sorting with a set of additive value functions', *European Journal of Operational Research*, **207**(3), 1455–1470, (2010).

[29] Miłosz Kadziński and Krzysztof Ciomek, 'Integrated framework for preference modeling and robustness analysis for outranking-based multiple criteria sorting with ELECTRE and PROMETHEE', *Information Sciences*, **352**, 167–187, (2016).

[30] Miłosz Kadziński, Salvatore Greco, and Roman Słowiński, 'Robust ordinal regression for dominance-based rough set approach to multiple criteria sorting', *Information Sciences*, **283**, 211–228, (2014).

[31] Miłosz Kadziński and Roman Słowiński, 'Parametric evaluation of research units with respect to reference profiles', *Decision Support Systems*, **72**, 33–43, (2015).

[32] Miłosz Kadziński, Roman Słowiński, and Salvatore Greco, 'Robustness analysis for decision under uncertainty with rule-based preference model', *Information Sciences*, **328**, 321–339, (2016).

[33] Miłosz Kadziński, Tommi Tervonen, and José Rui Figueira, 'Robust multi-criteria sorting with the outranking preference model and characteristic profiles', *Omega*, **55**, 126–140, (2015).

[34] Murat Köksalan, Vincent Mousseau, Özgür Özpeynirci, and Selin Bilgin Özpeynirci, 'A new outranking-based approach for assigning alternatives to ordered classes', *Naval Research Logistics*, **56**(1), 74–85, (2009).

[35] Murat Köksalan and Selin Bilgin Özpeynirci, 'An interactive sorting method for additive utility functions', *Computers & Operations Research*, **36**(9), 2565–2572, (2009).

[36] Helge Langseth and Thomas D Nielsen, 'Scalable learning of probabilistic latent models for collaborative filtering', *Decision Support Systems*, **74**, 1–11, (2015).

[37] Boying Li, Eugene Chng, Alain Yee-Loong Chong, and Haijun Bao, 'Predicting online e-marketplace sales performances: A big data approach', *Computers & Industrial Engineering*, **101**, 565–571, (2016).

[38] Jiexun Li, Xin Li, and Bin Zhu, 'User opinion classification in social media: A global consistency maximization approach', *Information & Management*, **53**(8), 987–996, (2016).

[39] Xiaoping Li, Tianze Jiang, and Rubén Ruiz, 'Heuristics for periodical batch job scheduling in a mapreduce computing framework', *Information Sciences*, **326**, 119–133, (2016).

[40] Jiapeng Liu, Xiuwu Liao, Wei Huang, and Jian-bo Yang, 'A new decision-making approach for multiple criteria sorting with an imbalanced set of assignment examples', *European Journal of Operational Research*, **265**(2), 598–620, (2018).

[41] Jiapeng Liu, Xiuwu Liao, and Jian-bo Yang, 'A group decision-making approach based on evidential reasoning for multiple criteria sorting problem with uncertainty', *European Journal of Operational Research*, **246**(3), 858–873, (2015).

[42] Jiapeng Liu, Xiuwu Liao, Wenhong Zhao, and Na Yang, 'A classification approach based on the outranking model for multiple criteria ABC analysis', *Omega*, **61**, 19–34, (2016).

[43] Francis Macary, Juscelino Almeida-Dias, Daniel Uny, and Anne Probst, 'Assessment of the effects of best environmental practices on reducing pesticide contamination in surface water, using multi-criteria modelling combined with a GIS', *International Journal of Multicriteria Decision Making 73*, **3**(2-3), 178–211, (2013).

[44] Daniel Mailly, Irène Abi-Zeid, and Steeve Pepin, 'A multi-criteria classification approach for identifying favourable climates for tourism', *Journal of Multi-Criteria Decision Analysis*, **21**(1-2), 65–75, (2014).

[45] James G March, 'Bounded rationality, ambiguity, and the engineering of choice', *The Bell Journal of Economics*, 587–608, (1978).

[46] Lucas Marin, David Isern, Antonio Moreno, and Aida Valls, 'On-line dynamic adaptation of fuzzy preferences', *Information Sciences*, **220**, 5–21, (2013).

[47] Donald Miner and Adam Shook, *MapReduce Design Patterns: Building Effective Algorithms and Analytics for Hadoop and Other Systems*, O'Reilly Media, Inc., 2012.

[48] Kevin P. Murphy, *Machine Learning: A Probabilistic Perspective*, MIT Press, 2012.

[49] Philippe Nemery and Claude Lamboray, 'Flowsort: a flow-based sorting method with limiting or central profiles', *Top*, **16**(1), 90–113, (2008).

[50] Thin Nguyen, Mark E Larsen, Bridianne ODea, Duc Thanh Nguyen, John Yearwood, Dinh Phung, Svetha Venkatesh, and Helen Christensen, 'Kernel-based features for predicting population health indices from geocoded social media data', *Decision Support Systems*, **102**, 22–31, (2017).

[51] Jorge Nocedal and Stephen J. Wright, *Numerical optimization*, Springer, 2004.

[52] Maria Franca Norese and Vincenzo Carbone, 'An application of ELECTRE Tri to support innovation', *Journal of Multi-Criteria Decision Analysis*, **21**(1-2), 77–93, (2014).

[53] Clara Rocha and Luis C Dias, 'An algorithm for ordinal sorting based on ELECTRE with categories defined by examples', *Journal of Global Optimization*, **42**(2), 255–277, (2008).

[54] Bernard Roy, *Multicriteria Methodology for Decision Aiding*, Springer US, 1996.

[55] Aydin Ulucan and Kazim Baris Atici, 'A multiple criteria sorting methodology with multiple classification criteria and an application to country risk evaluation', *Technological and Economic Development of Economy*, **19**(1), 93–124, (2013).

[56] Jun Zheng, Stéphane Aimé Metchebon Takougang, Vincent Mousseau, and Marc Pirlot, 'Learning criteria weights of an optimistic Electre Tri sorting rule', *Computers & Operations Research*, **49**, 28–40, (2014).

[57] Constantin Zopounidis and Michael Doumpos, 'Building additive utilities for multi-group hierarchical discrimination: The MHDIS method', *Optimization Methods and Software*, **14**(3), 219–240, (2000).