Omega 61 (2016) 19–34

Contents lists available at ScienceDirect

Omega

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

A classification approach based on the outranking model for multiple criteria ABC analysis ☆

Jiapeng Liu, Xiuwu Liao *, Wenhong Zhao, Na Yang

School of Management, The Key Lab of the Ministry of Education for Process Control & Efficiency Engineering, Xi'an Jiaotong University, Xi'an, 710049 Shaanxi, PR China

A R T I C L E I N F O

Article history:
Received 7 October 2014
Accepted 6 July 2015
Available online 16 July 2015

Keywords:
ABC inventory classification
Multiple criteria decision analysis
Clustering
Simulated annealing algorithm

A B S T R A C T

The multiple criteria ABC analysis is widely used in inventory management, and it can help organizations to assign inventory items into different classes with respect to several evaluation criteria. Many approaches have been proposed in the literature for addressing such a problem. However, most of these approaches are fully compensatory in multiple criteria aggregation. This means that an item scoring badly on one or more key criteria could be placed in good classes because these bad performances could be compensated by other criteria. Thus, it is necessary to consider the non-compensation in the multiple criteria ABC analysis. To the best of our knowledge, the ABC classification problem with non-compensation among criteria has not been studied sufficiently. We thus propose a new classification approach based on the outranking model to cope with such a problem in this paper. However, the relational nature of the outranking model makes the search for the optimal classification solution a complex combinatorial optimization problem. It is very time-consuming to solve such a problem using mathematical programming techniques when the inventory size is large. Therefore, we combine the clustering analysis and the simulated annealing algorithm to search for the optimal classification. The clustering analysis groups similar inventory items together and builds up the hierarchy of clusters of items. The simulated annealing algorithm searches for the optimal classification on different levels of the hierarchy. The proposed approach is illustrated by a practical example from a Chinese manufacturer. Furthermore, we validate the performance of the approach through experimental investigation on a large set of artificially generated data at the end of the paper.

1. Introduction

The efficient control of inventory can help firms improve their competitiveness [42]. As a basic methodology, the ABC analysis is widely used to manage a number of inventory items in organizations. The ABC analysis helps an inventory manager to divide the inventory items into three classes according to specific criteria. That is, items of high value but small in number are termed as class A, items of low value but large in number are termed as class C, and items that fall between these two classes are termed as class B. The ABC analysis provides a mechanism for identifying items that will have a significant impact on overall inventory cost while providing a method for pinpointing different categories of inventory that will require different management and control policies.

The traditional ABC classification method considers only one criterion, i.e., the annual dollar usage, to classify inventory items. This method is successful only when inventory items are fairly homogeneous and the main difference among items is in their annual dollar usages [36]. In practice, some organizations, such as P&G, Lenovo, and ZTE, have to control thousands of inventory items that are not necessarily homogeneous. Thus, the traditional ABC analysis may be counterproductive in real-world classification of inventory items. It has been recognized that other criteria, such as inventory cost, part criticality, lead time, commonality, obsolescence, substitutability, durability, repairability, and so on, are also important for inventory classification [21,33,36]. To solve such a multiple criteria ABC inventory classification (MCABC) problem, a great variety of methods has been proposed during the past decades. Bhattacharya et al. [3], Rezaei and Dowlatshahi [37], and Torabi et al. [44] have provided comprehensive reviews on the various MCABC approaches in the literature.

Despite the advantages of these approaches, it should be noted that most of them are fully compensatory in multiple criteria aggregation, i.e., a significantly weak criterion value of an item could be directly compensated by other good criteria values. Thus,

☆This manuscript was processed by Associate Editor Yeh.

* Corresponding author.
E-mail address: liaoxiuwu@mail.xjtu.edu.cn (X. Liao).

http://dx.doi.org/10.1016/j.omega.2015.07.004

an item scoring badly on one or more key criteria may be placed in good classes. Such classification may not be reasonable in some real-world applications because organizations do not need to manage these items that score badly on key criteria with more efforts and control. Therefore, it is necessary to consider the non-compensation in the multiple criteria ABC analysis. From our point of view, not enough attention has been paid to the MCABC problem with the non-compensation among criteria. Some exceptions include the studies developed by Zhou and Fan [45], Hadi-Vencheh [22], and Lolli et al. [29].

In this paper, we propose an alternative approach based on the outranking relation defined in ELECTRE [15] for the MCABC problem when the non-compensation among criteria should be considered. ELECTRE is a family of multiple criteria decision analysis methods that highlight the limited possibility of compensation through the construction of the outranking relation using tests of concordance and discordance [16]. Concerning the concordance test, the fact that item a outranks or does not outrank item b is only relevant to how much the performances of a are better or worse than the performances of b on the criteria. With regard to the discordance test, the existence of veto thresholds strengthens the non-compensation effect in ELECTRE methods. When the difference between the evaluations of a and b on a certain criterion is greater than the veto threshold, then no improvement of the performance of b or no deterioration of the performance of a, with respect to other criteria, can compensate this veto effect.

In the proposed approach, taking into account the requirement of the ABC analysis, the cardinality limitation of items in each class is specified in advance. We define an inconsistency measure based on the outranking relation and an average distance between items based on the multiple criteria distance to evaluate the classification. We pursue to find a solution that minimizes the two evaluation indices. Due to the relational nature of the outranking model, the search for the optimal classification is a complex combinatorial optimization problem. It is very time-consuming to solve such a problem using mathematical programming techniques, especially for large-size problems. In this paper, we combine the clustering analysis and the simulated annealing algorithm to search for the optimal classification. The clustering analysis aims to identify and group similar items into clusters. An agglomerative hierarchy clustering algorithm is employed to construct the hierarchy of the identified clusters. The hierarchy of clusters permits the search for the optimal classification to proceed on different levels of granularity. In the developed simulated annealing algorithm, the search for the optimal classification is performed according to the hierarchy of clusters from top to bottom. Such a combination of the two basic methods provides a tool to solve the combinatorial optimization problem in an efficient way.

Our work can be regarded as a new method of building the preference order of the identified clusters for the MCABC problem. Over the past few years, several similar techniques have been proposed for the clustering problem. Fernandez et al. [14], De Smet et al. [11], and Meyer and Olteanu [31] all combined an outranking model with a metaheuristic algorithm to explore the preference relation between clusters. Compared with these existing approaches, our work highlights the idea of “progressive refinement”, which is reflected in the hierarchy of clusters and the search strategy. Moreover, our algorithm can escape from a local optimum and avoid premature convergence, which is implemented by accepting a worse solution with a certain probability. Moreover, the cardinality of items in each class is incorporated into our algorithm, and we can specify the cardinality limitation in advance.

The approach presented in this paper is distinguished from the previous MCABC methods by the following new features. First, the non-compensation among criteria is considered in the MCABC problem. We develop an outranking-based classification model that limits the compensation among criteria. Second, the clustering analysis is incorporated into the ABC classification process. The clustering analysis groups similar items together and builds up the hierarchy of clusters. It permits a search for the optimal classification on different levels of granularity, and it contributes to simplifying the complex combinatorial optimization problem. Third, a simulated annealing algorithm is developed to search for the optimal solution according to the constructed hierarchy of clusters. The algorithm solves the combinatorial optimization problem efficiently, especially when the inventory size is large.

The remainder of this paper is organized as follows. In Section 2, we provide a literature review on the MCABC problem. In Section 3, we give a brief introduction to the outranking relation in the ELECTRE method and the multiple criteria distance. In Section 4, we combine the multiple criteria clustering analysis and the simulated annealing algorithm to address the MCABC problem. Section 5 demonstrates the approach using an example. Section 6 compares the approach with another two heuristic algorithms. The paper ends with conclusions and discussion regarding future research.

2. Literature review

The ABC analysis has been a hot topic of numerous studies on inventory management. Since Flores and Whybark [17,18] first stressed the importance of considering multiple criteria in the ABC analysis, various approaches for addressing the MCABC problem have been proposed in the literature. The existing work can be classified into the following six categories: (a) AHP [40], (b) artificial intelligence technique, (c) statistical analysis, (d) Data Envelopment Analysis (DEA)-like approaches [8], (e) weighted Euclidean distance-based approaches, and (f) UTADIS method [12].

Many researchers have applied the analytic hierarchy process (AHP) to the MCABC problem [19,34,35,4,29]. The basic idea is to derive the weights of inventory items by pairwise comparing the criteria and inventory items. However, the most important problem associated with AHP is the subjectivity of the decision maker (DM) involved in the pairwise comparisons. Moreover, when the number of criteria increased, the consistency of judgment will be very sensitive, and reaching a consistent rate will be very difficult.

Several approaches based on artificial intelligence techniques have also been applied to the MCABC problem. Güvenir [20] and Güvenir and Erel [21] proposed an approach based on genetic algorithm to learn criteria weight vector and cut-off points between classes A and B and classes B and C. Güvenir and Erel [21] reported that the approach based on genetic algorithm performed better than that based on the AHP method. Partovi and Anandarajan [33] applied artificial neural networks in the MCABC problem. They used backpropagation (BP) and genetic algorithm in their approach and brought out non-linear relationships and interactions between criteria. The approach was compared with the multiple discriminant analysis technique and the results showed that the approach had a higher predictive accuracy than the discriminant analysis. Tsai and Yeh [43] proposed a particle swarm optimization approach for the MCABC problem. In this approach, inventory items are classified based on a specific objective or on multiple objectives.

Statistical techniques, such as clustering analysis and principle component analysis, are also applied to the MCABC problem. Cluster analysis is to group items with similar characteristics together. Cohen and Ernst [9] and Ernst and Cohen [13] proposed an approach based on cluster analysis for the MCABC problem. The approach uses a full combination of strategic and operational attributes. Lei et al. [28] combined the principle component analysis with artificial neural networks and the BP algorithm to classify inventory items. The hybrid approach can overcome the shortcomings of input limitation in artificial neural networks and improve the prediction accuracy.

The DEA is a powerful quantitative analytical tool for evaluating the performance of a set of alternatives called decision-making units (DMUs). Ramanathan [36] proposed a DEA-like weighted linear optimization model for the MCABC problem. Zhou and Fan [45] extended the model of Ramanathan [36] to provide each item with two sets of weights that were most favorable and least favorable to the item. Then, a composite index was built by combining the two extreme cases into a balanced performance score. Chen [6] proposed an improved approach for the MCABC problem in which all items were peer estimated. In addition, Chen [7] proposed an alternative approach for the MCABC problem by using two virtual items and incorporating the TOPSIS [1]. Ng [32] proposed another DEA-like model that converted all criteria measures of an inventory item into a scalar score, assuming that the criteria were ranked in descending order for all items. Hadi-Vencheh [22] developed a model for the ABC analysis that not only incorporated multiple criteria but also maintained the effects of weights in the final solution. Torabi et al. [44] proposed a modified version of an existent common weight DEA-like model that could handle both quantitative and qualitative criteria. Hadi-Vencheh and Mohamadghasemi [23] integrated fuzzy AHP and DEA analysis to address the situation wherein inventory items were assessed under each criterion using linguistic terms.

Some researchers have proposed several weighted Euclidean distance-based approaches for the MCABC problem. On the basis of the TOPSIS model, Bhattacharya et al. [3] developed a distance-based, multi-criteria consensus framework based on the concepts of ideal and negative-ideal solutions. Chen et al. [5] presented a quadratic programming model using weighted Euclidean distances based on the reference cases. Their model may lead to a situation wherein the number of incorrect classifications is high and multiple solutions exist because their model aims to minimize the overall squared errors instead of the total number of misclassifications. Ma [30] proposed a two-phase, case-based distance approach for the MCABC problem to improve the model of Chen et al. [5]. The approach has several advantages, such as reducing the number of misclassifications, improving multiple solution problems, and lessening the impact of outliers. However, all of the weighted Euclidean distance-based models have a critical problem in setting up the positive and negative ideal points. No solid theoretical foundation regarding this issue is available and the classification results depend on the setting of the two ideal points sensitively.

In addition to these weighted Euclidean distance-based models, Soylu and Akyol [41] applied the UTADIS method to the MCABC problem in which the DM specifies exemplary assignment of a number of reference alternatives. Each inventory item is assigned to a suitable class by comparing its global utility with the threshold of each class. The parameters of the classification model developed by the UTADIS method, including the criteria weights, the marginal utility functions, and the thresholds of classes, are induced using the disaggregation–aggregation (or regression) paradigm.

These approaches proposed in the literature provide promising and useful tools for addressing the MCABC problem. However, we should focus on two critical aspects of the problem. One is that the approaches based on AHP and weighted Euclidean distance and the UTADIS method are compensatory in criteria aggregation. The weights of criteria in these approaches can be interpreted as substitution rates by which a gain on one criterion can compensate a loss on another criterion. Thus, an item scoring badly on one or more key criteria could be placed in good classes because these bad performances could be compensated by other criteria. The DEA-like models proposed by Ramanathan [36] and Ng [32] could also lead to a situation where an item with a high value on an unimportant criterion is inappropriately classified as a class A item. To avoid such classification, Zhou and Fan [45] and Hadi-Vencheh [22] presented extended versions of the Ramanathan-model and the Ng-model, respectively, and they obtained more reasonable classifications, both of which could be regarded as non-compensatory models for the MCABC problem. Another non-compensatory model was developed by Lolli et al. [29], who imposed a veto condition on each criterion that prevented an item evaluated as bad on at least one criterion to be top ranked in the global aggregation.

The other aspect associated with the MCABC problem is that the cardinality limitation of items in each class increases the complexity of the problem, especially when using the linear programming-based approaches. If we consider the cardinality limitation in building linear programs, integer variables need to be introduced to indicate the classification of items, and thus, it produces mixed integer linear programs that are very difficult to solve when the inventory size is large. The approaches proposed by Chen et al. [5], Ma [30], and Soylu and Akyol [41] built linear programs to infer the parameters of the classification models based on the disaggregation–aggregation (or regression) paradigm. They did not consider the cardinality limitation in the models, which could lead to an inappropriate outcome where good classes contained more items than bad ones. If we extend these models by adding integer variables in order to fulfill the cardinality limitation, it will take more time and expenditure to obtain the optimal classification.

3. Definitions and basic concepts

We consider a decision problem in which a finite set of alternatives $\mathbf{A} = \{a_1, a_2, ..., a_n\}$ is evaluated on a family of m criteria $G = \{g_1, g_2, ..., g_m\}$ with $g_j : \mathbf{A} \to \mathbb{R}$, $j = 1, 2, ..., m$. $g_j(a_i)$ is the performance of alternative a_i on criterion g_j, $j = 1, 2, ..., m$. Without loss of generality, we assume that all criteria $g_j \in G$ are to be maximized, which means that the preference increases when the criterion performance increases. $w = \{w_1, w_2, ..., w_m\}$ is the normalized weight vector indicating the importance of each criterion.

In this paper, we use the outranking relation defined in the ELECTRE III method [39] as the non-compensatory preference model to cope with the decision problem. Such a model is a binary relation on the set $\mathbf{A}$, which is usually denoted by aSb corresponding to the statement "alternative a is at least as good as alternative b". Given any pair of alternatives $(a, b) \in \mathbf{A} \times \mathbf{A}$, a credibility index $\sigma(a, b)$ will be computed to verify the statement aSb. $\sigma(a, b)$ is computed as follows:

- *computation of partial concordance indices*: the partial concordance index $c_j(a, b)$ indicates the degree to which the jth criterion agrees with the statement that a outranks b. It is a function of the difference of performances $g_j(b) - g_j(a)$, the indifference threshold q_j and the preference threshold p_j, such that $p_j \geq q_j \geq 0$. $c_j(a, b)$ is defined as follows:

$$c_j(a,b) = \begin{cases} 0 & \text{if } g_j(b) - g_j(a) \geq p_j, \\ \frac{p_j + g_j(a) - g_j(b)}{p_j - q_j} & \text{if } q_j < g_j(b) - g_j(a) < p_j, \\ 1 & \text{if } g_j(b) - g_j(a) \leq q_j. \end{cases}$$

  The indifference threshold q_j represents the largest difference $g_j(b) - g_j(a)$ that preserves indifference between a and b on criterion g_j. The preference threshold p_j represents the smallest difference $g_j(b) - g_j(a)$ compatible with a preference in favor of b on criterion g_j [16].
- *computation of the global concordance index*: the global concordance index is defined by aggregating all of the m criteria as

follows:

$C(a,b) = \sum_{j=1}^{m} w_j \cdot c_j(a,b).$

- *computation of partial discordance indices*: the partial discordance index $d_j(a,b)$ indicates the degree to which the jth criterion disagrees with the statement that a outranks b. It is a function of the difference of performances $g_j(b)-g_j(a)$, the preference threshold p_j and the veto threshold v_j, such that $v_j > p_j \geq 0$. $d_j(a,b)$ is defined as follows:

$$d_j(a,b) = \begin{cases} 1 & \text{if } g_j(b)-g_j(a) \geq v_j, \\ \frac{g_j(b)-g_j(a)-p_j}{v_j-p_j} & \text{if } p_j < g_j(b)-g_j(a) < v_j, \\ 0 & \text{if } g_j(b)-g_j(a) \leq p_j. \end{cases}$$

  The veto threshold v_j represents the smallest difference $g_j(b)-g_j(a)$ incompatible with the statement aSb. The consideration of the veto threshold reinforces the non-compensatory character of the outranking relation.
- *computation of the credibility index*: the results of the concordance and discordance tests are aggregated into the credibility index $\sigma(a,b)$, which is defined as follows:

$$\sigma(a,b) = C(a,b) \prod_{g_j \in F} \frac{1-d_j(a,b)}{1-C(a,b)}$$

  where $F = \left\{ g_j \in G \mid d_j(a,b) > C(a,b) \right\}$. The credibility index $\sigma(a,b)$ ranges in [0,1] and represents the overall credibility degree of the outranking relation aSb.

The outranking relation aSb is considered to hold if $\sigma(a,b) \geq \lambda$, where λ is the credibility threshold within the range (0.5,1). The negation of the outranking relation aSb is written as aS^cb. Note that the outranking relation is not transitive. That is, for any three alternatives $a,b,c \in \mathbf{A}$, if aSb and bSc, we cannot conclude that aSc. This means that for any two alternatives $a,b \in \mathbf{A}$, we should calculate the credibility index $\sigma(a,b)$ and verify whether the outranking relation aSb holds.

The outranking relation aSb is used to establish the following four possible outcomes of the comparison between a and b:

(a) a is indifferent to b: $aIb \Leftrightarrow aSb \ \wedge \ bSa$;
(b) a is preferred to b: $aPb \Leftrightarrow aSb \ \wedge \ bS^ca$;
(c) b is preferred to a: $bPa \Leftrightarrow aS^cb \ \wedge \ bSa$;
(d) a is incomparable to b: $aRb \Leftrightarrow aS^cb \ \wedge \ bS^ca$.

The indifference relation I is reflexive and symmetric, the preference relation P is asymmetric and the incomparability relation R is irreflexive and symmetric. The three relations $\{I,P,R\}$ make up a preference structure on $\mathbf{A}$.

De Smet and Guzmán [10] proposed a multiple criteria distance to measure the similarity between alternatives. The multiple criteria distance is based on the profile of an alternative. The profile of alternative $a \in \mathbf{A}$, denoted by $Q(a)$, is defined as a 4-tuple $\langle Q_1(a), Q_2(a), Q_3(a), Q_4(a) \rangle$, where $Q_1(a) = \{ b \in \mathbf{A} \mid aRb \}$, $Q_2(a) = \{ b \in \mathbf{A} \mid bPa \}$, $Q_3(a) = \{ b \in \mathbf{A} \mid aIb \}$, and $Q_4(a) = \{ b \in \mathbf{A} \mid aPb \}$. Then, the distance between two alternatives $a,b \in \mathbf{A}$ is defined as the difference between their profiles:

$$dist(a,b) = 1 - \frac{\sum_{k=1}^{4} |Q_k(a) \cap Q_k(b)|}{n}.$$

To apply the multiple criteria distance to the clustering analysis, the centroid of a cluster needs to be defined. The centroid of a cluster is defined as a fictitious alternative that is characterized by its profile. Let C_i be the ith cluster and b_i be the centroid of C_i. The profile $Q(b_i) = \langle Q_1(b_i), Q_2(b_i), Q_3(b_i), Q_4(b_i) \rangle$ will be defined as [10]

$$Q_k(b_i) = \{ a \in \mathbf{A} \mid \arg\max_{t \in \{1,2,3,4\}} \sum_{a' \in C_i} \chi_{\{a \in Q_t(a')\}} = k \}, \quad k = 1,2,3,4$$

where $\chi_{\{a \in Q_t(a')\}} = 1$ if the condition $a \in Q_t(a')$ is true and 0 otherwise.

4. The proposed approach

4.1. Problem description

Assume that a finite set of inventory items $\mathbf{A} = \{a_1, a_2, ..., a_n\}$ needs to be classified into the classes A, B, and C with A $\succ$ B $\succ$ C. The inventory items are evaluated with respect to a family of criteria $G = \{g_1, g_2, ..., g_m\}$. We employ the outranking relation of the ELECTRE III method as the preference model. The parameters of the outranking relation, including the criteria weights, indifference, preference, veto, and credibility thresholds, are assumed to be known a priori. The cardinality limitation of items in each class is specified in advance. The percentile interval of the cardinality limitation of items in class $s \in \{A,B,C\}$ is denoted by $[\sigma_s^{min-perc}, \sigma_s^{max-perc}]$, where $\sigma_s^{min-perc}$ and $\sigma_s^{max-perc}$ are the minimum and the maximum percentiles respectively. The set of items categorized into class s is denoted by T_s, and the number of items categorized into class s is denoted by n_s, $s \in \{A,B,C\}$.

The classification of items should fulfill the requirement of the following principle.

Definition 1. (Köksalan et al. [27]). Let $T(a)$ and $T(b)$ be the classification of items a and b, respectively, $T(a), T(b) \in \{A,B,C\}$. The classification is said to be consistent iff

$\forall a,b \in \mathbf{A}, \ aSb \Rightarrow T(a) \succeq T(b).$

This principle states that "if item a is as good as item b, a should be classified into a class at least as good as b". It is equivalent to

$\forall a,b \in \mathbf{A}, \ T(a) \prec T(b) \Rightarrow aS^cb.$

To evaluate the classification of all items, let us consider the following outranking degree between classes.

Definition 2. (Rocha and Dias [38]). The outranking degree of T_s over T_l, $s,l \in \{A,B,C\}$, represents the proportion of pairs of items $(a,b) \in T_s \times T_l$ such that a outranks b:

$$\theta_{sl} = \frac{\sum_{a \in T_s} \sum_{b \in T_l} \theta(a,b)}{n_s \times n_l}, \quad s,l \in \{A,B,C\},$$

with $\theta(a,b) = \begin{cases} 1 & \text{if } aSb \\ 0 & \text{if } aS^cb \end{cases}$

According to the classification principle, the outranking degree θ_{sl} of a worse class T_s over a better class T_l should be as small as possible. Therefore, we can define the following measure to evaluate the classification.

Definition 3. We quantify a classification solution Sol by an inconsistency measure which is defined as follows:

$\theta(Sol) = \theta_{BA} + \theta_{CB} + \theta_{CA}.$

It is obvious that the smaller $\theta(Sol)$ is, the better the classification solution Sol is.

To obtain the optimal classification solution, we can build the following optimization program:

$$\min \theta = \frac{\sum_{i-1}^{n}\sum_{j=1}^{n}\theta(a_i,a_j)\cdot x_B(a_i)\cdot x_A(a_j)}{\sum_{i=1}^{n}x_B(a_i)\cdot\sum_{j=1}^{n}x_A(a_j)} + \frac{\sum_{i-1}^{n}\sum_{j=1}^{n}\theta(a_i,a_j)\cdot x_C(a_i)\cdot x_B(a_j)}{\sum_{i=1}^{n}x_C(a_i)\cdot\sum_{j=1}^{n}x_B(a_j)} + \frac{\sum_{i-1}^{n}\sum_{j=1}^{n}\theta(a_i,a_j)\cdot x_C(a_i)\cdot x_A(a_j)}{\sum_{i=1}^{n}x_C(a_i)\cdot\sum_{j=1}^{n}x_A(a_j)}$$

$$\text{s.t. } n\cdot\sigma_s^{min\ -perc} \le \sum_{i=1}^{n} x_s(a_i) \le n\cdot\sigma_s^{max\ -perc}, \quad s=A,B,C,$$

$$x_s(a_i)\in\{0,1\},\quad i=1,2,\dots,n,\quad s=A,B,C,$$

where the binary variable $x_s(a_i)$ associated with item a_i indicates the classification of a_i such that $x_s(a_i)=1$ if a_i is categorized into the class s and 0 otherwise, $s\in\{A,B,C\}$.

We observe that the above model is a mixed integer nonlinear program, which is very difficult to solve using mathematical programming techniques, especially when the inventory size is large. Thus, we propose an alternative approach to solve the problem which is described as follows.

4.2. Outline of the proposed approach

The process of the proposed approach is depicted in Fig. 1, and its phases are described in detail below:

Step 1: Collect the data of the MCABC problem, including the set of inventory items **A**, the set of criteria G, and the items' performances on the criteria.

Step 2: Ask the DM from the organization to determine the following parameters: the percentile interval of the cardinality limitation $[\sigma_s^{min\ -perc},\sigma_s^{max\ -perc}]$, $s\in\{A,B,C\}$, the normalized weight vector of criteria $w=(w_1,w_2,\dots,w_m)$, the thresholds of criteria q_j,p_j, and v_j, $j=1,2,\dots,m$, and the credibility threshold λ. Refer to Roy [39] for more details about the setting of these parameters.

Step 3: Calculate the credibility index between items and establish the preference structure on **A**.

Step 4: Apply the agglomerative hierarchy clustering algorithm to group items into clusters and build up the hierarchy of clusters (see Section 4.3 for more details).

Step 5: Employ the simulated annealing algorithm to search for the optimal classification of the inventory items (see Section 4.4 for more details).

Step 6: The DM verifies the classification results. If the DM is satisfied with the recommendation, the process ends. Otherwise, the DM could revise the input data and the parameters and return to Step 1 to restart the classification process.

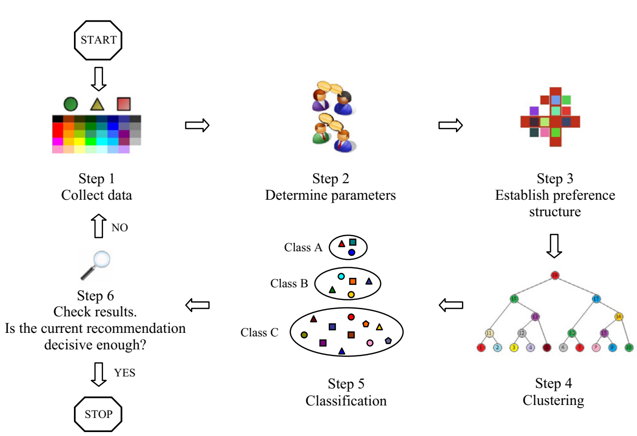

Fig. 1. General framework of the proposed approach.

In this process, we provide two evaluation indices for the classification of items, i.e., the inconsistency measure and the average distance between items in the solution (this is later defined in Section 4.4.1). We attempt to search for the optimal classification that minimizes the inconsistency measure and the average distance between items. Due to the large inventory size in practice and the relational nature of the outranking model, the search is a complex combinatorial optimization problem, and it is very time-consuming to solve the problem using mathematical programming techniques. Therefore, in the proposed approach, we combine the clustering analysis and the simulated annealing algorithm to search for the optimal solution. The clustering analysis groups similar items together and builds up the hierarchy of clusters. Then, the simulated annealing algorithm attempts to search for the optimal classification on different levels of the hierarchy. The algorithm starts with a randomly generated initial solution on a coarse-grained level. At a certain temperature of the search process, the algorithm generates a new solution from the neighborhood of the current solution by swapping the classification of two clusters of items. If the evaluation of the new solution is better than that of the current solution, the new one will replace the current one and the search process continues. A worse new solution may also be accepted as the new current solution with a small probability. As the temperature decreases, the current active clusters are split into more sub-clusters and the search will proceed on a fine level of granularity.

4.3. Clustering of the inventory items

In Step 4, similar items are grouped into clusters and the hierarchy of clusters is built by an agglomerative hierarchy clustering algorithm [2]. In the clustering algorithm, we apply the definition of the multiple criteria distance and the centroid of cluster introduced in Section 3. The clustering algorithm builds up a tree of clusters called a dendrogram. In the dendrogram, a parent cluster contains two sub-clusters and each fundamental cluster consists of one inventory item. The agglomerative hierarchy clustering algorithm allows exploring items on different levels of hierarchy. The general scheme of the agglomerative hierarchical algorithm is given in Algorithm 1.

Algorithm 1. The agglomerative hierarchical clustering algorithm

Inputs:
n initial clusters $C_i=\{a_i\}$,$i=1,2,\dots,n$, $a_i\in\mathbf{A}$;
The index of the new cluster to generate $iNewCluster=n+1$;
Begin
While There exist at least two clusters **do**
Determine the two clusters C_p and C_q such that the distance $dist(b_p,b_q)$ between the two centroids b_p and b_q is minimum;
Merge these two clusters C_p and C_q to form a new cluster $C_{iNewCluster}=C_p\cup C_q$;
Determine the centroid $b_{iNewCluster}$ of the new cluster $C_{iNewCluster}$;
$iNewCluster \leftarrow iNewCluster+1$;
End While
End
Output:
The number of the generated clusters $nCluster=iNewCluster-1$;
The hierarchical clustering scheme $\{C_1,C_2,\dots,C_{nCluster}\}$.

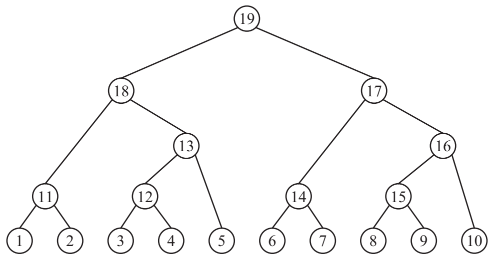

Fig. 2. An example of dendrogram.

Algorithm 1 starts with the initial clustering composed of n singleton-clusters, i.e., $C_i=\{a_i\}$, $i=1,2,\ldots,n$. Then, it iteratively generates a new cluster by choosing the nearest two clusters to merge until only one cluster exists.

Example 1. Let $n=10$ and $\mathbf{A}=\{a_1,a_2,\ldots,a_{10}\}$. Suppose that the hierarchy of clusters built by Algorithm 1 is depicted in Fig. 2.

In Fig. 2, the initial clustering is composed of 10 singleton-clusters, i.e., $C_i=\{a_i\}, i=1,2,\ldots,10$. The centroid b_i of each cluster C_i is a_i, $i=1,2,\ldots,10$. Then, the two clusters C_1 and C_2 with the minimum centroid distance $d(b_1,b_2)$ are merged to generate the new cluster $C_{11}=C_1\cup C_2$, and the centroid of the cluster C_{11} is generated by charactering its profile. Then, the new clusters $C_{12},C_{13},\ldots,C_{19}$ are generated iteratively. We observe that each fundamental cluster C_i, $i=1,2,\ldots,10$, contains only one item and each parent cluster C_i, $i=11,12,\ldots,19$, contains two sub-clusters.

4.4. Classification of the inventory items

On the basis of the hierarchical clustering scheme, we propose a simulated annealing algorithm to search for the optimal classification. The evaluation function, the initial solution, the neighborhood structure, and the parameters used in the proposed algorithm are discussed in the following subsections.

4.4.1. Evaluation function

We take the inconsistency measure $\theta(Sol)$ as an evaluation index for the classification solution Sol. Moreover, there might exist more than one classification solution corresponding to the minimum inconsistency measure θ^*. Therefore, the inconsistency measure $\theta(Sol)$ is not sufficient to quantify the classification solution. Let us consider another evaluation index for the classification solution.

Definition 4. The average distance between items in each class s is defined as follows:

$$AD(s)=\frac{\sum_{a,b\in T_s} dist(a,b)}{\binom{n_s}{2}},\quad s\in\{A,B,C\},$$

where $(n_s/2)$ is the 2-combination of the set of items in class s.

Definition 5. The weighted average distance between items of the classification solution Sol is defined as follows:

$$ACD(Sol)=\frac{\sum_{s\in\{A,B,C\}} n_s^{-1}\cdot AD(s)}{(n_A^{-1}+n_B^{-1}+n_C^{-1})}.$$

In this definition, we take n_s^{-1} as the weight of the average distance in class s because the importance of each class is heterogeneous. As we know, class A contains items of high value but small in number while class C contains items of low value but large in number. In practice, "A" items are very important and frequent value analysis of "A" items is required because of the high value of these items. In addition, an organization needs to choose an appropriate order pattern (e.g., "Just-in-time") for "A" items to avoid excess capacity. In contrast, "B" items are important and "C" items are marginally important, but of course less important than "A" items. Therefore, in order to better address the problem, $AD(s)$ in the metric $ACD(Sol)$ is weighted by the inverse of the number of items in each class.

Suppose that there are two classification solutions Sol and Sol' with $\theta(Sol)=\theta(Sol')$ and $ACD(Sol)<ACD(Sol')$. We argue that Sol is better than Sol' because Sol has better cohesion than Sol'.

To summarize, on the basis of Definition 3, the optimal classification solution should have the minimum inconsistency measure θ^*. In addition, according to Definition 5, when multiple classification solutions correspond to the minimum inconsistency measure θ^*, the one with the minimum weighted average distance ACD should be selected as the optimal classification solution.

4.4.2. Initial solution

On the basis of the hierarchy of clusters built by Algorithm 1, we can search for the initial classification by splitting a parent cluster into two sub-clusters and processing on different levels of hierarchy. The process is described in Algorithm 2. Note that the initial classification should fulfill the cardinality limitation of each class.

Algorithm 2. The algorithm to search for the initial classification solution

Input:
The set of items $\mathbf{A}=\{a_1,a_2,\ldots,a_n\}$;
The number of the generated cluster $nCluster$ (generated by Algorithm 1);
The hierarchical clustering scheme $\{C_1,C_2,\ldots,C_{nCluster}\}$ (generated by Algorithm 1);
The cardinality limitation $[\sigma_s^{min-perc},\sigma_s^{max-perc}]$, $s=A,B,C$;
The set of active clusters $\mathbb{C}_{active}=\{C_{nCluster}\}$;
Begin
The flag $isInitAssign\leftarrow false$;
The index of the current divisive cluster $iDivClu\leftarrow nCluster$;
While the flag $isInitAssign$ equals $false$ **do**
Split the cluster $C_{iDivClu}$ into two clusters $C_{iSubClu}$ and $C_{iSubClu}'$;
Remove $C_{iDivClu}$ from $\mathbb{C}_{active}$ and add $C_{iSubClu}$ and $C_{iSubClu}'$ to $\mathbb{C}_{active}$;
$iDivClu\leftarrow iDivClu-1$;
Check whether the current active clusters are possible to generate the initial classification which fulfills the cardinality limitation;
If there exists possible classification **do**
Generate the initial solution Sol;
$isInitAssign\leftarrow True$;
End
End While
End
Output:
The initial solution Sol;
The index of the current divisive cluster $iDivClu$;
The set of active clusters $\mathbb{C}_{active}$.

Algorithm 2 searches for the initial classification by splitting parent clusters in descending order of the cluster indices. Within each iteration of the loop, the current divisive parent cluster $C_{iDivClu}$ is split into two sub-clusters $C_{iSubClu}$ and $C_{iSubClu}'$, and then removed from the set of active clusters $\mathbb{C}_{active}$. Next, $C_{iSubClu}$ and $C_{iSubClu}'$ are added to $\mathbb{C}_{active}$. If the current active clusters are able to generate the classification that fulfills the cardinality limitation, formulate the initial classification and terminate Algorithm 2.

Table 1
List of the inventory items.

Inventory item	Criteria			
	Average unit cost (¥)	Annual RMB usage (¥)	Lead time (days)	Turnover (rate)
a_1	210.39	413,692.2	18	0.487
a_2	70.92	363,303.1	29	1.87
a_3	125.24	452,711.6	12	1.653
a_4	26.72	391,531.3	16	2.548
a_5	81.98	164,125.6	24	1.204
a_6	164.18	101,627	12	0.162
a_7	219.19	327,849.6	5	2.994
a_8	190.53	55,345.3	18	0.13
a_9	202.96	443,096.1	18	0.378
a_{10}	119.58	294,066.8	25	1.374
a_{11}	71.21	390,861.5	30	2.407
a_{12}	113.36	298,718.7	11	1.211
a_{13}	71.83	88,071.9	1	2.264
a_{14}	89.35	41,150.6	2	1.019
a_{15}	153.28	414,547.2	14	1.032
a_{16}	103.63	77,582	30	1.836
a_{17}	230.34	31,681.6	2	2.822
a_{18}	80.27	295,351.8	11	0.531
a_{19}	187.75	233,313.8	6	0.353
a_{20}	51.35	231,721.6	26	1.376
a_{21}	75.92	454,758.8	20	2.929
a_{22}	67.16	80,559.6	27	2.381
a_{23}	173.29	397,196.1	23	0.471
a_{24}	41.71	336,693	15	2.37
a_{25}	132.89	459,578.7	3	2.152
a_{26}	50.33	281,313.1	16	0.917
a_{27}	39.29	101,493.5	13	0.431
a_{28}	189.24	128,298.4	18	2.979
a_{29}	228.69	311,478.1	29	2.319
a_{30}	54.94	188,630.7	29	1.54
a_{31}	42.17	180,117	30	1.135
a_{32}	199.8	15,296.7	16	0.799
a_{33}	152.85	383,919.9	22	1.79
a_{34}	193.37	119,454.6	5	0.324
a_{35}	138.47	333,290.6	6	2.05
a_{36}	73.4	374,496.8	8	0.619
a_{37}	147.65	364,491.1	28	2.665
a_{38}	40.93	407,329.5	3	1.856
a_{39}	92.86	370,301.3	21	0.107
a_{40}	225.49	322,614	6	2.548
a_{41}	102.61	50,402.1	25	0.42
a_{42}	207.53	499,699.9	10	0.692
a_{43}	243.36	209,629.8	25	1.127
a_{44}	140.26	38,914.1	25	0.609
a_{45}	170.96	370,885.2	10	2.002
a_{46}	136.45	499,854.8	7	0.376
a_{47}	187.27	274,935.6	5	2.512
a_{48}	160.84	296,976.8	28	2.108
a_{49}	36.47	78,051.4	22	0.882
a_{50}	209.5	318,688.5	4	2.259
a_{51}	32.53	273,490.9	20	2.507
a_{52}	171.64	142,923	5	0.815
a_{53}	235.08	329,205.1	3	1.574
a_{54}	113.84	497,119.6	23	0.03
a_{55}	27.85	255,434	6	2.227
a_{56}	15.4	103,414	1	2.902
a_{57}	194.05	316,586	1	2.928
a_{58}	0.68	341,859.6	13	1.991
a_{59}	151.69	228,109	18	1.259
a_{60}	89.23	43,136.7	2	0.025
a_{61}	98.65	495,254.8	1	1.089
a_{62}	99.15	20,635.4	18	2.069
a_{63}	104.84	370,919.4	11	1.214

Otherwise, go to the next iteration of the loop until the initial solution is generated. Note that during the search process the items belonging to the same active cluster are classified into the same class.

In Algorithm 2 we need to check whether the current active clusters are possible to generate the initial classification which fulfills the cardinality limitation and this can be performed by solving the following linear program. Assume $\mathbb{C}_{active} = \{C_{r_1}, C_{r_2}, \dots, C_{r_K}\}$ and n_{r_t}, $t = 1, 2, \dots, K$, is the number of items belonging to cluster C_{r_t}. The cardinality limitation is $[\sigma_s^{min-perc}, \sigma_s^{max-perc}]$, $s = A, B, C$. Let us associate each cluster C_{r_t} with three binary variables $x_s(r_t)$, $s = A, B, C$, such that $x_s(r_t) = 1$ if items belonging to C_{r_t} are assigned to class s; otherwise, $x_s(r_t) = 0$. $\mathbb{C}_{active}$ is possible to generate the initial classification which fulfills the cardinality limitation if the maximum of the objective function of the following linear program is not less than zero, i.e., $f^* \geq 0$.

$$
\begin{aligned}
& f = \max \delta \\
& \text{s.t.} \ \sum_{t=1}^{K} n_{r_t} \cdot x_s(r_t) - \delta \geq n \cdot \sigma_s^{min-perc}, \quad s = A, B, C, \\
& \sum_{t=1}^{K} n_{r_t} \cdot x_s(r_t) + \delta \leq n \cdot \sigma_s^{max-perc}, \quad s = A, B, C, \\
& \sum_{s=A,B,C} x_s(r_t) = 1, \quad t = 1, 2, \dots, K, \\
& \sum_{t=1}^{K} x_s(r_t) \geq 1, \quad s = A, B, C, \\
& x_s(r_t) \in \{0, \ 1\}, \quad t = 1, 2, \dots, K, \quad s = A, B, C.
\end{aligned}
$$

For the above linear program, which is built on the level of clusters, the number of variables is small and it is not difficult to solve such a mixed integer linear program.

Example 2. Let us consider the hierarchy of clusters in *Example 1* and attempt to search for the initial classification solution. The cardinality limitation of items in each class is as follows: Class A should not contain more than 20% items; Class B should contain 20–50% items; Class C should contain at least 50% items.

Initially, the set of active clusters $\mathbb{C}_{active}$ contains only one cluster, i.e., C_{19}. We split the parent cluster C_{19} into two sub-clusters C_{17} and C_{18}. Then, the set of active clusters is updated and $\mathbb{C}_{active}$ becomes $\{C_{17}, C_{18}\}$. Obviously, there is no possible solution and thus we continue to the next iteration of the loop. We split the parent cluster C_{18} into two sub-clusters C_{11} and C_{13}. The set of active clusters is updated and $\mathbb{C}_{active}$ becomes $\{C_{11}, C_{13}, C_{17}\}$. C_{11} contains two items, C_{13} contains three items, and C_{17} contains five items, i.e., $n_{11} = 2$, $n_{13} = 3$ and $n_{17} = 5$. Thus the mixed integer linear program can be built as

$$
\begin{aligned}
& f = \max \delta \\
& \text{s.t.} \ 2x_A(11) + 3x_A(13) + 5x_A(17) - \delta \geq 10 \cdot 0\%, \\
& 2x_B(11) + 3x_B(13) + 5x_B(17) - \delta \geq 10 \cdot 20\%, \\
& 2x_C(11) + 3x_C(13) + 5x_C(17) - \delta \geq 10 \cdot 50\%, \\
& 2x_A(11) + 3x_A(13) + 5x_A(17) + \delta \leq 10 \cdot 20\%, \\
& 2x_B(11) + 3x_B(13) + 5x_B(17) + \delta \leq 10 \cdot 50\%, \\
& 2x_C(11) + 3x_C(13) + 5x_C(17) + \delta \leq 10 \cdot 100\%, \\
& x_A(11) + x_B(11) + x_C(11) = 1, \\
& x_A(13) + x_B(13) + x_C(13) = 1, \\
& x_A(17) + x_B(17) + x_C(17) = 1, \\
& x_A(11) + x_A(13) + x_A(17) \geq 1, \\
& x_B(11) + x_B(13) + x_B(17) \geq 1, \\
& x_C(11) + x_C(13) + x_C(17) \geq 1, \\
& x_A(11), x_A(13), x_A(17), x_B(11), x_B(13), x_B(17), \\
& x_C(11), x_C(13), x_C(17) \in \{0, 1\}.
\end{aligned}
$$

We solve the linear program and obtain the maximum of the objective function $f^* = 0$ and the solution $x_A(11) = x_B(13) = \ x_C(17) = 1$ and

$x_A(13)=x_A(17)=x_B(11)=x_B(17)=x_C(11)=x_C(13)\ =0$, which indicates that $\mathbb{C}_{active}$ is able to generate the initial classification, and the items belonging to C_{11}, C_{13}, and C_{17} are assigned to classes A–C, respectively. Therefore, the initial classification can be generated as $T_A=\{a_1,a_2\}$, $T_B=\{a_3,a_4,a_5\}$, $T_C=\ \{a_6,a_7,a_8,a_9,\ a_{10}\}$.

4.4.3. Neighborhood structure

We use the swap operator to generate the neighborhood structure of the current solution. Intuitively, swapping the classification of two clusters, between which the outranking degree is greater than the inconsistency measure θ of their corresponding classes, will improve the classification solution. Therefore, the swap operator randomly selects two clusters $C_i\subset T_s$ and $C_j\subset T_l$ from the set of current active clusters $\mathbb{C}_{active}$ such that $\theta_{C_i,C_j}>\theta_{sl}$. Moreover, because we only consider the outranking degree of a worse class over a better class, the selected two clusters C_i and C_j should fulfill one of the following conditions: (1) $C_i\subset T_B$ and $C_j\subset T_A$; (2) $C_i\subset T_C$ and $C_j\subset T_B$; (3) $C_i\subset T_C$ and $C_j\subset T_A$. Note that the swap operator should ensure that the neighborhood structure of the current solution fulfills the cardinality limitation of items in each class. Otherwise, the swap operator will continue to randomly select another two clusters until the neighborhood structure fulfills the cardinality limitation.

4.4.4. Setting of parameters

The proposed simulated annealing algorithm requires four parameters T_{init}, α, T_{final}, and *nInnerLoops*. T_{init} represents the initial temperature. α is a coefficient used to control the speed of the cooling schedule. T_{final} represents the final temperature. *nInnerLoop* is the number of iterations at which the search proceeds at a particular temperature.

With the decrease of temperature, the current divisive cluster will be split into two sub-clusters and the search will proceed on a fine level of granularity. We put the following condition on the initial temperature T_{init} and the coefficient used to control the temperature α:

$$\alpha\geq T_{init}^{-1/(nCluster-n)},$$

where *nCluster* is the number of clusters output by Algorithm 1. The above condition ensures that the search algorithm could arrive at the fundamental level of the hierarchy.

4.4.5. Simulated annealing procedure

At the beginning, the current temperature T is set to be T_{init}. The initial classification *Sol* is generated by Algorithm 2. The current optimal solution Sol_{best}, the optimal inconsistency measure θ_{best} and the weighted average distance between items ACD_{best} obtained so far are set to be *Sol*, $\theta(Sol)$ and $ACD(Sol)$, respectively.

At each iteration, a new solution *Sol*' is generated from the neighborhood of the current solution *Sol* and its $\theta(Sol')$ and $ACD(Sol')$ are calculated, respectively. Let $\Delta\leftarrow\theta(sol')-\theta(sol)$. If Δ is less than zero, *Sol* is replaced by *Sol*'. Otherwise, the probability of replacing *Sol* with *Sol*' is $e^{-\Delta/T}$. Sol_{best}, θ_{best} and ACD_{best} record the optimal solution obtained so far, as the algorithm progresses.

The current temperature T decreases after *nInnerLoops* iterations of inner loop, according to the formula $T=T\times\alpha$. After each temperature reduction, the local search procedure of the inner loop restarts to improve the optimal solution Sol_{best} obtained so far. The algorithm is terminated when the temperature T is less than the final temperature T_{final}. With the termination of the procedure, the optimal classification solution can be derived from Sol_{best}. The algorithm is described in Algorithm 3.

Table 2
The weights of the criteria and the thresholds.

	g_1	g_2	g_3	g_4
w_j	0.20	0.35	0.25	0.20
q_j	12.5	25,000	1.5	0.15
p_j	25.0	50,000	3.0	0.30
v_j	50.0	100,000	6.0	0.60
λ	0.60			

Table 3
The hierarchy of clusters.

Parent cluster	Sub-clusters	Number of items	Parent cluster	Sub-clusters	Number of items
C_{64}	C_7,C_{57}	2	C_{95}	C_{25},C_{92}	4
C_{65}	C_{16},C_{22}	2	C_{96}	C_{75},C_{95}	6
C_{66}	C_{12},C_{63}	2	C_{97}	C_{58},C_{85}	4
C_{67}	C_{19},C_{34}	2	C_{98}	C_{73},C_{82}	7
C_{68}	C_1,C_9	2	C_{99}	C_{81},C_{98}	9
C_{69}	C_2,C_{11}	2	C_{100}	C_{39},C_{99}	10
C_{70}	C_4,C_{51}	2	C_{101}	C_{96},C_{100}	16
C_{71}	C_{20},C_{30}	2	C_{102}	C_{28},C_{93}	6
C_{72}	C_{31},C_{71}	3	C_{103}	C_{43},C_{102}	7
C_{73}	C_5,C_{72}	4	C_{104}	C_{59},C_{103}	8
C_{74}	C_{40},C_{64}	3	C_{105}	C_{66},C_{76}	4
C_{75}	C_8,C_{32}	2	C_{106}	C_{97},C_{101}	20
C_{76}	C_{18},C_{36}	2	C_{107}	C_{56},C_{90}	4
C_{77}	C_{47},C_{50}	2	C_{108}	C_6,C_{79}	4
C_{78}	C_{53},C_{77}	3	C_{109}	C_{26},C_{105}	5
C_{79}	C_{52},C_{67}	3	C_{110}	C_{21},C_{69}	3
C_{80}	C_{23},C_{68}	3	C_{111}	C_{35},C_{45}	2
C_{81}	C_{41},C_{44}	2	C_{112}	C_{104},C_{106}	28
C_{82}	C_{62},C_{65}	3	C_{113}	C_{111},C_{112}	30
C_{83}	C_{74},C_{78}	6	C_{114}	C_{87},C_{113}	37
C_{84}	C_{15},C_{42}	2	C_{115}	C_{10},C_{110}	4
C_{85}	C_{24},C_{70}	3	C_{116}	C_{33},C_{115}	5
C_{86}	C_{13},C_{55}	2	C_{117}	C_3,C_{114}	38
C_{87}	C_{17},C_{83}	7	C_{118}	C_{116},C_{117}	43
C_{88}	C_{29},C_{48}	2	C_{119}	C_{108},C_{117}	47
C_{89}	C_{37},C_{88}	3	C_{120}	C_{27},C_{49}	2
C_{90}	C_{38},C_{86}	3	C_{121}	C_{108},C_{119}	51
C_{91}	C_{46},C_{54}	2	C_{122}	C_{109},C_{121}	56
C_{92}	C_{61},C_{91}	3	C_{123}	C_{120},C_{122}	58
C_{93}	C_{80},C_{84}	5	C_{124}	C_{89},C_{123}	61
C_{94}	C_{14},C_{60}	2	C_{125}	C_{94},C_{124}	63

Table 4
The classification results of the proposed approach.

Class	Inventory items
T_A	a_3,a_{21},a_{29},a_{33},a_{37},a_{43},a_{48}
T_B	a_1,a_2,a_7,a_8,a_9,a_{10},a_{11},a_{15},a_{17},a_{23}, a_{25},a_{28},a_{32},a_{39},a_{40},a_{42},a_{45},a_{46},a_{47},a_{50}, a_{53},a_{54},a_{57},a_{59},a_{61}
T_C	a_4,a_5,a_6,a_{12},a_{13},a_{14},a_{16},a_{18},a_{19},a_{20}, a_{22},a_{24},a_{26},a_{27},a_{30},a_{31},a_{34},a_{35},a_{36},a_{38}, a_{41},a_{44},a_{49},a_{51},a_{52},a_{55},a_{56},a_{58},a_{60},a_{62}, a_{63}

Algorithm 3. The simulated annealing algorithm to search for the optimal classification

Input:
The set of items $\mathbf{A}=\{a_1,a_2,...,a_n\}$;
The number of the generated cluster *nCluster* (generated by Algorithm 1);
The hierarchical clustering scheme $\{C_1,C_2,...,C_{nCluster}\}$ (generated by Algorithm 1);

The initial classification Sol (generated by Algorithm 2);
The set of active clusters $\mathbb{C}_{active}$ (generated by Algorithm 2);
The index of the divisive cluster $iDivClu$ (generated by Algorithm 2);
The initial temperature T_{init};
The decreasing rate of the temperature α;
The final temperature T_{final};
The number of iterations of inner loop $nInnerLoop$;

Begin
$T \leftarrow T_{init}$; $Sol_{best} \leftarrow Sol$; $\theta_{best} = \theta(sol)$; $ACD_{best} = ACD(Sol)$;
While $T > T_{final}$ **do**
$iIter \leftarrow 0$;
While $iIter < nInnerLoop$ **do**
Generate a new solution Sol' from the neighborhood of Sol;
$\Delta \leftarrow \theta(sol') - \theta(sol)$;
$prob \leftarrow \min(1, e^{-\Delta/T})$;
If $rand(0,1) \leq prob$ **do**
$Sol \leftarrow Sol'$
End If
If $(\theta(Sol') < \theta_{best})$ or $(\theta(Sol') = \theta_{best}$ and $ACD(Sol') < ACD_{best})$ **do**
$Sol_{best} \leftarrow Sol'$; $\theta_{best} = \theta(sol')$; $ACD_{best} = ACD(Sol')$;
End If
$iIter \leftarrow iIter + 1$;
End While
$T \leftarrow T \times \alpha$;
If $iDivClu > n$ **do**
Split the parent cluster $C_{iDivClu}$ into two sub-clusters $C_{iSubClu}$ and $C_{iSubClu}'$;
Remove $C_{iDivClu}$ from $\mathbb{C}_{active}$ and add $C_{iSubClu}$ and $C_{iSubClu}'$ to $\mathbb{C}_{active}$;
$iDivClu \leftarrow iDivClu - 1$;
End If
End While
End
Output:
The optimal classification Sol_{best}, θ_{best} and ACD_{best}.

5. Illustrative example

5.1. Problem description

In this section, a real-world MCABC problem is presented to illustrate the proposed approach. We consider the data set provided by a sports apparatus manufacturer in China. The data set consists of 63 inventory items used by the manufacturer and it is evaluated based on four criteria: (1) g_1: average unit cost (¥) ranging from 0.68 to 243.36; (2) g_2: annual RMB usage (¥) ranging from 15,296.7 to 499,854.8; (3) g_3: lead time (days) ranging from 1 to 30; and (4) g_4: turnover (rate) ranging from 0.025 to 2.994. Table 1 lists the 63 inventory items referred to as a_1–a_{63}.

The manufacturing manager specifies the cardinality limitation of items in each class as follows: class A should contain 10–20% items; class B should contain 20–50% items; class C should contain at least 50% items. The weights of the criteria and the thresholds are provided in Table 2.

5.2. Classification results

First, we calculate the outranking relation between every pair of items and build up the preference structure on the set of all items. On the basis of the preference structure, similar items are grouped into clusters and the hierarchy of clusters is built by Algorithm 1. Table 3 presents the hierarchy of these clusters. In Table 3, the 1st column lists the parent clusters. The 2nd column lists the two sub-clusters belonging to the corresponding parent cluster in the 1st column. The 3rd column lists the number of items belonging to the corresponding parent cluster in the 1st column. As the first 63 clusters $C_i = \{a_i\}$, $i = 1, 2, ..., 63$, are the fundamental clusters consisting of only one item, we do not present them in Table 3.

Next, Algorithms 2 and 3 are applied to classify the 63 inventory items into the classes A, B, and C based on the hierarchy of clusters. Concerning the parameter setting, the initial temperature is set to be $T_{init} = 100$, the decreasing rate of the temperature is set to be $\alpha = 0.93$, thus satisfying $\alpha \geq T_{init}^{-1/(nCluster-n)}$, the final temperature is set to be $T_{final} = 0.01$, and the number of iterations of inner loop is set to be $nInnerLoop = 1000$. Table 4 lists the obtained classification results.

It can be seen from Table 4 that seven items are classified into class A, 25 items are classified into class B, and 31 items are classified into class C, which fulfills the cardinality limitation of items in each class. Fig. 3 depicts the improvement of the inconsistency measure θ with the decrease of temperature. As seen from Fig. 3, θ decreases overall during the cooling procedure and we obtain an optimal solution $\theta_{best} = 0$ with $ACD_{best} = 0.4328$ when Algorithm 3 terminates. Although the swap operator may lead to a worse solution at a certain temperature, the search process can get back on track and escape from a local optimum.

5.3. Further analysis

5.3.1. Effect of the non-compensation between criteria

To examine the effect of the non-compensation between criteria, we make a comparison between the results when the compensation is considered or not. When the compensation between criteria is considered, we make the following modifications on the credibility index $\sigma(a,b)$: (1) the preference threshold p_j is set to be equal to the largest difference between items' performances on criterion g_j, $j = 1, 2, ..., m$; (2) partial discordance indices are not integrated into the credibility index, i.e., $\sigma(a,b) = C(a,b)$, and thus, no veto occurs. We apply the same procedure to classify these 63 items and the classification results are reported in Table 5.

One can observe that the classification of 18 inventory items differ when the compensation is considered, including a_2, a_3, a_8, a_{11}, a_{16}, a_{17}, a_{20}, a_{22}, a_{32}, a_{33}, a_{35}, a_{44}, a_{47}, a_{51}, a_{53}, a_{59}, a_{61}, and a_{63}. Note that the items a_{16}, a_{22}, and a_{44} do not score well on the key criterion g_2, but they are placed in class A when the compensation is considered. This is because their bad performances on g_2 are compensated by other criteria. However, when the compensation between criteria is limited, these items are classified into class C. This demonstrates the validity of the proposed approach when it is necessary to consider the non-compensation between criteria.

Moreover, Fig. 4 shows the number of outranking relations which exist between a particular item and others. The horizontal axis represents the items which have been grouped according to the classification results. The blue bar and the red bar count the number of outranking relations when the compensation is considered or not respectively. On one hand, we observe that when the compensation is considered, more outranking relations between items exist. On the other hand, the proposed approach classified the items which have more outranking relations into better classes overall. Thus, we may conclude that the classification results are coherent with the preference structure of the items.

5.3.2. Comparison among different methods

This section investigates how the proposed approach generates different classification in comparison with the aforementioned methods. Table 6 reports the classification results of Ramanathan

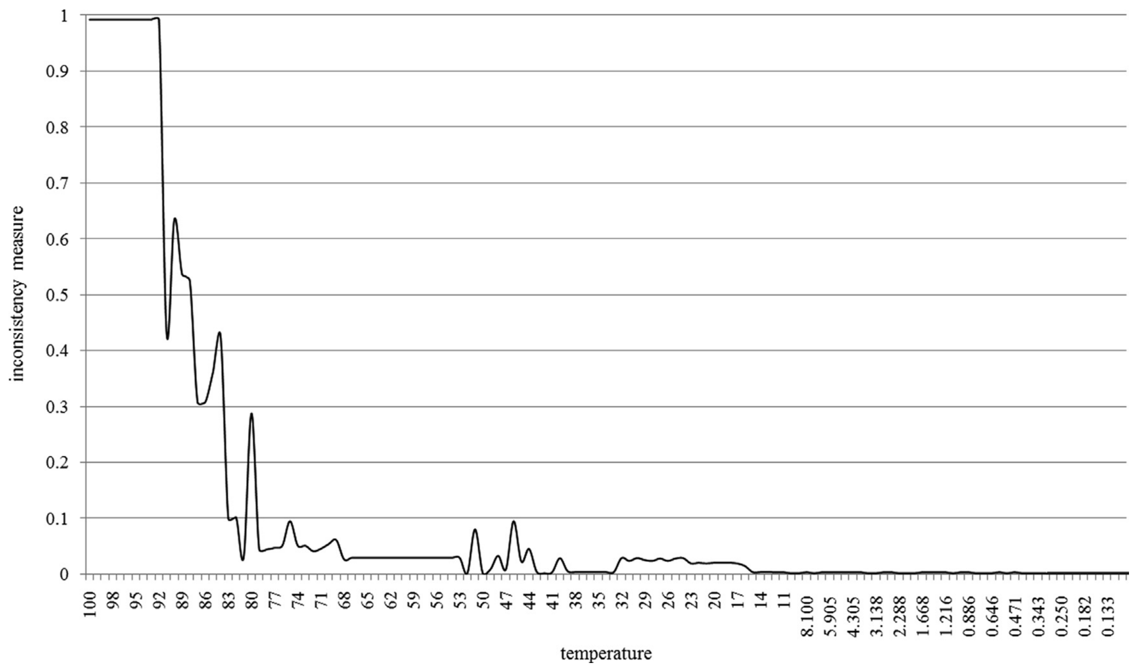

Fig. 3. The improvement of the inconsistency measure θ.

Fig. 4. The number of outranking relations. (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)

[36], Ng [32], Zhou and Fan [45], Hadi-Vencheh [22], and Lolli et al. [29]. In particular, Table 7 presents the numbers of items reclassified by the proposed approach (e.g., "A→B" indicates the number of items placed in class A by one of the methods and reclassified in class B by the proposed approach), as well as classified in the same classes (e.g., "A→A").

As we can observe from Tables 6 and 7, the classification obtained from Lolli et al. [29] is most similar to that of the proposed approach. However, we should mention that in some cases, the method of Lolli et al. [29] may lead to solutions that do not satisfy the cardinality limitation of items in each class. The reason for this lies in the *K*-means clustering procedure and the veto rules, incorporated by Lolli et al. [29], which put no restrictions on the number of items in each class. It is not reasonable to achieve solutions in which good classes contain more items than bad ones. This contradicts the principle of the ABC analysis.

The four DEA-like methods, i.e., Ramanathan [36], Ng [32], Zhou and Fan [45] and Hadi-Vencheh [22] differ substantially from the proposed approach. The numbers of variations between them and the proposed one are 22, 25, 20, and 17, respectively. As the weights of criteria in DEA-like methods are endogenously obtained by linear optimization models, it may amplify the effect of compensation between criteria. For example, the items a_{16} and a_{22} are poorly evaluated on criteria g_1 and g_2 but perform well on criterion g_3. The model of Ramanathan [36] places them in class A as their poor performances on criteria g_1 and g_2 are compensated by criterion g_3. In contrast, item a_{43} is classified into class C by the model of Ng [32] despite it performing moderately on all criteria. As Zhou and Fan [45] proposed an encompassing index that maintains a balance between the weights that are most favorable and least favorable for each item and Hadi-Vencheh [22] maintained the effects of weights in the final solution, the two models provide more reasonable classification than Ramanathan [36] and Ng [32]. The situations of "A→C" or "C→A"

Table 5
The classification results regarding the effect of non-compensation.

Inventory items	With non-compensation	With compensation
a_1	B	B
a_2	B	A
a_3	A	B
a_4	C	C
a_5	C	C
a_6	C	C
a_7	B	B
a_8	B	C
a_9	B	B
a_{10}	B	B
a_{11}	B	A
a_{12}	C	C
a_{13}	C	C
a_{14}	C	C
a_{15}	B	B
a_{16}	C	A
a_{17}	B	C
a_{18}	C	C
a_{19}	C	C
a_{20}	C	B
a_{21}	A	A
a_{22}	C	A
a_{23}	B	B
a_{24}	C	C
a_{25}	B	B
a_{26}	C	C
a_{27}	C	C
a_{28}	B	B
a_{29}	A	A
a_{30}	C	C
a_{31}	C	C
a_{32}	B	C
a_{33}	A	B
a_{34}	C	C
a_{35}	C	B
a_{36}	C	C
a_{37}	A	A
a_{38}	C	C
a_{39}	B	B
a_{40}	B	B
a_{41}	C	C
a_{42}	B	B
a_{43}	A	A
a_{44}	C	A
a_{45}	B	B
a_{46}	B	B
a_{47}	B	C
a_{48}	A	A
a_{49}	C	C
a_{50}	B	B
a_{51}	C	B
a_{52}	C	C
a_{53}	B	C
a_{54}	B	B
a_{55}	C	C
a_{56}	C	C
a_{57}	B	B
a_{58}	C	C
a_{59}	B	C
a_{60}	C	C
a_{61}	B	C
a_{62}	C	C
a_{63}	C	B

Table 6
The classification results obtained from other methods.

Inventory items	Ramanathan [36]	Ng [32]	Zhou and Fan [45]	Hadi-Vencheh [22]	Lolli et al. [29]	The proposed approach
a_1	B	B	A	A	B	B
a_2	B	B	B	B	B	B
a_3	B	A	B	A	A	A
a_4	C	B	C	C	C	C
a_5	C	C	C	C	C	C
a_6	C	C	C	C	C	C
a_7	A	B	A	A	B	B
a_8	C	C	C	B	C	B
a_9	A	B	B	B	B	B
a_{10}	C	C	B	B	C	B
a_{11}	A	B	B	B	B	B
a_{12}	C	B	C	C	C	C
a_{13}	C	C	C	C	C	C
a_{14}	C	C	C	C	C	C
a_{15}	C	B	B	B	B	B
a_{16}	A	C	B	B	C	C
a_{17}	A	C	C	C	B	B
a_{18}	C	C	C	C	C	C
a_{19}	C	C	B	B	C	C
a_{20}	C	C	C	C	C	C
a_{21}	A	A	B	B	A	A
a_{22}	A	C	B	C	C	C
a_{23}	B	B	B	B	B	B
a_{24}	C	B	C	C	C	C
a_{25}	B	A	B	B	B	B
a_{26}	C	C	C	C	C	C
a_{27}	C	C	C	C	C	C
a_{28}	B	C	B	B	B	B
a_{29}	B	B	A	B	A	A
a_{30}	B	C	B	C	B	C
a_{31}	B	C	B	B	C	C
a_{32}	B	C	C	C	B	B
a_{33}	B	B	B	B	A	A
a_{34}	C	C	B	B	C	C
a_{35}	C	B	B	B	C	C
a_{36}	C	B	C	C	C	C
a_{37}	B	B	A	B	A	A
a_{38}	C	B	C	C	C	C
a_{39}	C	B	C	C	B	B
a_{40}	B	B	A	B	B	B
a_{41}	C	C	C	C	C	C
a_{42}	B	A	B	B	B	B
a_{43}	B	C	A	A	A	A
a_{44}	C	C	C	C	C	C
a_{45}	B	B	B	B	B	B
a_{46}	B	A	B	B	B	B
a_{47}	C	C	B	B	B	B
a_{48}	B	B	A	A	A	A
a_{49}	C	C	C	C	C	C
a_{50}	B	B	B	B	B	B
a_{51}	B	C	C	C	C	C
a_{52}	C	C	C	C	C	C
a_{53}	B	B	B	B	B	B
a_{54}	B	A	B	B	B	B
a_{55}	C	C	C	C	C	C
a_{56}	B	C	C	C	C	C
a_{57}	B	B	C	B	B	B
a_{58}	C	B	C	B	C	C
a_{59}	C	C	C	C	C	B
a_{60}	C	C	C	C	C	C
a_{61}	B	A	C	C	A	B
a_{62}	C	C	C	C	C	C
a_{63}	C	B	C	C	C	C

do not occur in the classification of Zhou and Fan [45] and Hadi-Vencheh [22] in contrast to the proposed approach.

5.3.3. Sensitivity analysis

5.3.3.1. Sensitivity analysis of the parameters of the outranking model. In the proposed approach, the considered parameters of the outranking relation, including criteria weights, indifference, preference, veto, and credibility thresholds, are determined by the DM a priori. The choice of parameters may influence the classification results. We would like to investigate how sensitive the classification of items is to the modifications of these parameters. For each parameter, the following modifications are tested on the algorithm: $\pm 20\%$, $\pm 15\%$, $\pm 10\%$, $\pm 5\%$, $\pm 2\%$, and $\pm 1\%$. A ceteris paribus design is used for the experiments and only one parameter is modified each time. The algorithm runs 20 times for each parameter setting and the optimal classification is selected for comparison. Note that the modification range of the weight of one criterion is assigned to the other criteria proportionally in order to normalize the weights. Table 8 and Fig. 5 report the number of items that are classified into different classes with the modifications of the outranking model's parameters.

As shown in Table 8 and Fig. 5, the number of variations increases with the range of modification overall. At the level of $\pm 1\%$ and $\pm 2\%$, the number of variations are not more than five except for the modifications of w_2, w_3, and q_2. This indicates that the proposed approach can achieve robust classification despite the minor modifications of the parameters. At the level of $\pm 15\%$ and $\pm 20\%$, the numbers of variations for some modifications are more than 10, such as for the modifications of w_2, w_3, q_2, p_2, and v_2. This is because the preference structure varies significantly when such major parameter modifications are made.

It can also be seen from Table 8 and Fig. 5 that the modification of parameters on criteria g_2 and g_3 leads to major variations of the classification results. As these two criteria have the largest weights, the modifications of w_2 and w_3 change the preference structure more significantly than the other two criteria. Moreover, minor modifications in the indifference, preference and veto thresholds of these two criteria result in significant variation of the credibility index through multiplication of the corresponding criteria weight, which, in turn, changes the preference structure. Therefore, we suggest that the criteria weights in the proposed approach should be set carefully.

Table 7
The number of variations between the aforementioned methods and the proposed approach.

	A→A	B→B	C→C	Total same class	A→B	A→C	B→A	B→C	C→A	C→B	Total change
Ramanathan [36]	1	15	25	41	4	2	6	4	0	6	22
Ng [32]	2	13	23	38	5	0	4	8	1	7	25
Zhou and Fan [45]	4	15	24	43	3	0	3	7	0	7	20
Hadi-Vencheh [22]	3	18	25	46	2	0	4	6	0	5	17
Lolli et al. [29]	7	21	30	58	1	0	0	1	0	3	5

5.3.3.2. *Sensitivity analysis of the parameters of the simulated annealing algorithm.* In the simulated annealing algorithm, four parameters are required to be specified, including T_{init}, α, T_{final}, and *nInnerLoops*. We tested our approach with different settings of these parameters and then examined the variations of the classification results. We set up five levels for each parameter which are listed in Table 9. In each test, only one parameter is modified and the others remain the same. The algorithm runs 20 times on each modification, and the optimal classification is selected for comparison. Note that for each level of T_{init}, α is set to satisfy $\alpha \geq T_{init}^{-1/(nCluster-n)}$.

Table 10 reports the number of variations of the classification results. We observe that the modifications of T_{init}, α, and T_{final} result in few variations of the classification results. This is because in the simulated annealing algorithm, the parameters are set to ensure that the search for the optimal classification proceeds at each level of the hierarchy and thus the procedure is not sensitive to these three parameters. However, we also notice that when *nInnerLoops* is smaller than 1000, there are several variations of the classification results. We deem that a small *nInnerLoops* value does not guarantee finding the optimal classification at each level of the hierarchy. In this problem, the number of inventory items is 63 and *nInnerLoops* = 1000 seems to be sufficiently large to find the optimal classification. We need to properly set *nInnerLoops* for different inventory sizes in application of the approach.

5.3.4. Comparison between different clustering metrics

In the approach, we use the multiple criteria distance proposed by De Smet and Guzmán [10] as the clustering metric. To validate this clustering metric, we compare the classification results using different clustering metrics. The selected clustering metric for comparison is the weighted Euclidean distance which is defined as

$$dist(a,b) = \sqrt{\sum_{j=1}^{m}\left[w_j \cdot \left(\frac{g_j(a)-g_j(b)}{g_j^{*}-g_{j,*}}\right)^2\right]}$$

where g_j^{*} and $g_{j,*}$ are the maximum and minimum performances of items on criterion g_j, respectively, $j = 1, 2, ..., m$.

Table 11 lists the inconsistency measures θ of classification results using the two clustering metrics. We observe from Table 11 that the classification using the multiple criteria distance proposed by De Smet and Guzmán [10] is more consistent than the one using the weighted Euclidean distance. Moreover, if *nInnerLoops* is sufficiently

Table 8
The number of variations with the modification of the outranking model's parameters.

		−20%	−15%	−10%	−5%	−2%	−1%	+1%	+2%	+5%	+10%	+15%	+20%
Weight	w_1	2	9	1	5	4	3	2	3	7	7	7	5
	w_2	4	13	5	1	8	2	3	8	6	12	9	15
	w_3	11	7	1	4	4	2	1	7	5	10	6	12
	w_4	5	4	7	4	2	3	3	2	7	4	7	9
Indifference threshold	q_1	5	2	2	6	0	0	1	4	5	4	4	1
	q_2	10	1	1	6	1	3	3	6	2	7	8	14
	q_3	7	9	4	4	5	0	0	1	3	7	9	1
	q_4	4	5	3	4	3	1	1	0	6	2	1	3
Preference threshold	p_1	2	4	3	6	4	0	2	2	3	4	8	4
	p_2	14	5	8	7	4	3	3	2	2	5	3	1
	p_3	6	9	6	2	2	2	3	1	7	5	4	2
	p_4	4	4	1	4	0	0	2	2	2	4	3	7
Veto threshold	v_1	6	6	5	6	4	2	2	4	4	1	1	7
	v_2	7	12	3	6	5	1	3	3	1	6	3	2
	v_3	3	2	2	1	4	0	2	4	4	5	5	2
	v_4	4	7	6	2	0	2	2	2	2	6	1	1
Credibility threshold	λ	2	2	4	4	1	2	2	3	1	6	8	8

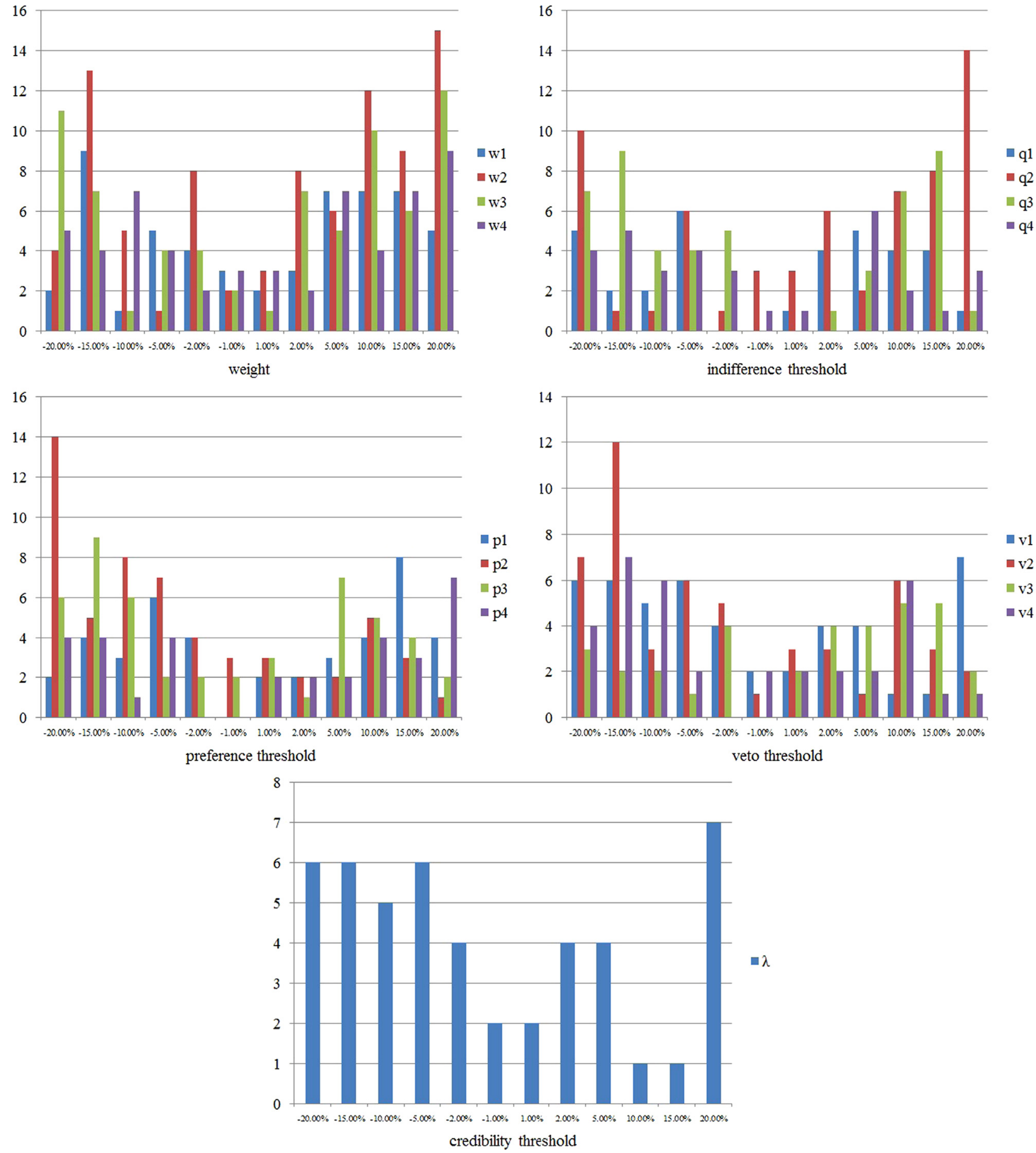

Fig. 5. The number of variations with the modification of the outranking model's parameters.

Table 9
The five levels for the parameters of the simulated annealing algorithm.

	Level 1	Level 2	Level 3	Level 4	Level 5
T_{init}	10	50	200	500	1000
α	0.94	0.95	0.96	0.97	0.98
T_{final}	0.001	0.005	0.02	0.05	0.10
nInnerLoop	100	500	2000	5000	10,000

Table 10
The number of variations with the modification of the parameters of the simulated annealing algorithm.

	Level 1	Level 2	Level 3	Level 4	Level 5
T_{init}	0	0	0	0	0
α	0	0	0	0	0
T_{final}	0	0	1	0	1
nInnerLoop	6	2	0	0	0

Table 11
The inconsistency measure using different clustering metrics.

	100	200	500	800	1000	2000
De Smet and Guzmán [10]	0.0365	0.0018	0	0	0	0
The weighted Euclidean distance	0.1093	0.0787	0.0431	0.0263	0.0018	0

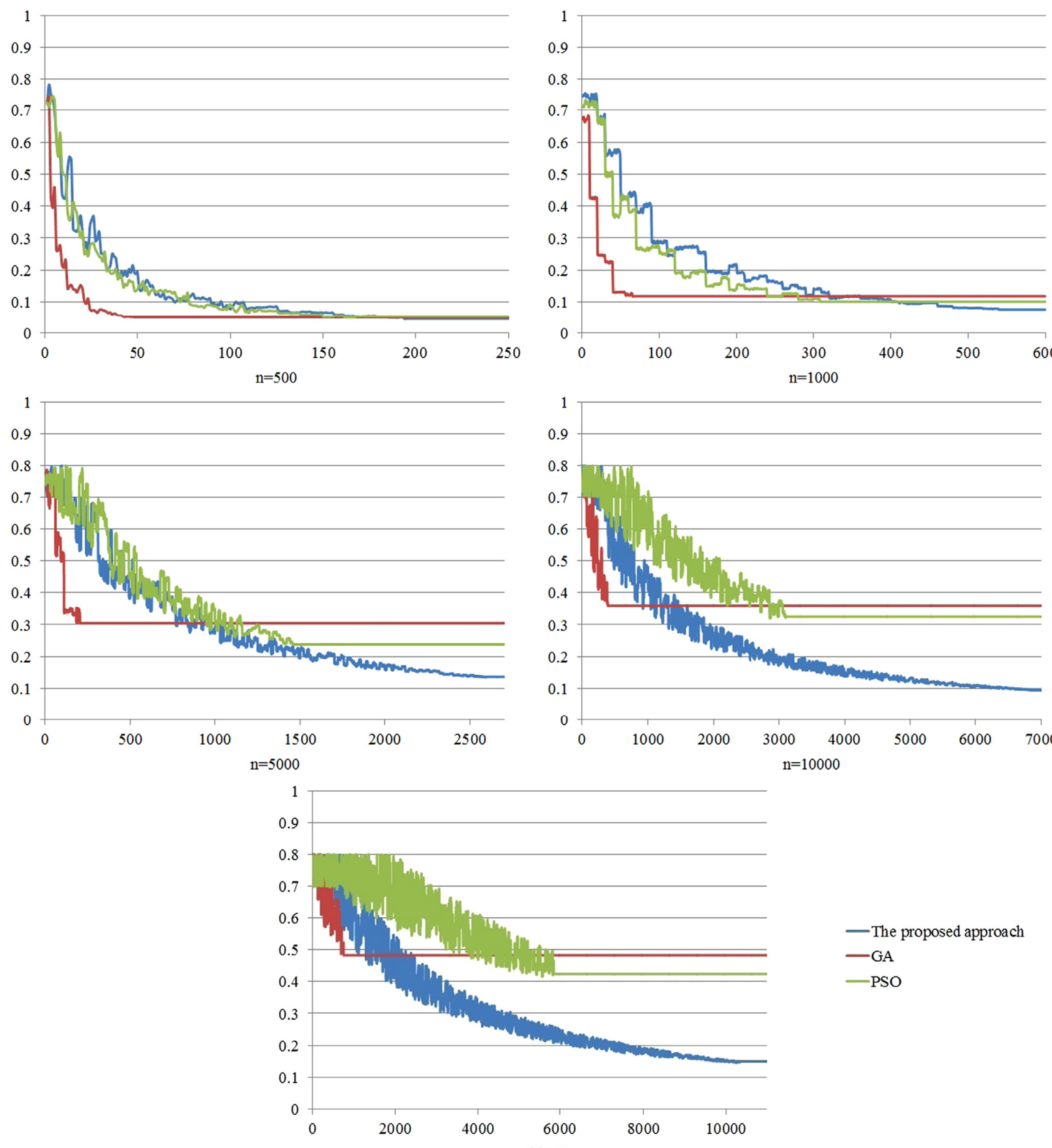

Fig. 6. The inconsistency measures θ of methods over time (s).

large, i.e., $nInnerLoops = 2000$, the clustering using the weighted Euclidean distance can also help to find the optimal classification. This is because when *nInnerLoops* is sufficiently large, the search for the optimal classification of clusters at a certain temperature is almost the enumeration of all possible solutions. Thus, the procedure can certainly find the optimal one. From another point of view, the multiple criteria distance proposed by De Smet and Guzmán [10] measures the similarity between items through the use of concepts

Table 12
The optimal inconsistency measures in different methods for datasets.

	500	1000	5000	10,000	20,000
The proposed approach	0.0455	0.0762	0.1365	0.0945	0.1498
GA	0.0501	0.1176	0.3045	0.3585	0.4819
PSO	0.0545	0.1008	0.2382	0.3248	0.4231

native to this field, namely the relations of indifference, preference, and incomparability. The clustering using the multiple criteria distance can organize items in a more consistent way, which will be useful to find the optimal classification in the following search procedure. In contrast, the clustering using the weighted Euclidean distance measures the similarity between items based on their geometric distance, and it does not describe the preference relation between items. The approach based on the clustering using the multiple criteria distance performs better in obtaining optimal classification than the one based on the clustering using the weighted Euclidean distance.

6. Experimental validation

This section studies the performance of the proposed approach for handling large-size datasets. We generate a series of datasets and conduct experiments to compare the classification of different methods.

6.1. Generating artificial datasets

We consider datasets containing 500, 1000, 5000, 10,000, and 20,000 items. Datasets of different sizes are generated to compare the performances of methods. The set of criteria in Section 5 is selected to evaluate items. The performances of items on criteria are generated artificially from the normal distribution with the following parameters:

Criterion g_1: $\mu=125$, $\sigma=30$,
Criterion g_2: $\mu=2.5\times10^5$, $\sigma=6\times10^4$,
Criterion g_3: $\mu=15$, $\sigma=4$,
Criterion g_4: $\mu=1.5$, $\sigma=0.4$.

The means of the normal distributions are set to be equal to the averages of the minimum and the maximum criteria performances in Section 5. The standard deviations are set to ensure that the probability density at zero is approximately zero. The criteria weights, indifference, preference, veto, credibility thresholds, and the cardinality limitation are the same as those specified in Section 5.

6.2. Methods considered for comparison

We compare the performance of the proposed approach with two popular heuristic algorithms – the genetic algorithm (GA) and the particle swarm optimization (PSO) [26]. The two methods are used to search for the optimal classification with respect to the inconsistency measure. The term "performance" in this section refers to the inconsistency measure.

The details about the implementation of GA are described as follows: (1) We adopt the 0–1 integer coding and the length of chromosome is $3\times n$. Every three consecutive genes represent the classification of the corresponding item; (2) the population size is 100; (3) the crossover probability is 0.9; and (4) the mutation probability is 0.05. PSO is used with the following parameters: (1) the size of the swarm consists of 100 particles; and (2) the parameters in the update equation for the velocity of the particle are set as $w=0.5$ and $c_1=c_2=2$. The above parameter settings are commonly encountered in the existing literature. Note that all of the methods are allowed to run for same amount of time to make the comparison fair.

6.3. Experiment results

The platform for conducting the experiment is a PC with a 2.4 GHz CPU and 4 GB RAM. All of the programs are coded in C language. The three methods run 20 times for each dataset and the classification with the minimum inconsistency measure is selected for comparison. Fig. 6 depicts the inconsistency measures of the methods over time and Table 12 reports the obtained optimal inconsistency measures.

One can observe that with the increase in inventory size, all of the methods require more computational time to obtain the optimal classification. For the same-sized problem, GA spends the least time while the proposed approach spends the most. When the size is quite large, i.e., $n=10,000$ and 20,000, the amount of computational time for the proposed approach is more than one hour. This is because for large-size datasets, the coefficient α in the approach is so large that the procedure requires more time to cool from a high temperature.

With respect to the optimal inconsistency measure, the one obtained by the proposed approach is significantly less than those from the other two methods, especially when the inventory size is large. Thus, the proposed approach performs better than the others in obtaining the optimal classification. It is obvious that GA and PSO may relapse into a local minimum for large-size datasets. GA and PSO also do not capture the relations between items, and therefore, their search abilities deteriorate when the inventory size is large. In contrast, the clustering in the proposed approach organizes items in a hierarchy that incorporates the preference relations between items. The proposed approach helps the following search procedure to find the optimal classification in a more efficient way.

7. Conclusions

In this paper, we present a classification approach based on the outranking relation for the multiple criteria ABC analysis. The non-compensation among criteria is considered in the approach through the construction of the outranking relation so that an inventory item scoring badly on one or more key criteria will not be placed in good classes no matter how well the item is evaluated on other criteria. This is an innovative feature in a literature where most the prevailing approaches addressing the MCABC problem do not consider the non-compensation in multiple criteria aggregation.

To obtain the optimal classification, we combine the clustering analysis and the simulated annealing algorithm to solve such a combinatorial optimization problem. The clustering analysis groups similar items together and permits the manipulation of items on different levels of granularity. The simulated annealing algorithm searches for the optimal solution according to the constructed hierarchy of clusters. It has been proven to obtain the optimal solution efficiently and it is suitable for handling large-size problems.

This paper also proposes two indices for quantifying the classification solution, i.e., the inconsistency measure and the average distance between items. The notion of the inconsistency measure is based on the consistency principle of classification and it indicates the proportion of how items of a worse class outrank items of a better class. The average distance between items refines the final classification when there are multiple solutions corresponding to the minimum inconsistency measure.

Our work provides an alternative avenue for solving multiple criteria classification problems. Most multiple criteria methods in the literature use mathematical programming techniques and thus are not suitable for large-size problems, especially when the outranking model is incorporated and the cardinality limitation should be considered. The idea of combining the clustering analysis and the simulated annealing procedure in our approach can be easily applied to general multiple criteria classification problems, not just the MCABC problem. It will help to obtain a satisfactory classification for a large-size problem in an efficient way.

An acknowledged limitation of the approach is that the DM is required to set the parameters of the outranking relation, including the criteria weights and thresholds. This involves a certain degree of subjectivity, and, as the sensitivity analysis shows, the preference structure changes with the variation of parameters, which in turn leads to different classification results. Thus, we suggest that the parameters of the outranking model should be set carefully. Moreover, in general, it is a difficult task for the DM to provide the preference information in such a direct way. One possible method of overcoming the difficulty is to adopt another preference elicitation technique known as indirect elicitation. For the indirect elicitation technique, the DM may provide assignment examples of some items which the DM is familiar with. Then, a compatible outranking relation model could be induced using the regression paradigm from these assignment examples. One can refer to the work of Kadziński et al. [24,25] for more details about the indirect elicitation technique for the outranking model.

Acknowledgment

The research is supported by the National Natural Science Foundation of China (#71071124 and #71132005). The authors are grateful to the anonymous reviewers for their constructive and detailed comments on the manuscript, which have helped improve the quality of the paper to its current standard.

References

[1] Behzadian M, Otaghsara SK, Yazdani M, Ignatius J. A state-of art survey of TOPSIS applications. Expert Systems with Applications 2012;39:13051–69.
[2] Berkhin P. A survey of clustering data mining techniques. Grouping Multidimensional Data 2006:25–71.
[3] Bhattacharya A, Sarkar B, Mukherjee SK. Distance-based consensus method for ABC analysis. International Journal of Production Research 2007;45(15):3405–20.
[4] Cakir O, Canbolat MS. A web-based decision support system for multi-criteria inventory classification using fuzzy AHP methodology. Experts Systems with Applications 2008;35(3):1367–78.
[5] Chen Y, Li KW, Kilgour DM, Hipel KW. A case-based distance model for multiple criteria ABC analysis. Computers & Operations Research 2008;35(3):776–96.
[6] Chen JX. Peer-estimation for multiple criteria ABC inventory classification. Computers & Operations Research 2011;38(12):1784–91.
[7] Chen JX. Multiple criteria ABC inventory classification using two virtual items. International Journal of Production Research 2012;50(6):1702–13.
[8] Coelli T, Prasada Rao DS, O'Donnell C, Battese GE. Data envelopment analysis. An introduction to efficiency and productivity analysis. Springer; 2005. p. 161–181.
[9] Cohen MA, Ernst R. Multi-item classification and generic inventory stock control policies. Production and Inventory Management Journal 1988;29(3):6–8.
[10] De Smet Y, Guzmán LM. Towards multicriteria clustering: an extension of the *k*-means algorithm. European Journal of Operational Research 2004;158(2):390–8.
[11] De Smet Y, Nemery P, Selvaraj R. An exact algorithm for the multicriteria ordered clustering problem. Omega 2012;40(6):861–9.
[12] Doumpos M, Zopounidis C. Multicriteria decision aid classification methods. Kluwer Academic Publishers; 2002.
[13] Ernst R, Cohen MA. Operations related groups (ORGs): a clustering procedure for production/inventory systems. Journal of Operations Management 1990;9(4):574–98.
[14] Fernandez E, Navarro J, Bernal S. Handling multicriteria preferences in cluster analysis. European Journal of Operational Research 2010;202(3):819–27.
[15] Figueira J, Mousseau V, Roy B. ELECTRE methods. In: Figueira J, Greco S, Ehrgott M, editors. Multiple criteria decision analysis: state of the art surveys. Berlin: Springer; 2005. p. 133–62.
[16] Figueira J, Greco S, Roy B, Słowińsiki R. An overview of ELECTRE methods and their recent extensions. Journal of Multi-Criteria Decision Analysis 2013;20(1-2):61–85.
[17] Flores BE, Whybark DC. Multiple criteria ABC analysis. International Journal of Operations and Production Management 1986;6(3):38–46.
[18] Flores BE, Whybark DC. Implementing multiple criteria ABC analysis. Journal of Operations Management 1987;7(1):79–84.
[19] Flores BE, Olson DL, Dorai VK. Management of multi-criteria inventory classification. Mathematical and Computer Modeling 1992;16(12):71–82.
[20] Güvenir HA. A genetic algorithm for multicriteria inventory classification. In: Pearso DW, Steele NC, Albrecht RF, editors. Proceedings of the international conference on artificial neural nets and genetic algorithms. Berlin: Springer; 1995. p. 6–9.
[21] Güvenir HA, Erel E. Multicriteria inventory classification using a genetic algorithm. European Journal of Operational Research 1998;105(1):29–37.
[22] Hadi-Vencheh A. An improvement to multiple criteria ABC inventory classification. European Journal of Operational Research 2010;201(3):962–5.
[23] Hadi-Vencheh A, Mohamadghasemi A. A fuzzy AHP-DEA approach for multiple criteria ABC inventory classification. Expert Systems with Applications 2011;38(4):3346–52.
[24] Kadziński M, Greco S, Słowińsiki R. Extreme ranking analysis in robust ordinal regression. Omega 2012;40(4):488–501.
[25] Kadziński M, Tervonen T, Figueira JR. Robust multi-criteria sorting with the outranking preference model and characteristic profiles. Omega 2015;55:126–40.
[26] Kennedy J, Eberhart RC. Particle swarm optimization. In: Proceedings – IEEE international conference on neural networks, vol. 4; 1995. p. 1942–8.
[27] Köksalan M, Mousseau V, Özpeynirci Ö, Özpeynirci SB. A new outranking-based approach for assigning alternatives to ordered classes. Naval Research Logistics 2009;56(1):74–85.
[28] Lei Q, Chen J, Zhou Q. Multiple criteria inventory classification based on principle components analysis and neural network. Advances in Neural Networks 2005;3498:1058–63.
[29] Lolli F, Ishizaka A, Gamberini R. New AHP-based approaches for multi-criteria inventory classification. International Journal of Production Economics 2014;156:62–74.
[30] Ma LC. A two-phase case-based distance approach for multiple-group classification problems. Computers and Industrial Engineering 2012;63(1):89–97.
[31] Meyer P, Olteanu A-L. Formalizing and solving the problem of clustering in MCDA. European Journal of Operational Research 2012;227(3):494–502.
[32] Ng WL. A simple classifier for multiple criteria ABC analysis. European Journal of Operational Research 2007;177(1):344–53.
[33] Partovi FY, Anandarajan M. Classifying inventory using an artificial neural network approach. Computers and Industrial Engineering 2002;41(4):389–404.
[34] Partovi FY, Burton J. Using the analytic hierarchy process for ABC analysis. International Journal of Production and Operations Management 1993;13(9):29–44.
[35] Partovi FY, Hopton WE. The analytic hierarchy process as applied to two types of inventory items. Production and Inventory Management Journal 1994;35(1):13–9.
[36] Ramanathan R. ABC inventory classification with multiple-criteria using weighted linear optimization. Computers & Operations Research 2006;33(3):695–700.
[37] Rezaei J, Dowlatshahi S. A rule-based multi-criteria approach to inventory classification. International Journal of Production Research 2010;48(23):7107–26.
[38] Rocha C, Dias LC. MPOC: an agglomerative algorithm for multicriteria partially ordered clustering. 4OR 2013;11(3):253–73.
[39] Roy B. The outranking approach and the foundations of ELECTRE methods. Theory and Decision 1991;31:49–73.
[40] Saaty T. The analytic hierarchy process. New York: McGraw-Hill; 1980.
[41] Soylu B, Akyol B. Multi-criteria inventory classification with reference items. Computers & Industrial Engineering 2014;69:12–20.
[42] Silver EA, Pyke DF, Peterson R. Inventory management and production planning and scheduling. New York: Wiley; 1998.
[43] Tsai CY, Yeh SW. A multiple objective particle swarm optimization approach for inventory classification. International Journal of Production Economics 2008;114(2):656–66.
[44] Torabi SA, Hatefi SM, Saleck Pay B. ABC inventory classification in the presence of both quantitative and qualitative criteria. Computers & Industrial Engineering 2012;63(3):530–7.
[45] Zhou P, Fan L. A note on multi-criteria ABC inventory classification using weighted linear optimization. European Journal of Operational Research 2007;182(3):1488–91.