Lexicographic constrained multicriteria ordered clustering

Jean Rosenfeld, Dimitri Van Assche, Yves De Smet

Université libre de Bruxelles, Ecole polytechnique de Bruxelles, 50 avenue Franklin
Roosevelt, 1050 Bruxelles, yves.de.smet@ulb.ac.be

Abstract. Researchers in the multicriteria analysis community have recently started to study the application of clustering techniques. Different methods have been proposed leading to complete or partial multicriteria partitions. Recently, a method based on the exploitation of the ordinal properties of a valued preference relation has been developed in order to obtain a totally ordered clustering. The underlying idea is to find a partition that minimizes the inconsistencies between the obtained ordered clusters and the input preference matrix. The exact algorithm gives the guarantee to achieve this goal based on a lexicographic order. In this contribution, we propose to further extend their approach to the case where minimum and maximum capacities are imposed on the cluster sizes. This leads to add a constraint satisfaction test (that is based on the levels and the ranks of the graph built during the procedure). The algorithm is evaluated regarding its execution time and illustrated on the academic ranking of world universities.

Keywords: Multiple Criteria Decision Aid, Multicriteria Clustering, Capacity Constraint.

1 Introduction

Over the last decades, multicriteria decision aid has been extensively used to manage real world problems. In this context, three main problem settings are usually considered: choice, ranking and sorting [18]. Let A denote a set of alternatives that are evaluated according to a set of criteria $F = \{f_1, f_2, \dots, f_q\}$. The *choice* problem consists in the selection of a subset $A' \subseteq A$. This subset should be as small as possible and be composed of the alternatives considered as the best ones. In a procurement context, the selection of the best offers (according to the price, time of delivery, quality of the supplier, etc.) is a typical example. The *ranking* problem consists in determining a complete or partial pre-order over the set of alternatives [16]. The ranking of universities (such as the Academic Ranking of World Universities - also commonly called *Shanghai* ranking according to academic performances), countries (regarding the human development or environmental performances) are common examples of this kind of problems. Sorting consists in assigning each alternative to a predefined ordered category. The set of alternatives is then divided in several groups. For example, patients

suffering from cancer can be sorted in four categories (defined by the World Health Organization (WHO)) depending on the disease severity. In this case, categories are supposed to be known a priori. Recently, a new kind of problem has been investigated by the multicriteria decision aid community; multicriteria clustering. Here the aim consists in detecting groups of alternatives as well as potential relations between these groups by explicitly taking into account the multicriteria nature of the problem.

Clustering is a well-known topic in data mining. Many methods have been developed (k -means, hierarchical approaches, etc.) [1, 10]. These generally rely on a distance measure in order to form groups as homogeneous as possible and that are as heterogeneous as possible between themselves. By definition the distance used in this context is a symmetric measure. Here relies the main difference with multicriteria decision aid where the comparison between alternatives leads most of the time to non symmetric (preference) relations. This observation was at the origin of the works on multicriteria clustering. This distinctive feature opens new perspectives such as, for instance, the existence of preference relations between the clusters. Finally, let us note that the main difference with sorting methods lies in the fact that categories are not supposed to be known beforehand. On the contrary, the objective is here to identify these groups (therefore, multicriteria clustering could be viewed as a pre-process to sorting methods).

To the best of our knowledge, De Smet and Guzmán [4] were the first who investigated the multicriteria clustering problem based on binary preferences. Their approach relies on the definition of a particular distance which includes the multicriteria preferential information of the alternatives comparison. The model was limited to the detection of nominal clusters. De Smet and Eppe [3] developed an extension of this first work to build relations between the clusters. De Smet and al. [5] also developed an exact algorithm to detect a totally ordered clustering based on the lexicographic optimization [6] of a penalty vector. More precisely they start by searching totally ordered partitions that minimize the worst penalty. Among these, they refine the search to those which minimize the second highest penalty and so on until an unique partition is obtained. Another method of multicriteria clustering has been proposed by Rocha et al [15]. It distinguishes first the clustering approach and then a multicriteria application to find relations between the clusters. Finally, Meyer and Olteanu [12] recently summarized and formalized this new topic.

In this contribution, we will focus on totally ordered clustering based on ordinal preferences existing between each pair of alternatives. We will consider the method developed by De Smet et al. [5] and propose an extension in order to manage capacity constraints on the clusters sizes. Let us point out that capacity constraints have already received some attention in sorting contexts [11].

Imposing constraint capacities on clusters sizes in a multicriteria context might have a number of applications. Let us consider the following illustrative examples:

- **Junior football player:** A famous international football team is proceeding to the recruitment of young players. After passing a series of standard

physical tests, 100 children between 5 and 7 years old have to be grouped in 5 classes based on their performances. The first class is assumed to represent the best players and so on. Each class is managed by two coaches. Therefore, its size should be less than or equal to 25 children (in order to properly manage the players) and should be at least equal to 11 (in order to be able to form a team in each group);

- **Disaster management:** After a major earthquake, 150 victims have to be urgently treated. Depending on the severity of their injuries, they will be sent to an hospital (with a capacity constraint of 60 people maximum), to mobile medical units (with a total capacity of 80 people). The victims who will not be treated directly will have to wait under the supervision of nurses. Within this last group, a ranking will have to be established in order to provide a priority of treatments when spaces become available.

Finally, let us note that the problem of limiting the sizes of the groups in traditional clustering has been developed since 1984 [13]. The problem is known as the capacitated clustering problem and one of its applications is the sales forces area distribution where the clients are distributed among clusters that should be similar but with a limit on the size in order to have a balance between the sales forces [8]. Often those problems can be solved by integer mathematical programming but at the cost of important computation times. This is the reason why heuristics have been commonly proposed to work on these problems. This is the case, for instance, in [14] where they use tabu search and simulated annealing, or in [19]. A modified version of the k -means is proposed in [7] where a constraint is added in the assignment step (in order to stop the assignment when the cluster size limit is reached).

In this contribution, we propose to develop an exact algorithm in the context of multicriteria clustering. The paper is organised as follow. First, we formalize the problem in section 2. Then, a dedicated algorithm is presented in section 3. This approach is illustrated in section 4. Finally, we conclude and give directions for future research.

2 The problem

Let $A = \{a_1, a_2, \dots, a_n\}$ be the set of alternatives and let $F = \{f_1, f_2, \dots, f_q\}$ be the set of criteria. In the following model, the number of clusters K and the preference matrix π are considered as inputs. The value π_{ij} represents the strict preference value of alternative a_i over alternative a_j . When there is no preference between a_i and a_j , we assume that $\pi_{ij} = 0$. In the following we assume that all preference values are different. If it is not the case, identical values can be differentiated by adding, for instance, an arbitrary small value ϵ .

An ordered partition of A into K clusters, noted $P_K(A) = [C_1, C_2, \dots, C_K]$ is defined as follows:

- $A = \bigcup_{i=1, \dots, K} C_i$,
- $\forall i \neq j : C_i \cap C_j = \emptyset$,

$$- C_1 \succ C_2 \succ \dots \succ C_K ,$$

where $C_i \succ C_{i+1}$ denotes that cluster C_i has a lower rank than cluster C_{i+1} (in other words, C_i is considered to be better than C_{i+1}).

The output of the method represents the cluster which each alternative belongs to. Two main assumptions can be made about this partition:

- Two alternatives belonging to the same cluster should be considered as being indifferent or similar. Therefore, their mutual preference values should be as low as possible;
- If an alternative a_i is assigned to a cluster which is better than the cluster of another alternative a_j , a_i is considered to be better than a_j . Therefore π_{ji} should be as low as possible.

In addition, we assume that clusters are subject to capacity constraints. More formally, we assume that:

$$x_h \leq |C_h| \leq X_h \quad (1)$$

where X_h and x_h represent, respectively, the maximum number and the minimum number of alternatives in the cluster C_h (to be consistent, we have $x_h \leq X_h$ and $\sum_h x_h \leq n \leq \sum_h X_h$).

The problem consists in determining the best partition among a number of possibilities (which increases very fast with the number of alternatives). To do so, one will try to minimize the inconsistencies with respect to the two previous conditions. Two types of inconsistencies have been distinguished:

- $a_i \in C_l$ and $a_j \in C_m$ with $l < m$ while $\pi_{ji} > 0$. Indeed, in an ideal case, we should have $\pi_{ij} > 0$ and $\pi_{ji} = 0$;
- $a_i, a_j \in C_l$ while $\pi_{ij} > 0$ or $\pi_{ji} > 0$. Indeed, in an ideal case, we should have $\pi_{ij} = \pi_{ji} = 0$.

Of course the severity of these inconsistencies will depend on the values of π_{ij} . There are different ways to minimize these inconsistencies. For instance, one could consider to minimize the sum of all inconsistencies. In this contribution, we follow the model proposed by De Smet et al. [5]. The idea is to base the optimization on the ordinal properties of the preference matrix. We will first consider all the K ordered partitions that minimize the worst inconsistency value. Among them, we will then identify those that minimize the second worst inconsistency. We repeat this step until a unique K ordered partition remains.

3 A dedicated algorithm

In this section, we present an extension of the algorithm proposed by De Smet et al. [5] in order to address capacity constraints. However, we decided to use different notations in order to simplify the presentation.

Let $I = \{1, 2, \dots, n\}$ be a set of indices and $H : \pi \rightarrow I \times I : \pi_{ij} \rightarrow (i, j)$. The H function allows to determine the ordered pair of elements corresponding

to a given preference value. Let ρ be the $n \cdot (n - 1)$ dimensional vector of the elements of π sorted in decreasing order (let us recall that all these elements are assumed to be different).

The proposed method consists of successively considering the values of ρ and verify if the corresponding elements can be put in clusters such that the first elements is in a better cluster than the second one (in other words to have an ordered partition that is as compatible as possible with the π matrix). Therefore, we will consider a binary matrix M . $M_{ij} = 1$ means that alternative a_i is put in a better cluster than alternative a_j . M is initially a null matrix. Obviously, we have $M_{H(\rho(1))} = 1$. Indeed, at the beginning, the elements corresponding to the highest preference value of the preference matrix should be placed in such a way that the induced K ordered partition is compatible with this value. The elements of M give the relative positions of elements. It will be built step by step. Of course it should respect constraints:

- M should be acyclic since we want to obtain a total order on the set of clusters;
- the longest path within the M matrix should not exceed length $(K - 1)$ since the number of clusters is limited to K ;
- capacity constraints have to be respected.

In other words, one will successively consider the elements of the ρ vector in a descending order. At iteration k , we will investigate if the pair of elements (a_i, a_j) given by $H(\rho(k)) = (i, j)$ are such that a_i can be put in a better cluster than a_j . This will be authorized if the three previous conditions are simultaneously satisfied. Let M^k be the binary matrix at iteration k . We have:

1. $M^0 = 0_{n \times n}$;
2. $M^1 = M^0$ and $M^1_{H(\rho(1))} = 1$;
- ...
3. $M^{k+1} = M^k$ and $M^{k+1}(H(\rho(k+1))) = 1$.

If M^{k+1} simultaneously satisfies the three conditions, the considered preference $\rho(k+1)$ is considered to be compatible with the partition that is currently built. If it is not the case, the preference value is added to the penalty vector and $M^{k+1}(H(\rho(k+1))) = 0$.

In order to manage these constraints, let us define the following notions. At first, since we consider a given iteration, we will omit the index k in the M matrix in order to simplify the notations. For a given $a_i \in A$, let us define $W(a_i) = \{a_j \in A | M_{ij} = 1\}$, the set of alternatives that are currently put in a worse cluster than a_i . Similarly, let us define $B(a_i) = \{a_j \in A | M_{ji} = 1\}$, the set of alternatives that are currently put in a better cluster than a_i . Let $w(0) = \{a_i \in A | W(a_i) = \emptyset\}$ and $w(k) = \{a_i \in A | W(a_i) \subset w(0) \cup w(1) \cup \dots \cup w(k-1)\}$. In other words, $w(0)$ denotes the subset of alternatives that have no *successors* in the M matrix. These alternatives can potentially be put in the worst cluster. It is easy to see that $a_i \in w(k)$ is equivalent to say that the longest path starting from a_i is of length k . For instance, let us consider a problem with $K = 5$. If $a_i \in w(3)$

it means there is a path of length 3 starting from a_i in the M matrix. As a consequence a_i could be placed in C_1 or C_2 . Otherwise the maximum number of clusters would not be satisfied. It is easy to see that when w can be determined for all the elements of A , the M matrix is acyclic. In a similar way, let us define $b(0) = \{a_i \in A \mid B(a_i) = \emptyset\}$ and $b(k) = \{a_i \in A \mid B(a_i) \subset b(0) \cup b(1) \cup \dots \cup b(k-1)\}$. The meaning of $b(k)$ is similar to $w(k)$ except that we now consider *predecessors* (instead of successors). For instance, if $a_i \in b(2)$ it could not be placed in C_1 or C_2 (since there exists a path of length 2 with a_i as the last node). Let us note that b and w are the so-called *ranks* and *levels* of the graph induced by the M adjacency matrix.

As already said, the M matrix will be built progressively. To summarize, when $a_i \in b(k)$ and $a_i \in w(l)$, this alternative could not be placed in clusters that are characterized by ranks greater than $K - l$ or lower than $k + 1$. All the clusters from C_{k+1} to C_{K-l} are possible assignments that are compatible with the desired number of clusters K . Finally, when $k + 1 = K - l$, the assignment of a_i is precisely determined. During the execution of the algorithm the possible assignments related to a_i will be progressively refined.

These notions help us verifying at each step that the M matrix is acyclic and respect the desired number of clusters. The partition of A with respect to b and w allows us to build a K ordered partition. Now we still have to take into account the constraints on capacities.

Let n_{kl} be the number of alternatives that could be placed in $C_k \cup \dots \cup C_l$ (with $k < l$) during the execution of the algorithm. For the sake of simplicity, let us consider the case $K = 3$. The conditions for maximum capacity constraints can be elicited as follows:

$$\left\{ \begin{array}{l} n_1 \leq X_1 \\ n_2 \leq X_2 \\ n_3 \leq X_3 \\ n_1 + n_2 + n_{12} \leq X_1 + X_2 \\ n_2 + n_3 + n_{23} \leq X_2 + X_3 \\ n_1 + n_2 + n_3 + n_{12} + n_{23} + n_{13} \leq X_1 + X_2 + X_3. \end{array} \right. \quad (2)$$

It is obvious that the simultaneous satisfaction of all these constraints leads to an incomplete partition that is compatible with the maximum capacities. Of course, some of the alternatives are not yet assigned to a precise cluster (especially during the first iterations) but it leaves room for a future assignment that will respect the maximum capacities.

The general condition can be written as follows: $\forall k \in I, \forall l \in I, k \leq l$:

$$\sum_{J \subseteq \{k, k+1, \dots, l\}, |J| \leq 2} n_J \leq \sum_{k \leq m \leq l} X_m. \quad (3)$$

In a similar way the conditions for minimum capacity constraints can be formulated as follows:

$$\begin{cases} n_1 + n_{12} + n_{13} \geq x_1 \\ n_2 + n_{12} + n_{23} + n_{13} \geq x_2 \\ n_3 + n_{23} + n_{13} \geq x_3 \end{cases} \quad (4)$$

More generally, we have: $\forall k \in I$

$$\sum_{J \subseteq I, |J| \leq 2, J \cap \{k\} \neq \emptyset} n_J \geq x_k. \quad (5)$$

At each step of the algorithm, these conditions will be verified in order to be sure that the capacity constraints could be satisfied in the future steps.

Algorithm 1 Determine K constrained ordered partition

Require: n, π, K and $(x_h, X_h) \forall h = 1, \dots, K$

Create ρ {a decreasing vector constituted by the elements of π }

Create $M^0 : 0_{n \times n}$

$M^1 = M^0$ and $M_{H(\rho(1))}^1 = 1$

$p \leftarrow \emptyset$ {a penalty vector}

$k = 1$

while $k \leq n \cdot (n - 1)/2$ **and** $\rho(k + 1) > 0$ **do**

$k \leftarrow k + 1$

$M^k = M^{k-1}$ and $M_{H(\rho(k))}^k = 1$

 Based on M^k , compute b and w

if M^k has a cycle **or** M^k has a path bigger than $K - 1$ **or** M^k does not respect constraints 3 and 5 **then**

$M_{H(\rho(k))}^k = 0$

$p(\text{end}+1) \leftarrow \rho(k)$

end if

end while

Algorithm 1 allows to obtain a K ordered partition that satisfies the capacity constraints. In addition, there is no other K ordered partition that lexicographically minimizes the penalty vector. More formally, let p be the penalty vector obtained at the end of Algorithm 1 and $\tilde{p}$ be the penalty vector associated to another K ordered partition. Let $T = \min\{|p|, |\tilde{p}|\}$, we have:

$$p \succ_{\mathcal{L}} \tilde{p} \Leftrightarrow p_i < \tilde{p}_i, i = \min[k \in \{1, 2, \dots, T\} | p_k \neq \tilde{p}_k] \quad (6)$$

Without loss of generality, we may assume the k first components of p and $\tilde{p}$ are the same (k might be equal to 0). By assumption, we have $p(k+1) > \tilde{p}(k+1)$. Following our approach, the value $p(k+1)$ was put in the penalty vector due to (at least) one of these conditions: M has a cycle, M has a path longer than $K - 1$ or M does not respect the capacity constraints.

Since the value $p(k+1)$ does not appear in the penalty vector of the new partition, we have $M_{H(\tilde{p}(k+1))} = 1$ and the induced graph is not a K ordered partition satisfying the capacity constraints.

Let us point out that sometimes Algorithm 1 can lead to imprecise assignments. This happens, for instance, when there are many zero values in the π matrix (leading to a situation where the alternatives cannot be properly discriminated). Finally, it is important to note that this approach is based on the assumption that preference values are different. As already stressed, this can easily be satisfied if identical values are discriminated by adding an arbitrary small value to these preferences. This step should not impact the penalty vector in a significant way (and therefore the partition structure). Nevertheless, we suggest to repeat different trials in order to assess the solution robustness.

4 Validation

Obviously, a minimal requirement is being able to replicate the results of the algorithm developed by De Smet et al. [5]. Let us consider the optimal solution for a given problem without constraint with $K = 3$. This solution assigns n_1 (respectively n_2, n_3) for the clusters 1 (respectively 2, 3). If we propose capacity constraints that are compatible with these values, the results given by the new approach should be the same. Many tests have been done with several datasets (and different parameters). Fortunately, results have always been the same.

In what follows, we will investigate computational time issues in subsection 4.1 and study qualitative results based on a real dataset in subsection 4.2.

4.1 Computation time

The proposed algorithm was tested on different artificial datasets for different numbers of alternatives and clusters. In these cases, all the π_{ij} values have been drawn from a $[0, 1]$ uniform random distribution (theses value being assumed to be stochastically independant). The performances are shown in Table 1.

Table 1. Execution time in seconds on artifical datasets.

# alternatives	3 clusters	4 clusters	5 clusters	6 clusters
100	2	2	2	2
300	71	70	87	62
500	381	382	327	311
700	900	952	1104	879

In addition, tests have also been conducted on real datasets:

- The Human Development Index (HDI) [9];
- The Academic Ranking of World Universities (ARWU) [17].

In Table 2, execution time in seconds is shown on randomly sampled subsets from the HDI dataset (those obtained on the ARWU dataset being similar).

Table 2. Execution time in seconds on HDI dataset.

# alternatives	3 clusters	4 clusters	5 clusters	6 clusters
10	0.01	0.01	0.01	0.02
50	0.6	1.14	1.35	1.80
100	6.9	14	19	22.4
150	31	62	87	103

For both tests in Table 1 and 2, the capacity constraints have been set as the same minimum and maximum capacities:

$$\min_capacity = \frac{\text{number_of_alternatives}}{(\text{number_of_clusters}) * 1.5} \quad (7)$$

$$\max_capacity = \frac{\text{number_of_alternatives} * 1.5}{(\text{number_of_clusters})} \quad (8)$$

4.2 Qualitative results

In this section, we show the results obtained while applying the algorithm on a real dataset: the Academic Ranking of World Universities (ARWU). In order to compute the input preference matrix we decided, without loss of generality, to use the linear PROMETHEE preference function [2]. For the sake of simplicity, the indifference and the preference thresholds correspond respectively to the second and the eighth decile of the differences of evaluations for each criterion. Here we consider the 100 first institutions of the Shanghai ranking. Weight values are the same as in the initial ranking. When the model is tested without constraints for $K = 4$, we get the results represented on Figure 1. It is represented in the PCA two-dimensional space. This ordered partition puts Harvard, the first university in the ranking, alone in a single cluster. This shows how Harvard is better than the other alternatives (which is coherent with the very high evaluations of this university on all the criteria). This partition assigns 1 alternative to C_1 , 9 alternatives to C_2 , 25 to C_3 and 65 to C_4 .

The obtained partition is clearly unbalanced (a single alternative in the first cluster and 65 in the worst one). In order to better balance this output, one would like to investigate a partition that respects the following conditions; at least 5 universities in the best cluster and at most 60 in the last cluster. The results are presented on Figure 2. Its distribution is as follows: 5 alternatives in C_1 , 7 alternatives in C_2 , 28 in C_3 and 60 in C_4 .

The confusion matrix F is defined as follows:

$$F = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 4 & 5 & 0 & 0 \\ 0 & 2 & 23 & 0 \\ 0 & 0 & 5 & 60 \end{pmatrix}$$

Fig.1. Model with no size constraints applied to the 100 first institutions of the Shanghai Ranking of World Universities, $K=4$ - PCA representation. We have $C_1(\square)$, $C_2(\triangleleft)$, $C_3(\circ)$ and $C_4(\triangleright)$.

where $F(i, j)$ denotes the number of elements belonging in C_i in the unconstrained version of the method that are placed in C_j when constraints are taken into account. As expected, these new constraints imply a reduction of the quality of the clustering in terms of lexicographic optimization. Figure 3 shows the evolution of the 100 first terms of the inconsistency matrix. One can see the higher lexicographic values of the model without constraints. In fact, the first term of the inconsistency matrix is equal to 0.57 (respectively 0.83) for the unconstrained (respectively constrained) model.

Fig.2. Model $x_1 = 5$ and $X_4 = 60$ applied to the 100 first institutions of the Shanghai Ranking of World Universities, $K=4$ - PCA representation. We have $C_1(\square)$, $C_2(\triangleleft)$, $C_3(\circ)$ and $C_4(\triangleright)$.

Fig. 3. Comparison of the inconsistency values for the models with and without constraints - Shanghai ranking

5 Conclusion

In this contribution, we have proposed an exact method to address capacity constraints in the context of multicriteria ordered clustering. The distinctive feature of this approach is to minimize a vector of inconsistencies based on a lexicographic order following the work of De Smet et al. [5]. The execution time issue of the proposed method has been studied and results have been illustrated on a real data set. Multicriteria clustering is a recent research topic. If some works have tried to formalize it, there are still a number of questions to be addressed. To the best of our knowledge this work is the first to investigate capacity constraints in this specific context (while similar questions have already been raised in multicriteria sorting methods or in *traditional* clustering). A common question in clustering is to properly determine the number of clusters to create. Some indicators might help the analyst answering this question. Here the question is even more complex since the input information is much richer; minimum and maximum constraints may be provided. To our point of view, these two kinds of parameters are related and need further investigations. The proposed solution heavily relies on the fact that we only use the ordinal properties of the preference matrix (by using a lexicographic order). Of course, other objectives can be used (such as minimizing the sum of inconsistent preferences, etc.). In these cases, managing capacity constraints would become more difficult (and the solution based on the computation of the levels and the ranks of a graph would not be appropriate anymore).

References

1. Paul S. Bradley and Usama M. Fayyad. Refining Initial Points for K-Means Clustering. In *Proceedings of the Fifteenth International Conference on Machine Learning*, ICML '98, pages 91–99, San Francisco, CA, USA, 1998. Morgan Kaufmann Publishers Inc.

2. Jean-Pierre Brans and Yves De Smet. PROMETHEE methods. In *Multicriteria Decision Analysis: State of the Art Surveys*, International Series in Operations Research and Management Science, pages 187–219. Springer-Verlag, 2016.
3. Yves De Smet and Stefan Eppe. Relational multicriteria clustering: The case of binary outranking matrices. In M. et al. Ehrgott, editor, *Evolutionary Multi-Criterion Optimization. Fifth international conference, EMO 2009. Proceedings*, volume 5467 of *Lecture Notes in Computer Science*, pages 380–392. Springer Berlin, 2009.
4. Yves De Smet and Linett Montano Guzmàn. Towards multicriteria clustering: An extension of the k-means algorithm. *European Journal of Operational Research*, 158(2):390 – 398, 2004. Methodological Foundations of Multi-Criteria Decision Making.
5. Yves De Smet, Philippe Nemery, and Ramkumar Selvaraj. An exact algorithm for the multicriteria ordered clustering problem. *Omega*, 40(6):861 – 869, 2012. Special Issue on Forecasting in Management Science.
6. Matthias Ehrgott. *Multicriteria Optimization*. Springer-Verlag, Berlin, Heidelberg, 2005.
7. N. Ganganath, C. Cheng, and C. K. Tse. Data clustering with cluster size constraints using a modified k-means algorithm. In *2014 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery*, pages 158–161, 2014.
8. Frank Höppner and Frank Klawonn. Clustering with size constraints. In *Computational Intelligence Paradigms*, pages 167–180. Springer, 2008.
9. Human Development Index. <http://hdr.undp.org/en/content/human-development-index-hdi>, 2015.
10. Tapas Kanungo, David M. Mount, Nathan S. Netanyahu, Christine D. Piatko, Ruth Silverman, Angela Y. Wu, Senior Member, and Senior Member. An efficient k-means clustering algorithm: Analysis and implementation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24:881–892, 2002.
11. Murat Köksalan, Vincent Mousseau, and Selin Özpeynirci. Multi-criteria sorting with category size restrictions. *International Journal of Information Technology & Decision Making*, 16(01):5–23, 2017.
12. Patrick Meyer and Alexandru-Liviu Olteanu. Formalizing and solving the problem of clustering in {MCDA}. *European Journal of Operational Research*, 227(3):494 – 502, 2013.
13. John M Mulvey and Michael P Beck. Solving capacitated clustering problems. *European Journal of Operational Research*, 18(3):339–348, 1984.
14. Ibrahim H Osman and Nicos Christofides. Capacitated clustering problems by hybrid simulated annealing and tabu search. *International Transactions in Operational Research*, 1(3):317–336, 1994.
15. Clara Rocha, Luis C. Dias, and Isabel Dimas. Multicriteria classification with unknown categories: A clustering–sorting approach and an application to conflict management. *Journal of Multi-Criteria Decision Analysis*, 20(1-2):13–27, 2013.
16. Bernard Roy. The outranking approach and the foundations of Electre methods. In Carlos A. Bana e Costa, editor, *Readings in Multiple Criteria Decision Aid*, pages 155–183, Berlin, Heidelberg, 1990. Springer Berlin Heidelberg.
17. ShanghaiRanking Consultancy. <http://www.shanghairanking.com>, 2015.
18. Philippe Vincke. *Multicriteria Decision-Aid*. J. Wiley, New York, 1992.
19. Shunzhi Zhu, Dingding Wang, and Tao Li. Data clustering with size constraints. *Knowledge-Based Systems*, 23(8):883–889, 2010.