Journal of Informatics and Mathematical Sciences
Volume 2, Number 1 (2010), pp. 51–62

http://www.rgnpublications.com

A Method for Selecting Pareto Optimal Solutions in Multiobjective Optimization

Mohamed Cheikh, Bassem Jarboui, Ta cir Loukil, and Patrick Siarry

Abstract. In many real-life multiobjective optimization problems and particularly in combinatorial ones, a common methodology is to determine a Pareto optimal set. These sets can be extremely large or may contain an infinite number of solutions. It is then difficult to choose among these solutions, one that corresponds to the best compromise according to the decision-maker's preferences. In this paper, we propose a model to select a restricted set of solutions. These sets should be structurally different and the most representative of the Pareto-optimal solutions. Our model is implemented using hierarchical algorithms and variable neighborhood search metaheuristics.

1. Introduction

Operations research is mainly made up starting from models which postulate the existence of a single objective function [1]. Although single objective decision models are sufficient for some decision making processes, there are many situations where decisions have multiple objectives [2]. In these situations, the aim is to simultaneously optimize a group of conflicting objectives.

As opposed to single objective optimization problems, in multiobjective optimization problems, there is no single optimal solution, but rather a set of alternative solutions. The approaches to find the best feasible solution to be implemented can be quite challenging for the decision-maker. In this kind of problems the analyst either determines a single solution or identifies a set of nondominated solutions, often referred to as Pareto-optimal set.

Determination of a single solution for multiobjective problems is performed using methods such as the weighted sum method, utility theory, goal programming, etc [3]. The other general approach is the determination of a set of nondominated solutions, i.e., a Pareto optimal set. Pareto optimal sets are often preferred to single solutions because they can be practical when considering real-life problems, since

Key words and phrases. Partitional clustering; Hierarchical algorithms; Variable neighborhood search; Multicriteria analysis.

the final solution of the decision maker is almost always a trade-off between crucial parameters.

In many real-life multiobjective optimization problems and particularly in combinatorial ones, the Pareto-optimal set can contain an infinite number of solutions or can be extremely large. It is then difficult to choose among these solutions, one that corresponds to the best compromise according to the decision-maker's preferences.

The selection of one solution may be quite a challenging problem since, in the absence of subjective or judgmental information, none of the corresponding trade-offs can be said to be better than the others. This is particularly true if we consider that the size of the Pareto-optimal set increases proportionally to the number of objectives.

This discussion makes clear that there is a need to achieve a set of promising solutions with a smaller cardinality. Thus, the motivation for the current work stems from challenges encountered during the post-Pareto analysis phase. To fully make the Pareto-optimal set easier to consider for the decision-maker, we propose to select k representative solutions of the Pareto optimal frontier.

Some methods have been proposed to select a representative solution in Pareto optimal frontier. Taboada *et al.* [4] proposed two methods, the first method is a pseudo-ranking scheme that helps a decision maker to select a solution that reflects his objectives functions priorities. They generate a set of random ranked weights, these weight sets are used to repeatedly combine the scaled objectives into a single objective function. The best solution for each combined objective is recorded from the set of the Pareto optimal solutions available. This is repeated with the next set of weights and the best solution for that combination is identified. This process is repeated many times, and the end results is a "prunned" Pareto optimal set.

In the second approach, they used data mining clustering techniques to group the data by using the k-means algorithm to find clusters of similar solutions. One way for the decision maker to pick one solution, among the solutions contained in a cluster, is to identify what solution is closest to its centroid.

Malakooti *et al.* [5] proposed the selection in two steps. Firstly, they used unsupervised learning clustering artificial neural network with variable weights for clustering of alternatives. In the second step they used feedforward artificial neural network for the selection of the best alternatives for each cluster of alternatives. Malakooti *et al.* [6] developed theories and procedures for clustering discrete multiple criteria alternatives, and the selection of the best alternative for each clustered group can be performed using existing methods.

In this work, we propose an approach to select a restricted set of solutions. These solutions will have to be structurally different and are the most representative of the whole set of the Pareto-optimal solutions. Our approach is based on the clustering methods [7]. These methods consist in discovering, from

a population, clusters or groups of individuals presenting common features. The number and the definition of clusters are not necessarily given a priori. The set of solutions will be partitioned in clusters that present some similar features. Then, the representative solutions of every cluster in this partition will be extracted.

The advantage of these approaches is to decrease the number of criteria (one or more criteria in the same group are equal or very close) and facilitate the decision making by the decision-maker, who will be able to understand his problem and possibly to choose, more easily, the solution, that interests him, among these representative solutions of the multicriteria problem at hand.

The other advantages of our proposition is that the clustering and the selection are performed in one phase. For the competing approaches, two phases are required.

The organization of this paper is as follows: we present in section 2, a modelization of the problem based on the minimization of the distances to ideal points. In sections 3, 4 and 5 we describe the steps of the algorithm that we propose to solve this problem, then we report experimental results in section 6. We conclude and propose some ways for future research in section 7.

2. **Modelization**

Let us consider a set of Pareto optimal solutions; our aim is to find a set of k representative points. We suppose that we have a set of N solutions $a_i \in A$, where A represents a Pareto frontier $A = \{a_1, a_2, \ldots, a_N\}$. Each solution a_i is characterized by q distinct attributes $(a_i^1, \ldots, a_i^q)$. Without loss of generality, we assume that all these attributes have to be minimized.

We say that b_i dominates b_j, denoted by $b_i \succ b_j$ if $b_i^t \leq b_j^t \ \forall \ t = 1, \ldots q$ and $\exists \ l \in \{1, \ldots, q\}$ such that $b_i^l < b_j^l$.

The ideal point of a subset of points is the virtual point that has a minimal evaluation for each attribute. Let us consider A' a subset of $A : A' \subseteq A$, $A' = \{a_1, \ldots, a_r\}$, the ideal point of the subset A' is:

$$\mathrm{id}(A') = \min_{i \in \{1,\ldots,r\}} a_i^t, \quad \forall \ t \in \{1, \ldots, q\}.$$

Our objective is to find k couples (a_{i_j}, b_j) $(j = 1, \ldots, k)$ for which b_j is a virtual point that dominates a_{i_j}, where a_{i_j} is the best solution among all the solutions dominated by b_j, and $a_i \in A$, there exists at least one b_j that dominates a_i.

We are going to find k couples that minimize a sum of Euclidean distances relative to every couple (a_{i_j}, b_j), and the set of points a_{i_j} are the representative points in Pareto optimal frontier:

$$\min \sum_{j=1}^{k} d^2(b_j, a_{i_j})$$
$$\forall \ a_i \in A \ \exists \ b_j (j = 1, \ldots, k) : b_j \succ a_i \, .$$

The point b_k represents the ideal point of the subset that it dominates. $\forall \ b_k$ there exists a subset C such that the ideal point of C is equal to b_k, $\mathrm{id}(C) = b_k$.

Proof. Let $\widetilde{C}$ be a set of points which are dominated by b_k :

If: $\tilde{C} \neq \emptyset$.

$$\begin{aligned} & \forall\ a_i \in \tilde{C} : b_k \succ a_i \\ \Rightarrow \quad & \forall\ t = 1, \ldots, q, \quad b_k^t = \min_{a_i \in \tilde{C}} a_i^t \\ \Rightarrow \quad & \forall\ l \in \{1, \ldots, q\}, \quad b_k^l \leq \min_{a_i \in \tilde{C}} a_i^l. \end{aligned}$$

Suppose that $\exists\ \varepsilon > 0$: such that $b_k^l = \mathrm{id}(\tilde{C})^l - \varepsilon$.

$$\Rightarrow \quad \forall\ a_i \in \tilde{C} : d(a_i, \mathrm{id}(\tilde{C})) \leq d(a_i, b_k)$$

The set of points in $\widetilde{C}$ remain dominated by $\mathrm{id}(\widetilde{C})$. Therefore, for $a_i \in \tilde{C} \Rightarrow d(a_{i_k}, b_k) > d(a_{i_k}, \mathrm{id}(\tilde{C}))$, from where the contradiction on the fact that b_k is the best point that minimizes the relative distance to the representative points a_{i_k} of a set $\widetilde{C}$.

3. **Proposition**

As it was stated in the previous section, a representative point in a set of solutions, is the point that has the minimal distance to the ideal point of this subset.

From where the decision of partitioning the set of solutions into k clusters that represent similar features, where k represents the number of representative solutions of the Pareto front that the decision-maker wants to find, and every set of k clusters represents a partition.

Figure 1 illustrates an academic example of our problem. Let us consider a Pareto front composed of 10 non dominated solutions, we have partitioned it in two clusters, each composed of 5 solutions. As representative points of every cluster, we find the two points that are represented by the small full circles, which correspond to the minimal distance to the ideal point of the sets C_1 and C_2.

We denote by P_i a partition of a set of solutions in k clusters C_k. The evaluation of a cluster C_k is given by the minimal distance among all the distances between the points in this cluster and its ideal point. We denote the class evaluation by:

$$Q(C_k) = \min_{a_i \prec \mathrm{id}(C_k)} d^2(a_i, \mathrm{id}(C_k)).$$

We evaluate a quality of a partition P_i, by the sum of squares of all its cluster's evaluation to eliminate the compensation between qualities of clusters, $Q(P_i) = \sum_{C_k \in P_i} Q(C_k)$. The best partition in the set of possible partitions P, is the partition that minimizes $Q(P_i)$.

4. **Hierarchical algorithm**

The total number of partitions that can be built, for a given number of criteria and a given number of actions, is such that, in most cases, a complete enumeration is impossible. Therefore, heuristics are needed to find a partition or a set of k

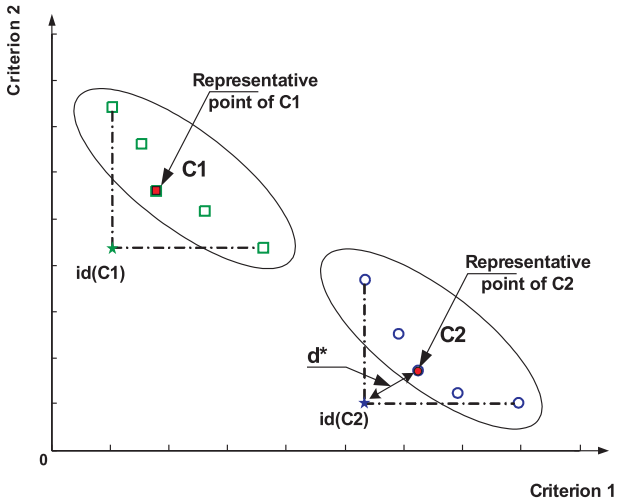

Figure 1. Example of 2 representative points in Pareto frontier

representative points. Some clustering techniques are available in the literature. Among them, the *k*-means algorithm [8]. It is an iterative algorithm where the obtained solution depends on the selection of the initial partition and may converge to a local minimum of the criterion's value if the initial partition is not properly chosen.

A hierarchical clustering proposed by Johnson [9], is a sequence of partitions in which each one is nested into the next partition in the sequence. The proposed clustering algorithm dictates how a proximity matrix should be interpreted to merge two trivial clusters, thus nesting the trivial clustering into a second partition. The process is repeated to form a sequence of nested clusterings in which the number of clusters decreases as the sequence progresses until a single cluster containing all objects is obtained. A branch and bound algorithm is proposed by Koontz *et al.* [10] to find the globally optimal clustering. However, it takes much computation time. Simulated annealing (SA) algorithm for the clustering problem is proposed by Selim *et al.* [11]. It is shown that this algorithm can reach slowly the optimal solution, because the search for optimal results requires the temperature to be decreased slowly from one iteration to another.

The evolution strategies are used to solve the clustering problem by Babu and Murty [12]. The clustering is viewed as an optimization problem that optimizes the clustering objective function. Artificial neural networks have been extensively used over the past three decades for both classification and clustering by Jain and Mao [13].

Genetic algorithms (GAs) have been most frequently used in clustering by Maulik and Bandyopadhyay [14]. GAs represent points in the search space as

binary strings, and rely on the crossover operator to explore the search space. Mutation is used in GAs for the sake of completeness, that is, to endeavour that no large part of the search space is left unexplored.

Among these clustering techniques, we choose to apply a hierarchical scheme to our problem. The algorithm is presented hereafter:

Step 1: Initialize cluster set P; each individual $a_i \in A$ constitutes a distinct cluster.

Step 2: If the number of clusters is greater than k, then go to *Step* 3, else Stop.

Step 3: Calculate the distance of all possible pairs of clusters. The distance $d(C_I, C_J)$ of two clusters C_I and $C_J \in P$ is given as the minimal distance between the nearest individuals to the ideal point of $C_I \cup C_J$:

$$d(C_I, C_j) = \min_{a_i \prec \mathrm{id}(C_I \cup C_J)/I \neq J} d(a_i, \mathrm{id}(C_I \cup C_J)).$$

Step 4: Determine two clusters with minimal distance $d(C_I, C_J)$. Merge these clusters into a larger one. Go to *Step* 2.

At each step of the algorithm, two clusters are merged such that the distance between the ideal point and the nearest objects is minimal. Starting with the partitions where each object is considered to form a cluster on its own, the number of clusters of the partitions is then iteratively reduced.

Of course, even if the simplicity of an hierarchical approach is appealing, a number of bottlenecks have been stressed. Among them, once two clusters have been merged, they can never be split again. Hence, a locally optimal merging may mislead the algorithm.

5. Variable Neighborhood Search

Contrary to other metaheuristics, the VNS [15] based on local search methods, does not follow a trajectory but explores increasingly distant neighborhoods of the current incumbent solution, and jumps from this solution to a new one if and only if an improvement has been made.

In the following we explain the basic rules of VNS metaheuristic. We then apply it to our problem.

5.1. *Rules*

Let us denote by N_V $(V = 1, \ldots, V_{\max})$ a set of neighborhood structures, with $N_V(x)$ the set of solutions in the Vth neighborhood of x.

- *Initialization*: Find an initial solution x; Select the set of neighborhood structures N_V, $(V = 1, \ldots, V_{\max})$, that will be used in the search; choose a stopping criterion;
- Repeat the following until the stopping condition is met:
  (1) set $V \rightarrow 1$;
  (2) Until $V = V_{\max}$, repeat the following steps:

1	2	1	1	2	2

Figure 2. Partition representation

(a) generate at random a point x_0 from the Vth neighborhood of ($x' \in N_V(x)$);
(b) apply some local search method with x' as initial solution, denote by x'' the so obtained best solution;
(c) if this best solution is better than the incumbent solution then move ($x \leftarrow x''$) and continue the search with $N_1(V \leftarrow 1)$; otherwise, set $V \leftarrow V + 1$.

The stopping condition may be, reaching the maximum number of iterations, the maximum CPU time, or the maximum number of iterations between two improvements.

5.2. *VNS for k representative points problem*

In our problem, we consider an encoding method where a partition is encoded as a string of length N (where N is the number of points in A). The ith element of the string denotes the group number assigned to point a_i. Thus, each string represents a possible cluster configuration.

Example. Assume that we have two clusters containing the following solutions: $C_1 = \{a_1, a_3, a_4\}$, $C_2 = \{a_2, a_5, a_6\}$, the encoding of the chromosome will be as shown in Figure 2.

Given an initial partition P, generate at random a partition P' from the Vth neighborhood of P ($P' \in N_V(P)$) by randomly selecting a solution a_i that belongs to the cluster C_l in the current solution and assigning it to another cluster $C_J, J \neq l$.

In the next step we apply a local search method with P' as initial solution, the object a_i that belongs to the cluster C_l in the current solution is assigned to some other cluster C_J, $J \neq l$, we explore all such possible exchanges (for all l and J). P'' is the obtained local optimum. A move is made if the local optimum P'' in the neighborhood is better than the current one. Otherwise, the procedure is stopped. The pseudo code of the VNS algorithm is given in Figure 3.

These operations may create two problems. The first one is that the generated partition may have fewer groups than the initial one.

For example, let us consider a partition $P = \{1, 2, 2, 3, 2, 3\}$, if we remove the object a1 from the cluster 1 and we assign it to the cluster 3, the new partition will be $P' = \{3, 2, 2, 3, 2, 3\}$. Note that this new partition has only two groups instead of three. To avoid this problem, we use objective function $Q(P) + \delta^* * (\text{nbcluster}(P) - \text{nbcluster}(P'))$, where ($\delta^* \ggg 0$).

Algorithm 2. VNS algorithm to find k representative points

Initialization: Find an initial partition P'

While $\{Time \geq T_{\max}\}$
  $i = \text{rand}(1, N)$; $J = \text{rand}(1, K)/(J \neq l)_{/a_{i \in C_l}}$;
  $P' = \text{insert}(P, a_i, C_J)$; (insert a_i in the cluster C_j)
  $P'' = \text{local\_search}(P')$; (apply a local search to the partition P')
if $\{Q(P'') < Q(P)\}$
  $P \leftarrow P''$ **endif**
endwhile

Figure 3. Pseudo code of VNS algorithm.

Table 1. Quality of partition of Hierarchical algorithm and Variable Neighborhood Search for two criteria, two and three representative points

Test	HA		VNS	
	2	3	2	3
10	0.048	0.0142	0.048	0.0140
20	0.158	0.0726	0.152	0.0720
30	0.131	0.0712	0.122	0.0593
40	0.169	0.0910	0.161	0.0800
50	0.163	0.0870	0.153	0.0860
100	0.205	0.1140	0.203	0.1090

The second problem is the case where we find a cluster that is dominated by another one. To avoid this problem we use objective function $Q(P) + \delta *$ (number of do min ated clusters), where ($\delta \ggg 0$).

6. Experimental results and discussion

This section is devoted to the empirical analysis of the model and the algorithm proposed in the previous sections. We tested both algorithms on artificial data that we generated according to a uniform distribution $U(0, 1)$. The number of criteria q, varies from 2 to 3. The number of clusters k, varies from 2 to $q + 1$.

We have tested the hierarchical algorithm (HA) and variable neighborhood search (VNS) on instances of size 10, 20, 30, 40, 50 and 100 objects.

Table 1 and 2 gather the results obtained through the VNS and hierarchical algorithms. From this experimentation we conclude that the VNS algorithm gives a better quality of partition than the hierarchical algorithm for almost all problems.

Partitioning an instance with 30 solutions of Pareto frontier to 3 clusters is shown in Figure 4. With VNS algorithm cluster 1 contains 9 solutions, there are 8 solutions in cluster 2 and 13 in cluster 3.

Table 2. Quality of partition of Hierarchical algorithm and Variable Neighborhood Search for three criteria, two, three and four representative points

Test	HA			VNS		
	2	3	4	2	3	4
10	0.152	0.093	0.054	0.137	0.082	0.052
20	0.207	0.181	0.133	0.184	0.149	0.117
30	0.234	0.212	0.169	0.218	0.174	0.148
40	0.247	0.232	0.189	0.225	0.188	0.162
50	0.268	0.24	0.228	0.255	0.211	0.187
100	0.343	0.318	0.291	0.298	0.249	0.234

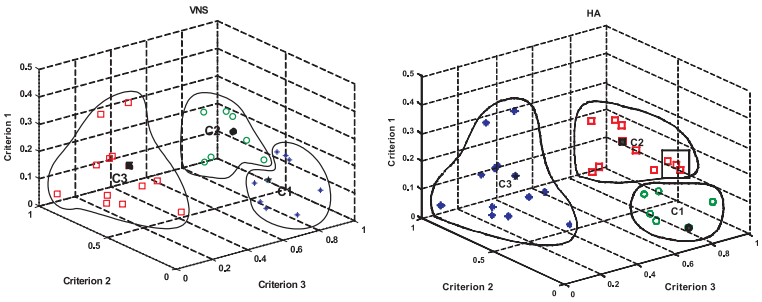

Figure 4. Example of 3 representative points of Pareto frontier

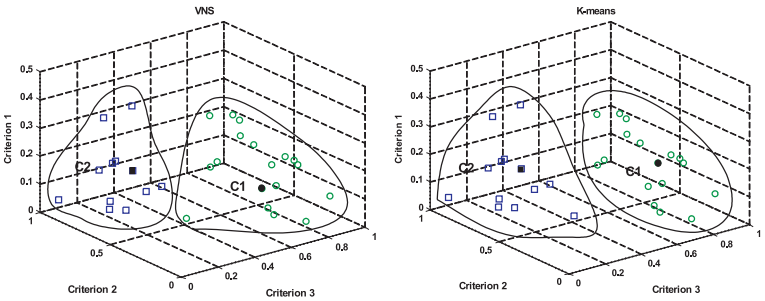

Figure 5. Example of 2 representative points of Pareto frontier

With Hierarchical algorithm cluster 1 contains 6 solutions, there are 11 solutions in cluster 2 and 13 in cluster 3.

One way for the decision maker to pick one solution, among the solutions contained in a cluster, is to identify what solution is the closest to its ideal point. In Figure 4 the representative point of each cluster is the point filled with black. Tables 3 and 4 show the results obtained with the cluster analysis; each solution is shown with its corresponding evaluation for each criterion.

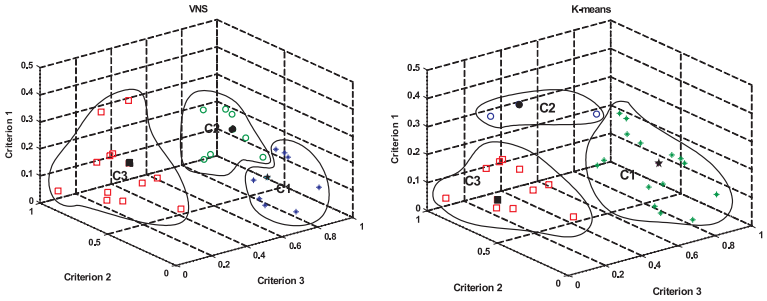

Figure 6. Example of 3 representative points of Pareto frontier

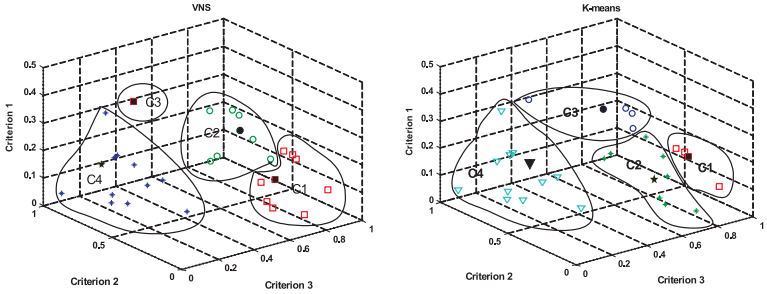

Figure 7. Example of 4 representative points of Pareto frontier

Table 3. Representative points with VNS for 30 solutions in Pareto front

	# of solutions	Criterion 1	Criterion 2	Criterion 3
Cluster 1	9	0.168	0.177	0.654
Cluster 2	8	0.384	0.165	0.449
Cluster 3	13	0.206	0.598	0.196

Table 4. Representative points with HA for 30 solutions in Pareto front

	# of solutions	Criterion 1	Criterion 2	Criterion 3
Cluster 1	6	0.168	0.177	0.654
Cluster 2	11	0.384	0.165	0.449
Cluster 3	13	0.206	0.598	0.196

With the information from tables 3 and 4 the decision maker now has a small set of solutions, and it is thus easier to make his/her choice regarding the importance of the different objectives.

Tables 3 and 4 show that the representative points in the three clusters are the same with VNS and Hierarchical algorithms. But the composition of clusters 1 and 2 is not the same. With Hierarchical algorithm the 3 points in the square in Figure 4 are assigned to cluster 2, but with VNS they are assigned to cluster 1.

Table 5. Representative points with K-means for 30 solutions in Pareto front

	# of solutions	Criterion 1	Criterion 2	Criterion 3
Cluster 1	16	0.259	0.140	0.60
Cluster 2	3	0.495	0.437	0.068
Cluster 3	11	0.064	0.749	0.168

For partition with Hierarchical algorithm these three points in cluster 2 are also dominated by the ideal point of cluster 1. We conclude that Hierarchical algorithm created an overlap partition, what is not the case for the VNS algorithm. In Table 5 we illustrate the results of selecting representative points by the proposed method for Taboada *et al.* In Figures 5, 6 and 7 we compare the resulted partition for VNS and k-means method respectively.

In these figures we can see that the cluster compositions are very different when the number of clusters increases. For many instances, if we consider the ideal point of a cluster in the k-means algorithm, we can find in some instances, ideal point of one cluster dominates many points in another cluster.

7. Conclusion

In this paper, we have motivated and formalized the notion of k representative points of Pareto frontier. In this approach, the solutions in the Pareto optimal set are clustered so that the Pareto optimal front is partitioned into k clusters. Each cluster consists of solutions with similar properties, and therefore the decision-maker only has to investigate one solution per cluster, this solution being the closest to its ideal point.

The model proposed still needs to be completed. More precisely, careful attention should be paid to the evaluation of the partitions.

In this work we have considered the sum of squared distances between the ideal point of clusters and a nearest point to this ideal point. However, other aggregation operators could be considered and should be tested.

References

[1] M.R. Anderberg, *Cluster Analysis for Applications*, Academic Press, New York, 1973.

[2] B. Roy, Des crit res multiples en recherche opérationnelle: pourquoi?, *Operational Research* 1987, 829–842.

[3] M. Ehrgott, *Multicriteria Optimization*, Berlin, Springer, 2000.

[4] L.R. Keeney and H. Raiffa, *Decisions with Multiple Objectives: Preferences and Value Tradeoffs*, John Wiley and sons, 1976.

[5] H.A. Taboada, F. Baheranwala and D.W. Coit, Practical solutions for multi-objective optimization: an application to system reliability design problems, *Reliability Engineering and System Safety* **92** (2007), 314–322.

[6] B. Malakooti and V. Raman, Clustering and selection of multiple criteria alternatives using unsupervised and supervised neural networks, *J. Intell. Manuf.* **11** (2000), 435–453.

[7] B. Malakooti and Z. Yang, Clustering and group selection of multiple criteria alternatives with application to space-based networks, *IEEE Transactions On Systems, Man and Cybernetics*, Part B: Cybernetics **34**(2004), 40–51.
[8] A.K. Jain, M.N. Murty and P.J. Flynn, Data clustering: a review, *ACM Computing Surveys* **31**(1999), 264–323.
[9] S.Z. Selim and M.A. Ismail, *K*-means-type algorithm: generalized convergence theorem and characterization of local optimality, *IEEE Trans. Pattern Anal. Mach. Intell.* **6** (1984), 81–87.
[10] S.C. Johnson, Hierarchical clustering schemes, *Psychometrika* **2**(1967), 241–254.
[11] W.L. Koontz, P.M. Narendra and K. Fukunaga, A branch and bound clustering algorithm, *IEEE Trans. Comput.* **24**(1975), 908–915.
[12] S.Z. Selim and K.S. Al-Sultan, A simulated annealing algorithm for the clustering problem, *Pattern Recogn.* **24**(1991), 1003–1008.
[13] G.P. Babu and M.N. Murty, Clustering with evolution strategies, *Pattern Recogn.* **27**(1994), 321–329.
[14] A.K. Jain and J. Mao, Neural networks and pattern recognition, in *Computational Intelligence: Imitating Life*, J.M. Zurada, R.J. Marks and C.J. Robinson (editors), 1994, 194–212.
[15] U. Maulik and S. Bandyopadhyay, Genetic algorithm-based clustering technique, *Pattern Recognition* **33**(2000), 1455–1465.
[16] P. Hansen and N. Maldenobić, Variable neighborhood search: principles and applications, *European Journal of Operational Research* **130**(2001), 449–467.

Mohamed Cheikh, *GIAD, FSEGS, route de l'a roport km 4 Sfax 3018, Tunisie.*
E-mail: mohamedc@gmail.com

Bassem Jarboui, *GIAD, FSEGS, route de l'a roport km 4 Sfax 3018, Tunisie.*
E-mail: bassem_jarboui@yahoo.fr

Ta cir Loukil, *LOGIQ, ISGIS, route Mharza Km 1.5 BP. 954, Sfax 3018, Tunisie.*
E-mail: taicir.loukil@fsegs.rnu.tn

Patrick Siarry, *Lissi, universit de Paris 12, 61av. du G n ral de Gaulle, 94010 Cr teil, France*.
E-mail: siarry@univ-paris12.fr

Received July 17, 2009
Accepted February 22, 2010