J-MEANS: A New Local Search Heuristic for Minimum Sum-of-Squares Clustering

Pierre HANSEN[1] **and Nenad MLADENOVIĆ**[1,2]

[1] *GERAD and École des Hautes Études Commerciales*
3000 chemin de la Côte-Sainte-Catherine
Montréal, Canada H3T 2A7

[2] *SMG and ISRO, Université Libre de Bruxelles*
Boulevard du Triophe,
B-1050 Bruxelles, Belgium

Abstract

A new local search heuristic, called J-MEANS, is proposed for solving the minimum sum-of-squares clustering problem. The neighborhood of the current solution is defined by all possible centroid-to-entity relocations followed by corresponding changes of assignments. Moves are made in such neighborhoods until a local optimum is reached. The new heuristic is compared with two other well-known local search heuristics, K-MEANS and H-MEANS as well as with H-MEANS+, an improved version of the latter in which degeneracy is removed. Moreover, another heuristic, which fits into the Variable Neighborhood Search metaheuristic framework and uses J-MEANS in its local search step, is proposed too. Results on standard test problems from the literature are reported. It appears that J-MEANS outperforms the other local search methods, quite substantially when many entities and clusters are considered.

1 Introduction

Consider a set $X = \{x_1, \ldots, x_N\}$, $x_j = (x_{1j}, \ldots, x_{qj}) \in R^q$ of N entities (or points) in Euclidean space R^q. The minimum sum-of-squares clustering (MSSC) problem is to find a partition P_M of X into M disjoint subsets (or clusters) C_i such that the sum of squared distances from each object x_ℓ to the centroid $\overline{x}_i$ of its cluster C_i is minimum. This problem is among the most studied in cluster analysis and has numerous applications in engineering, medicine and both natural and social sciences. It is known to be NP-hard (Brucker, 1978).

Let $\mathcal{P}_M$ denote the set of all partitions of X. Then MSSC can be expressed as follows:

$$\min_{P_M \in \mathcal{P}_M} \sum_{i=1}^{M} \sum_{x_\ell \in C_i} \|x_\ell - \overline{x}_i\|^2, \tag{1}$$

where $\|.\|$ denotes the Euclidean norm and

$$\overline{x}_i = \frac{1}{|C_i|} \sum_{\ell : x_\ell \in C_i} x_\ell,$$

for $i = 1, 2, \ldots, M$.

Beside this combinatorial formulation there are several mathematical programming ones, described in du Merle *et al.* (1997). Exact methods of branch-and-bound type (Koontz, Narendra and Fukunaga, 1975, Diehr, 1985), allow solution of small problems only, unless the clusters are far apart. A recent algorithm (du Merle *et al.*, 1997) combines several tools of mathematical programming (column generation, the ACCPM interior point method of Goffin, Haurie and Vial, 1992, hyperbolic and quadratic 0-1 programming, and variable neighborhood search) and leads to exact solution of MSSC problem with up to 150 entities (including the famous iris of Fisher, 1936).

However, numerous data sets have several hundred or thousand entities and hence heuristics are still needed. Moreover, heuristics are also important components of exact methods. Among the many heuristics for MSSC proposed in the literature the best known, and most used, appear to be K-MEANS (Jancey 1966, Mac Queen, 1967) and H-MEANS (Howard, 1966).

Recently, heuristics for MSSC which are not blocked in the first local optimum found and which fit into various metaheuristics frameworks have been proposed. They use simulated annealing (Klein and Dubes, 1989), tabu search (Al-Sultan, 1995), genetic search (Babu and Murty, 1993) or variable neighborhood search (du Merle *et al.*, 1997).

In this paper we first remain in the simpler framework of local search methods. Such methods proceed from an initial solution through a series of local improvements, to a locally optimal solution. A brief revision of the steps of K-MEANS and H-MEANS completes this introduction. In the next section we propose an improved version of H-MEANS which removes the degeneracy (empty clusters) problem.

In Section 3 we propose a new descent local search heuristic called J-MEANS. Here a cluster centroid x_i is relocated to some unoccupied entity location. Since this move corresponds to several reassignments (or K-MEANS moves) and, in contrast with the small and often ineffective centroid moves entailed by a single reassignment, can be large we refer to it as a *jump* move. Obviously, solutions obtained could be improved by H-MEANS+ and/or K-MEANS heuristics. It turns out that the most efficient hybrid is with both H-MEANS+ and K-MEANS. We denote it with J-MEANS+. In Section 4, we embed this heuristic into the variable neighborhood search metaheuristic framework (Mladenović 1995, Mladenović and Hansen 1997, Hansen and Mladenović 1998). Two variants are developed, which use J-MEANS or J-MEANS+ in their local search step. In Section 5 computer results on four standard test problems from the literature are reported, while Section 6 concludes the paper.

K-MEANS works as follows. An initial partition $(C_1, \ldots, C_m)$ is chosen at random. Then reassignments of one entity at a time are considered. Assume an entity x_j that belongs to cluster C_ℓ in the current solution is reassigned to some other cluster C_i, $(\ell \neq i)$. The centroids of these new clusters can be easily obtained from the following updating formulas (Späth, 1980):

$$\bar{x}_\ell \leftarrow \frac{n_\ell \bar{x}_\ell - x_j}{n_\ell - 1}; \quad \bar{x}_i \leftarrow \frac{n_i \bar{x}_i + x_j}{n_i + 1}. \tag{2}$$

where $n_i = |C_i|$ and $n_\ell = |C_\ell|$. The change in the objective function value caused by this move is

$$v_{ij} = \frac{n_i}{n_i + 1} \|\bar{x}_i - x_j\|^2 - \frac{n_\ell}{n_\ell - 1} \|\bar{x}_\ell - x_j\|^2, x_j \in C_\ell. \tag{3}$$

Such changes are computed for all possible reassignments. If they are all non-negative the heuristic stops with a locally minimum partition. Otherwise, the reassignment reducing most the objective function value is performed and the procedure iterated.

H-MEANS works as follows. An initial partition $(C_1, \ldots, C_M)$ is chosen at random and centroids $\bar{x}_1, \ldots, \bar{x}_M$ of its clusters are computed. Then each entity x_j $(j = 1, \ldots, N)$ is assigned (reallocated) to its closest centroid $\overline{x}_i$ $(i = 1, \ldots, M)$; if no change in assignments occurs, the heuristic stops with a locally minimum partition. Otherwise, the centroids are updated and the procedure iterated.

It should be noted that the solution obtained by K-MEANS cannot be improved by H-MEANS (or H-MEANS+), but that the solution obtained by H-MEANS can be improved by K-MEANS. Indeed, if no entity can be profitably reallocated to some other cluster than its own in the K-MEANS solution, it is assigned to its closest cluster centroid, hence the partition is a local minimum for H-MEANS too. This suggests to use H-MEANS+ followed by K-MEANS and not the reverse. We shall denote such a two-phase heuristic by HK-MEANS.

2 Modification of H-MEANS (H-MEANS+)

Since the iterations of the H-MEANS heuristic consist of alternate entity allocation (or assignment) and centroid relocation phases, it is similar to Cooper's (1963) *Alternate* heuristic for the Multisource Weber (or Location-Allocation) Problem and to Maranzana's heuristic (1963) for the p-Median problem, both well-known in Operations Research. Let us begin by stating precisely the steps of H-MEANS:

Step 1. Initialization. Let C_i, $i = 1, \ldots, M$, be the initial partition of the set X, and let $\overline{x}_i$ be the corresponding centroids.

Step 2. Assignment. Assign (allocate) each entity x_j $(j = 1, \ldots, N)$ to its closest centroid $\overline{x}_i$ $(i = 1, \ldots, M)$.

Step 3. Local optimality test. If no change in assignments occurs, a locally optimal partition is found and Stop.

Step 4. Updating. Update centroids $\overline{x}_i$ of each cluster C_i, and return to Step 2.

It is well known that the H-MEANS heuristic can stop in a so-called degenerate solution, i.e., with a partition having less than M non-empty clusters (Späth, 1985 p. 68). However, such a solution can easily be improved by some insertion strategy (as shown in Mladenović and Brimberg, 1996 for Cooper's *Alternate* heuristic).

Assume that the current H-MEANS solution is degenerate, i.e., that the number of clusters in the current solution is $M - t$ $(t > 0)$. Then we find the t entities with largest squared distances to their cluster's centroid (i.e., the t largest contributions to the objective function value) and form t new single point clusters with them. The so-obtained proper (i.e., non-degenerate) solution is obviously better than the degenerate one, but could possibly be further improved, so H-MEANS iterations continue. The modified H-MEANS heuristic is referred to as H-MEANS+ and uses the following two last steps instead of Steps 3 and 4 above:

Step 3′. *Local optimality test.* If there are changes in assignments, go to Step 4′. Otherwise, a locally optimal partition is found; if it is proper, Stop; if it is degenerate with t empty clusters, select the t points farthest from their centroids, insert them into the solution as single point clusters and return to Step 2;

Step 4′. *Updating.* Update centroids $\overline{x}_i$ of each cluster C_i, and go to Step 2.

As creation of new clusters reduces the objective function value and the number of partitions is finite, the H-MEANS+ heuristic converges to a proper locally optimal solution.

3 J-MEANS heuristic

In some problem instances (particularly when M is large), existing points could be centroids of some clusters in the current solution. We shall refer to them as *occupied points.* We next present the rules of the J-MEANS heuristic.

In order to get a neighboring solution of the current one, the centroid $\overline{x}_i$ of a cluster C_i (and not an entity, as in K-MEANS) is relocated to some unoccupied entity location and all entities of C_i relocated to their closest centroid. All possible such moves constitute the *jump neighborhood* of the current solution.

Step 1. (*Initialization*). Let $P_M = \{C_i\}$, $(i = 1, \ldots, M)$, $\overline{x}_i$, $(i = 1, \ldots, M)$ and f_{opt}, be the initial partition of the set X, the corresponding centroids, and the current objective function value, respectively.

Step 2. (*Occupied points*). Find unoccupied points, i.e., entities which do not coincide with a cluster centroid (within a small tolerance).

Step 3. (*Jump Neighborhood*). Find the best partition P'_M and corresponding value f' in the jump neighborhood of the current solution P_M.

Step 4. (*Termination or move*) If $f' > f_{opt}$, Stop (a local minimum was found in the previous iteration); otherwise, move to the best neighboring solution P'_M ($P_M := P'_M$, $f_{opt} = f'$) and return to Step 2.

The main step of J-MEANS is Step 3, and its efficient implementation is crucial. We therefore present it in more detail:

Step 3. (*Jump Neighborhood*).

- *Exploring the Neighborhood.*
  For each j ($j = 1, \ldots, N$) repeat the following steps:
  a) (*Relocation*). Add a new cluster centroid $\overline{x}_{M+1}$ at some unoccupied entity location x_j and find the index i of the best centroid deletion; denote with v_{ij} the change in the objective function value;
  b) (*Keep the best*). Keep the pair of indices i' and j', where v_{ij} is minimum;

- *Move.* Replace centroid $\overline{x}_{i'}$ by $x_{j'}$ and update assignments accordingly to get the new partition P'_M; set $f' := f_{opt} + v_{i'j'}$.

It should be noted that the efficiency of the J-Means heuristic is largely dependent on the fact that the *relocation* step (Step 3a) can be implemented in $O(N)$ time. Similar results have previously been reported for solving the p-median (Whitaker, 1983, Hansen and Mladenović, 1997) and multisource Weber problem (Brimberg *et al.*, 1997).

Observe also that J-Means can be viewed as an extended *Greedy* heuristic. Indeed, assume that all points are initially assigned to the same cluster, i.e., all M centroids are located at the same far away point (for example at origin). Then, in each iteration a new centroid is added, and one deleted from the origin. However, the *Greedy* heuristic stops when the number of origin centroids becomes zero, while J-Means could continue the search.

As mentioned above, one can improve each *jump* neighborhood solution (after completion of Step 3) by using K-Means, H-Means, HK-Means, or some other heuristic. We got the best results (within the same computing time) with HK-Means (as an improving procedure within J-Means) and we denote the resulting hybrid heuristic with J-Means+.

4 VNS heuristic

Variable neighborhood search (VNS) is a recently proposed metaheuristic for solving combinatorial problems (Mladenović, 1995, Mladenović and Hansen 1997, Hansen and Mladenović 1998). The basic idea is to proceed to a systematic change of neighborhood within a local search algorithm. The set of neighborhoods are usually induced from one metric function introduced into the solution space. The algorithm centers the search around the same solution until another solution better than the incumbent is found and then jumps there. So it is not a trajectory method as are simulated annealing or tabu search. Neighborhoods are usually ranked in such a way that solutions increasingly far from the current one are explored. We may view VNS as an optimization process with a random perturbation routine, where movement to a neighborhood further from the current solution corresponds to a larger perturbation. Unlike random restart, VNS allows a controlled increase in the level of the perturbation.

In the solution space $\mathcal{P}_M$ (the set of all partitions of X), we now introduce a distance $\rho(P'_M, P''_M)$ between any two solutions $P'_M, P''_M \in \mathcal{P}_M$. We first equivalently represent each solution P_M as a M-star graph G_M, where vertices correspond to entities and centroids and the entities from the same cluster are connected to the same vertex. Each such graph has obviously N edges (possibly with loops if some centroids coïncide with entities). Let us denote with G'_M and G''_M M-star graphs that correspond to the solutions P'_M and P''_M

respectively. Then we say that $\rho(P'_M, P''_M) = k$ if and only if G'_M and G''_M differ in k of their edges. The set of neighborhoods is induced from $\rho(P'_M, P''_M)$ as follows:

$$P''_M \in \mathcal{N}_k(P'_M) \Longleftrightarrow \rho(P'_M, P''_M) = k.$$

Note that all points from $\mathcal{N}_1(P_M)$ correspond to the neighborhood of P_M that is used in the K-MEANS heuristic.

To solve the MSSC problem, we implement a basic VNS that uses a single parameter (the number of neighborhoods that will be used in the search, k_{max}). This VNS heuristic has the following steps:

Step 1. Initialization. Let $P_M = \{C_i\}$ and f_{opt} be the initial partition of the set X and the current objective function, respectively. Choose some stopping condition, and a value for a parameter k_{max}.

Step 2. Termination. If the stopping condition is met, Stop.

Step 3. First Neighborhood. Set $k = 1$.

Step 4. Inner loop. If $k > k_{max}$, return to Step 2.

Step 5. Perturbation. Draw at random a point from $\mathcal{N}_k(P_M)$, i.e., reassign any k entities from X to other clusters than their own; denote the so-obtained partition with P'_M

Step 6. Local search. Apply the J-MEANS local search (with P'_M as initial solution); denote the resulting solution and objective function value with P''_M and f'', respectively.

Step 7. Move or Not. If $f'' < f_{opt}$, then recenter the search around the better solution found ($f_{opt} = f''$ and $P_M = P^{``}_M$) and go to Step 3. Otherwise, set $k = k + 1$ and go to Step 4.

The stopping condition may be e.g. maximum CPU time allowed (t_{max}), maximum number of iterations, or maximum number of iterations between two improvements. In Section 5 below we use t_{max}. Observe that point x' is generated at random in Step 5 in order to avoid cycling, which might occur if any deterministic rule was used.

As a local optimum within some neighborhood is not necessarily one within another, change of neighborhoods can be performed during the local search phase too. In the computer results section below, we also tested a VNS heuristic (VNS+), that uses the J-MEANS+ heuristic in Step 6 (instead of J-MEANS).

5 Computational results

The six local search methods, (e.g. K-MEANS, H-MEANS, H-MEANS+, HK-MEANS, J-MEANS and J-MEANS+), are compared on the basis of equivalent CPU times, i.e., each local search heuristic is restarted until a given time elapses. VNS and VNS+ start with the solution obtained by J-MEANS and J-MEANS+ respectively, and use the same CPU time in their search. Thus, their time limit is twice larger.

All heuristics are coded in FORTRAN 77 and run on a SUN Ultra I System (143-MHz). In order to decrease running time, compiling is done with the optimizing option, i.e., *f77 -cg92 -O4*.

The sets of problem instances used in testing are: (i) the well-known 3-dimensional 89-Bavarian postal zones of Späth, (1980); (ii) the famous 4-dimensional 150 iris of Fisher, (1936); (iii) 1060 and (iv) 3038 points in the plane, taken from the TSP-LIB data base (Reinelt, 1991). Results on those problems are presented in Tables 1, 2, 3 and 4 respectively.

	Optimal	*Deviation from optimal solution in* t_{max} *=1 s.*						t_{max} =2 s.	
M	*solution*	K-MEANS	H-MEANS	H-MEANS+	HK-MEANS	J-MEANS	J-MEANS+	VNS	VNS+
2	$6.02547\ 10^{11}$	0.00	7.75	7.75	0.00	0.00	0.00	0.00	0.00
3	$2.94506\ 10^{11}$	23.48	23.48	20.02	20.02	0.00	0.00	0.00	0.00
4	$1.04474\ 10^{11}$	0.00	166.86	0.08	0.00	0.00	0.00	0.00	0.00
5	$5.97615\ 10^{10}$	0.00	334.58	0.00	0.00	0.00	0.00	0.00	0.00
6	$3.59085\ 10^{10}$	27.65	593.23	27.65	27.65	0.00	0.00	0.00	0.00
7	$2.19832\ 10^{10}$	69.39	69.39	0.68	0.00	0.00	0.00	0.00	0.00
8	$1.33854\ 10^{10}$	141.13	163.43	0.00	0.00	0.00	0.00	0.00	0.00
9	$8.42374\ 10^{9}$	259.69	276.55	0.00	0.00	0.00	0.00	0.00	0.00
10	$6.44647\ 10^{9}$	350.65	351.95	0.00	0.00	0.00	0.00	0.00	0.00
11	$5.19798\ 10^{9}$	436.36	436.73	1.22	1.22	0.00	0.00	0.00	0.00
12	$3.95052\ 10^{9}$	2.42	597.51	1.61	1.61	0.00	0.00	0.00	0.00
13	$2.77802\ 10^{9}$	869.18	884.82	20.65	20.65	0.00	0.00	0.00	0.00
14	$2.11554\ 10^{9}$	48.48	1186.22	8.36	0.00	0.00	0.00	0.00	0.00
18	$9.80686\ 10^{8}$	155.73	2588.15	5.61	0.00	0.00	0.00	0.00	0.00
22	$5.42137\ 10^{8}$	258.59	4726.68	24.14	8.16	0.00	4.71	0.00	0.00
26	$2.82234\ 10^{8}$	91.05	9065.67	35.24	15.92	1.56	0.00	0.00	0.00
30	$1.71381\ 10^{8}$	149.86	14595.74	17.54	12.26	0.29	0.01	0.00	0.00
Average error		169.63	2121.69	10.03	6.32	0.05	0.28	0.00	0.00

Table 1: Min Sum of squares clustering in 89-Bavarian postal zones data.

Optimal solutions for Späth's postal zones data are taken from du Merle *et al.* (1997) or obtained with the algorithm described there for $M > 10$; Optimal solutions for Fisher's iris data (second column of Table 2) are taken from du Merle *et al.* (1997). In order to get best known solutions for the two other data sets (in Tables 3 and 4), we first run VNS+ with t_{max} equal to 10 and 50 minutes for each instance of (iii) and (iv) data respectively.

The % errors reported in the tables are calculated as $(f - f_{opt})/f_{opt} \cdot 100$, where f and f_{opt} denote the solution found by the heuristic and the optimal solution (in Tables 1 and 2) or the best known solution (in Tables 3 and 4).

The parameter k_{max} is set to 10 in all tests. The maximum time allowed for each run (t_{max}) is chosen to be approximately the computing time when no further improvement in the solution of the fastest heuristic, i.e. K-MEANS is observed (but possibility of some improvement with a much larger t_{max} cannot be ruled out).

Tables 1 to 4 suggest the following conclusions:

(i) Regarding solution quality, the eight methods can be clearly ranked into four groups of two heuristics: 1. VNS and VNS+, which perform best in all sets of instances; 2. J-MEANS and J-MEANS+, which are best among local search methods and outperform

M	Optimal solution	Deviation from optimal solution in $t_{max} = 1$ s.						$t_{max} = 2$ s.	
		K-MEANS	H-MEANS	H-MEANS+	HK-MEANS	J-MEANS	J-MEANS+	VNS	VNS+
2	153.3470	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
3	78.8514	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
4	57.2284	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
5	46.4461	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
6	39.0399	6.83	8.19	0.00	0.00	0.07	0.00	0.00	0.00
7	34.2982	9.60	9.03	0.36	0.00	0.29	0.00	0.00	0.00
8	29.9889	18.65	21.40	0.01	0.00	0.00	0.00	0.00	0.00
9	27.7860	21.52	2.70	1.25	0.00	0.84	0.00	0.00	0.00
10	25.8340	24.99	9.09	1.58	0.50	1.73	0.00	0.00	0.00
Average error		9.07	5.60	0.35	0.06	0.33	0.00	0.00	0.00

Table 2: Min Sum of squares clustering in 150 Fisher data.

other methods quite substantially when many clusters are considered; 3. H-MEANS+ and HK-MEANS, which perform much better than the classical local search heuristics K-MEANS and H-MEANS; 4. K-MEANS and H-MEANS.

M	Best known VNS+ (10 m.)	Deviation from the best known solution in t_{max} =10 s.						t_{max} =20 s.	
		K-MEANS	H-MEANS	H-MEANS+	HK-MEANS	J-MEANS	J-MEANS+	VNS	VNS+
10	1.75484 10^9	0.03	0.04	0.00	0.00	0.09	0.19	0.04	0.03
20	7.91794 10^8	3.96	5.84	1.98	1.38	3.43	0.04	0.83	0.04
30	4.81251 10^8	10.51	24.13	11.20	8.53	4.19	1.82	0.42	0.05
50	2.55509 10^8	16.58	74.76	38.42	25.59	5.34	3.84	1.70	0.32
60	1.97273 10^8	30.47	118.48	38.32	34.24	4.15	4.84	1.30	0.11
70	1.58450 10^8	50.44	152.31	58.16	50.83	3.72	3.13	0.82	0.19
80	1.28890 10^8	50.96	183.09	64.60	56.94	3.92	4.56	0.19	0.16
90	1.10417 10^8	57.30	216.97	67.11	57.19	4.01	3.67	0.56	0.36
100	9.63781 10^7	65.13	231.52	47.68	37.32	3.86	3.41	1.06	0.51
110	8.48458 10^7	65.72	282.31	48.89	43.92	3.18	4.07	0.44	0.13
120	7.55997 10^7	56.17	335.49	53.83	41.41	5.30	3.96	1.63	1.02
130	6.75542 10^7	65.48	375.72	56.77	51.76	6.93	6.32	0.93	0.92
140	6.11216 10^7	58.31	390.57	45.40	37.35	5.29	5.14	1.49	0.52
150	5.59256 10^7	66.50	402.19	42.90	37.40	5.17	4.69	1.02	1.54
Average error		40.99	188.54	39.76	33.17	4.07	3.46	0.89	0.42

Table 3: Min Sum of squares clustering in n=1060 TSP-LIB data.

(ii) Large errors are observed for K-MEANS and H-MEANS in the relatively small 89-Bavarian postal zones problem. They can be explained by the 'unpleasant' structure of this problem. Clusters with a single entity quite often occur in the optimal partition of this problem (Muenchen zone for example for $M = 3$ and 6, or Muenchen, Augsburg, Nurenberg and Wuersburg, for $M = 13$). As a consequence, this solution is in a deep valley, surrounded and 'protected' by local minima with poor values. It appears that this deep valley is very hard to find by edge exchanges or the alternate heuristic;

(iii) results obtained by H-MEANS are the worst in two among four data sets, but its modification H-MEANS+ performs much better than both K-MEANS and H-MEANS;

this is specially the case when the number of clusters is large and the degree of degeneracy is large too; removing degeneracy thus plays an important role in HK-Means hybrid, J-Means+ descent and VNS+ too.

	Best known	*Deviation from the best known solution in t_{max} =150 s.*						t_{max} =300 s.	
M	VNS+ (50 m.)	K-MEANS	H-MEANS	H-MEANS+	HK-MEANS	J-MEANS	J-MEANS+	VNS	VNS+
10	5.60251 10^8	0.00	0.00	0.00	0.00	0.58	0.00	0.00	0.00
20	2.66812 10^8	2.38	1.58	0.07	0.13	0.05	0.18	0.05	0.00
30	1.75598 10^8	1.00	3.98	1.26	0.94	0.51	0.83	0.51	0.83
40	1.26107 10^8	7.43	8.52	4.23	1.93	1.71	1.09	1.38	1.09
50	9.89439 10^7	11.18	16.04	3.50	3.06	1.68	1.86	1.68	1.53
100	4.77197 10^7	48.60	48.30	13.52	8.35	4.53	3.72	3.21	1.77
150	3.05573 10^7	101.27	89.44	21.60	15.08	4.95	3.84	2.84	2.27
200	2.19186 10^7	160.26	132.73	35.51	17.27	5.65	4.57	3.65	3.01
250	1.66603 10^7	215.14	185.41	45.96	31.83	6.07	3.34	2.96	1.73
300	1.33540 10^7	255.90	230.19	47.19	33.16	6.45	4.01	3.43	1.93
350	1.10979 10^7	289.04	259.65	36.01	25.82	6.39	4.46	2.61	2.64
400	9.41168 10^6	305.20	308.01	48.29	31.92	7.09	4.99	4.38	4.38
450	8.22641 10^6	308.49	345.80	44.45	29.59	6.63	4.64	3.23	3.37
500	7.23506 10^6	320.70	353.34	38.83	26.69	6.58	3.98	3.28	2.51
Average error		131.22	125.36	23.20	15.31	4.02	2.89	2.30	1.89

Table 4: Min Sum of squares clustering in n=3038 TSPLIB data.

In Table 5 the number of restarts of each method, within 1 second, for Fisher's problem is given. We do not report on number of restarts for other test problems, because similar trends are observed. Note first that the number of restarts corresponds to the number of local minima found (not necessary all different) within a given time limit. It can be seen that the fastest local search descent is K-Means, followed by H-Means. However, the solution quality of these two methods is worst. The slowest among the eight methods compared is J-Means, i.e., it performs more iterations in one restart than others. Comparing the rate of convergence to a local minimum of J-Means and J-Means+, one can see that J-Means+ is twice faster, despite the fact that it needs much more time for one iteration (J-Means+ uses HK-Means in addition to J-Means). Since the solutions obtained by J-Means+ are of better quality than those of J-Means, we conclude that change of the neighborhood structures within local search can be useful. Similar trends can be observed when comparing VNS and VNS+.

In Table 5 it can also be seen that VNS (which uses J-Means in its Step 6) finds much more local optima than J-Means alone within the same computing time (compare 33 and 113, the average number of restarts of J-Means and VNS respectively). This is one of the main reasons why the VNS metaheuristic is effective. It exploits the fact that local minima of good quality are close one to another. In order to get a new local minimum, instead of performing many iterations for each random restart, in VNS we perform only a few iterations, since the incumbent is already in a 'deep valley' of the solution space.

We observe also the so-called *central limit catastrophe* (Baum, 1986) in the previous local search heuristics' performance: when problems grow large, random local minima are almost surely of 'average' quality and increasing the number of restarts does not help much. This weakness of the multi-start approach has been observed earlier for other combinatorial problems (see for example Boese *et al.*, 1994, for travelling salesman and graph bisection

	Number of local searches (restarts) in 1 second							
M	K-MEANS	H-MEANS	H-MEANS+	HK-MEANS	J-MEANS	J-MEANS+	VNS	VNS+
2	1425	1329	1339	1094	61	113	164	222
3	1317	609	553	488	28	98	121	208
4	721	497	459	393	46	92	134	195
5	796	462	488	385	34	62	110	156
6	654	438	387	307	30	64	101	160
7	505	402	348	271	26	45	106	114
8	364	353	316	242	25	40	92	118
9	286	312	292	220	27	45	99	122
10	257	280	280	205	25	40	89	121
Average	700	527	504	406	33	66	113	157

Table 5: Number of restarts in 1 second for Fisher data.

problems, Hansen and Mladenović, 1997, for p-Median problem and Brimberg *et al* 1997, for Multisource Weber problem). In Table 6, typical trends are shown on the 150 and 1060-entities problems. Each row of the table presents the average error for a different value of t_{max}. Like in previous tables, the maximum running time for VNS and VNS+ are again twice larger than for the other heuristics, i.e., they start with multistart J-MEANS and J-MEANS+ solutions and try to improve them in the same CPU time. It appears that VNS and VNS+ are able to improve substantially the quality of the solution when t_{max} increases, while other methods do not.

N	t_{max}	K-MEANS	H-MEANS	H-MEANS+	HK-MEANS	J-MEANS	J-MEANS+	VNS	VNS+
	0.1	9.28	15.76	0.92	0.82	1.47	0.34	0.18	0.15
	0.2	9.28	10.51	0.66	0.36	0.50	0.06	0.00	0.06
	0.5	9.09	8.55	0.46	0.16	0.33	0.06	0.00	0.00
150	1.0	9.07	5.60	0.35	0.06	0.33	0.00	0.00	0.00
	2.0	9.07	3.22	0.20	0.06	0.12	0.00	0.00	0.00
	5.0	9.00	2.27	0.20	0.01	0.05	0.00	0.00	0.00
	10.0	9.00	1.22	0.09	0.01	0.01	0.00	0.00	0.00
	2.0	46.24	202.54	46.81	36.76	4.96	4.41	4.48	3.87
	5.0	44.21	189.43	43.78	34.55	4.41	3.53	3.47	2.66
	10.0	40.99	188.54	39.76	33.17	4.07	3.46	2.99	2.00
1060	20.0	38.23	174.50	38.23	31.36	3.17	2.70	1.94	1.35
	50.0	35.50	172.08	34.84	28.29	2.80	2.14	1.14	0.95
	100.0	34.47	169.50	33.37	26.17	2.46	2.03	0.73	0.59

Table 6: Min Sum-of-squares clustering for N=150 Fisher and for N=1060 TSPLIB data: average errors for different maximum running time t_{max} (in seconds).

6 Conclusions.

The well-known K-MEANS and H-MEANS heuristics for MSSC can give very poor results when one seeks partitions into many clusters. Indeed, it is not uncommon to get objective function values several times larger than the optimal ones. Moreover, experiments show that this may happen even for data sets of moderate size (e.g. the 89 Bavarian postal zones).

New heuristics, based on a different type of move, i.e., centroid-to-entity relocation give much better results. For moderate size problems they get the optimal solution in most cases. Moreover, embedding these heuristics in the VNS metaheuristic framework, yields improved methods which always gave optimal solution for the two data sets considered (89 postal zones and Fisher's 150 iris) where they are known. For larger data sets, very good solutions (within 1 or 2 percent of the best known ones) are obtained in moderate computing time (they can be improved a bit by running the heuristics for a much longer time). Average errors (relative to the best known solution) with the best heuristic (VNS+) are over 60 times smaller than those of H-MEANS and K-MEANS. So J-MEANS and the related J-MEANS+ and VNS+ appear to be good alternatives.

References

[1] K.H. Al-Sultan. A Tabu Search Approach to the Clustering Problem. *Pattern Recog.* **28**, 1443–1451 (1995).

[2] G.P. Babu and M.N. Murty. A Near-Optimal Initial Seed Value Selection in k-Means Algorithm Using Genetic Algorithm. *Pattern Recog. Lett.* **14**, 763–769 (1993).

[3] E.B. Baum. Toward Practical 'Neural' Computation for Combinatorial Optimization Problems. In J. Denker (eds.), *Neural Networks for Computing*, American Institute of Physics. (1986).

[4] K.D. Boese, A.B. Kahng & S. Muddu. A New Adaptive Multi-Start Technique for Combinatorial Global Optimizations. *Opns. Res. Lett.*, **16**, 101–113 (1994).

[5] J. Brimberg, P. Hansen, N. Mladenović and É. Taillard. Improvements and Comparison of Heuristics for Solving the Multisource Weber Problem. *Les Cahiers du GERAD G-97-37*, Montréal, Canada (1997) (to appear in *Opns Res.*).

[6] P. Brucker. On the Complexity of Clustering Problems. *Lecture Notes in Economics and Mathematical Systems* **157**, 45–54 (1978).

[7] L. Cooper. Location - Allocation Problems. *Opns. Res.* **11**, 331–343 (1963).

[8] G. Diehr. Evaluation of a Branch and Bound Algorithm for Clustering. *SIAM J. Sci. Statist. Comput.* **6**, 268–284 (1985).

[9] O. du Merle, P. Hansen, B. Jaumard and N. Mladenović. An Interior Point Algorithm for Minimum Sum of Squares Clustering. *Les Cahiers* du GERAD G–97–53, Montreal, Canada, (1997) (to appear in *SIAM J. Sci. Comput.*).

[10] R.A. Fisher. The Use of Multiple Measurements in Taxonomic Problems. *Ann. Eugenics* **VII** part II, 179–188 (1936).

[11] J.-L. Goffin, A. Haurie and J.-P. Vial. Decomposition and Nondifferentiable Optimization with the Projective Algorithm. *Mgmt. Sci.* **38** 284–302 (1992).

[12] P. Hansen and N. Mladenović. Variable Neighborhood Search for the p-Median, *Location Sci.*, **5**, 207–226 (1997).

[13] P. Hansen and N. Mladenović. An Introduction to Variable Neighborhood Search. In S. Voss *et al.* (eds.), *Proceedings of the 2nd International Conference on Metaheuristics* - MIC97, Kluwer, Dordrecht (1998).

[14] R. Howard. Classifying a Population into Homogeneous Groups. In Lawrence, J.R. (eds.), *Operational Research in the Social Sciences.* Tavistock Publ., London (1966).

[15] R.C. Jancey. Multidimensional Group Analysis. *Australian J. Botany* **14**, 127–130 (1966).

[16] R.W. Klein and R.C. Dubes. Experiments in Projection and Clustering by Simulated Annealing. *Pattern. Recogn.* **22**, 213–220 (1989).

[17] W.L.G. Koontz, P.M. Narendra and K. Fukunaga. A Branch and Bound Clustering Algorithm. *IEEE Trans. Computers* **C–24**, 908–915 (1975).

[18] J.B. MacQueen. Some methods for classification and analysis of multivariate observations. *Proceeding of 5th Berkeley Symposium on Mathematical Statistics and Probability* **2**, 281–297 (1967).

[19] F.E. Maranzana. On the Location of Supply Points to Minimize Transportation Costs. *Opns. Res. Quart.* **12**, 138–139 (1964).

[20] N. Mladenović. A Variable Neighborhood Algorithm — A New Metaheuristic for Combinatorial Optimization. Abstracts of papers presented at Optimization Days, Montréal, p. 112 (1995).

[21] N. Mladenović and J. Brimberg. A Degeneracy Property In Continuous Location-Allocation Problems. *Les Cahiers du GERAD* G–96–37, Montreal, Canada, 1996.

[22] N. Mladenović and P. Hansen. Variable Neighborhood Search. *Comps. in Opns. Res.* **24**, 1097–1100 (1997).

[23] G. Reinelt. TSP-LIB-A Traveling Salesman Library. *ORSA J. Comput.* **3**, 376-384 (1991).

[24] H. Späth. *Cluster Analysis Algorithms for Data Reduction and Classification of Objects*, (Chichester: Ellis Horwood, 1980).

[25] R. Whitaker. A Fast Algorithm for the Greedy Interchange for Large-Scale Clustering and Median Location Problems. *INFOR* **21**, 95–108 (1983).