Les Cahiers du GERAD ISSN: 0711–2440

Complement to a Comparative Analysis of Heuristics for the *p*-Median Problem

P. Hansen
N. Mladenović

G–2007–24

March 2007

Complement to a Comparative Analysis of Heuristics for the p-Median Problem

Pierre Hansen
GERAD and HEC Montréal
3000, chemin de la Côte-Sainte-Catherine
Montréal (Québec) Canada H3T 2A7
pierre.hansen@gerad.ca

Nenad Mladenović
GERAD and School of Mathematics
Brunel University
United Kingdom
nenad.mladenovic@brunel.ac.uk

March 2007

Les Cahiers du GERAD
G–2007–24

Abstract

A recent comparison of evolutionary, neural network, and scatter search heuristics for solving the p-median problem is completed by (i) gathering or obtaining exact optimal values in order to evaluate errors precisely, and (ii) including results obtained with several variants of a variable neighborhood search (VNS) heuristic. For a first, well-known, series of instances, the average errors of the evolutionary and neural network heuristics are over 10% and more than 1000 times larger than that of VNS. For a second series, this error is about 3% while the errors of the parallel VNS and of a hybrid heuristic are about 0.01% and that of parallel scatter search even smaller.

Key Words: p-median, metaheuristics, evolutionary algorithm, genetic algorithm, neural networks, scatter search, variable neighborhood search.

Résumé

Une comparaison récente entre algorithmes évolutionnistes, réseaux neuronaux et recherche dispersée pour la résolution du problème de la p-médiane est complétée par (i) la réunion ou l'obtention de valeurs optimales exactes permettant d'évaluer précisement les erreurs et (ii) l'inclusion de résultats obtenus par plusieurs variantes d'une heuristique à voisinage variable. Pour une première série d'instances, bien connue, les erreurs moyennes des heuristiques évolutionnistes et des réseaux neuronaux sont de plus de 10% et plus de 1000 fois plus large que celle de la recherche à voisinage variable. Pour une seconde série, cette erreur est d'environ 3% tandis que les erreurs de la RVV parallèle et d'une approche heuristique sont d'environ 0.01% et celle d'une recherche dispersée parallèle encore plus petite.

Mots clés : p-médiane, métaheuristique, algorithme évolutionniste, recherche heuristique, réseaux neuronaux, recherche dispersée, recherche à voisinage variable.

1 Introduction

The p-median problem (PMP for short) is a basic model of location theory. It consists, given a set of n points (or customer locations) and a matrix of distances (or costs) between all pairs of them, to select p points (or facility locations) in order to minimize the sum for all (demand) points of their distance to the closest chosen point (or facility). In matrix terms, it corresponds to the problem of selecting p rows of a square matrix in such a way that the sum of minima in each column belonging to these rows is minimized. In a variant, more often encountered in practice, a separate set of m points, among which the p points are to be chosen, is given together with a $m \times n$ matrix of distances between points of both sets.

In this last and more general case, the PMP can be expressed mathematically as follows:

$$\text{Minimize } \sum_{i=1}^{m}\sum_{i=1}^{n} d_{ij}x_{ij} \tag{1}$$

subject to

$$\sum_{i=1}^{m} x_{ij} = 1, \qquad \forall j = 1, 2, \ldots, n \tag{2}$$

$$o \leq x_{ij} \leq y_i \qquad \forall i = 1, 2, \ldots, m; \forall j = 1, 2, \ldots, n \tag{3}$$

$$\sum_{i=1}^{m} y_i = p \tag{4}$$

$$y_i \in \{0, 1\} \qquad i = 1, 2, \ldots, m \tag{5}$$

The matrix $d = (d_{ij})$ expresses the distances between potential facility locations i and demand points j. The variable x_{ij} corresponds to assignment of demand point i to facility j ($x_{ij} = 1$) or not ($x_{ij} = 0$). The indicator variable y_i expresses that a facility is established at i ($y_i = 1$) or not ($y_i = 0$). The objective function (1) is to minimize the sum for all demand points of the distance to their closest facility. Constraint (2) expresses that all demand points are to be assigned. Constraints (3) ensure that no demand point j is assigned to point i unless there is a facility there. Constraint (4) expresses that exactly p facility locations are to be chosen among the m potential ones. Finally constraint (5) expresses that facility should be located at point i entirely or not at all.

The PMP is due to Hakimi (1965) and was shown to be NP-hard by Kariv and Hakimi (1979). Therefore, the existence of an algorithm for PMP taking in the worst case a computing time polynomial in the size of the input is very unlikely.

The PMP has numerous applications in Operations research, Telecomunications, Medicine, Pattern recognition, and other fields. Consequently, a very large amount of work has been devoted to the design of algorithms and heuristics for its exact or approximate solution. Surveys are provided by, among others, Brandeau and Chiu (1989), Labbé and Louveaux (1997), Mladenović et al. (2007).

The recent work of Alba and Dominguez (2006) presents an empirical comparison of several heuristics for PMP fitting in various metaheuristic frameworks: *constructive genetic algorithm* (consGA), *generational genetic algorithm* (genGA), *celular genetic algorithm* (cGA) *neural network algorithm* (NA) and *replicated parallel scatter search* (RPSS).

The purpose of the present note is to complete the comparative study of Alba and Dominguez (2006) by gathering or providing (i) exact optimal values to test problems, in order to evaluate precisely the error made by the heuristic, and (ii) results obtained with several Variable Neighborhood Search (VNS) heuristics for the two main sets of test problems of that paper. As will be seen, conclusions of this expanded comparison differ from those of the previous one.

The note is organized as follows. The basic principle of VNS (Mladenović and Hansen 1997, Hansen and Mladenović 2001), which is simple and largely applicable, is presented in Section 2. The VNS heuristics for PMP already described in the literature are also presented in that section. They include:

- an early application of the basic scheme (Hansen and Mladenović 1997);
- a more sophisticated heuristic that uses two-level VNS, called Variable Neighborhood Decomposition Search - VNDS (Hansen, Mladenović and Perez-Brito (2001);
- parallel implementations of VNS - PVNS (García-López et al. 2002, Crainic et al. 2004, Moreno, Hansen and Mladenović 2005).

Computational results are presented in Section 3 for the two main data sets considered in Alba and Dominguez (2006). They include the VNS results and the optimal values for those test instances for which they were not available, i.e., for two cases within the second series of test problem. These optimal values were obtained with a variant of a recent primal-dual VNS algorithm for the uncapacitated facility location problem (Hansen et al. 2007). Conclusions are drawn in Section 4 together with several proposals for future research.

2 Variable neighborhood search for the p-median problem

The Variable neighborhood search metaheuristic is a framework for building heuristics to solve approximately combinatorial and global optimization problems. It exploits system-

atically change of neighborhoods within the search for a globally optimal (or near optimal) solution. VNS is based on the following simple observations: (i) An optimum for one neighborhood structure is not necessarily one for another; (ii) A global optimum is a local optimum with respect to all neighborhood structures; (iii) Empirical evidence shows that for many problems all local optima are relatively close one to the other.

The first property is exploited by using increasingly complex moves in a so-called Variable neighborhood descent (VND) in order to find local optima. The second property suggests using more neighborhoods if the local optima found are of poor quality. Finally, the third property allows, once a local optimum has been reached, exploiting the corresponding information to find a better local optimum in its vicinity.

The basic scheme of VNS is presented in Figure 1.

Initialization. Select the set of neighborhood structures $\mathcal{N}_k$, $k = 1, \ldots, k_{max}$, that will be used in the search; find an initial solution x; choose a stopping condition;

Repeat the following sequence until the stopping condition is met:

(1) Set $k \leftarrow 1$;

(2) Repeat the following steps until $k = k_{max}$:

(a) Perturbation. Generate a point y at random from the k^{th} neighborhood of x ($y \in \mathcal{N}_k(x)$);

(b) Local search. Apply some local search method with y as initial solution, to obtain a local optimum given by y';

(c) Neighborhood change. If this local optimum is better than the incumbent, move there ($x \leftarrow y'$), and continue the search with $\mathcal{N}_1$ ($k \leftarrow 1$); otherwise, set $k \leftarrow k + 1$.

Figure 1: Steps of the basic VNS

As many other heuristics for PMP the descent heuristic of the basic VNS for that problem relies on interchange moves. Once an initial solution has been chosen at random, all exchanges between a chosen point and a non chosen one are considered. The exchange corresponding to the largest decrease in objective function value is performed, and the procedure iterated until no improvement is possible. Then the current solution is perturbed and the descent phase applied again. If no improved solution is found, the perturbation is increased (unless it is already at its maximum value, in which case one returns to the smallest perturbation). Otherwise, the incumbent (or best solution found so far) is updated and the search is re-centered there. The heuristic stops according to some rule such as maximum computing time.

Data structures used in coding the descent phase are of great importance. They are discussed in detail in the papers of Whitaker (1983), Hansen and Mladenovic (1997), Resende and Werneck (2004) as well as in the survey of Mladenović et al. (2006).

Improved results are obtained for problems with large p by decomposition of the solution space, as explained in detail in Hansen et al. (2001). To this effect a set of $k < p$ chosen points are selected together with the non-chosen points assigned to them. The smaller PMP so defined is solved again by VNS. If an improved solution is obtained, it replaces the corresponding part of the incumbent. Possible reallocation of non-chosen points may then further improve the solution.

Finally, VNS can be parallelized in various ways:

1. *Multi-direction.* It is a variant of the classical multi-thread, independent search concept (Crainic and Toulouse 2003): Several searches, each a standard VNS (or variant) are started from the same point but in different directions. If the best solution found by a thread at the end of its search improves the incumbent, this last one is updated. The process then re-starts its search either from the incumbent (if it has been updated) or from a randomly selected point.
2. *Parallel Neighborhoods.* Each search thread selects randomly a neighborhood and explores it. When all searches stop, the best solution serves as starting point for the new round.
3. *Cut-off Parallel Neighborhoods.* Same as the previous strategy, but all searches are cut off as soon as one thread identifies an improved solution and the search is re-started.
4. *Co-operative Parallel Neighborhoods.* Initiated with the same basic strategy as *Parallel Neighborhoods*. But (i) threads are allowed to finish their individual searches; (ii) the best global solution is continuously updated. When a thread finishes its search, it compares its best solution to the incumbent and, if it is better, it restarts its exploration from the updated incumbent and the first neighborhood. Otherwise, it continues the search from its current solution and the next un-assigned neighborhood.

3 Computational results

3.1 OR-Lib test instances

A well-known data set for PMP, with 40 test problems, has been provided by Beasley (1985). They have n =100 to 900 points of which $p = 5$ to 200 are to be chosen. Exact solutions for all of them were obtained with a branch-and-bound algorithm due to that author on a Cray-1S parallel computer.

The basic VNS for PMP has been applied to these 40 instances in Hansen and Mladenović (1997). We add these results to those of Alba and Dominguez (2006) in Table 1. As these last authors separated the 20 first and the 20 last instances, in their Tables 3 and 5, we present in separate lines average errors for the 20 first, for the 20 last and for all of the

Table 1: Beasly instances: Percentage error comparison for various heuristics

Problems		Percentage Errors (%)				
Name	(n, p)	consGA	genGA	cGA	NA	VNS
pmed1	(100,5)	0.00	0.00	0.00	0.28	0.00
pmed2	(100,10)	0.00	0.00	0.29	1.73	0.00
pmed3	(100,10)	0.00	0.00	0.00	0.33	0.00
pmed4	(100,20)	0.00	4.85	3.00	2.99	0.00
pmed5	(100,33)	0.36	22.73	12.18	23.03	0.00
pmed6	(200,5)	0.00	0.00	0.00	0.18	0.00
pmed7	(200,10)	0.00	0.00	0.44	1.05	0.00
pmed8	(200,20)	0.20	5.31	5.13	3.82	0.00
pmed9	(200,40)	0.73	19.24	13.50	7.53	0.00
pmed10	(200,67)	0.15	44.14	44.62	29.54	0.00
pmed11	(300,5)	0.00	0.00	0.08	0.32	0.00
pmed12	(300,10)	0.04	0.09	1.10	0.57	0.00
pmed13	(300,30)	n/a	3.96	6.10	3.29	0.00
pmed14	(300,60)	n/a	25.40	22.41	8.77	0.03
pmed15	(300,100)	n/a	45.58	44.13	26.95	0.00
pmed16	(400,5)	n/a	0.15	0.00	1.00	0.00
pmed17	(400,10)	n/a	1.99	0.00	3.64	0.00
pmed18	(400,40)	n/a	6.97	13.45	9.67	0.00
pmed19	(400,80)	n/a	27.14	28.72	20.77	0.04
pmed20	(400,133)	n/a	56.51	57.24	47.18	0.00
pmed21	(500,5)	n/a	0.00	0.00	1.71	0.00
pmed22	(500,10)	n/a	0.00	2.05	2.51	0.00
pmed23	(500,50)	n/a	15.50	13.92	9.46	0.00
pmed24	(500,100)	n/a	31.81	31.00	20.67	0.00
pmed25	(600,167)	n/a	59.30	61.27	51.75	0.00
pmed26	(600,5)	n/a	0.00	0.07	2.43	0.00
pmed27	(600,10)	n/a	0.04	1.36	4.89	0.00
pmed28	(600,60)	n/a	18.32	14.21	11.07	0.00
pmed29	(600,120)	n/a	33.60	33.56	21.07	0.00
pmed30	(600,200)	n/a	60.53	57.97	46.56	0.15
pmed31	(700,5)	n/a	0.00	0.00	0.55	0.00
pmed32	(700,10)	n/a	0.04	0.52	5.10	0.00
pmed33	(700,70)	n/a	18.83	18.26	12.64	0.00
pmed34	(700,140)	n/a	38.43	37.37	24.79	0.00
pmed35	(800,5)	n/a	0.00	0.64	0.00	0.00
pmed36	(800,10)	n/a	0.00	0.67	2.57	0.00
pmed37	(800,80)	n/a	22.17	21.49	10.22	0.00
pmed38	(900,5)	n/a	0.00	0.69	2.21	0.00
pmed39	(900,10)	n/a	0.06	0.69	3.39	0.00
pmed40	(900,90)	n/a	20.85	20.53	12.56	0.04
Average of pmed1-20		0.12	13.20	12.62	9.63	0.00
Average of pmed20-40		n/a	15.97	15.81	12.31	0.01
Average of pmed1-40		0.12	14.59	14.22	10.97	0.01

40 instances. Conclusions are similar for both subsets, and hence to those for the whole set.

The basic VNS solves exactly 36 out of 40 instances. Its average error is less than 0.01%. The average error of consGA, but only for the 12 smallest instances, the only ones for which results are reported, is 0.12%. Average errors for the other heuristics are much larger, and equal to 14.59%, 14.22% and 10.97% for genGA, cGA and NA respectively, i.e., more than 1000 times the average error of the basic VNS.

3.2 TSP-Lib test instances

The instances with $n = 1400$ of the TSP-Lib (Reinelt 1991) have been used as a data set for the PMP by several researchers, including Alba and Dominguez (2006). Exact solutions for instances with $p = 10, 20, \ldots, 70$ and 90 were obtained previously with an algorithm based on stabilized column generation (du Merle et al. 1999, Crainic et al. 2004). We confirmed those values with our primal-dual VNS algorithm and completed them by the optimal values for $p = 80$ and p=100.

Alba and Dominguez (2006) give values of heuristic solutions obtained by two variants of cGA and by the parallel scatter search heuristic RPSS (García-López et al. 2003). These instances were also solved by parallel versions of VNS for PMP (García-López et al. 2002, Crainic et al. 2004), as well as by a hybrid heuristic (HYB) of Resende and Werneck (2004). The corresponding errors relative to the optimal values are given in Table 2.

It appears that parallel VNS as well as HYB attain the optimal solution in most cases and have a very small average error of about 0.01%. The best results are those of RPSS, that are optimal in 9 cases out of 10, and at 0.05% of the optimum in the last one. This

Table 2: TSP-Lib 1400-customer problem: Percentage error comparison for various heuristics

	Optimal	cGA		VNS			Hybrid	SS
p	value	(16×16)	(32×32)	VNS	VNDS	PVNS	HYB	RPSS
10	101249.47	0.274	0.186	0.000	0.000	0.000	0.000	0.000
20	57857.55	0.621	0.641	0.000	0.000	0.000	0.001	0.000
30	44013.02	1.597	1.356	0.167	0.170	0.000	0.001	0.000
40	35002.02	1.837	1.887	0.011	0.030	0.000	0.002	0.000
50	29089.71	2.865	2.606	0.139	0.000	0.000	0.004	0.000
60	25160.40	3.906	3.457	0.064	0.023	0.023	0.014	0.000
70	22125.46	4.843	4.903	0.274	0.000	0.000	0.003	0.000
80	19870.28	4.678	4.649	0.153	0.038	0.000	0.032	0.000
90	17987.91	4.684	4.676	0.378	0.000	0.000	0.004	0.000
100	16551.62	4.443	4.838	0.000	0.212	0.080	0.050	0.005
Average error		2.975	2.920	0.119	0.047	0.010	0.011	0.001

contrasts with the notable average errors of cGA on a 16 × 16 grid (2.975 %) and on a 32 × 32 grid (2.920%).

4 Conclusions

Comparing performance of evolutionary and neural network algorithms on one side and of VNS, HYB, and scatter search on the other, on the two main sets of test problems considered by Alba and Dominguez (2006), very clearly shows the superiority of the latter group over the former one in terms of accuracy of the solutions obtained. There are other criteria, though: difficulty of understanding and using the heuristics and time to find the best solution. Regarding simplicity of the methods, it appears that celular GA is easy to understand and versatile in its potential applications. The same is true for VNS which indeed has been applied in many contexts. VNDS and Parallel VNS are more complicated, but remain easy to use, particulary due to the fact that they require a very small number of parameter values to be chosen.

Regarding computing time, no discussion could be made here as Alba and Dominguez (2006) do not mention which computer was used, nor the times expended for the computation. Details on times for the VNS heuristics can be found in Table 3 and in the cited references.

These good and less good results are empirical ones and caution should be exercised in generalizing conclusions. This suggest several avenues for future research on evolutionary and neural networks heuristics as well as on VNS and scatter search, and possibly some combinations thereof.

Table 3: 1400-customer problem: CPU times of VNDS, VNS and Parallel VNS with a different number of processors (5, 10, 20 and 40) on a SUN Sparc 10 computer

			Parallel VNS			
p	VNDS	VNS	5	10	20	40
10	9.2	392.7	408.1	242.0	123.3	64.7
20	13.6	724.2	717.8	415.4	230.3	143.2
30	18.6	855.8	1012.1	606.2	307.3	180.9
40	25.7	1126.7	947.3	570.5	296.9	203.5
50	21.7	936.0	1396.6	808.5	424.1	267.8
60	31.4	1478.1	1278.6	762.4	402.3	260.7
70	96.9	1662.8	1508.3	907.7	492.7	304.0
80	50.1	1837.0	1245.7	720.3	462.4	293.9
90	46.8	1946.5	1281.2	809.4	439.3	299.6
100	39.4	2353.8	3734.4	2057.5	1105.6	605.4
Average	35.4	1331.4	1353.0	790.0	428.4	262.4

First, one might study the influence of various characteristics of the test instances instead of confining oneself, most of the time, to two dimensional instances with Euclidean distances as costs. Parameters of importance appear to be the dimension of the space to which points belong, variability of points density in that space (or presence of easy-to-find clusters) and satisfaction or not of the triangular inequality. As an example, p-median problems with distances randomly generated from some distribution are notoriously difficult. Which heuristics would perform best, or well, on such instances?

Second, one might study the evolution of the error as a function of computing time. Indeed, some heuristics that give poor results in the long run might be better, or even best, in a context where computing time is limited and accuracy not the major concern.

Third, as both celular GA and VNS are simple and versatile it would be worthwhile to explore for which problems they give better results one than the other or for which they give comparatively close results. Possibly, this could lead to better understanding of their specific strengths and weaknesses, and suggest corrective steps.

Fourth, as suggested by Alba and Dominguez (2006), celular GA could be tailored to specific problems. This avenue has been explored in Memetic algorithms (Moscato, 2003) that include local search phases within the genetic search. Gain in accuracy may then be at the cost of difficulty in understanding the resulting hybrid heuristic or of using it, due to the possible multiplication of parameters. In any event, if such a path is to be followed it would be worthwhile to check whether present sophisticated algorithms provide consistently optimal or very close to optimal solutions to test instances currently under study. Results given above show that this is the case for the p-median problem and usual test instances with $n \leq 1400$. It might be a different story for test instances with $n = 10,000$ or even $n = 100,000$ points and more, as considered in data mining.

References

Alba E. and Domínguez E. 2006. Comparative analysis of modern optimization tools for the p-median problem. Statistical Computing 16: 251–260.

Beasley J.E. 1985. A note on solving large p-median problems. European Journal of Operational Research 21: 270–273.

Brandeau M.L. and Chiu S.S. 1989. An overview of representative problems in location research. Management Science 35: 645–674.

Crainic T., Gendreau M., Hansen P., and Mladenović N. 2004. Cooperative parallel variable neighborhood search for the p-median. Journal of Heuristics 10: 289–310.

Crainic T. and Toulouse M. 2003. Parallel Strategies for Metaheuristics. In Glover, F. and Kochenberger, G. (Eds.) Handbook of Metaheuristics. Kluwer Academic Publishers. pp. 475–513.

du Merle O., Villeneuve D., Desrosiers J., and Hansen P. 1999. Stabilized Column Generation. Discrete Mathematics 194: 229–237.

García-López F., Melián Batista B., Moreno Pérez J.A., and Moreno Vega J.M. 2002. The parallel variable neighborhood search for the p-median problem. Journal of Heuristics 8: 375–388.

García-López F., Melián Batista B., Moreno Pérez J.A., and Moreno Vega J.M. 2003. Parallelization of the scatter search for the p-median problem. Parallel Computing 29: 575–589.

Hakimi S.L. 1965. Optimum distribution of switching centers in a communication network and some related graph theoretic problems. Operations Research 13: 462–472.

Hansen P., Brimberg J., Urošević D., and Mladenović N. 2007. Primal-dual variable neighborhood search for the simple plant location problem. INFORMS J. on Computing (in press).

Hansen P. and Mladenović N. 1997. Variable neighborhood search for the p-Median. Location Science 5: 207–226.

Hansen P. and Mladenović N. 2001. Variable neighborhood search: Principles and applications. European Journal of Operational Research 130: 449–467.

Kariv O. and Hakimi S.L. 1979. An algorithmic approach to network location problems; Part 2. The p-medians. SIAM Journal on Applied Mathematics 37: 539–560.

Krasnogor N., Aragon A., and Pacheco J. 2006. Memetic Algorithms, Chapter 12, In Metaheuristics in Neural Networks Learning, (Eds. Rafael Marti and Enrique Alba), Kluwer.

Labbé M. and Louveaux F.V. 1997. Location problems, Annotated bibliography in Combinatorial Optimization. DellAmico, M.,Maffoli, F. and Martello S. (Eds.). Wiley. pp. 261–281.

Mirchandani P. and Francis R. (eds.) 1990. Discrete location theory. Wiley-Interscience.

Mladenović N. and Hansen P. 1997. Variable neighborhood search. Computers and Operations Research 24: 1097–1100.

Mladenovic N., Brimberg J., Hansen P., and Moreno-Perez J. 2007. The p-median problem: A survey of metaheuristic approaches. European Journal of Operational Research 179: 927–939.

Moreno-Perez J.A., Hansen P., and Mladenović N. 2005. Parallel variable neighborhood search. In Alba, E. (Ed.) Parallel Metaheuristics: A new class of algorithms. Wiley. pp. 247–266.

Reinelt G. 1991. TSPLIB – A traveling salesman problem library. ORSA Journal on Computing 3: 376–384.

Resende M.G. and Werneck R.F. 2004. A hybrid heuristic for the p-median Problem. Journal of Heuristics 10: 59–88.