Multilevel Approaches applied to the Capacitated Clustering Problem

D. Rodney[1], A. Soper[1] and C. Walshaw[1]

[1] School of Computing and Mathematical Sciences, University of Greenwich, London England SE10 9LS

Abstract - *This paper presents two multilevel refinement algorithms for the capacitated clustering problem. Multilevel refinement is a collaborative technique capable of significantly aiding the solution process for optimisation problems. The central methodologies of the technique are filtering solutions from the search space and reducing the level of problem detail to be considered at each level of the solution process. The first multilevel algorithm uses a simple tabu search while the other executes a standard local search procedure. Both algorithms demonstrate that the multilevel technique is capable of aiding the solution process for this combinatorial optimisation problem.*

Keywords: Multilevel refinement, Clustering, Tabu search, P-median, Coarsening, Segment transfers.

1 Introduction

The field of location science addresses the problem of how to optimally locate resources. Important applications of location analysis are found in the fields of data mining [34] and clustering [26], [13]. Data clustering is a technique central to pattern recognition [20] and knowledge discovery [29] among others. It is concerned with the partitioning of n data points in m-dimensional space into k clusters, to maximise similarities between data of the same cluster. This conforms to the model of various location problems for example, the p-median problem.

The p-median problem [10], [11] is known to be NP-Hard [15]. The problem models locating p medians on a network of n nodes while attempting to minimize the total cost of connections between medians and nodes. The capacitated clustering problem, also frequently referred to in the literature as the capacitated p-Median problem (CPMP) [19], [2], has direct applications in vehicle routing [6], [16] and communication network design [23] among others. The clustering aspect has applications in fields as diverse as biology and economics [22]. The problem extends the p-median problem with the addition of capacity constraints [12] and in the case of fixed medians, it reduces to the generalised assignment problem [27], [25], [33], [14] and is known to be NP complete [9]. The aim of the CPMP is to partition n demand nodes into p disjoint clusters such that a maximum capacity constraint imposed for a cluster is not exceeded and the total cost is minimised. A formal representation of the CPMP is provided by Fleszar and Hindi [7] and the problem continues to be an area of active research [1], [7], [27].

1.1 Multilevel technique

The multilevel technique encompasses three phases: *coarsening*, *extension* and *refinement*. The technique uses recursive coarsening to create a hierarchy of approximations to a given problem. In many cases, since the problem is coarsened to the maximum point allowed by the problem constraints, the coarsest approximation can then be used as an initial solution, which is repeatedly extended (coarsest to finest) and iteratively refined, generating a final solution [31].

If the coarsening process is exact [32], a feasible solution exists at each level. Additionally, individual moves implemented on the coarsened problem in the refinement phase typically correspond to larger moves around the solutions space than individual moves implemented on an uncoarsened problem. By this process, the multilevel algorithm is able to quickly get to the detailed problem in the lower levels with a high quality solution in place. The hierarchical view of the problem taken by the multilevel algorithm imparts a more global view than can be seen by local search heuristics acting alone [31].

The multilevel technique has been successfully applied across a wide range of problem domains [30]. The recent review by Walshaw [32] provides comprehensive analysis of numerous such instances. These include the application of the multilevel technique to: graph colouring [31], covering design [4], biomedical feature selection [21], capacitated multicommodity network design [3], the graph partitioning problem [9] and the traveling salesman problem [28]. Further, the recent application of the technique to the vehicle routing problem has produced encouraging results [24]. Generic multilevel and single-level algorithms are shown in the figure 1. The call to refine in these algorithms executes the refinement algorithms of figure 4, section 3.4.

There exist a number of enhancements for the multilevel technique. These enhancements can be incorporated into the multilevel framework to improve performance [32]. *Solution-based recoarsening* [32], one of the more powerful enhancements, allows for the coarsening of a solution to a given problem. Restrictions are placed on the coarsening process, ensuring that the desirable features of the solution are still present after it is re-coarsened. In the case of a clustering solution, coarsening is applied to nodes belonging to the same cluster. The refinement algorithm in place then treats the re-coarsened solution as an initial solution and searches for further improvements and this forms the basis of iterated multilevel algorithms [31].

```
set level counter i := 0
set problem = P_i
while (P_i can be coarsened)
  P_{i+1} = coarsen(P_i)
  i := i + 1
end
Set initial solution S_i = P_i
while ( i ≥ 0)
  i := i - 1
  S_temp = extend(S_{i+1})
  S_i = refine (S_temp)
end
```

```
set problem = P
construct initial solution S for P
while (improvement found)
  refine (S)
end
```

Figure 1: Generic multilevel and single-level algorithms.

2 Multilevel technique for the CPMP

The multilevel technique applied to the CPMP attempts to aid the solution process by filtering solutions from the search space at each level and reducing the amount of detail at each level. The technique has found success in filtering solutions from the search space and this is reflected in the plot of figure 3. It can be seen from this figure that the technique is capable of obtaining improvements to the solution cost in the upper levels of the refinement process, when the solution is in a coarsened state and the number of possible solutions are restricted.

The following points can be made about the application of the multilevel technique to the CPMP.

- *Search space filtering*: As the problem is coarsened at each level, the number of nodes in the solution is reduced. Medians can only be located at the node locations present at any given level when the levels are revisited in the refinement phase. By this process, the technique filters solutions from the search space: The means by which the multilevel technique makes its main impact for the CPMP.
- *Reducing the level of detail available at each level*: While the multilevel technique approximates the solution space (reducing the number of possible median locations), the technique does not approximate the problem space i.e. cost calculations are done on actual node locations to produce accurate solutions at each level. Preliminary tests were done for the CPMP using approximate cost calculations, thus allowing more detail to be filtered from the problem. However, these results were not encouraging.
- *Impact of the node and edge weights during the coarsening process*: The data for the CPMP can be represented by a weighted graph and the solution seeks to minimise the sum of the edges connecting the nodes to their medians. Construction heuristics typically evaluate decisions based on edge weights as opposed to the weight of the nodes. However as the CPMP is capacitated and the number of medians is predetermined, when constructing a solution, node weights have to be actively considered in order to produce a feasible solution. This is especially true for an agglomerative process like the multilevel technique's coarsening phase, since as the nodes are coarsened feasibility becomes more difficult to guarantee.

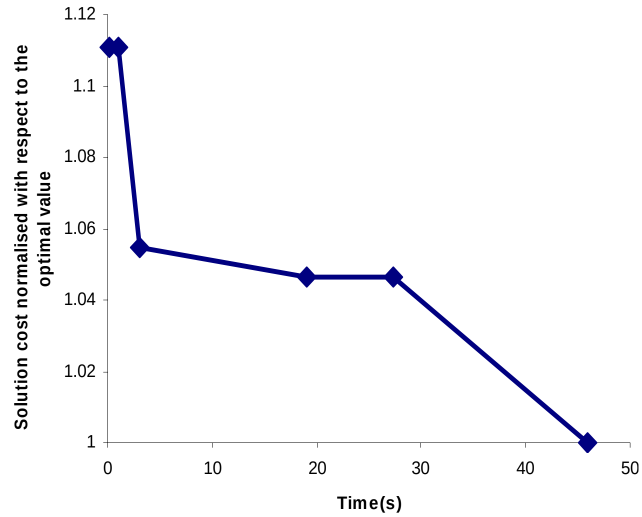

Figure 3: The change in the quality of solution throughout the refinement process as the multilevel technique is applied to instance 1 of the OC instances [22]. Refinement done to reduce the cost to approximately 1.04 above the optimal values is done in the upper levels of refinement. This corresponds to the refinement done within the first 30s of the solution process.

3 Multilevel algorithm for the CPMP

At each level, the coarsening algorithm iteratively merges node locations to form partial cluster segments, simplifying the problem. The refinement process then extends and refines the initial solution, created at the end of coarsening, until an optimised solution to the original problem is obtained.

We define a *segment* as a section of a cluster having a cost, a demand, and a location (represented by x, y coordinates). At level zero, a segment represents a single node. A segment in the upper level is created by merging a pair of existing segments. The new segment represents its constituting segments as a single location. When two segments are merged, one of these segments' locations is randomly chosen as the location of the new segment. By using one of the original segments' locations, the heuristic ensures that the search for median locations in the refinement stage occurs at locations corresponding to node locations in the original problem. The cost of an upper level segment is equivalent to the cost of connecting all its constituting segments to the location assigned to the upper level segment and its demand is equal to the total demand of all its constituting segments.

Two types of coarsening are implemented for the CPMP. The first type uses coarsening to construct the initial solution, while the second type uses a two-phase coarsening approach.

3.1 Coarsening used to create the initial solution for the CPMP

At each level, segments are matched in pairs by the coarsening algorithm and a new segment created which replaces each pair of matched segments. This continues while there are pairs of unmatched segments at the current level that can be used to create a new segment that respects the constraints. The new segments plus segments that could not be matched are included in the next level and the process repeated, until the problem is represented by p segments. This stopping condition is chosen, since p clusters are required to be served by p medians with each median belonging to exactly one cluster.

Because the problem is capacity constrained and the number of medians is predetermined, the segments are ordered by demand at each level and merged in pairs by decreasing demands. The solution produced by this heuristic may be an infeasible one, i.e. there may exist more than p medians. In this case, feasibility is enforced during the refinement phases using the inter cluster heuristic. However, merging the segments based on demand as opposed to cost reduces the possibilities of constructing infeasible solutions.

3.2 Two-phase coarsening for the CPMP

The two-phase coarsening process constructs a feasible initial solution and then coarsens the solution while respecting the clusters i.e. all segments produced belong entirely to only one cluster. Two construction heuristics were implemented for the CPMP. The first, termed the *grouping heuristic*, is modelled on the three-phase heuristic proposed by Osman and Christofides [22] and modified by Franca et al. [8].

The heuristic commences by selecting two initial median locations, these being the locations of the two nodes farthest apart. If the number of medians p is equal to two, the heuristic terminates. If, however p is greater that two, additional medians are chosen until $p-1$ medians are obtained, such that each new median maximizes the product of the distances between itself and all previously located medians. The last median is chosen, satisfying p, such that the distance between itself and all previously located medians is minimised.

In the second phase, the nodes are assigned to medians in increasing order of a calculated quotient, while the capacity constraints can be respected. The quotients are calculated by dividing the distance between each node and each median, by the node's demand. The third phase recalculates the median of each cluster at the end of all assignments. If a new set of medians is found, they become the initial medians, and the second and third phase are repeated until a stable set of medians emerges.

The second construction heuristic implemented, termed the *greedy heuristic*, selects p initial medians randomly, as done by Mulvey and Beck [19]. The nodes are assigned by increasing cost from their nearest available median, as done by Osman and Christofides [22]. However, after each insertion the median locations are updated for the affected cluster. The heuristics are modified to handle capacity overflows.

3.2.1 Coarsening the clusters

Coarsening is applied to each cluster in the solution in turn, calculating the cost between all unique pairs of segments in the cluster. The segments are merged in pairs starting with the two closest segments, then the next pair of closet segments and so on while there are pairs of unmerged segments at the current level. Segments are merged once at a given level. The new segments are then included in the next level along with any unmerged segment and the process repeated until the cluster is represented by one segment.

3.3 CPMP local search procedure - Simple segment transfer

Simple segment transfer is an inter-cluster heuristic [22],[8] that defines two move types for generating neighbourhoods: transfer and interchange. Transfer moves consider the insertion of segment(s) from one cluster into another. Interchange moves consider the exchange of segment(s) between a pair of clusters. All segments in the clusters are considered for transfer or interchange and a parameter, λ, specifies how many segments can be removed or added to a cluster at once.

An arbitrary ordering is defined on all the clusters in a solution S, the heuristic then sequentially searches all pairs of clusters in S. The search is conducted first for improving interchange moves then repeated for improving transfer moves. A first improvement strategy is used. Since the heuristic searches the clusters in pairs, the size of the search neighbourhood is determined by the number of clusters in S and the value of λ. Hence, typically λ equal to one is used to reduce the size of the search neighbourhood.

3.4 Refinement strategies for the CPMP

Two refinement strategies are implemented for the CPMP. The first, termed *simple search*, outlined at the top of figure 4 iteratively expands and then refines the clusters at each level using the simple segment transfer heuristic. When an improving move is implemented, the affected clusters are optimised, determining the best median locations within those clusters. The second strategy guides the refinement process using a *tabu search* heuristic

3.4.1 Tabu search refinement strategies

The multilevel algorithm using tabu search refinement is outlined in figure 4. Simple segment transfer is used to iteratively refine the solutions at each level, however a tabu search mechanism is added which allows the accepting of non-improving moves and the rejection of tabu-ed moves. This guides the refinement process to areas of the search space inaccessible to the simple search algorithm. The tabu search concepts of the algorithm of figure 4 are modelled on the work of France et al.[8] but the algorithm has been modified for the multilevel framework.

Synopsis of the tabu search mechanism: edges that are a part of moves implemented during the refinement phase are tabu-ed for a given number of iterations. The solution is then prevented from visiting solutions containing a given number of tabu-ed edges. An aspiration criterion can be defined such that if a tabu-ed move identifies a solution better in quality than any previously obtained, the tabu state is ignored and the move implemented. If no improving move was found during the last iteration, the least non-improving move is accepted. These non-improving moves are allowed for a stated number of iterations.

Analysis of the Tabu search mechanism: A transfer move involves two edges, while an interchange move involves four edges. At each level, a tabu range is defined, stating the lower and upper bounds for the number of iterations for which to tabu an attribute. The tabu range is calculated by dividing the number of segments at a level by four and two, giving the minimum and maximum values respectively. When a move is implemented, all the involved edges are tabu-ed and assigned a random value chosen between the tabu range. The tabu-ed edges record the segments that are transferred or interchanged. Where these segments are upper level segments, while the edges are tabu-ed and the segments have not been extended, the relevant moves are tabu-ed. Extending the segments however, means moves involving those segments are no longer tabu. Intensification and diversification of the search process is driven by: the tabu range, the acceptance of uphill and downhill moves and the controlling of when moves are tabu-ed. An integer tolerance parameter [8] is defined stating the maximum number of tabu-ed edges allowed in a move. Moves containing a number of tabu-ed edges exceeding this tolerance parameter value are disqualified.

At each level, the simple segments transfer heuristic, as used in the simple search algorithm, attempts to relocate each segment in the solution. Since there is no history of moves made at previous levels, computational resources could potentially be wasted. This can occur when expensive cost calculations are made in attempting to relocate two segments to the exact locations they exchanged as a part of an improving move for example. If the aspiration criterion is not implemented, the tabu search algorithm potentially reduces the waste of computational resources. This is due to the fact that computationally less expensive checks are made on the edges of proposed moves against those edges that have been tabu-ed. This procedure then identifies moves of the types described above.

```
expand clusters
do
   simple segments transfers
   If improvement compute median locations
while( improvement found)
```

```
set iteration counter x := 0
set maximum number of iterations tni := an integer value.
expand clusters
do
   do
      simple segments transfer
      If improvement found compute median locations
   while( improvement found)
   store best solution found to date
   if( no improvement found)
   accept least non improving move.
   compute median locations
   x := x + 1
while( x < tni)
```

Figure 4: Simple search followed by the tabu search refinement algorithm executed at each level for the CPMP

4 Results for the CPMP

The algorithms for the CPMP have been implemented in Java and tested on a Pentium-4, 3GHz PC using a number of standard test instances from the literature. The Osman and Christofides (OC) instances [22], encompass two sets (A and B) of problems totalling 20 instances. Set A contains 10 instances of sizes 50 nodes (n) and 5 medians (p), while set B contains 10 instances with n = 100 and p = 10. Maniezzo et al. [18] provided optimal solution values (opt) for these instances using a branch and bound algorithm. Tests are also performed on a set of larger instances (the San Jose dos Campos (SJC) instances [17]), that use data collected from the Brazilian city of San Jose dos Campos. This test suite consists of six instances of dimensions (n , p) equal to (100, 10), (200, 15), (300, 25), (300, 30), (402, 30) and (402, 40). Best-known values were produced by a hybrid scatter search [5].

The λ-interchange parameter, λ = 1, is employed for all tests. When tabu search refinement is used, the following additional parameters are employed. The number of iterations for which to search for improving solutions (value set to 4), the tabu range (at each level the range is given by the number of segments at that level divided by 2 and 4) and the tolerance parameter, set equal to 1 for transfers and 3 for interchanges.

4.1 Construction and coarsening heuristics testing

Three methods of coarsening were implemented for the CPMP. The pair of two-phase coarsening methods and the coarsening algorithm used to create the initial solution. These methods were experimentally evaluated for both tabu search and simple search algorithms. However, the tabu search values were more informative and outperformed the simple search values. The tabu search values are summarized in table 1 below.

Two-phase coarsening with the greedy heuristic is advocated as the means of creating initial solutions for the CPMP multilevel algorithm. As can be seen from Table 1, it outperforms the other two-phase coarsening methods when viewed across both test suites. Where coarsening is used to create the initial solutions, at the end of coarsening, the average results are 190.46% above the optimal values for the OC instances, compared with 24.68 % and 25.45% for the pair of two-phase coarsening methods. The results produced at the end of the refinement phase are better than the best produced when the other coarsening methods were employed. These results are 2 % above the optimal values compared to 2.33% for the two-phase coarsening methods applied to the OC instances. However, the runtimes incurred by the heuristic to recover from these poor starting results are approximately 40% more than the runtimes experienced for the better starting results. The fact that the problem is capacity constrained and the number of medians predetermined means that using coarsening to create the initial solution is biased towards the demand considerations as opposed to the cost considerations. For this reason, this form of coarsening is not advocated for this problem. This point is further illustrated in the result section on the San dos Campos instances.

Table 1: Results at the end of the solution process using tabu search refinement. Solution quality represented as averages across the test suite with respect to the optimal solution values (*opt*) or the best-known solutions values (*bks*).

Method of Coarsening	ML - Quality (% above *opt*) OC instances	ML - Quality (% above *bks*) SJC instances
Two phase Coarsening (Grouping heuristic)	2.33	7.13
Two phase Coarsening (Greedy heuristic)	2.33	4.49
Coarsening used to create the initial solution	2.00	4.05

4.2 Refinement approaches applied to the OS instances

The simple search and tabu search multilevel and single-level algorithms performance are compared and the results displayed in Tables 2 and 3. Both sets of results show that irrespective of refinement type the multilevel algorithms outperform the single-level versions. The difference in performance of the algorithms is most pronounced when the simple search is used. In this case, the single-level results are 64% worse when compared to the multilevel's. In the case of the tabu search, the single-level results are 40% worse when compared to the multilevel's. The results also show that the tabu search algorithms outperform their equivalent simple search counterparts. The simple search multilevel algorithm returns results 21.89 % above the tabu search multilevel algorithm.

The iterated multilevel (It.ML) algorithm results are also presented in Tables 2 and 3. The It.ML algorithm consists of applying solution-based recoarsening using the coarsening algorithm of section 3.2.1 to the solutions initially produced by the multilevel algorithm. The new solutions were then refined using the refinement algorithms of section 3.4. This procedure was repeated for ten iterations. In the best case the It.ML algorithm, produced results 0.49% above the optimal values.

Table 2: Results for the Osman and Christofides instances. The refinement phase uses the tabu search algorithm. Solution quality is represented as percentage of the solution cost above the optimal values.

Instances	N	P	Quality (% above opt)			Average runtime (S)		
			SL	ML	It.ML	SL	ML	It.ML
Set A	50	5	1.20	1.09	0.17	20.97	63.37	216.95
Set B	100	10	5.32	3.57	0.81	136.20	295.34	1610.20
Average			**3.26**	**2.33**	**0.49**	**78.59**	**179.36**	**913.52**

Table 3: Results for the Osman and Christofides instances. The refinement phase uses the simple search algorithm. Solution quality is represented as percentage of the solution cost above the optimal values.

Instance	N	P	Quality (% above opt)			Average runtime (S)		
			SL	ML	It.ML	SL	ML	It.ML
Set A	50	5	3.55	1.43	1.29	6.53	11.71	29.83
Set B	100	10	5.76	4.25	3.89	46.82	63.09	85.35
Average			**4.66**	**2.84**	**2.59**	**26.68**	**37.4**	**48.94**

4.3 Results for San Jose dos Campos city instances

As shown in table 4, when the algorithms were applied to the larger instances, the multilevel algorithm outperformed the single-level algorithm by a factor of 1.72. This compares to the case of the smaller OC instances where the multilevel algorithm outperformed the single-level algorithm by a factor of 1.4.

When two-phase coarsening was compared to the process of using coarsening to create the initial solutions for the OC instances, at the end of the refinement phase, the best results were found when coarsening was used to create the initial solutions. The investigation was extended to the San Jose dos Campos instances and a comparison presented in table 5. When coarsening is used to create the initial solutions for the San Jose dos Campos instances, at the end of the solution process better results are obtained compared to the case where two-phase coarsening is used. However, the starting result produced using coarsening to create the initial solution was 392% above the bks at the end of coarsening. This resulted in the multilevel algorithm requiring 1345.59 minutes to refine an average problem in the test suite. The solutions produced were an average 4.05 % above the bks; however, the iterated multilevel algorithm using two phase coarsening was able to produce superior results, 2.46 % above the bks in significantly less time, that of 655.58 minutes (see Table 4).

Table 4: Results for the San Jose dos Campos instances [18] using tabu search refinement.

	Quality (% above bks)			Time (min)		
	SL	ML	It.ML	SL	ML	It.ML
Average	**7.73**	**4.49**	**2.46**	**93.79**	**216.09**	**655.58**

Table 5: Results for the San Jose dos Campos instances [18] using the tabu search multilevel algorithm.

Method of creating initial solution	Quality (% above bks)	Time (min)
Two phase Coarsening (Greedy heuristic)	4.49	216.09
Coarsening used to create the initial solution	4.05	1345.59

5 Conclusion

Multilevel refinement is able to significantly aid the performance of local search algorithms applied to the CPMP. Due to the capacitated nature of the problem and the fact that the number of medians is predetermined, the multilevel practitioner can benefit from separating the solution construction from the process of approximating the search space. This can be done using a two-phase coarsening process as presented in this paper. This can significantly reduce the runtime compared to the case where coarsening is used to construct the initial solutions, as it produces initial solutions of higher quality. A tabu search component to the multilevel refinement algorithm can further improve the solutions, and combined with solution-based recoarsening, the technique is highly effective at producing solutions to the CPMP.

6 References

[1] Ahmadi, S. and Osman, I. (2005) Greedy random adaptive memory programming search for the capacitated clustering problem, European Journal of Operational Research, Vol. 162, pp. 30 – 44

[2] Bozkaya, B. Erkut, E. and Laporte, G. (2003) A tabu search heuristic and adaptive memory procedure for political districting, European Journal of Operational Research, Vol. 144, No. 1, pp. 12 – 26.

[3] Crainic, T. Li, Y. and Toulouse, M. (2006) A first multilevel cooperative algorithm for capacitated multicommodity network design, Computers and operations research, Vol. 33, No. 9

[4] Dai, C. Li, P. and Toulouse, M. (2006) A multilevel cooperative tabu search algorithm for the covering design problem, Department of computer science, university of Manitoba.

[5] Diaz, J. and Fernandez, E. (2006) Hybrid scatter search and path relinking for the capacitated p-median problem, European Journal of Operational Research, Vol. 169 pp. 570–585.

[6] Fisher, M. and Jaikumar, R. (1981) A generalized assignment heuristic for the vehicle routing problem, Networks, Vol. 11, No. 2, pp. 109 – 124.

[7] Fleszar, K. and Hindi, K. (2007) An effective VNS for the capacitated p-median problem, European Journal of Operational Research, doi:10.1016/j.ejor.2006.12.055

[8] Franca, P. Sosa, N. and Pureza, V. (1999) An adaptive tabu search algorithm for the capacitated clustering problem, International transaction in operation research, Vol. 6, pp. 665 – 678.

[9] Garey, M and Johnson, D. (1979) Computers and intractability: a guide to the theory of NP-completeness, New York: W. H. Freeman.

[10] Hakimi, S. (1964) Optimal locations of switching centres and the absolute centres and medians of a graph, Operations Research, Vol. 12, pp. 450 – 459.

[11] Hakimi, S. (1965) Optimal distribution of switching centres in a communication network and some related theoretic graph theoretic problems, Operations Research Vol. 13, pp. 462 – 475

[12] Hansen, P. and Mladenovic´, N. (1997) Variable neighbourhood search for the p-median, Location Science, Vol. 5, pp. 207 – 226.

[13] Hansen, P. and Jaumard, B. (1997) Cluster analysis and mathematical programming, Mathematical Programming, Vol. 79, pp. 191 – 215.

[14] Hindi, K. and Pienkosz, K. (1999). Efficient solution of large scale single-source capacitated plant location problems, Journal of the Operational Research Society, Vol. 50, pp. 268 –274.

[15] Kariv, O. and Hakimi, S. (1979) An algorithmic approach to network location problems Part II: The p-median, SIAM Journal of Applied Mathematics, Vol. 37, pp. 539 – 560.

[16] Koskosidis, Y. and Powell, W. (1992) Clustering algorithms for consolidation of customer orders into vehicle shipments, Transportation Research, Vol. 26 , No. 5, pp. 365 – 379.

[17] Lorena, L. and Senne, E. (2004) A column generation approach to capacitated p-median problems, Computers & Operations Research, Vol. 31, pp. 863–876

[18] Maniezzo, V. Mingozzi, A. and Baldacci, R. (1998) A Bionomic Approach to the Capacitated p-Median Problem, Journal of Heuristics, Vol. 4. No. 3, pp. 263 – 280.

[19] Mulvey, J. and Beck, M. (1984) Solving capacitated clustering problems, European Journal of Operational Research, Vol. 18, No. 3, pp. 339 – 348.

[20] Nieddu, L. and Patrizi, G. (2000) Formal methods in pattern recognition: A review, European Journal of Operational Research, Vol, 120, pp. 459 - 495

[21] Oduntan, I. Toulouse, M. Baumgartner, R. Bowman, C. Somorjai, R and Crainic, T. (2008) A multilevel tabu search algorithm for the feature selection problem in biomedical data, Computers and Mathematics with Applications, Vol. 55, No. 5, pp. 1019–1033

[22] Osman, I. and Christofides, N (1994) Capacitated Clustering Problems by Hybrid Simulated Annealing and Tabu Search, International Transactions in Operational Research, Vol.1, No. 3, pp. 317 – 336

[23] Pirkul, H. (1987) Efficient algorithms for the capacitated concentrator location problem, Computers and Operations Research, Vol. 14, No. 3, pp.197 – 208.

[24] Rodney, D. Soper, A. and Walshaw, C. (2007), Multilevel Refinement Strategies for the Capacity Vehicle Routing Problem, Int. J. IT & IC, Vol.2, No.3

[25] Ross, G. and Soland, R. (1977) Modelling Facility Location Problems As Generalized Assignment Problems, Management Science, Vol. 24, No. 3, pp. 345 - 357.

[26] Schaeffer, S. (2007) Survey Graph clustering, Computer Science review, Vol. 1, pp. 27 - 64

[27] Scheuerer, S. and Wendolsky, R. (2006) A scatter search heuristic for the capacitated clustering problem, European Journal of Operational Research, Vol.169, pp. 533–547

[28] Schrijver, A. (2005) On the history of combinatorial optimization (till 1960), In Aardal, K. Nemhauser, G. Weismantel, R. editor, Handbook of Discrete Optimization , Elsevier, Amsterdam, 2005, pp. 1 - 68.

[29] Stumme, G. (2003) Off to newshores: conceptual knowledge discovery and processing, Int. J. Human-Computer Studies, Vol. 59, pp. 287 – 325

[30] Teng, S. (1999). Coarsening, sampling and smoothing: elements of the multilevel method. In: Heath MT, Ranade A and Schreiber RS (eds). Algorithms for Parallel Processing. IMA Volumes in Mathematics and its Applications, Vol. 105, Springer Publishing Co. Inc., New York, pp 247–276.

[31] Walshaw, C. (2004) Multilevel Refinement for Combinatorial Optimisation Problems, Annals of Operations Research. Vol. 131, pp. 325–3

[32] Walshaw, C. (2007) Multilevel Refinement for Combinatorial Optimisation: Boosting Metaheuristic Performance. In C. Blum et al, editor, Hybrid Metaheuristics – An emergent approach for optimization. Springer. (Invited Chapter – to appear).

[33] Yagiura, M. Ibaraki, T. and Glover, F. (2006) A path relinking approach with ejection chains for the generalized assignment problem, European Journal of Operational Research, Vol. 169 No. 2, pp. 548 – 569.

[34] Zhou, Z. (2003) Three perspectives of data mining, Artificial Intelligence, Vol. 143, pp. 139–146