Proceedings of the 1995 ACM/SIGAPP Symposium on Applied Computing, February 26-28, 1995, Nashville, TN, ACM Press, pp. 293-299.

Approximation Techniques for Variations of the p-Median Problem*

Cory J. Hoelting, Dale A. Schoenefeld, and Roger L. Wainwright

Department of Mathematical and Computer Sciences, The University of Tulsa
[hoeltc,dschoen,rogerw]@euler.mcs.utulsa.edu

Abstract

In this paper we investigate the genetic algorithm (GA) as a heuristic technique for obtaining near optimal solutions to location problems that are variations of the p-median problem. The results have applications in a variety of contexts including telecommunications network design.

The p-median problem is to find an optimal set of K vertices from a connected, undirected graph with N vertices (K<N). Each of the K selected vertices is a service center for other nearby vertices. An optimal selection is one that minimizes the sum, over all vertices, of the path length from a vertex to the nearest service center. The p-median problem, also known as the min-sum multicenter problem, is a classical NP-complete network design problem.

A more general location problem involves service centers that are not vertices and various strategies for their placement. One variation investigated by this research allows for the service centers to be located at arbitrary points in the Cartesian plane. A second variation allows other terms, including the cost of service centers and the cost of interconnecting service centers, to contribute to the function being optimized. Techniques for locating a service center corresponding to a subset of vertices include a centroid strategy and a bounding circle strategy.

Introduction

This research investigates strategies for finding near optimal solutions for several variations of the p-median problem. The p-median problem for a graph G = (V,E) with N vertices is to find an optimal selection of K points on G (K<N), where a point on G can be either a vertex $v \in V$ or a point on an edge $e \in E$. Each of the K selected points serves as a service center for nearby vertices. An optimal selection is one that minimizes the sum, over all vertices, of the path lengths from a vertex to the nearest service center. Given that Q is the optimal set of K points, it can be shown that there is no loss of generality in restricting Q to be a subset of V. We refer to the resulting problem as the restricted p-median problem. In either formulation, the p-median problem is a classical NP-complete network design problem [11]. The p-median problem is also known as the min-sum multicenter problem.

We assume that we are given a collection of vertices, V, in the Cartesian plane and that an edge can be constructed between any two of these points. Edge cost is given by Euclidean distance. One variational strategy investigated by this research concerns the location of service centers. On the one hand, we can insist, as in the classical p-median problem, that the service centers are necessarily points in V. We refer to this problem as the Restricted Median Placement (RMP) problem. On the other hand, we can allow the service centers to be arbitrarily located in the Cartesian plane. Thus, the service center locations may or may not be in V. We refer to this problem as Generalized Median Placement (GMP) problem. A second variational strategy investigated by this research concerns the terms contributing to total cost. On the one hand, we can simply accumulate the total cost of the edges where each edge connects a vertex to a corresponding service center as in the p-median problem. We refer to this model as the Island (I) model. On the other hand, we can accumulate the cost implied by the island model along with the cost of the service centers and the cost of interconnecting the service centers with a minimum spanning tree. We refer to this model as the Connection (C) model. By considering either the RMP problem or the GMP problem in combination with the I model or the C model, the four variations of the p-median problem considered in this paper are: (1) the Restricted Median Placement problem with the Island model (RMPI) where we assume that each service center is necessarily located at one of the given vertices and that the total cost is obtained by simply summing the distance from each vertex to the nearest service center, (2) the Generalized Median Placement problem with the Island model (GMPI) where we assume that each service center can be located at any point in the Cartesian plane and that the total cost is obtained by summing the distances from each vertex to the associated (not necessarily nearest) service center, (3) the Restricted Median Placement problem with the Connection model (RMPC) where we assume that each service center is necessarily located at one of the given vertices and that the total cost is obtained by summing the distances from each vertex to the nearest service center, the cost of the service

* Research supported by OCAST Grant AR2-004 and Sun Microsystems, Inc.

centers, and the cost of interconnecting the service centers, and (4) the Generalized Median Placement problem with the Connection model (GMPC) where we assume that each service center can be located at any point in the Cartesian plane and that the total cost is obtained by summing the distances from each vertex to the associated (not necessarily nearest) service center, the cost of the service centers, and the cost of interconnecting the service centers. The RMPI problem is the classical Euclidean p-median problem for a complete graph [11].

The Problem

Formally, the Generalized Median Placement problem with the Connection model (GMPC) can be stated as follows (see Figure I):

Given a set $V \subseteq \Re \times \Re$ of N points and a positive integer K (K<N), find a set of K points, say $Q = \{q_1, q_2, ..., q_K\} \subseteq \Re \times \Re$ and a partition of V with K subsets, say $\Omega = \{P_1, P_2, ..., P_K\}$ where q_i is associated with P_i for each i = 1, 2, ..., K, such that the following cost is a minimum:

$$
\begin{aligned}
Cost(Q,\Omega) = \quad & \Sigma_{\,i=1,..,K}\, [w * star(q_i,P_i)\,] \\
& + \Sigma_{\,i=1,..,K}\, [fixedcost\,] \\
& + \Sigma_{\,i=1,..,K}\, [varcost(\mathrm{card}(P_i))\,] \\
& + W * \mathrm{mst}(Q).
\end{aligned}
$$

We refer to V as the set of N vertex locations. We refer to Q as the set of K service centers. The factor $star(q_i,P_i)$ is the total Euclidean edge length of the "star" tree connecting each point of P_i to the associated service center, q_i. The factor w is a constant weight indicating the cost per unit length of connectivity between any point in P and its associated service center. The *fixedcost* term is used to reflect the fixed cost of installing a service center at each q_i. The $varcost(\mathrm{card}(P_i))$ function is included to reflect a variable cost of installing a service center at q_i, which is dependent upon the cardinality of the collection of associated vertices. The $mst(Q)$ factor is the total length of the minimum spanning tree that interconnects the service centers. Finally, the factor W is a constant weight indicating the cost per unit length of "backbone" connectivity between any two points in Q. The factor W and the factor w are not necessarily equal.

The RMPC problem is the same as the GMPC problem except that we assume that the service center locations are necessarily vertex locations and that each vertex is associated with the nearest service center. The GMPI problem is the special case of the GMPC problem obtained by assuming that the $w * star(q_i,P_i)$ terms are the only possible non-zero terms in $Cost(Q,\Omega)$. That is, for the GMPI problem, the only cost is connectivity to a service center. There is no cost associated with establishing a service center and there is no cost associated with interconnecting the service centers. Similarly, the RMPI problem is a special case of the RMPC problem (i.e., the service center locations are necessarily vertex locations and each vertex is associated with the nearest service center) where the $w * star(q_i,P_i)$ terms are the only possible non-zero terms in $Cost(Q,\Omega)$.

Background and Related Problems

Location theory is concerned with the problem of selecting the best location in a specified region for a service center such as a shopping center, a warehouse, or a network switching center. The mathematical structure of a location problem depends on the selection of a location from the available sites and the strategy for evaluating the quality of a location. Evans and Minieka [10,20] present a classification scheme for their median, general median, absolute median and general absolute median location problem. A median of a graph is a vertex x of the graph having the smallest possible total distance from x to all other vertices. A general median of a graph is a vertex x with the smallest total distance to each edge, where the distance from a vertex to an edge is taken to be the maximum distance from the vertex to all points on the edge. An absolute median is any point on any edge whose total distance to all vertices is as small as possible. It can be shown that there is always a vertex that is an absolute median. A general absolute median is any point on any edge with the property that the total distance from it to all edges is as small as possible. Again, the distance from a point to an edge is taken as the maximum distance from the point to all points on an edges. Evans and Minieka also present a survey of solution techniques for each of their median, general median, absolute median and general absolute median location problems. One can generalize each of the four median problems presented in Evans and Minieka to p-median problems by allowing for the selection of more than one service center location. However, the p-median problems become complex. Historically, the techniques for finding multimedian solutions rely upon integer programming techniques. Such techniques are presented in a book by Handler and Mirchandani [17] and in surveys by Tansel et al. [23,24]. Hakimi [15,16] has shown, using the terminology of Evans and Minieka, that there is a p-absolute median that consists entirely of vertices. Hence, the problem in our research that we refer to as the Island model of the Restricted Median Placement (RMPI) problem is the same problem as Evans and Minieka's p-median problem for a complete graph on N vertices and Hakimi's p-absolute median problem for a complete graph on N vertices. As stated previously, the problem is NP-complete [11]. The GA techniques developed in our research for finding a near optimal solution to the RMPI problem are alternatives to integer programming.

The RMPC problem considered by this research does not specifically correspond to any of the problems in the classification of Evans and Minieka. They do not consider the cost issue of establishing service centers and do not consider the cost of interconnecting the service centers. The GMPI and GMPC problems considered by this research also do not correspond to any of the problems in the classification by Evans and Minieka. All of their variations assume that the multimedian points are either vertices or points along the edges.

Genetic Algorithms

Several researchers have investigated the benefits of solving combinatorial optimization problems using genetic algorithms[1,2,4,6,9,25]. Davis, Goldberg and Rawlings provide an excellent in-depth study of genetic algorithms [7,8,13, 21]. It is assumed that the reader is familiar with the fundamentals of genetic algorithms (GA). The GA package used in this research is LibGA [5]. The authors are not aware of other research applying heuristic genetic algorithm techniques to these variations of the p-median problem.

Service Center Encoding for the Restricted Median Placement Problem

To apply genetic algorithms, one must define a chromosome encoding strategy, an evaluation function, and a recombination strategy. For the two restricted median placement problems (RMPI and RMPC) with N vertex locations where K of the vertex locations are selected as service centers, a natural chromosome encoding is to use a string of N bits where there are exactly K bits set to 1 and exactly N-K bits set to 0. If a 1 indicates that a vertex is selected as a service center, then there is a one-to-one correspondence between the set of all bit strings of length N having exactly K bits set to 1 and the set of possible selections of K service centers from among the N vertices.

A more compact encoding [6], particularly when the number of service center locations is small in comparison to the number of vertex locations, is to let a chromosome consist of a list of K distinct integers in the range from 1 to N where each integer in the list identifies the index of a vertex selected as a service center. For example, if there are N = 10 vertices and the objective is to select a subset of K = 3 of the vertices to serve as service centers, the chromosome given by, say, (8 2 5) represents that the three service center locations are at vertices with labels 8, 2, and 5. Each chromosome consisting of a list of K distinct integers uniquely determines a selection of K service centers selected from the N vertices. However, a selection of K service centers from the N vertices does not correspond to a unique chromosome since any two permutations of K distinct integers determine the same selection of service centers. This encoding will be referred to as the *service center* encoding. The *service center* encoding is the only encoding that we used for the RMP problem.

Since every chromosome with the *service center* encoding is a feasible chromosome and represents a selection of K service centers from the N vertex locations, no penalty term is used in an evaluation function. The evaluation function used for the *service center* encoding is simply $Cost(Q,\Omega)$, the function that we are trying to minimize.

The recombination operation used with the *service center* encoding is a variant of the two parent order-based crossover. The crossover works by taking two parents and performing a regular uniform crossover (for each chromosome position , randomly choose a value from the corresponding position in either parent 1 or parent 2). Afterwords, duplicate values must be removed from the children by randomly selecting an index that does not already appear in the chromosome [6].

Locating a Service Center for a Subset of Vertices

The generalized median placement problem (GMPI and GMPC), where each of the N vertices are serviced by one of the K service centers located at arbitrary points in the Cartesian plane, can be subdivided into two sub-problems. The first problem is a set partitioning problem. A scheme is required to determine the subcollection of vertices to be served by each of the service centers. The second problem is one of appropriately locating a service center for any given subcollection of vertices. The two problems are not necessarily independent and, in fact, our chromosome encoding schemes will illustrate partition representation strategies where the iterative assignment of vertices to partition elements is dependent upon the greedy strategy for assigning service centers.

As a preliminary for developing several chromosome encoding schemes for the GMP problem, we present two techniques that we utilized for assigning a "good" service center location to a given subcollection of vertices. The techniques are referred to as the *centroid* strategy and the *bounding circle* strategy. For a set of vertices given by

$$U = \{(u_{x_1},u_{y_1}), \ldots,(u_{x_s},u_{y_s})\},$$

the *centroid* strategy assigns the service center location to be the point $c = (c_x,c_y)$ where

$$c_x = (\Sigma_{i=1,\ldots,s} u_{x_i}) / s \text{ and, similarly,}$$
$$c_y = (\Sigma_{i=1,\ldots,s} u_{y_i}) / s.$$

The *bounding circle* strategy finds the center of the smallest circle containing all points of U. A simple, exact algorithm for computing the smallest bounding circle for the points in U was devised by E. Grassmann and Rokne and can be found in [3]. Although not used in our two dimensional context, it is interesting to note that a near optimal $\vartheta(N)$ algorithm for computing a bounding sphere for N points in three dimensional space was devised by Ritter and can be found in [12].

Clearly, the *centroid* strategy produces a reasonable service center location for any given subcollection of vertices. On the other hand, it is easy to devise a subcollection of vertices for which the center of the *bounding circle* seems to be a very poor choice for the service center. The motivation for proceeding with the *centroid strategy* for the GMP problem is to assist any evolutionary heuristic strategy to do the best that it can with a given partition, although, in the process, evolution of a better partition may be inhibited. The motivation for proceeding with the *bounding circle* technique for the GMP problems is the anticipation that genetic pressure will cause a better partition of the vertices to evolve. Thus, both techniques have merit.

Group Number Encoding for the Generalized Median Placement Problem

Jones and Beltramo [19] investigated partitioning problems using genetic algorithms. A common problem in applied research is to partition a collection of objects into a fixed number of groups to optimize an objective function. Solving this problem is difficult because the number of partitions of N objects into K groups increases exponentially with K^N [18].

To solve partitioning problems with GAs, we must encode partitions in a way that allows manipulation by genetic operators. Our first chromosome encoding strategy for representing a partition of N vertices and a selection of K service centers for the Generalized Median Placement (GMP) problems is to use an N-string whose i^{th} element is the group number assigned to object i. As in Jones and Beltramo [19] we refer to the encoding strategy as the *group number* encoding. For example, the partition of a set of N=10 vertices with labels A, B, ... , and J into K=3 subsets given by

$$\{\{A,E,H\}, \{C,G,I\}, \{D, B, F,J\}\}$$

is represented by the string

```
(1 3 2 3 1 3 2 1 2 3).
```

Encoding partitions as strings of group numbers allows the use of standard single-point and uniform crossover operators. These operators, however, create two problems. First the child can have fewer groups that the parents. For example, if we cross strings

```
(1 2 2 3 3 3 3 3 3 1) and
(1 3 3 2 2 2 2 2 2 1)
```

after the third position with single point crossover, one child will be

```
(1 2 2 2 2 2 2 2 2 1)
```

which contains two groups rather than three. This problem also illustrates another problem. Both parents encode the same partition,

{{A,J}, {B,C}, {D,E,F,G,H,I}};

the only difference is how the groups are numbered. The child, on the other hand encodes an entirely different partition. Normally, except for mutation, we would like a child of "good" and identical parents to more strongly resemble the parents [19].

Every chromosome resulting from the group number encoding corresponds to a partition of the set of vertices. During one set of experiments, we used the *centroid* strategy for determining the service center associated with each group in the partition. That is, for a given group in the partition, the service center was assumed to be located at the centroid of the vertices in the group. During a second set of experiments, we used the *bounding circle* strategy for determining the service center associated with each group in the partition. That is, for a given group in the partition, the smallest circle containing all of the points in the group is determined and its center is assumed to be the location of the service center. In either case, once the vertices in each group of a partition are determined by the *group number* encoding scheme and the service center for each group in the partition is identified, the evaluation function used for the group number encoding is $Cost(Q,\Omega)$, the function that we are trying to minimize.

Encoding a partition as a string of group numbers allows the use of single-point and uniform crossover operators. However, two parents, each having K groups, can produce children that have fewer than K groups. In our experiments, children having fewer than K groups did not survive. In other literature, a rejection method is used to ensure that two parents with K groups produce children with K groups. For example, uniform crossover considers each vertex separately, giving the child the group number of either parent 1 or parent 2 depending on the flip of a coin. A rejection method is used if the children do not represent K groups [19].

Davis Encoding for the Generalized Median Placement Problem

Our second chromosome encoding strategy for the GMP problem is referred to as the *permutation* encoding. The general technique is attributed to Davis by Jones and Beltramo [19]. The technique has been used successfully in other research [2,4]. A chromosome is a permutation of the N vertices to be partitioned into K subsets according to a greedy assignment strategy. The evaluation function assumes that the first K vertices are assigned to K different groups in a partition as suggested by the Davis *permutation* encoding strategy. In particular, the first vertex is assigned to the first group, the second vertex is assigned to the second group, etc. It is possible that the first K vertices might be close together. However, intuitively it would be more desirable for a set of K vertices, one in each group, to be more widely dispersed. The remaining N - K vertices are examined individually and iteratively. As each new vertex is considered, the cost consequence of assigning it to each of the K groups is determined. The vertex is assigned to the subgroup for which the cost consequence is a minimum. This process continues until all of the vertices have been placed into a group of the partition.

As with the group number encoding, during one set of experiments, we used the *centroid* strategy for determining the service center associated with each group in the partition and, during a second set of experiments, we used the *bounding circle* strategy for determining the service center associated with each group in the partition. To illustrate the Davis *permutation* encoding for a partition with the *bounding circle* service center selection strategy, consider the set of N = 10 vertices with labels 1, 2, 3, ..., and 10 where K = 3. The interpretation of the chromosome given by

```
(9 1 8 2 3 10 6 4 7 5)
```

is that the vertex with labels 9, 1, and 8 belong to three different groups, say A, B, and C, respectively. The cost consequence of tentatively assigning vertex 2 to group A is computed by, first, finding the smallest bounding circle containing vertex 9 and vertex 2 and, then, finding the center of this bounding circle to serve as the service center for vertex 9 and vertex 2. Then, the fitness function is evaluated for vertex 9, vertex 2, and the computed service center in group A; for vertex 1 and the computed service center (which is vertex 1) in group B; and for vertex 8 and the computed service center (which is vertex 8) in group C. Similarly, the cost consequence of tentatively assigning vertex 2 to group B is computed by, first, finding the smallest bounding circle containing vertex 1 and vertex 2 and, then, finding the center of this bounding circle to serve as the service center for vertex 1 and vertex 2. Then, the fitness function is evaluated for vertex 9 and the computed service center (which is vertex 9) in group A; for vertex 1, vertex 2, and the computer service center in group B; and for vertex 8 and the computed service center (which is vertex 8) in group C. Similarly, the cost consequence of tentatively assigning vertex 2 to group C is computed. Vertex 2 is, then, assigned to the group, either A, B, or C, that produces the minimum value of the cost function. The chromosome interpretation continues with the determination of the group for vertex 3, then vertex 10, 6, 4, 7, and, finally, vertex 5.

By encoding a partition as a permutation, one can use any of the crossovers that have been designed for permutations. Much work has been done with such crossover operators applied to the traveling salesman problem [4,14]. In this research, we used the partially matched crossover (PMX) operator. Two crossover sites are selected randomly and the elements between the two starting positions in one of the parents are directly inherited by one of the offspring. Each element between the two crossover points in the alternate parent are mapped to the position held by this element in the first parent. Then the remaining elements are inherited from the alternate parent. The PMX crossover is described in [14,22] and has been used successfully in other network design optimization problems [1].

Test Cases and Results

Various genetic algorithm strategies (GA) were applied to a data set consisting of sixty (60) vertices with the objective of identifying eight (8) service centers. Each of the sixty vertices is a coordinate pair of real numbers randomly placed in [0,100] x [0,100] $\subseteq \Re \times \Re$. Each GA variation was repeated three (3) times. The set of sixty vertex sites remained the same, but the initial population of chromosomes was generated with a different random seed for each of the three replications. For each of the GA trial runs, a generational GA was used and the initial population pool had size 100. Each of the GA trial runs was executed until convergence. The number of generations until convergence varied from approximately fifty to over five hundred. Roulette selection, a mutation rate of 0.1, and a crossover rate of 1.0 were used A mutation simply swapped two alleles for the *group number* encoding and the Davis *permutation* encoding. For the *service center* encoding, an allele was replaced by a value not present in the chromosome.

Table I and Table II summarize our results for executing various GA algorithms. Table I shows the results for the Island (I) cost model. That is, the cost calculation is given by $Cost(Q,\Omega)$ where $w = 1$, $fixedcost = 0$, $W = 0$, and *varcost* is always 0. The only contributions to cost are from edges constructed from each vertex to its associated service center. The *service center* results presented in the first row of Table I are near optimal solutions to the RMPI problem where each service center is required to be a vertex. Each of the other rows in Table I represents results for the GMPI problem. A "*" in each of the tables indicates the best result. A "**" indicates the second best result.

Table II shows the result for the Connection (C) model. That is, the cost calculation is given by $Cost(Q,\Omega)$ where $w = 1$, $fixed\ cost = 5$, $W = 5$, and $varcost(\text{card}(Pi))$ is the square root of the number of connections to the service center. Although the values for the parameters are rather arbitrary, the selection of $w = 1$ and $W = 5$ is intended to reflect that the a "backbone" network is more expensive per unit length than "local" network links. A *fixedcost* of 5 simply reflects the fixed cost of a service center. Using the square root function for *varcost* is intended to assign an incremental cost for each vertex serviced by a service center. Again the *service center* results presented in Table II represent near optimal solutions to the RMPC problem where each service center is required to be a vertex. Each of the other rows in Table II represent near optimal solutions to the GMPC problem.

For the island model problem presented in Table I, the best result for each of the three trials was produced by the Davis *permutation* encoding with the *centroid* strategy for locating service centers. The second best was the *service center* encoding strategy where, necessarily, each service center was one of the vertices. The third best result was the Davis *permutation* encoding with the *bounding circle* strategy for locating service centers. For the connection model problem presented in Table II, the Davis *permutation* encoding with the *centroid* strategy and the Davis *permutation* encoding with the *bounding circle* strategy reversed positions for the first best and the third best performance as compared to Table I. The *service center* encoding remained the second best. In general, the best performance was obtained using the Davis *permutation* encoding with PMX and either the *bounding* circle or the *centroid* strategy.

The *group number* encodings did NOT outperform the *service center* encoding or the Davis *permutation* encoding for either the island model or the connection model. The Davis *permutation* encoding outperforming the group number encoding is consistent with the results of Jones and Beltramo [19].

When interpreting the results in either table, note that the first row represents results for a slightly different problem than the remaining six rows. The first row (RMP) assumes that service centers are placed at vertex locations and the remaining six rows (GMP) do not. Although the distinction is sensitive to the choices for other parameters, the fact that there are results on either side of the RMP result in either of the two tables, along with other theoretical considerations, suggests that the distinction might not be significant for some vertex sets or for at least some parameter choices.

Future Research

Future research might include a larger number of data sets and a greater number of trials for each data set. Real data sets that are not randomly generated should be considered. Attention should be given to investigating more variation in the assignment of parameters. For example, very high values of W in comparison to w tend to make it desirable to locate all service centers centrally and close to each other in order to minimize the cost of the backbone interconnection. The assignment of a *fixedcost* of 5 for a service center and a *varcost* given by the square root of the number of attached vertices is rather arbitrary. A realistic constraint would be to specify an upper bound on the number of vertices that can be attached to a service center. This research assumed a fixed number of service centers for a given collection of vertices. It would be useful to investigate a variable number of service centers for a given collection of vertices. Finally, for this initial research, the GA chromosome encodings and crossover strategies were selected from among those that have become quite standard. There are enhanced chromosome encodings and more tailored crossover strategies that can be applied to the RMP and the GMP problems [4].

Acknowledgments

This research has been supported by OCAST Grant AR2-004. The authors also wish to acknowledge the support of Sun Microsystems Inc.

References:

[1] Abuali, F. N., Schoenefeld, D. A. and Wainwright, R. L. "The Design of a Multipoint Line Topology for a Communication Network Using Genetic Algorithms," *Proceedings of the Seventh Oklahoma Conference on Artificial Intelligence*, November, 1993.

[2] Abuali, F. N., Schoenefeld, D. A. and Wainwright, R. L., "Terminal Assignment in a Communication Network Using Genetic Algorithms," *Proceedings of the Twenty Second Annual Computer Science Conference*, March, 1994.

[3] Arvo, James, editor, *Graphical Gems II*, Academic Press Inc., 1991.

[4] Blanton, J. L. and Wainwright, R. L. "Multiple Vehicle Routing with Time and Capacity Constraints using Genetic Algorithms", *Proceedings of the Fifth International Conference on Genetic Algorithms* (ICGA-93), Stephanie Forrest, Editor, Morgan Kaufmann Publisher, 1993, pp. 452-459.

[5] Corcoran, A. L. and Wainwright, R. L., "LibGA: A User-Friendly Workbench for Order-Based Genetic Algorithm Research", *Proceedings of the 1993 ACM/SIGAPP Symposium on Applied Computing*, February, 14-16, 1993, pp. 111-117, ACM Press.

[6] Crawford, K.D., Wainwright, R.L., and Vasicek, D.J., "Detecting Multiple Outliers in Multidimensional Data Using Genetic Algorithms," Computer Science Technical Report No. UTULSA-MCS-92-4, The University of Tulsa, July 1992.

[7] Davis, L. ed., *Genetic Algorithms and Simulated Annealing*, Morgan Kaufmann Publisher, 1987.

[8] Davis, L. ed., *Handbook of Genetic Algorithms,* Van Nostrand Reinhold, 1991.

[9] De Jong, K. A. and Spears, W. M., "Using Genetic Algorithms to Solve NP-Complete Problems", *Proceedings of the Third International Conference on Genetic Algorithms*, June, 1989, pp. 124-132.

[10] Evans, J. R. and Minieka, E., *Optimization Algorithms for Networks and Graphs*, Second Edition, Marcel Dekker, Inc., 1992.

[11] Garey, M. R. and Johnson, D. S., *Computers and Intractability: A Guide to the Theory of NP-Completeness*. Freeman, New York, 1979.

[12] Glassner, A., Editor, *Graphics Gems*, Academic Press Professional, 1990.

[13] Goldberg, D. E., *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley, 1989.

[14] Goldberg, D. and Lingle, R., "Alleles, loci, and the Traveling Salesman Problem," *Proceedings of the First International Conference on Genetic Algorithms*, pages 154-159, 1985.

[15] Hakimi, S. L., "Optimum Locations of Switching Centers and the Absolute Centers and Medians of a Graph," *Operations Research*, 12, 1964, pp. 450-459.

[16] Hakimi, S. L., "Optimum Distribution of Switching Centers in a Communications Network and some Graph Theoretic Problems," *Operations Research*, 13, 1965, pp. 462-475.

[17] Handler, G., and Mirchandani, P., *Location of Networks: Theory and Algorithms*, MIT Press, 1979.

[18] Hartigan, J., *Clustering Algorithms*, John Wiley and Sons, 1975.

[19] Jones, D. R. and Beltramo, M. A, "Solving Partitioning Problems with Genetic Algorithms", *Proceedings of the Fourth International Conference on Genetic Algorithms*, pp. 442-449, Morgan Kaufmann, 1989.

[20] Minieka, E., "The Centers and Medians of a Graph," *Operations Research*, 25(4),1977, pp. 641-650.

[21] Rawlins, G., ed., *Foundations of Genetic Algorithms*, Morgan Kaufmann Publishers, 1991.

[22] Starkweather, T., McDaniel, S., Mathias, K., Whitley, D., and Whitely, C., "A Comparison of Genetic Sequencing Operators," *Proceedings of the Fourth International Conference in Genetic Algorithms and their Applications*, pp.69-76, 1991.

[23] Tansel, B. C., Francis, R. L., and Lowe, T. J., "Location on Networks: A Survey. Part I: The p-Center and the p-Median Problems," *Management Science*, 29(4), 1983, pp. 482-497.

[24] Tansel, B. C., Francis, R. L., and Lowe, T. J., "Location on Networks: A Survey. Part II:, Exploiting Tree Network Structure," *Management Science*, 29(4), 1983, pp. 498-511.

[25] Whitley, D., Starkweather, T, and Fuquat, D., "Scheduling Problems and Traveling Salesman: The Genetic Edge Recombination Operator", *Proceedings of the Third International Conference on Genetic Algorithms,* June, 1989.

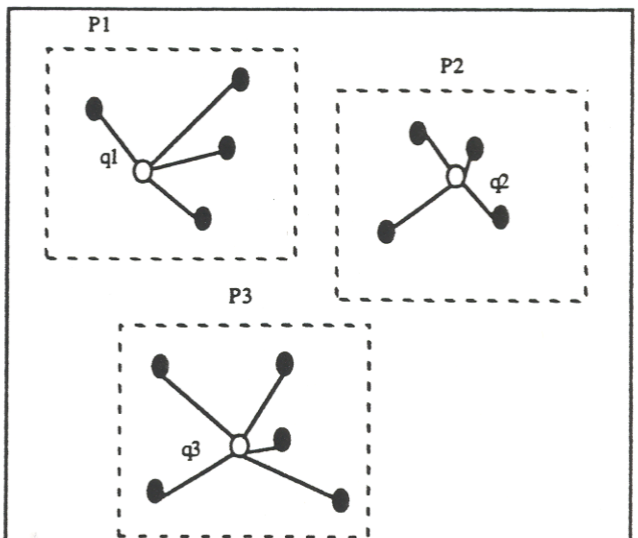

Figure I. GMPI Model
The GMPC Model interconnects the q_i's.

* best cost ** second best cost			Trial Numbers		
			I	II	III
chromosome representation	service center location strategy	crossover			
RMPI service center	a vertex	altered uniform	** 808	** 803	** 803
GMPI group number	bounding circle	simple	1376	1236	1290
group number	bounding circle	uniform	1281	1194	1164
group number	centroid	simple	1211	1064	1048
group number	centroid	uniform	919	997	1067
permutation	bounding circle	pmx	817	814	814
permutation	centroid	pmx	* 768	* 768	* 768

Table I. Island Cost Model, *w* = 1, *fixedcost* = 0, *W* = 0.
varcost is 0.
Cost of service centers is NOT included.
Cost of interconnecting service centers is NOT included.
N = 60 vertices, K = 8 service centers.

* best cost ** second best cost			Trial Numbers		
			I	II	III
chromosome representation	service center location strategy	crossover			
RMPC service center	a vertex	altered uniform	** 1906	** 1916	** 1908
GMPC group number	bounding circle	simple	2259	2263	2134
group number	bounding circle	uniform	2205	2037	2179
group number	centroid	simple	2234	2049	2097
group number	centroid	uniform	2104	2034	2082
permutation	bounding circle	pmx	* 1864	* 1897	* 1842
permutation	centroid	pmx	2001	2023	2024

Table II. Connection Cost Model, *w* = 1, *fixedcost* = 5, *W* = 5.
varcost = sqrt(number of vertices in group).
Cost of service centers is included.
Cost of interconnecting service centers is included.
N = 60 vertices, K = 8 service centers.