PHILIPPINE ENGINEERING JOURNAL
PEJ 2004; **Vol. 25 No. 1**:61–87

An Opportunity Cost-Based Modified Genetic Algorithm for the $P - k$ Median Problem

Joanna Z. Resurreccion[1,*] and Augustus C. Resurreccion[1]

[1] *Department of Engineering Sciences*
University of the Philippines Diliman
Quezon City 1101 PHILIPPINES

ABSTRACT

Median problems are combinatorial problems that associate the allocation cost of demand points to the selection of different location sites for a number of facilities that satisfy the total demand. This study focuses on the $P - k$ median problem of minimizing the total weighted distance between n demand points and these location sites when the number of existing facilities, k, on a given network is increased to P. Initial combinations of possible locations for the additional $P - k$ facilities are iteratively improved using a proposed modified genetic algorithm. The algorithm implements a new opportunity cost-based child reproduction procedure for the generation of better solutions with biased parent selection probabilities. This creates the best possible offspring without affecting the locations of existing facilities while current information from having the existing k facilities simplifies the choice of locations for increasing the number of facilities from k to P. The generated combinations of facility locations are tested on the Galvao-100 median set deriving 30 $P - k$ median problems from the Lagrangian relaxation solutions. Average percentage difference from the optimal solution found at 0.52% outperforms the neighborhood search improvement made on the myopic algorithm at higher values of $P - k$.

KEY WORDS: facility location, genetic algorithm, median problems, opportunity cost

1. INTRODUCTION

The P-median problem is the cost minimization problem of serving the demands of a network by finding the location of P facilities that will serve these demands. Hakimi [13] was among the first to present the formulation of the median problem and used a simple enumeration procedure for solving the problem [22].

The solution set of the P-median problem consists of combinations of possible locations of the P facilities called medians. The closest among the facilities to each of the demand points is assigned to serve the corresponding demand. A solution provides the locations of these facilities

*Correspondence to: Department of Engineering Sciences, University of the Philippines Diliman, 304 Melchor Hall Quezon City 1101 PHILIPPINES

Received 31 May 2004
Revised 5 January 2005
Accepted 16 February 2005

and the demand points served by each facility. While the P-median problem can be solved in polynomial time on a tree network, the problem is NP-hard on a general graph ([17] and [11]).

Facility location literature has provided solutions to the P-median problem using several heuristic algorithms with varying performance with respect to finding the optimal solution. M. Daskin [4] discusses three classes of heuristic algorithms: the myopic algorithm, exchange heuristic and neighborhood search algorithm. These algorithms estimate at their best a 0.172% average difference and can go as far as 4.208% from the optimal solution given by the Lagrangian relaxation approach [2]. When coupled with any of the three heuristic algorithms, the Lagrangian approach has been considered superior than many proposed algorithms with respect to the quality of solution obtained though it requires the most computational time [4].

Review of literature also shows that there has not been much attention placed on possible existing facilities in the P-median problem. This study, however, takes into account the number of existing facilities, k, on a given network that is different from the P-median problem and will be referred to as the $P-k$ median problem. The solution to this combinatorial median problem of finding $P-k$ more facility locations is determined from the minimum cost associated with the weighted distances of the demand points from the candidate facility locations as well as the existing facilities. The computational complexity of the P-median problem extends to the $P-k$ median problem where the number of combinations of facility locations is also polynomial in n for constant values of P and k where $n$is the number of demand points ([4] and [20])

Ideal for the complexity of the P-median and the $P-k$ median problems is the use of genetic algorithms where the natural selection of solutions under controlled and well understood conditions is developed [15]. The concept of genetic algorithms came from the Darwinian theory of evolution and natural selection. Its two primary processes, namely, natural selection and sexual reproduction, determine the survival and reproduction of better individuals or solutions. Mixing and recombination among the genes of their offspring generate a set of individuals from a very large and complex set of candidate solutions randomly [12].

In recent related studies on the application of genetic algorithm for the solution to the P-median problem, a solution represented as a string of n binary variables is checked for feasibility using a compatibility-based genetic algorithm [7]. The algorithm verifies that exactly a number P of these binary variables representing the locations of the medians (variable value equal to one) are found in the chromosome of the generated offspring upon combination of the parents' strings. Combined parents' strings having more or less than the required P facility locations are considered infeasible. A child is then randomly chosen from among the feasible solutions. Similar to earlier applications, biased parent selection probabilities were used giving higher chances of selection to better fit individuals. In other related studies, a set of P indices corresponding to P candidate locations is used to represent a solution instead of using a string of n binary variables. Each index is a number between 1 to n representing one of the P chosen candidate locations of the solution. The genes of the parents are merged to a maximum of $2P$ indices and reduced to only P locations using a greedy selection algorithm [1] or a merge-drop operator [3]. These studies assumed to restriction on the number of demand points that a facility can satisfy (noncapacitated) and no existing facilities.

In the use of genetic algorithm for solving the $P-k$ median problem, the reduction of a set of $2*(P-k)$ indices to $P-k$ additional facility locations forming a child or solution can be done by applying the concept of *opportunity costs.*

Consider a network with only k existing facilities serving total demand. The minimum cost in serving a unit demand of a node is equal to the shortest distance of the node from among the k existing facilities. A node's demand must be satisfied by the existing facility closest to the node. The weighted distance in satisfying this demand is the product of the demand and the distance of the closest existing facility to the node. A current assignment of the nodes whose demands are satisfied by each existing facility with this corresponding minimum cost is available if not easily obtainable since the facility locations are known.

The addition of $P - k$ more facilities to the same network creates increased chances of finding a serving facility within a shorter distance from a demand node than when there are fewer facilities. The distance of a node from the nearest existing facility, for example, may be greater compared to the distance of the same node from some of the candidate locations. The distances of these candidate locations become relevant in the computation of new costs as they can reduce the current minimum cost when they are made to satisfy the demands of the nodes instead of an existing facility.

In increasing the number of facilities, the best opportunity to decrease the cost associated in satisfying the demand of a specific node is by placing a facility at that same node since the distance of the node from itself is at its minimum of zero. The corresponding reduction in cost is equal to the weighted distance of the node from the closest existing facility. The choice of a single facility location to be added to a network with already k existing facilities ($P = k + 1$), therefore, is the node having the greatest weighted distance from its closest existing facility. Choosing any other candidate foregoes the best opportunity of minimizing cost. The savings lost from foregoing this best opportunity called *opportunity costs.*

The choice of $P - k$ facility locations, however, from among several candidates cannot be evaluated solely from a candidate's capacity to reduce its own weighted distance to zero. The same candidate may also be the closest location for a facility to satisfy the demand of another node. In choosing this candidate, the weighted distance associated in satisfying the demand of the other node is also decreased. The net savings realized in placing a facility at this candidate location is the sum of the reductions in weighted distances associated in satisfying the demands of all the nodes for which the candidate is the closest location for a facility to these nodes. This net savings become the opportunity cost incurred in not placing a facility at that candidate location.

This study proposes that the solutions to the $P - k$ median problem be 1. generated as combinations of $P - k$ location sites from the $n - k$ candidate locations using genetic algorithm due to the combinatorial complexity of the discrete solution set; and 2. the generation of a child solution including $P - k$ indices be the result of an evaluation of the candidate facility locations based on their combined realized savings in satisfying the demand of n nodes in the form of *opportunity costs,* or the cost of the best rejected of foregone opportunity [8].

This study applies the proposed opportunity cost-based modified genetic algorithm to the non-capacitated $P - k$ median problem and is organized as follows: Section 2 presents the integer programming formulation of the P-median problem and the development in the approximations used for its solution. Section 3 introduces the proposed opportunity cost-based genetic algorithm. Section 4 reports some computational results. Lastly, conclusions are given in Section 5 .

2. THE P-MEDIAN PROBLEM

2.1. The Zero-One P-median Problem formulation

The P-median problem requires the cost minimization in serving the demands of a network by finding the location of P facilities. In this class of facility location problems, "the cost (benefit) associated in satisfying the demand of a node from a facility increases (decreases) gradually with the distance between the demand node and the nearest facility," Daskin [4]. However, other facility location problems such as covering and center problems assume that a demand node only receives complete benefits from a facility if it is within a certain coverage distance.

From the integer programming formulation proposed by Revelle and Swain [21], the following inputs and decision variables were used by M. Daskin and are adapted as:

The inputs and indices are

$$\begin{aligned}
i &= \text{index of demand nodes ranging from 1 to } n \\
j &= \text{index of candidate locations ranging from 1 to } n \\
h_i &= \text{demand at node } i \\
d_{ij} &= \text{distance between demand node } i \text{ and candidate site } j \\
P &= \text{number of facilities to locate}
\end{aligned}$$

The decision variables are represented as

$$x_j = \begin{cases} 1, & \text{if a facility is to be located at candidate site } j \\ 0, & \text{if not} \end{cases}$$

$$y_j = \begin{cases} 1, & \text{if demand at node } i \text{ is served by a facility at node } j \\ 0, & \text{if not} \end{cases}$$

In formulating the objective function, the total cost of serving all the demand points is to be minimized. The cost associated if a demand point i is served by facility j is given by the product of the demand at point or node i and the distance of node i from facility j. To minimize cost, the facility j closest to the demand point i must serve the demand of i.

In 0-1 model form,

$$\text{minimize } \mathbf{z} = \sum_{i=1}^{n} \sum_{j=1}^{n} h_i d_{ij} y_{ij} \tag{1a}$$

Subject to:

$$\sum_{j=1}^{n} y_{ij} = 1 \qquad \forall \quad I \tag{1b}$$

$$\sum_{j=1}^{n} x_j = P \tag{1c}$$

$$y_{ij} - x_j \leq 0 \qquad \forall \quad i,j \tag{1d}$$

$$x_j = 0,1 \qquad \forall \quad i,j \tag{1e}$$

$$y_{ij} = 0,1 \qquad \forall \quad i,j \tag{1f}$$

The *evaluation* function is the *cost function, z,* the total demand-weighted distance between each of the demand nodes and the nearest facility to each node. The first set of equality constraints is a guarantee that exactly one facility will be able to serve the demand of each node. The second set of equation constraints is a system constraint pertaining to the total number of facilities required to serve the total demand of the network. The third set of inequality constraints eliminates the allocation of demands at node i from source j if a facility at location j is nonexistent. The last two sets of constraints are standard binary constraints for the decision variables.

[14] has shown that for the P-median problem at least one optimal solution can be found in locating each facility on the network's demand nodes. For n demand nodes and no existing facilities, each node is a candidate location for placing a facility or all demand points are potential medians. This makes the number of x_j variables equal to n. From a weighted demand-distance matrix, there is a number of n x n $y_i j$ variables. All in all, these as binary variables, x_i's and $y_i j$'s, has a total of $n + n^2$ for the P-median problem.

2.2. Solution of the P-median Problem Using Heuristic Algorithms

The three classes of heuristic algorithms presented by Daskin [4] for the solution of the P median problem are the myopic algorithm, neighborhood search and exchange algorithms.

The myopic algorithm is a construction algorithm. It builds a solution from scratch and increases the number of facilities by one every time the algorithm is applied. The myopic algorithm finds the location of the P^{th} facility that minimizes cost Daskin [4] given that there are already $P-1$ facilities that either exist or whose locations are predetermined from previous applications of the algorithm. The $P-1$ predetermined locations can no longer be changed when finding the location of the P^{th} facility.

For a 1-median problem ($P = 1$), the solution is found by evaluating the cost function by placing the facility at every candidate location j setting $x_j = 1$ to find z_j. The location with the minimum cost z_j, is the solution to the 1-median problem. This location becomes the location of the first facility in the 2^{nd} median problem ($P = 2$) with the location of the 2^{nd} facility unknown. The remaining n-1 candidates are independently evaluated for the second location with the facility closer to node i between the first facility and the candidate satisfying the demand of the node. The candidate providing the least cost determines the second facility location. These two locations, however, can no longer be changed in finding other myopic medians, ($P > 2$).

In general, consider X_{P-1} to be the set of the known locations of the $P-1$ facilities. The location of the P^{th} facility can be found by selecting the node j that minimizes the sum of weighted distances

$$z = \Sigma_i h_i d\,(i, j\; UX_{P-1}) \qquad (2)$$

where

$d(i, jUX_{p-1})$	=	the shortest distance of the demand node i from the closest facility among the set jUX_{p-1},
jUX_{p-1}	=	the union of the given set of existing locations X_{p-1} and the location of the candidate facility j

It can be seen that by holding the previous $k,\ k+1,\ \ldots,\ P-1$ facility locations fixed as the known elements of the set X_{P-1} each time a single facility location candidate is selected, no combination among the remaining $n-P+1$ candidate locations is considered in any solution. This greatly affects local optimality of the solution of large networks. This results in finding only the best location for the P^{th} facility given the locations of $P-1$ facilities. Daskin [4] shows that the myopic algorithm approximation does not always yield the global optimal solution if there is more than one facility to be added. However, it is useful as an initial solution to the Lagrangian relaxation algorithm and the two other heuristic algorithms, the neighborhood search and exchange algorithms.

The neighborhood search and exchange algorithms are considered improvement algorithms of solutions having already the total number of required facilities, P. These algorithms find new combinations of P locations to further minimize cost.

The exchange or substitution algorithm improves a given solution by finding the best among the $n-P$ unselected candidate locations to replace one of the assigned locations in that solution. The reallocation of demands to determine which locations to exchange is done for each of the P locations, every time choosing the best replacement from among $n-P$ unselected candidates.

The neighborhood search improvement algorithm [18] considers the effect of changing a facility's location on the demand points that are served by that same facility. A starting solution having P facility locations is divided into P non-overlapping neighborhoods with each neighborhood composed of all the nodes i whose demands are satisfied by a common facility j. Each neighborhood is subjected to a first myopic median in an attempt to find if the current facility provides the minimum weighted distance among the demand nodes *within the neighborhood.* This only considers the effect the effect on the members of the neighborhood but accepts the new location without noting the potential benefit to nodes outside each neighborhood [4].

2.3. An Optimization-Based Lagrangian Algorithm for the P-median Problem

Two optimization-based approaches in solving the P-median problem involve relaxations of the zero-one P-median problem. Daskin [4] discusses the relaxation of either constraint (1b) or constraint (1d) using Lagrangian relaxation. Briefly, the changes in the objective function as constraint (1b) is relaxed give the Lagrangian function as

$$\underset{\lambda}{\text{Max}}\ \underset{X,Y}{\text{Min}}\ \sum_i \sum_j h_i d_{ij} y_{ij} + \sum_i \lambda_i \left[1 - \sum_j y_{ij}\right] = \sum_i \sum_j \left(h_i d_{ij} - \lambda_i\right) Y_{ij} + \sum_i \lambda_i \tag{3}$$

For the relaxation of constraint (1d) , the Lagrangian function is

$$\begin{aligned} &\underset{\lambda}{\text{Max}}\ \underset{X,Y}{\text{Min}}\ \sum_i \sum_j h_i d_{ij} y_{ij} + \sum_i \sum_j \lambda_{ij} \left(y_{ij} - x_j\right) \\ &= \sum_i \sum_j \left(h_i d_{ij} + \lambda_{ij}\right) y_{ij} - \sum_j \left(\sum_i \lambda_{ij}\right) X_j \end{aligned} \tag{4}$$

These new functions must be minimized for fixed values of Lagrange multipliers, λ_{ij}. The solution to the Lagrangian problem provides a lower bound and the resulting primal feasible solution using the Lagrangian solution in the P-median objective function provides an upper

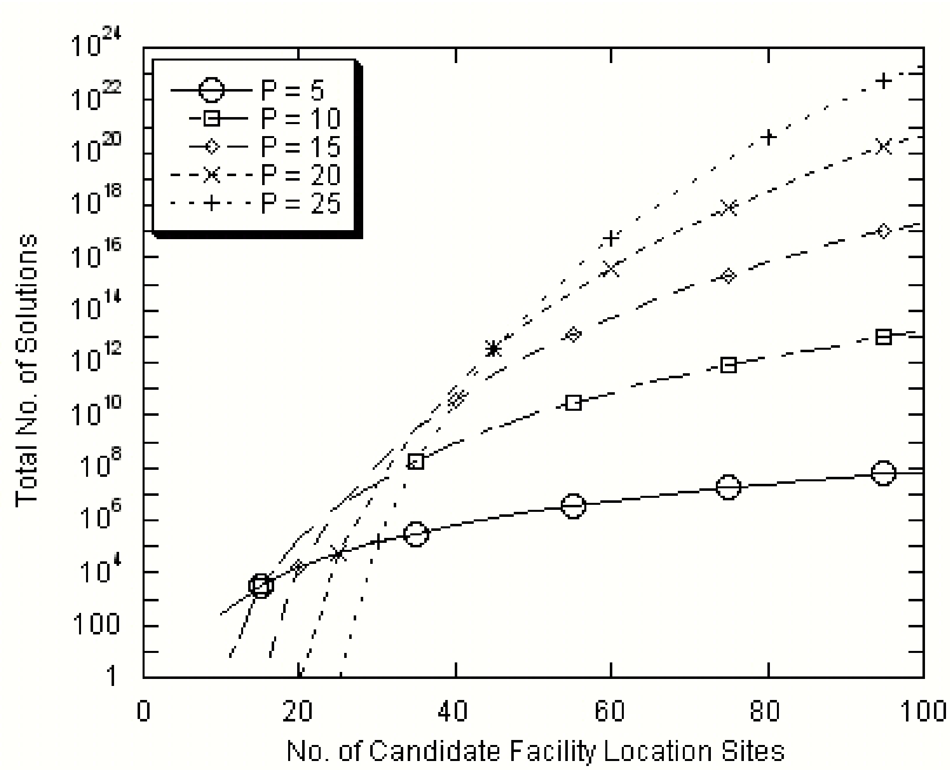

Figure 1. Total Number of Solutions vs P

bound on the P-median problem. The Lagrange multipliers are revised and updated using a standard subgradient optimization procedure until the prespecified number of iterations has been reached or when the upper and lower bounds are equal. For a detailed discussion of the Lagrangian relaxation approach, refer to Daskin [4].

2.4. Computational Complexity and Solution space of the P-median Problem

The total number of solutions of the P-median problem is equal to the number of ways of choosing P out of n given possible locations where a facility may be placed or

$$C_n^P = \frac{n!}{P!(n-P)!} = \binom{n}{P} \tag{5}$$

where

n = number of nodes of the network (or number of positions where a facility can be placed

P = number of facilities to be located

For a constant value of P, the number in (5) is $O(n^P)$ which is polynomial in n. Without any existing facility, a network with 10 demand points and 5 required facilities has a total of

252 solutions. Increasing to 100 and 1000 demand points for a 5 facility-network increases the total number of solutions to more than 75 million and 8.25029E+12, respectively. In terms of computational time, evaluating 10^4 solutions per second will only take 2.5^{-04} second to solve the minimum cost by enumeration if there are 10 demand nodes. Increasing to 100 and 1000 nodes, it will take 752.9 seconds and 22,917 hours (2.616 years), respectively, to evaluate the cost function of every solution. This combinatorial explosion leads to the computational complexity of the P-median problem and is also depicted in Figure 1 below. Figure 1 also demonstrates the faster increase in the size of the solution set as n is increased for higher values of P. Solving by enumeration for any given value of n, the time required to solve any P-median problem is exponential in n Daskin [4] and is given by the value

$$\Sigma_{P=1}^{n} C_n^P = 2^n - 1 = O(2^n) \tag{6}$$

Figures 1 and 2 reflect that the total number of solutions increases with the number of demand points, n, of a given network for the same value of P. For the same value of n (Figure 2), the number of solutions where $P = n/2$ is the greatest and therefore the most difficult to solve by enumeration to find the optimal solution. In fact, the total number of solutions of the P-median problem is polynomial in n for moderate values of P Daskin [4]. From equation (5), whenever P is increased by one, a factor of $(n-P)/(P+1)$is placed for $P < n/2$ while a factor of $(P+1)/(n-P)$ is removed for $P > n/2$. With both these factors positive for the corresponding ranges of P, the number of solutions increase by a factor of $(n-P)/(P+1)$ for $P < n/2$ and decreases by a factor of $(P+1)/(n-P)$ for $P > n/2$ whenever P is increased by 1 (Figure 2).

As the number of demand points, n, increases for constant values of P, there is an increase in the combinatorial and computational complexity of the P-median problem. This has been a major concern for the development of heuristics in solving the P-median problem Teitz [23].

The computational complexity of the myopic algorithm relies in the task of adding one facility at a time. This requires one major comparison of each candidate location to each of the n demand nodes and one multiplication for each of the weighted demand distances to find the cost. That will be $O(n^2)$ elementary operations per additional facility or $O(Pn^2)$ operations in finding P facilities.

In the exchange algorithm, the number of exchanges which can be made is $P \times (n-P)$. The number of elementary operations required to determine if an exchange is better than the current solution involves taking the minimum products of the demands and distances of n demand nodes, so that the number of basic operations is $O(nP(n-P)$ considering all possible exchanges. For values of P almost equal to $n/2$, this number is $O(n^3)$. A network with 5 required facilities and no existing facility, for example, increases in computational complexity from 475 to 94,525 to 9,945,025 for 10, 100 and 1000 demand nodes, respectively. Enumerating almost all possible combinations for the exchanges, again illustrates the complexity of the P-median problem in terms of its combinatorial explosion.

Lastly, the computational complexity of the neighborhood search algorithm can only be measured per iteration as the number of iteration for the solution to converge to its optimal is not definite Daskin [4]. The algorithm performs a 1-median (myopic) search on each of the P smaller non-overlapping neighborhoods per iteration with a total of n demand nodes for all neighborhoods. From the myopic algorithm, each iteration in the neighborhood search algorithm has a computational complexity of $O(n^2)$.

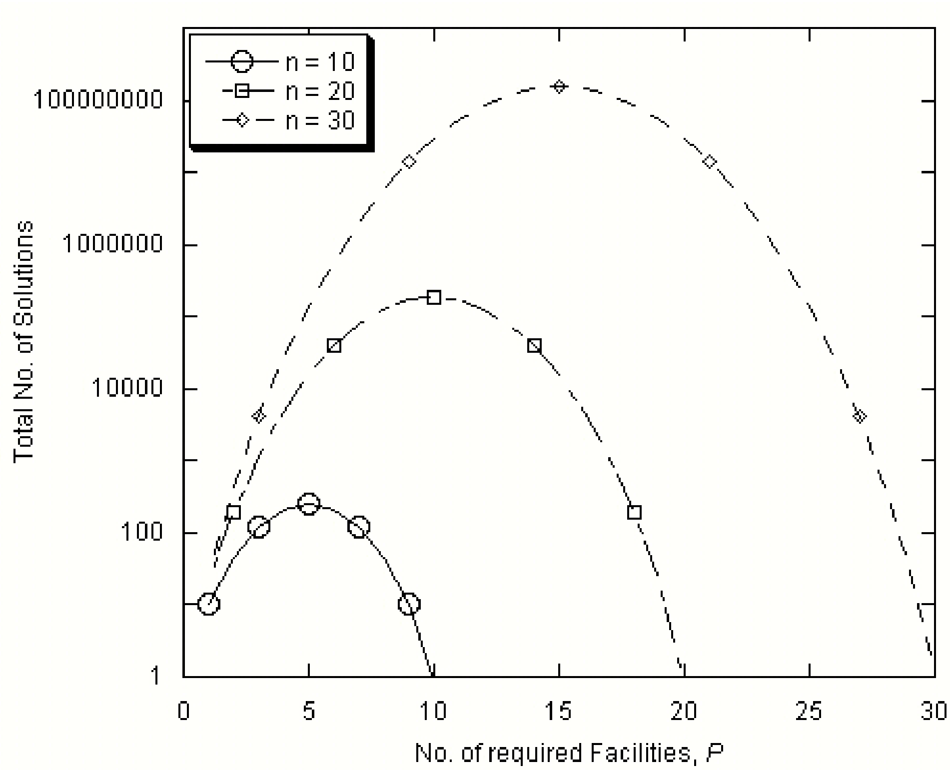

Figure 2. Total Number of Solutions vs n

3. THE $P-k$ MEDIAN PROBLEM AND THE PROPOSED GENETIC ALGORITHM

3.1. Genetic Algorithms

Genetic algorithms are based on the natural processes of selection and evolution. They work well in any search space by randomly generating a set of solutions from a very large and complex set of candidate solutions. The foundation of such algorithms is the controlled evolution of a structured population Lorena [16] from a fixed population size.

Changes in the members of the solution set or sample *population* are made as a result of the use of an evaluation function. The performance of an individual or solution based on this function is referred to as the survival 'fitness' of the individual where the individuals that perform better than others are assigned greater chances of survival. As a result, better fit individuals reproduce to create offspring/s that will be part of future generations of solution sets. Certain traits or characteristics are inherited and may be mutated randomly until a more suitable solution has been found.

In this study, the evaluation function is the cost function z that minimizes the sum of weighted distances. While the traits of an individual are the location indices j corresponding to x_j values equal to one where the $P-k$ facilities must be placed.

Recall that X_{P-1} from the myopic algorithm is the set of known locations of $P-1$ facilities. Since the indices j are locations and x_j is binary, then the indices for which $x_j = 1$ complete a combination of locations for the required P facilities defining an individual or solution in the proposed genetic algorithm. The solution in the form $X_p = [j_1, j_2, ..., j_k, j_{k+1}, j_{k+2}, ..., j_P]$ includes the k locations of the existing facilities, $j_1, j_2, ..., j_k$ and the $P-k$ candidate locations $j_k+1, j_k+2, ..., j_p$. An individual for a 10-median problem, for example, with 50 nodes and fixed facilities at locations j=12, 27 and 40 may be represented as $X = [12, 27, 40, 7, 19, 20, 33, 38, 43, 47]$. This satisfies the required number of facilities of the $P-k$ median problem in every generated parent and child while the variables remain binary. Using indices to represent the solution will be much less tedious than enumerating all the n binary values, x_j, per solution especially if $n >> P$. Other combinations of 10 facility locations for this example may be found but may not always be chosen as members of any of the different generations used by the genetic algorithm.

3.2. *Solution space of the $P-k$ median Problem*

In the $P-k$ median problem, holding k facility locations fixed reduces the x_j variables to a number $n-k$ from the candidate locations while y_{ij} still represents the allocation variables of the demand nodes to n possible medians of fixed and non-fixed facility locations. For a number k of existing facilities, $P-k$ candidates locations must be chosen to complete a solution while the total number of candidate locations is $n-k$. So that the total number of ways of choosing $P-k$ candidates from among $n-k$ is

$$S = C_{n-k}^{P-k} = \frac{(n-k)!}{(P-k)!(n-P)!} \tag{7}$$

where S = total number of solutions
k = number of facilities held fixed
P = total number of facilities to satisfy network demand
n = total number of demand nodes or location sites

Similar to equation (5), the total number of solutions of the $P-k$ median problem is polynomial in n for constant values of P (7). The same combinatorial explosion is therefore experienced by the $P-k$ median problem as shown by the plot of S against n in Figure 3. Given the same number of demand nodes, there are more combinations or solutions at lower values of k. Moreover, for smaller values of k, there is a relatively faster increase in the number of solutions as the number of demand nodes increases.

As explained, it is expected in the $P-k$ median problem that for the same number of existing facilities, k, the total number of solutions is greatest when as $P-k$ is half of $n-k$ for different values of n (Figure 4).

3.3. *Generation of the Initial Members of the Population*

The representation of traits in the generation of the members of the initial population of solutions and population size are important factors to consider in using genetic algorithms.

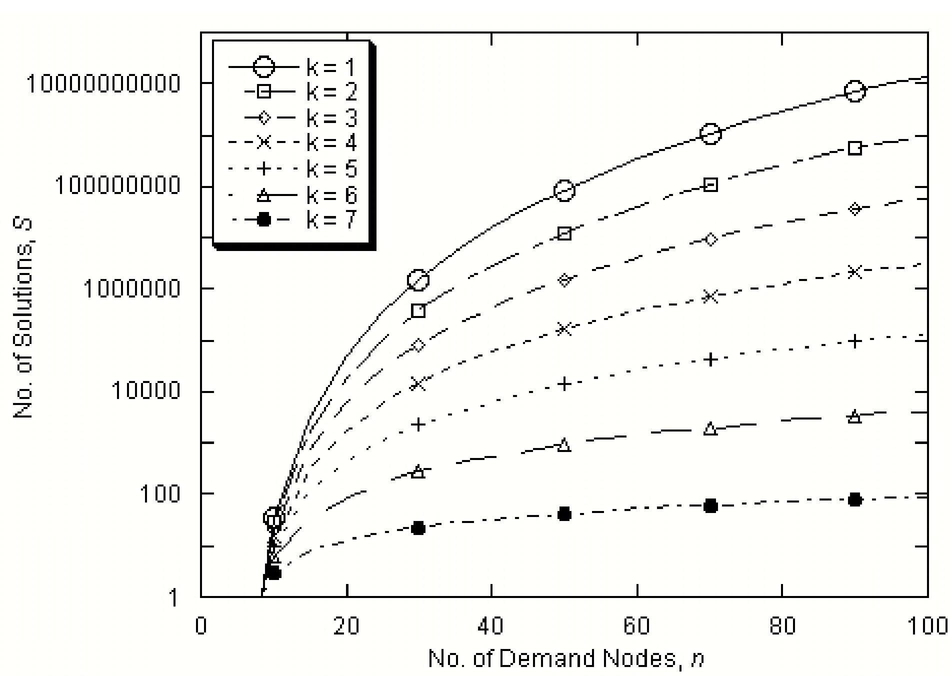

Figure 3. Total Number of Solutions, S vs Number of Demand (P=8)

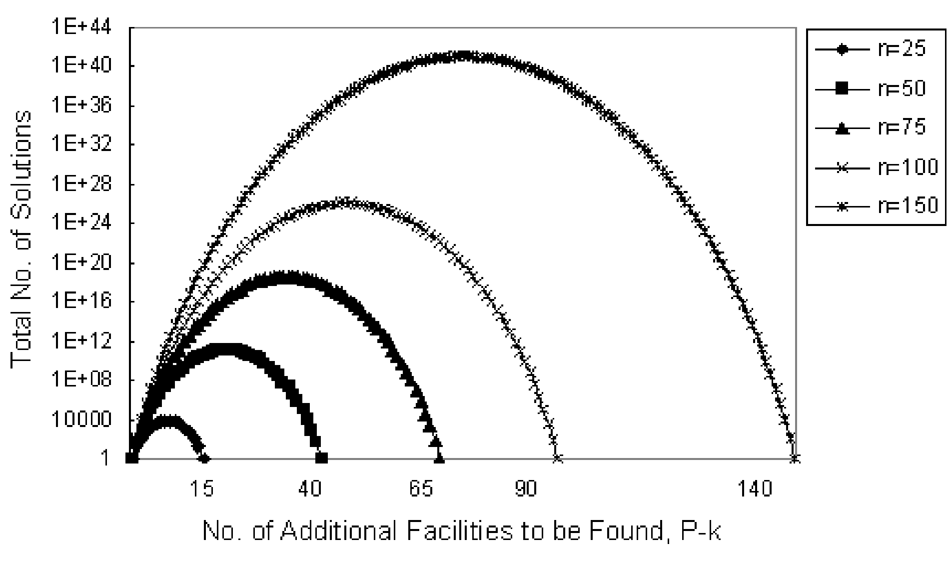

Figure 4. Total Number of Solutions S as P-k increases at k = 10

Achieving genetic diversity is inversely proportional in relation to the speed of the generation of initial members of the population. Also, the number of members comprising a population is dependent on the size of the solution space. Every trait being represented in the initial population helps prevent premature convergence.

Alp [1] used a formula for the generation of population sizes and initial population given by $D(n,p) = max\left\{2, \left[\frac{n}{100} \cdot \frac{lnS}{d}\right]\right\} \times d$ where D is the population size from the P-median problem of n nodes and P required facilities. d is the smallest integer greater than or equal to the ratio of n over P corresponding to the minimum number of members representing each gene in the initial population.

Modifying the equation for the $P-k$ median problem, the population size is

$$D(n-k, P-k) = max\left\{2, \frac{n-k}{100} \cdot \frac{lnS}{d}\right\} d \tag{8}$$

$$\text{and } d = (n-k)/(P-k) \tag{9}$$

The lnS term used provides a slow increase in the value of D as a proportion of S (Figure 5).

Genetic diversity is supported by the uniform distribution of traits across the initial D members of the population. D is first divided into $C = D/d$ groups of individuals or members where each group is assigned a group number, c, ranging from 1 to C. Then, all the $n-k$ candidate locations are uniformly assigned to each group combining $P-k$ indices with the fixed k locations to create the individuals belonging to each group. The group number corresponds to the interval between the j indices taken to complete the combination of locations identifying an individual.

The d members of the first group ($c = 1$) have $P-k$ indices distributed to each member according to the sequence $j, j+1, j+2, j+3...$ For $c = 2$, the distribution of indices is according to the sequence $j, j+2, j+4, ...$ and the pattern goes for other c values. The $P-k$ indices are combined to the indices of the k existing facilities. If the indices corresponding to existing facility locations fall into the sequence of a group, they do not disrupt the sequence. Finally, if the sequence has exhausted all the $n-k$ candidates but has not completed all the members of the same group, it starts again from the index value following the first index value last used by the sequence from the candidate locations.

A 52-node network with fixed facilities occupying locations j= 12, 27 and 40 and with P=10 will have for its initial population the members of two groups ($C = 2$) to be

$$c=1 \quad \begin{array}{l}
[\ 1,2,3,4,5,6,7,12,27,40\ ] \\
[\ 8,9,10,11,12,13,14,15,27,40\ ] \\
[\ 12,16,17,18,19,20,21,22,27,40\ ] \\
[\ 12,23,24,25,26,27,28,29,30,40\ ] \\
[\ 12,27,31,32,33,34,35,36,37,40\ ] \\
[\ 12,27,38,39,40,41,42,43,44,45\ ] \\
[\ 12,27,40,46,47,48,49,50,51,52\ ]
\end{array}
\quad c=2 \quad \begin{array}{l}
[\ 1,3,5,7,9,11,12,13,27,40\ ] \\
[\ 12,15,17,19,21,23,25,27,29,40\ ] \\
[\ 12,27,31,33,35,37,39,40,41,43\ ] \\
[\ 12,27,40,45,47,49,51,2,4,6\ ] \\
[\ 8,10,12,14,16,18,20,22,27,40\ ] \\
[\ 12,24,26,27,28,30,32,34,36,40\ ] \\
[\ 12,27,38,40,42,44,46,48,50,52\ ]
\end{array}$$

3.4. Parent Selection and the Cost Function

In most applications of the genetic algorithm for the solution to P-median problems, the selection procedures for finding candidate parents for the generation of new solutions require

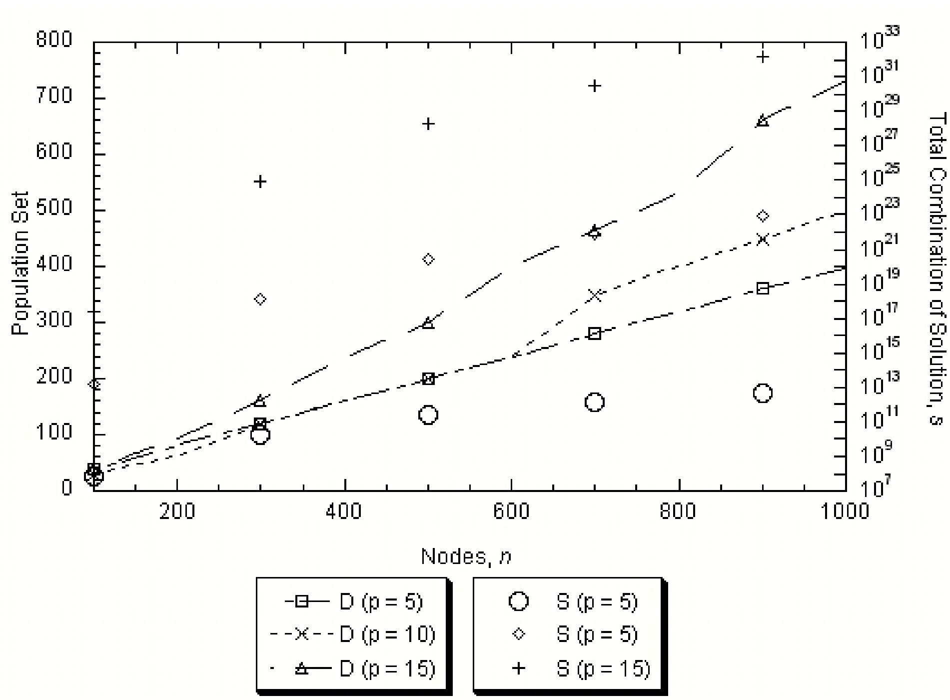

Figure 5. Population Size (D) and Total Number of Solutions (S)

the ranking of the members of the population according to fitness value such as the ranking-based and the roulette wheel selection by Dvorette [7], among others. In the work of Alp [1], however, no biased selection mechanism was applied.

The chance or probability of an individual to be selected as parent is directly proportional to its fitness or evaluation function, z. Higher probability may be associated to a solution having a lower cost function value or a higher return function value as the case may be. The z values are computed for all the members of the population. A biased random selection or giving higher probability to a better value of z is called the roulette wheel type of parent selection. With the minimization objective function for the $P-k$ median problem, the probability assigned to each member, X_{Pi}, is obtained by first taking the inverse of the objective function z_i, z_i^{-1}. Then the sum of the inverses of the entire population is computed and the probability for selection of a member X_{Pi} is given by

$$P[X_i] = \frac{Z_i^{-1}}{\Sigma_{i=l}^{D} Z_i^{-1}} \tag{10}$$

The fitness values of the members of the initial population need not be arranged in ascending order so as to decrease computational time. Bias on the fitness value is achieved by associating the obtained probabilities of each solution using equation (10) to the fraction of the range of the random numbers assigned to each solution. The random numbers are distributed to the members of the population according to these probabilities. The solution is chosen as parent provided that the random number drawn for parent selection falls within the allocated random numbers of the solution. The best and worst values of z are kept as basis for the behavior of the current generation and the acceptance of a candidate offspring into the new generation. The probability distribution changes as cost functions change with new members constituting the population.

The random number generator used for the selection of parents was taken from minimal standard generator and is implemented in this piece of code,

```
double random (double seed)
{
const a = 16087;
const m = 2147483647;
seed = (a * seed) mod m;
return seed/ m;
)
```

3.5. Child Generation

3.5.1. Merging of Parents' Traits From the selected parents, a new individual may be formed through *crossover* or the swapping of traits between two parents. Single point crossover is the most commonly used solution where a 'child' inherits a portion of each of its parents' traits.

Each pair of *parent* solutions in this study generates a maximum of $2 * (P - k)$ combined candidate location indices or traits and k fixed locations as a result of the k existing facilities. This is when there is no duplication in the indices coming from both parents. Should duplication exist, the child will inherit the duplicated index/indices.

The reproduction of a new individual as well as its survival depends on

1. the fitness of the candidate child to reproduce, in turn, a better fit offspring;
2. retention of the locations of the fixed facilities; and
3. choosing exactly $P - k$ locations among the combined traits of the parents.

3.5.2. Reduction of Location Indices to P The combined unduplicated location indices from the parents are to be reduced to the number of required indices to fill the remaining P locations using opportunity costs. This is to generate the best offspring from each parent-pair. With at most $2*(P-k)$ unduplicated indices from both parents, these form a subset of the total $n-k$ candidate locations of the given network. The additional $P-k$ facility locations of an offspring will come from this subset. Further, the pair of parents is chosen so that the parents are not exactly the same individuals so as to improve the current population.

The cost of having only k facilities will be reduced when a location index j chosen from the subset of candidates and added to the network is better than any of the existing facilities with respect to the distance from a demand node i (where i is not necessarily equal to j). As explained earlier, the decision not to include a candidate j in the $P - k$ facilities when it is closest to the demand node i among all other candidates as well as the existing facilities foregoes the best opportunity to lower costs and incurs an *opportunity cost*. If an existing facility is the nearest facility to a node i, no opportunity to lower cost is lost in not choosing any candidate j to satisfy the demand of i or $y_{ij} = 0$ and $x_j = 0$ for all candidates. Lastly, choosing a candidate j that is not any nearer to another demand node i than any of the existing facilities only decreases cost by satisfying its own demand. Notice that finding the best opportunity in satisfying the demand of each node given a solution can be made without reference to the required number of additional facilities.

In quantifying the capacities or the opportunity foregone by removing a candidate from the subset of candidate location indices, the following notations are applied:

X_{P1} = set of location indices of parent 1 including existing facilities

X_{P2} = set of location indices of parent 2 including existing facilities

X = a subset of location indices from$X_{P1} \cup X_{P2}$

X_P = set of location indices of the resulting child or individual having reduced X into P locations indices

OC_j = opportunity cost of candidate location index j

$j*$ = the location index with the lowest opportunity cost among candidates; the location index removed as a candidate from X

$d(i, X)$ = the shortest distance of the demand node i from the closest facility among the set, X

$b(i, X)$ = the location index j from the set X corresponding to $d(i, X)$

$\tilde{X}$ = the set of location indices X excluding the location index $b(i, X)$

$d'(i, \tilde{X})$ = the shortest distance of the demand node i from the closest facility among the set, $\tilde{X}$

$b'(i, \tilde{X})$ = the location index j from the set $\tilde{X}$ corresponding to $d(i, \tilde{X})$

$X*$ = the reduced subset of location indices X after the removal of $j*$

The union of the sets of location indices from the two parents, $X_{P1} \cup X_{P2}$, represents the existing facility locations and the range of choices for the locations of the $P - k$ facilities for the resulting child solution. All duplicated location indices are automatically included in the set of indices of the generated child so that they are no longer evaluated as candidates. The number of candidate locations from this union is equal to $2*(P-k-r)$ where r is the number of duplications. The subset of locations indices, X, starts with elements from this union and is reduced every time a location index is eliminated as a candidate. The reduction criterion is the candidate location's opportunity cost and terminates when there are only P remaining indices in the set X corresponding to the generated child.

If the set X is reduced by an element, the opportunity of incurring the least cost in satisfying the demand of node i is foregone only if the closest facility to node $i, b(i, X)$, is the one removed from the set X. Otherwise, the option to have the least cost to satisfy the demand of node i from the closest facility is available and no opportunity cost is incurred.

Upon the removal of $b(i, X)$, the next best opportunity from the set $X, B'(i, \tilde{X}$, is still present in the choices for the locations of the $P-k$ facilities of the child. Then the opportunity foregone in not choosing the closest facility to node i in satisfying its demand is the additional cost associated in choosing the next closest facility, $b'(i, \tilde{X})$. This additional cost is equal to $d'(i, \tilde{X}) - d(i, X)$. It is also the opportunity cost of not choosing the location index $j = b(i, X)$ that best satisfies the demand of node i.

The demand nodes have different location indices $b(i, X)$ corresponding to the best choice of facility location j for each node. The evaluation of the opportunity cost of candidate location, j, is therefore considered for all the demand nodes having j as the closest facility to these nodes or for all i's satisfying $b(i, X) = j$. The opportunity cost of not putting a facility at a location index j is equal to

$$OC_j = \sum_{b(i,X)=j} d'(i, \tilde{X}) - d(i, X) \tag{11}$$

The location index, $j*$, that has the lowest value of opportunity cost OC_j* is considered the worst candidate as a facility location among the given candidates and will no longer be part of the $P - k$ additional facilities of the resulting child of the given parents X_{P1} and X_{P2}. And the removal of a location index as candidate creates the shorter list of indices, $X*$.

The computation of opportunity costs after a first location index has been removed becomes less complicated since the best, $b(i, X)$, and next best, $b'(i, \tilde{X})$, locations of facilities for satisfying the demand of node i have already been identified. Further, the removal of an index may or may not affect the previously computed opportunity costs of the remaining candidates or indices in the set $X*$. The opportunity cost computed for a candidate j only changes and increases when

1. the removal of $j*$ from X makes j the new best candidate location from $X*$ for the demand of any node i previously allocated to $j*$
2. j is the best candidate for the demand node i and $j*$ is the next best candidate location for the same node *prior* to $j*$'s removal as a candidate

In the first case, $b(i, X) = j*$ and $b'(i, \tilde{X}) = j$. The demand node/s, i satisfying this condition is/are sure to minimize cost to satisfy its/their demands if it/they are reallocated to j after its/their best alternative $j*$ has been removed. An increase in the opportunity cost of candidate j is realized since not retaining j now as candidate foregoes the opportunity of having this new best source satisfy the demand of node i previously allocated to $j*$. Hence, for a demand node i,if $b(i, X) = j*$, the next best choice $b'(i, \tilde{X}) = j$ from X becomes the best choice $b(i, X*)$,in the remaining set of indices $X*$ that satisfies the demand of i. A new index, $b'(i, \tilde{X}*)$, will represent the next best location for i from $X*$. The increase in opportunity cost of index j is equal to $d'(i, \tilde{X}*) - d(i, X*)$.

In the second case, the index j remains the best source in the smaller set of candidates, $X*$, or $b(i, X) = b(i, X*) = j$ for the demand of i upon the removal of $j*$. But the opportunity cost of j will still increase since not retaining j forces the demand of i to be satisfied by another candidate from $X*$ that is farther from j than $j*$.

3.5.3. Computational Example of the Child Generation using Opportunity Costs The computation of opportunity costs is illustrated using the two parent set solutions:

$$X_{P1} = [1, 2, 3.4, 5] \quad X_{P2} = [1, 2, 6, 7, 8].$$

i, j	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
1	**0**	810	875	465	915	320	457	580
2	520	**0**	385	785	430	346	543	390
3	700	235	**0**	560	750	320	643	**228**
4	**120**	200	170	**0**	255	185	330	465
5	269	380	727	**198**	**0**	510	609	425
6	700	520	400	395	**165**	**0**	547	250
7	1100	1080	630	459	**270**	550	**0**	490
8	1230	970	650	410	814	730	**360**	**0**
9	841	624	1405	562	471	**285**	657	**312**
10	623	**390**	**150**	600	520	462	487	546

Table I. Demand-Weighted Distances Between Demand Node i and Facility Location j

The duplicated location indices are 1 and 2 which could both reflect existing facilities or merely duplications in candidate facility locations. In any case, they will both be included in the set of indices of the resulting child. The resulting subset of location indices prior to the elimination of any unduplicated index as a candidate is $X = [1, 2, 3, 4, 5, 6, 7, 8]$ from a network with 10 demand nodes. In addition to the 2 duplicated indices, 3 more indices are to be selected from the 6 unduplicated candidate locations to complete the 5 required facilities for the child.

The opportunity costs of the 6 candidate locations are computed from the demand-weighted distances for each demand node i satisfied by each facility j (Figure I).

The best sources for the demands of nodes 1 to 8 are from each of their own locations, or for i=1 to 8, $b(i, X) = j$ whenever $i = j$. The best, $b(i, X)$, and second best, $b'(i, \tilde{X})$, sources for the demands of each node are shown in bold numbers on Table I except for nodes 1 and 2 where the existing facilities are located. Their second best choices are not significant to identify since the opportunity cost in not choosing any candidate locations to satisfy the demands of nodes 1 and 2 is **zero**. If an existing facility j is or will become the best facility to satisfy the demand of any other node i, the opportunity to choose location j to satisfy the demand of i will always be available. Hence no additional opportunity cost is incurred in the removal of any candidate location once the demand of a node is satisfied by an existing facility. The best and second best facility locations to satisfy the demands of nodes 3 to 10 are shown in table 2. The last column shows the corresponding opportunity costs when the best facility locations are not chosen to serve the demand of each demand node. Table 3 summarizes these opportunity costs according to the indices of the location candidates.

The candidate location that has the least opportunity cost from Table III is index $j* = 4$. It will be removed from the list of candidate locations for the resulting child from among the 6 candidates to incur the least opportunity cost to the network. The removal of index 4 as a candidate affects only node 4 where it was the best source for its own demand, $b(4, X) = j* = 4$. The next best source for node 4, as shown from Table II, is the existing facility 1 which now becomes the new best source to satisfy the demand of node 4, $b'(4, X) = 1 = b(i, \tilde{X})$. Since this is not a candidate node but an existing facility the removal of another location index as candidate among the remaining indices 3,5,6,7 and 8 will no longer affect the demand of node

i	$b(i,X)$	$b'(i,X)$	$d'(i,X\sim) - d(i,X)$
3	3	8	228
4	4	1	120
5	5	4	198
6	6	5	165
7	7	5	270
8	8	7	360
9	6	8	27
10	3	2	240

Table II. Best and 2^{nd} Best Facility Locations

$b(i,X) = j$	OC_j
3	228 + 240 = 468
4	***120***
5	198
6	165 + 27 = 192
7	270
8	360

Table III. Opportunity Costs of Candidates

j	OC_j
3	468 + 0 = 468
5	198 + 0 = 198
6	192 + 0 = ***192***
7	270 + 0 = 270
8	360 + 0 = 360

Table IV. Opportunity Costs of Remaining Five Candidates

4. So that no additional opportunity cost is incurred in removing the next candidate which is index $j* = 6$ as shown in Table IV.

The removal of index 6 as a location candidate leaves the set of indices $X* = [1, 2, 3, 5, 7, 8]$. This affects the demands of nodes 6 and 9 (Table II) where their demands will have to be satisfied by their second best facility locations namely, $b'(6, X) = 5$ and $b'(9, X) = 8$, respectively. These location indices, 5 and 8, become the new best facility locations for the corresponding demands under the set $X*$. Unlike the removal of index 4, however, facility locations 5 and 8 do not have existing facilities. The new second best facility locations for the demands of nodes 6 and 9 are the location indices $b'(6, \tilde{X*}) = 8$ and $b'(9, \tilde{X*}) = 5$, respectively. The additional opportunity lost from the removal of the new best facility locations 5 and 8 as candidates are

j	OC_i
3	$468 + 0 = 468$
5	$198 + 85 = 283$
7	$270 + 0 = \boldsymbol{270}$
8	$360 + 159 = 519$

Table V. Opportunity Costs of Remaining Four Candidates

$$d'(6, \tilde{X*}) = 250 - 165 = 85$$
$$d'(9, \tilde{X*}) - d(9, X*) = 471 - 312 = 159.$$

There are no changes in the opportunity costs in removing indices 3 and 7 since neither is a new best source for the demands of the nodes left unsatisfied by the removed index $j* = 6$. With the opportunity cots of these candidates remaining the same, the process of determining which among the remaining candidate locations a facility must be placed is simplified especially in larger networks where $n >> P$.

Candidate facility location 7 has the least opportunity cost (Table V) which leaves the indices 3, 5 and 8 to be combined to the duplicated indices 1 and 2 forming the resulting child $X_P = [1, 2, 3, 5, 8]$. This child solution is ready for an evaluation of its acceptance into the population of the next generation.

3.5.4. Improvement of the Population A candidate solution or child is accepted as a new member of the next generation provided that its fitness value is better than the worst fitness value of the current population. If the child is accepted, new worst and best values of z are checked for changes in values while parent-selection probabilities of the members of the new population are recomputed. If the generated offspring is rejected, another parent-pair is obtained for the generation of another candidate solution.

3.6. Mutation

Genetic diversity can be maintained using the process of mutation where a trait or candidate location index is randomly selected and changed as necessary. The chance of its occurrence in nature is estimated by Skinner, M. at 10%. Lorena [16] introduced their own interchange heuristic for a local search mutation for the capacitated P-median problems. "Mutation alone does not generally advance the search for a solution, but it does provide insurance against the development of a uniform population incapable of further evolution." Holland [15]. Sufficient balance between selective pressure and genetic diversity must be achieved to prevent premature convergence. In the proposed genetic algorithm, mutation rate is taken to be at a 10 percent level.

3.7. Termination

The generation of offspring is terminated until a number of children have failed to replace the best solution of the population. Termination is done when there is no change in the current value of the best solution for $(n - k)/\sqrt{p - k}$ times Alp [1]. Other termination criteria may be used to determine the maximum number of iterations to be implemented. The pseudocode is provided as appendix.

4. RESULTS AND DISCUSSION

This section presents the summary of the solution and performance of the proposed opportunity cost-based genetic algorithm for the $P-k$ median problem along with the solutions provided by the Lagrangian relaxation to the P-median problem and solutions of the myopic algorithm with an applied neighborhood search improvement to the $P-k$ median problem.

4.1. Computational Analysis and Model Application

Using the Lagrangian relaxation approach to solve a P-median problem does not result in the solution of any other median problem Daskin [4]. So that it was not used to solve the $P-k$ median problem of increasing the number of facility locations from k to P. However, to evaluate the performance of the proposed algorithm in terms of the quality of solution against the myopic algorithm with neighborhood improvement, the known optimal solutions of the Lagrangian relaxation approach for various values of P with no existing facilities ($k = 0$) for the Galvao-100 test problem set with 4950 links were used. These also serve as basis to generate the different location indices representing the candidates and the k existing facilities. The data input files can be downloaded from www.bus.ualberta.ca/eerkut/testproblems and were modified to suit input file format in the source code.

The performance with respect to the quality of solution for the addition of $P-k$ locations to the k existing facilities is compared between the proposed genetic algorithm and the myopic algorithm. Recall that the myopic search algorithm allows increases in the number of facility locations. Improvement in the solution of the myopic algorithm was made through the application of the neighborhood search algorithm. The same number of iterations was set for the termination of both the myopic algorithm with neighborhood improvement and the proposed genetic algorithm. Both the proposed genetic and improved myopic algorithms were programmed with the source code of these models compiled and run using the built-in C/C++ compiler in Linux system.

The optimal solution is still based on the solution of the Lagrangian relaxation approach. Having set the same termination criterion for both algorithms, the performance of each of the programmed algorithms was evaluated solely on the values of their evaluation or cost functions for chosen ranges of P and k. With the number of required facilities for the $P-k$ median problem most critical for values of $P-k$ near $(n-k)/2$, only solutions at this critical level were the proposed opportunity cost-based algorithm tested. The specific chosen values of P and k are summarized in Table VI. The $P-k$ median problems were solved using 10 combinations from each set of $P-k$ median problem whenever possible and performed 30 replications per combination.

4.2. Model Performance

TableVI presents the performance summary of the proposed opportunity cost-based genetic algorithm and the improved myopic algorithm for the Galvao-100 test problem set. It includes the best and worst cost functions obtained by the two algorithms, the mean cost functions and

P	**k**	**Optional Solution z***	**Myopic w/ Neighborhood Search**				**Proposed Opportunity Cost-based Genetic Algorithm (GA)**			
			Z_{ave}	%ave diff from z*	Best solution	Worst solution	Best solution	Worst solution	Z_{ave}	%ave diff from Z*
10	4	4426	4455.6	0.668%	4426	4510	4426	4506	4472.929	1.060%
	5		4447.0	0.474%	4426	4510	4426	4488	4460.333	0.780%
	6		4429.4	0.077%	4426	4440	4426	4440	4430.667	0.110%
	7		***4446.8***	***0.470%***	***4426***	***4594***	***4426***	***4440***	***4429.111***	***0.070%***
15	4	3893	3923.7	0.788%	3896	3946	3893	***3983***	3912.500	0.501%
	5		3912.1	0.490%	***3893***	3935	3897	3918	3913.200	0.520%
	6		3911.6	0.478%	3895	3957	3897	3947	3910.700	0.455%
	7		3919.5	0.681%	3896	3996	3899	3951	3908.700	0.403%
	8		3915.2	0.570%	3894	3994	3898	3937	3903.300	0.265%
	9		3922.2	0.750%	3895	3994	3897	3923	3902.600	0.247%
	10		3928.4	0.909%	***3893***	***4087***	***3893***	3898	3895.100	0.054%
	11		3940.5	1.220%	***3893***	***4087***	***3893***	3927	3897.900	0.126%
20	7	3565	3614.0	1.369%	3580	3701	3575	***3607***	3577.867	0.361%
	8		3614.0	1.363%	3586	3702	3581	3596	3582.800	0.499%
	9		3614.0	1.363%	3586	3702	3585	3598	3587.067	0.619%
	10		3616.0	1.442%	3579	3725	3572	3589	3578.600	0.380%
	11		3613.0	1.335%	***3574***	3725	3582	3593	3583.933	0.531%
	12		3618.0	1.477%	3577	***3733***	3582	3606	3582.800	0.499%
	13		3617.0	1.459%	3575	***3733***	3581	3592	3582.267	0.484%
	14		3606.0	1.144%	3575	3706	3575	3589	3577.267	0.344%
	15		3617.0	1.459%	3575	3706	***3572***	3598	3578.000	0.360%
25	9	3291	3326.7	1.805%	3315	3342	3310	3336	3314.467	0.710%
	10		3326.8	1.088%	3318	3342	3311	3344	3316.667	0.780%
	11		3325.6	1.051%	3316	3342	***3305***	3344	3312.067	0.640%
	12		3330.0	1.185%	3312	3371	3319	3324	3319.667	0.871%
	13		3330.0	1.170%	***3311***	3373	3309	***3322***	3310.733	0.600%
	14		3330.0	1.170%	***3311***	3370	3315	3323	3316.600	0.778%
	15		3342.0	1.557%	3316	***3426***	3315	3335	3315.267	0.737%
	16		3328.4	1.288%	3320	3373	3316	3355	3319.600	0.869%
	17		3328.2	1.282%	3318	3371	3315	3333	3318.400	0.833%

Table VI. Galvao-100 Test Problem Performance

the difference of their solutions as a percentage of the optimal values of $z*$. From the results of 30 $P-k$ median problems with 10 or more required facilities and using the same combination of indices for the locations of existing facilities (Table VI), only at 4 instances has the myopic algorithm produced better results in its addition of $P-k$ locations for the given network than the proposed genetic algorithm. This can be observed from the lower average evaluation function values, Z_{ave}, for the first three values of k used for $P=10$ and at $k=5$ for $P=15$. Both algorithms were able to provide the optimal solutions at lower values of P decreasing in frequency as P is increased and k is decreased. However, at $P=5$, the myopic algorithm obtains the optimal solution at any value of k and at any combination of the k number of existing facilities. The best among the best and the worst among the worst values obtained for each case of P considered are highlighted in Table VI. For the 30 problems solved the best among the best and the worst among the worst evaluation function vales for different P's from Table VI show that the proposed genetic algorithm is never inferior to the improved myopic search algorithm. This is also the case when the best among the worst for each value P is compared between these two algorithms. In fact, for $P=20$, the worst solution of the genetic algorithm for each k value considered is always better than the worst solution obtained by the improved myopic algorithm. It is only the comparison of the worst among the best values that neither algorithm completely reigns superior over the other. The %difference among these values is within 0.077% of the higher value.

It can be deduced from Table VII that the same number of required facilities, P, there are no drastic changes in the percentage difference from the optimal solution as k is increased. It

	P, No. of Required Facilities				
k	**10**	**15**	**20**	**25**	**Mean % difference**
5	0.780	0.520			0.650
6	0.110	0.455			0.283
7	0.070	0.403	0.361		0.278
8		0.265	0.499		0.382
9		0.247	0.619	0.710	0.525
10		0.054	0.380	0.780	0.405
11		0.126	0.531	0.640	0.432
12			0.499	0.871	0.685
13			0.484	0.600	0.542
14			0.344	0.778	0.561
15			0.360	0.737	0.549
16				0.869	0.869
17				0.833	0.833
mean % difference	0.320	0.296	0.453	0.758	

Table VII. Percent Difference from the Optimal Solution

only has a -0.041% average decrease in this percentage each time k is increased. This reflects that although an increase in the total number of combinations or solutions, S, is expected as k decreases as shown in Figure 3, the fraction of the solution space covered by the generated population D is sufficient in providing near optimal results for this problem instance. This relationship between S and D has earlier been shown on Figure 5. Moreover, the considerd values of P and $P-k$ for the application of the $P-k$ median problem have an average increase in the percent difference from the optimal value of the cost function of only 0.241% each time P is increased by 5 for constant values of k.

Figure 6 illustrates that for the chosen $P-k$ median problem of $P = 10$ and $k = 6$ over 30 replications, the proposed genetic algorithm found solutions with regularity over different locations of existing facilities. That is, for different starting combinations of indices for the existing facilities, the proposed algorithm for the 8 out of the 10 combinations shown in Figure 7 converged at almost the same% difference from the optimal solution. Only in combinations 4 and 5, at one replication each from these combinations has the evaluation function of these runs produced a noticeable % difference from the average evaluation function of the problem instance. The solution of the proposed genetic algorithm to the $P-k$ median problem illustrated in Figure 7 has not been greatly affected by the considered combination of locations or indices of the existing facilities.

The important feature of the opportunity cost-based genetic algorithm of improving the solution set quickly is depicted in Figure 7 where the fitness value for $P = 10$ has improved by almost 96% after the 20^{th} iteration. The average cost function for different k values (Figure 8) has declined faster for lower values of k even if a larger number of solutions, S, is associated with a lower k value for the same number of required facilities P. So that all values of k from Figure 8 converged at almost the same number of iterations even though the size of their solution spaces vary. And the average cost function improves from an 11.8% at 25 iterations

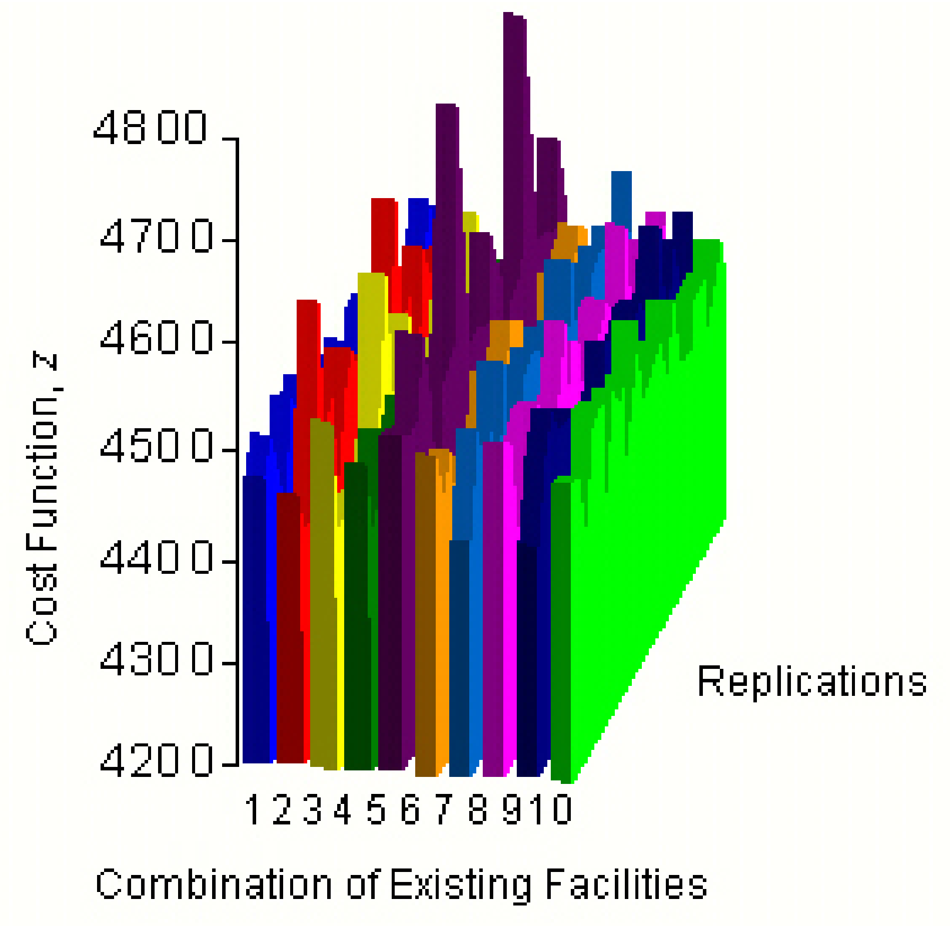

Figure 6. Cost Function for P=10, k=6

to almost within 5.06% difference from the optimal solution after the first 50 iterations.

The slow decrease in the average fitness values after 50 iterations indicate a reduction in the variance of the sample population as solution quality improve near the Lagrangian optimal solution. Most importantly, with only the myopic algorithm among the available heuristic algorithms capable of increasing the number of facilities from $P-1$ to P and from k to P, the proposed algorithm with its average of 0.52% difference from the Lagrangian optimal solution outperforms the myopic algorthm's average of 1.05% difference for the 30 problems presented in this study.

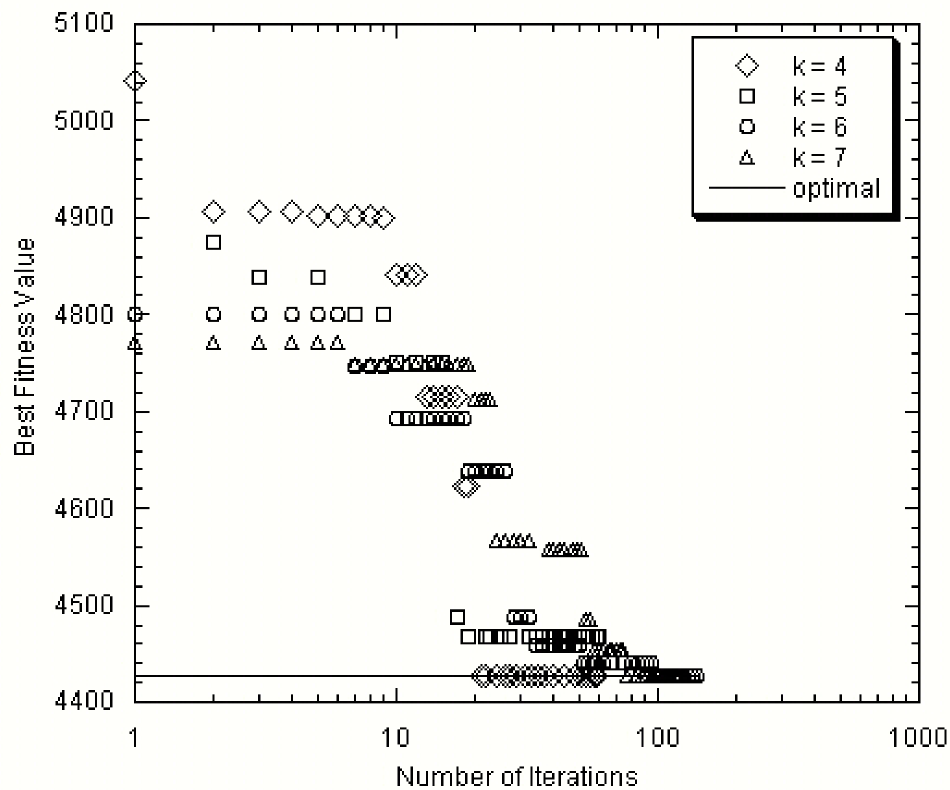

Figure 7. Convergence to the Optimal Solution with *different* k values at $P = 10$

5. CONCLUSIONS

The use of the opportunity cost-based genetic algorithm in solving the combinatorial problem of assigning $P - k$ facilities over the $n - k$ candidate locations for the 30 instances presented has proven to generate better solutions than the improved myopic algorithm. The proposed algorithm can be very useful for systems constrained by the locations of existing facilities and for which the network is relatively large so as not to cover all possible solutions nor predict solution behavior. The computation for determining which locations will incur greater opportunity costs is simple. Weighted demand-distances need not be computed for nodes not affected by previous decisions since knowing the best and second best location alternatives provide easier access to information on minimum costs.

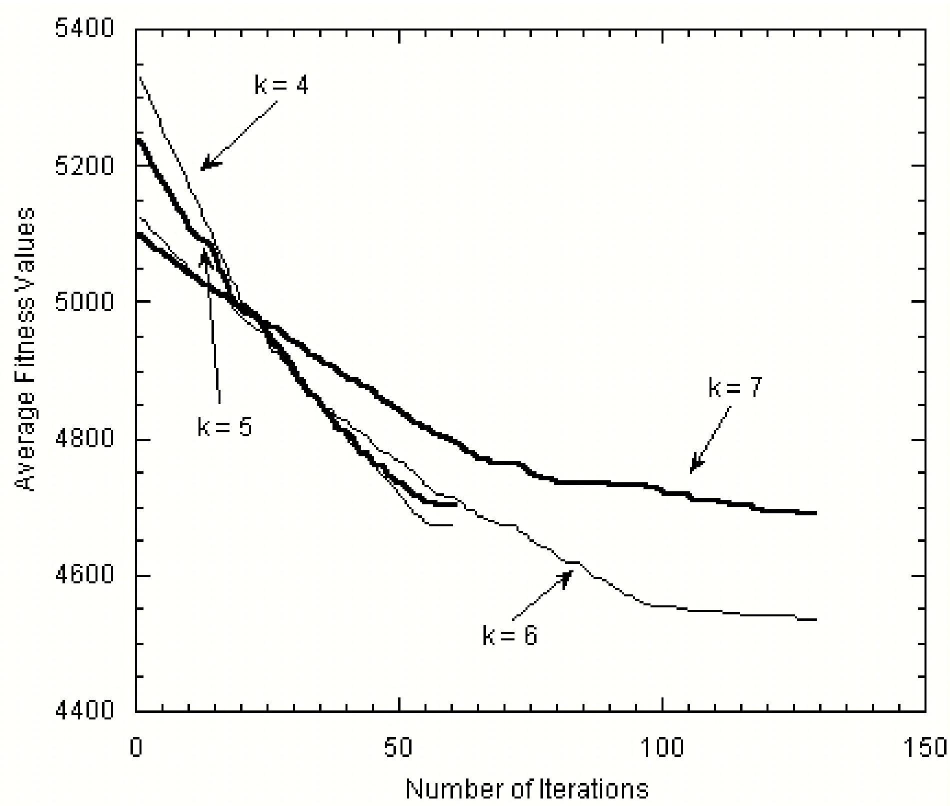

Figure 8. Average Fitness Values per Generation

6. RECOMMENDATIONS AND AREAS FOR FURTHER STUDY

Further improvement on this research may include unidirectional network problems where a number of nodes may have no link or the flow is only in one direction; or two-directional network problems where the demand distances differ depending on which will be the demand source between two nodes. With the cost/benefit still dependent on the associated distance between a facility-demand nodes pair, the demand-distance matrix may be modified to incorporate improbable links for undirectional flow or differentiate associated distances for two-directional network problems. The proposed opportunity cost-based child generation can be made into a decision making tool should the opposite of the problem arise such that a need to reduce instead of increase the number of facilities is required. In which case, the reduction of $P - k$ indices to the desired number of facilities less than P requires the handling of only one child (from the current system with P existing facilities) and opportunity costs associated with each of thye P facilities may be used to decrease the number of these existing facilities. Sensitivity analysis as the demands of some nodes change may also be another area of application using unit opportunity costs on the candidate locations to evaluate critical levels of demand when it is necessary to change the current assignment of demands to facilities.

REFERENCES

1. Alp, E., Erkut, E. and Drezner, Z., 2003. "'An Efficient Genetic Algorithm for the $P-$Median Problem,"' Annals of Operations Research, 122, pp. 21-42.
2. Beasley, J.E., 1985. "'A Note on Solving Large p-median Problems,"' European Journal of Operations Research, 21, 270-273.
3. Berman, O. Drezner, Z. and Wesolowsky, G. O., 2002. "'Locating Unreliable Service Facilities that are Distance Sensitive."' Computers and Operations Research, in press.
4. Daskin, M. S., 1995. *Network and Discrete Location: Models, Algorithms and Applications.* New York: John Wiley and Sons, Inc.
5. Daskin, M.S., 1998. CITATION, *Software package and Data sets.*
6. Daskin, M., 2000. "'A New Approach to Solving the Vertex P-Center Problem to Optimality: Algorithm and Computational Results,"' Northwestern University, IL.
7. Dvorett, J., 1999. *Compatibility-Based Genetic Algorithm: A New Approach to the P-Median Problem,* Northwestern University IL.
8. DeGarmo, E.P., Sullivan, W. and Bontadelli, J., 1993. Engineering Economy, 9^{th} ed. Macmillan Publishing Company, New York.
9. Erkut, E., Bozkaya, B. and Zhang, J., 2001. An Effective Genetic Algorithm for the p-median Problem, In press.
10. Galvao, R.D. and ReVelle, C., 1996. "'A Lagrangian Heuristic for the Maximal Covering Location Problems,"' European Journal of Operations Research, 88, 114-123.
11. Garey, M.R. and Johnson, D.S., 1979. *Computers and Interactibility: A Guide to the Theory of NP-Completeness.* Freeman, San Francisco, California.
12. Goldberg, D. E., 1989. Genetic Algorithms in Search, Optimization and Machine Learning, Addison-Wesley, Reading, MA.
13. Hakimi, S.L., 1964. "'Optimum Location of Switching Centers and the Absolute Centers and Medians of a Graph,"' Operations Research, 12, pp. 462-475.
14. Hakimi, S.L., 1965. "'Optimum Location of Swithching Centers in a Communication Network and Some Rated Graph Theoretic Problems,"' Operations Research, 13, pp. 450-459.
15. Holland, J.H., 1975. Adaptation in Natural and Artificial Systems, University of Michigan Press, Ann Arbor, MI.
16. Lorena, L.A.N. and Furtado, J. C., 2001. "'Constructive Genetic Algorithm for Clustering Problems,"' Evolutionary Computation 9(3), Massachusetts Institute of Technology.
17. Kariv, O. and Hakimi, S.L., 1979. "'An Algorithm Approach to Network Location Problems. II: The p-Medians,"' SIAM Journal on Applied Mathamtics, 37, 539-560.
18. Maranzana, F.E., 1964. "'On the Location of Supply Points to Minimize Transport Costs,"' Operational Research Quarterly, 15, 261-270.
19. Park, S. and Miller, K., 1988. "'Random Number Generators: Good Ones are Hard to Find,"' Communications of the ACM, vol.30, no. 10, 1192-1201.
20. Mirchandani, P.B. and R.L. Francis, 1990, *Discrete Location Theory,* Wiley, New York.
21. Revelle, C. and Swain, R., 1970. Central Facilities Location, *Geographical Analysis,* 2, 30-42.
22. Senne, L.F. and Lorena L.A.N., 2000. Lagrangian/Surrogate Heuristic for p-Median Problems. In Laguna, M. and Gonsalez-Velarde, J. L., editors, *Computing Tools for Modeling, Optimization and Simulation: Interfaces in Computer Science and Operations Research,* 115-130, Kluwer Academic Publishers, London, UK.
23. Teitz, M.B. and Bart, P., 1968. *Heuristic Methods for Estimating Generalized Vertex Median of a Weighted Graph*, Operations Research, 16, 955-961.

APPENDIX

Pseudo-code and flow of source code

- Read Data file for node distances, and existing k facilities

- Calculate the population size of the solution set
- Set all initialize the distribution of c group solution set (the existing k facilities are already a part of the solution so as to make them fixed genes in all solution set)
- Set itermax $= (n-k)/\sqrt{p-k}$;

do{
 - Compute Z value of the solution set
 - Locate best and worst solution among the solution set
 - Compute Probability Distribution $P[V_i]$
 do{
  Loop:
 - Call random generator to select parents in generating child

 Until: selected parents are not the same *(i.e. father is not the same as the mother)*
 - (Generation of Child) Combine the parent solutions into subset of "size" $2P$ indices
 - Separate the duplicated parent traits m including the k existing facility indices (*The remaining genes are called free genes of size* $2P-m$)
 - Among the candidate location indices, delete the index from the subset that when removed incurs the least oportunity cost
 - the resulting P indices after reducing the subset to k existing facility locations and $P-k$ candidates is the child generated from the parents

}**While** Child generated is same as parent

- Call random generator to determine need for mutation;
 If mutation must be implemented
 - all random generator to determine index j to be replaced;
 - Call random generator to determine new location to replace j;

 - Check if child generated improves the current solution set by comparing the child's z value with the z value of the worst solution **Until** the current best $z-value$ remains the same for *itermax*

Output the solution

