

Solving maximal covering location problem using genetic algorithm with local refinement

Soumen Atta¹ · Priya Ranjan Sinha Mahapatra¹ · Anirban Mukhopadhyay¹

Published online: 20 April 2017
© Springer-Verlag Berlin Heidelberg 2017

Abstract The maximal covering location problem (MCLP) deals with the problem of finding an optimal placement of a given number of facilities within a set of customers. Each customer has a specific demand and the facilities are to be placed in such a way that the total demand of the customers served by the facilities is maximized. In this article an improved genetic algorithm (GA)-based approach, which utilizes a local refinement strategy for faster convergence, is proposed to solve MCLP. The proposed algorithm is applied on several MCLP instances from literature and it is demonstrated that the proposed GA with local refinement gives better results in terms of percentage of coverage and computation time to find the solutions in almost all the cases. The proposed GA-based approach with local refinement is also found to outperform the other existing methods for most of the small as well as large instances of MCLP.

Keywords Facility location problem · Covering location problem · Maximal covering location problem (MCLP) · Genetic algorithm (GA) · Local refinement

1 Introduction

One of the most popular *facility location problems* is the *covering location problem* (Hamacher and Drezner 2002). The

objective of a simple version of this problem is to determine the optimal location and number of facilities in order to cover all customers. A customer is said to be *covered* or *served* by a facility if it falls within the *service distance* of that facility. The covering location problem has several applications in emergency and military services as well as in public services such as locating police stations, schools, plants, bus stops, fire stations to name a few (Chung 1986; Schilling et al. 1993). An important and related problem in this context is the *maximum covering location problem (MCLP)* (Adenso-Diaz and Rodriguez 1997; Church and ReVelle 1974; Galvão et al. 2000; Hamacher and Drezner 2002; Lorena and Pereira 2002; Resende 1998; ReVelle and Eiselt 2005; Rodriguez et al. 2012). MCLP is considered when it is not feasible to cover all customers due to many problems like budgetary constraints or insufficient resources. For a simple model of MCLP, the objective is to locate a given number of facilities such that the number of customers covered by these facilities is maximized.

Since MCLP is NP-hard in general (Berg et al. 2009; Gary and Johnson 1979; Megiddo et al. 1983), *genetic algorithms (GAs)* (Goldberg 1989) could be a better choice for solving MCLP. GAs are popular search and optimization strategies guided by the principle of Darwinian evolution. Here, the parameters of the search space are encoded in the forms of strings (*chromosomes*). The population consisting of a set of chromosomes is initialized randomly. The goodness of a chromosome is measured using a *fitness function* which is usually related to the objective function to be optimized. Some biologically inspired operators such as *selection*, *crossover* and *mutation* are applied to create the child population. Subsequently, elitism is performed to evolve the next generation population using the most-fit chromosomes of the parent and child populations. The algorithm terminates if some specific criterion is met or the maximum

✉ Anirban Mukhopadhyay
anirban@klyuniv.ac.in

Soumen Atta
soumen.atta@klyuniv.ac.in

Priya Ranjan Sinha Mahapatra
priya@klyuniv.ac.in

¹ Department of Computer Science and Engineering,
University of Kalyani, Kalyani, West Bengal 741235, India

Algorithm 1 Pseudocode of genetic algorithm

```

1: procedure GENETIC-ALGORITHM
2:    $i = 1$ ;
3:   initialize( $P(i)$ );      ▷ Initialize population with chromosomes
4:   evaluate( $P(i)$ );        ▷ Evaluate fitness of initial chromosomes
5:   while notTerminated do
6:      $M(i) = \text{selection}(P(i))$ ;      ▷ Perform selection to create
       mating pool
7:      $C(i) = \text{crossover}(M(i))$ ; ▷ Perform crossover on mating pool
8:      $T(i) = \text{mutation}(C(i))$ ; ▷ Perform mutation on the offspring
       solutions
9:     evaluate( $T(i)$ );      ▷ Evaluate fitness of new population
10:     $P(i + 1) = \text{createNextGenerationFrom}(P(i), T(i))$ ;      ▷
       Create next generation population by elitism
11:     $i = i + 1$ ;
12:  end while
13:  return(bestChromosome( $P(i)$ ));      ▷ Return the best
       chromosome of last generation
14: end procedure

```

generation limit is reached. The best-fit chromosome of the final generation provides the solution. Algorithm 1 shows the pseudocode of a typical GA.

Suitably designed GAs are known to perform well to produce near-global optimal results for NP-hard problems. However, these techniques are often criticized for their relatively high computation time. GAs were used by Jaramillo et al. (2002) for solving different types of location problems including MCLP. Fazel Zarandi et al. (2011) also developed a customized GA to solve MCLP and they applied this to large instances of MCLP with up to 2500 nodes. However, some studies have indicated that although classical GA is very good in quickly locating high performance regions in a large and complex search place, it is not good enough in fine-tuning the solutions that are very near to some optimal solution. GA spends most of the time in refining the initially obtained good solutions to convert them into best ones. The performance of GA can be improved substantially by incorporating some local improvement strategies applied on the solutions encoded in chromosomes (García-Martínez and Lozano 2007). Motivated by this, in this article, an improved GA-based approach is proposed to solve MCLP. In this approach, the chromosomes of GA encode possible locations of facilities. Fitness of each chromosome is computed by the percentage of coverage by the solution encoded by it. A local refinement strategy is adopted to improve the chromosomes locally for faster convergence of GA by quickly fine-tuning the solutions. The results obtained using the proposed GA-based technique with local refinement are compared with the results given by Lagrangian/surrogate heuristic-based approach in Lorena and Pereira (2002) as well as the customized GA-based approach in Fazel Zarandi et al. (2011). It is found to outperform both the approaches in terms of percentage of coverage and computational time in most of the cases. Moreover to emphasize the effectiveness

of the local refinement strategy, the results obtained using the proposed GA without this strategy are also mentioned in this article.

The rest of the article is organized as follows. Related works are discussed in Sect. 2. Section 3 formally defines MCLP. In Sect. 4, the proposed GA-based technique is described in detail. In Sect. 5, the experimental results are reported and discussed. Finally, Sect. 6 concludes the article.

2 Literature review

Since the introduction of MCLP by Church and ReVelle on a network (Church and ReVelle 1974), many of its alternative versions have been considered in facility location literature (Assis Corrêa et al. 2009; Atta and Mahapatra 2013; Berman et al. 2010; Davari et al. 2011; Drezner and Hamacher 2001; Farahani et al. 2012; Karasakal and Karasakal 2004; Karmakar 2011; Lee and Lee 2010; Mahapatra 2012; Moore and ReVelle 1982; Pereira et al. 2015; ReVelle and Eiselt 2005; Spieker et al. 2016). In the present work, we consider the model of MCLP proposed by Galvão and ReVelle (1996). In this model, each customer is considered as a point in a plane and each of them has an associated demand. These demands can also be considered as the weights of the points. The potential location of each facility is restricted to be the locations of customers. Each facility has its own service distance, within which only, it can provide its service. The *service area* of each facility is considered to be circular with the service distance as the radius of the circle. The service distance, a parameter of MCLP, is taken to be same for all the facilities. Therefore, the objective of MCLP under this model is to locate a given number of facilities such that the maximal possible demands of customers is served within a given service distance. Moreover, the geometric version of MCLP in this model can be stated as follows. Given a set of n weighted points on a plane, a number of facilities k , and a service distance r , find a placement of k circles, each of having radius r , so that the sum of the weights of the input points covered by k such circles is maximized.

Many researchers focus on different heuristic approaches for solving this problem. Galvão and ReVelle developed a *Lagrangian heuristic* for MCLP in Galvão and ReVelle (1996) where they tested their heuristic in a network of up to 150 vertices. They improved both the lower and upper bounds of the problem and hence provided a better approximate solution for MCLP compared to Weaver and Church (1984). Galvão et al. compared the solutions of MCLP obtained by the two heuristics based on *Lagrangian* and *surrogate relaxation* in Galvão et al. (2000). Comparison was done using 331 test problems available in the literature with number of vertices ranging from 55 to 900. Later on, Lorena

and Pereira (2002) developed another Lagrangian heuristic using *Unified Linear Model* given by Hillsman (1984) to obtain better solution for MCLP in reduced computational time. They got better *percentage of coverage* than that obtained by Galvão and ReVelle (1996) and Galvão et al. (2000). Other approaches used for MCLP are *dubbed greedy adding* (Church and ReVelle 1974), *greedy adding with substitution* (Church and ReVelle 1974), *greedy randomized adaptive search* (Resende 1998), genetic algorithms (GAs) (Fazel Zarandi et al. 2011; Jaramillo et al. 2002). Although MCLP is NP-hard in general, recently deterministic algorithms were proposed to solve the problem using the tools of computational geometry (De Berg et al. 2000; Preparata and Shamos 1993) when the number of given facilities is *small* and service area of each facility is either *circular* (Berg et al. 2009) or *square* (Mahapatra 2007; Mahapatra et al. 2015) or *rectangular* (Mahapatra 2012) in shape.

3 Problem definition

Let us assume that the customers are represented by a set P of m points in a plane. Each point $p_j \in P$, $j = 1, 2, \dots, m$, has a nonnegative demand which is considered as the weight of the corresponding point. The service area (or *coverage area*) of each facility is circular in shape, each having the radius r , which is a constant. This distance r is known as the service distance. The center of the circle with radius r is the location of the facility to be installed. Let the given number of facilities to be installed be k , $1 \leq k \leq m$ and the set of centers of these k facilities be $\{c_1, c_2, \dots, c_k\}$. Here, the potential location of each of the k facilities is restricted to the locations of the customers itself. This implies that $c_i \in P$, $1 \leq i \leq k$.

A point $p_j \in P$ is said to be covered (enclosed) by a circle with center at c_i , $1 \leq i \leq k$, if and only if the *Euclidean distance* between p_j and c_i , denoted as $d(p_j, c_i)$, is at most r ; otherwise the point p_j is not covered by the corresponding circle. If a point is covered by more than one circle then the corresponding customer can avail the service from any one of the facilities which are installed at the centers of such circles. However, we assume that a customer cannot avail the service from more than one facility. Note that the objective of MCLP is to find the locations of k facilities such that the sum of the demands of the customers covered by k facilities is maximized.

Let r be the service distance for MCLP, $P = \{p_1, p_2, \dots, p_m\}$ be the set of customers, $I = \{c_1, c_2, \dots, c_m\}$ be the set of potential facility sites, and f_j be the demand of customer p_j . Suppose $a_{ij} = 1$ if customer $p_j \in P$ can be covered by a facility located at $c_i \in I$, and $a_{ij} = 0$ otherwise. Let k be the number of facilities to be established. Also $x_j = 1$ if customer p_j is covered, and $x_j = 0$ otherwise; $y_i = 1$ means

that a facility must be located at site $c_i \in I$, and $y_i = 0$ otherwise. The mathematical formulation of the objective function of MCLP, $v(MCLP)$, given by Galvão et al. (2000) is as follows:

$$v(MCLP) = \max \sum_{p_j \in P} f_j x_j \quad (1)$$

subject to the constraints

$$\sum_{c_i \in I} a_{ij} y_i - x_j \geq 0, \quad p_j \in P, \quad (2)$$

$$\sum_{c_i \in I} y_i = k, \quad (3)$$

$$x_j \in \{0, 1\}, \quad p_j \in P, \quad (4)$$

$$y_i \in \{0, 1\}, \quad c_i \in I. \quad (5)$$

In this formulation, the objective function denotes the total demands covered and the goal is to maximize the value of the objective function. Constraint (2) denotes that a customer $p_j \in P$ is covered if and only if there is at least one facility $c_i \in I$ such that $d(p_j, c_i) \leq r$. Constraint (3) restricts the number of facilities to be exactly k . Constraints (4) and (5) guarantee the 0 – 1 nature of the decision variables of the problem. In this work, we consider the locations of the customers as the potential locations of the facilities to be placed. Hence, the set I and the set P are analogous for the problem considered in this article.

4 Proposed GA for MCLP

In this section, the proposed GA-based solution for the aforementioned MCLP is described. The overall procedure of the proposed GA-based solution of MCLP is demonstrated in Fig. 1. In the following subsections, we describe each step of the proposed method in detail.

4.1 Chromosome encoding

In GAs, possible solutions for the optimization problem in hand are encoded in the form of strings called chromosomes. A possible solution of MCLP is a set of potential locations of k facilities to be chosen from the set of m customers $P = \{p_1, p_2, \dots, p_m\}$. Thus a chromosome here encodes an integer string $\{t_1, t_2, \dots, t_k\}$ of length k representing the indices of the k customers chosen as the facilities. Each element $p_{t_i} \in P$, $i = 1, 2, \dots, k$, since the potential locations of k facilities are restricted to the locations of the customers itself. For example, if $m = 100$ and $k = 5$, then a chromosome $\{2, 34, 75, 87, 92\}$ implies that the customers $p_2, p_{34}, p_{75}, p_{87}$ and p_{92} are chosen as the potential locations of facilities as per the solution encoded in the chromosome.

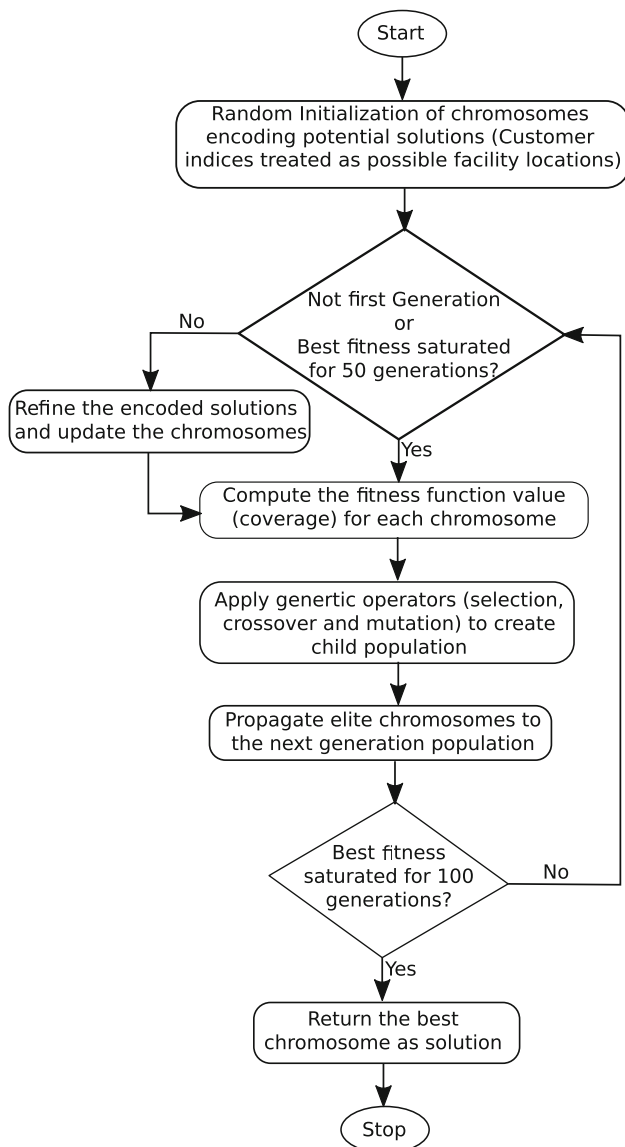


Fig. 1 Flowchart of the proposed GA-based method for solving MCLP

4.2 Population initialization

The initial population consists of \mathcal{P} chromosomes where \mathcal{P} is a user defined parameter called population size. Each chromosome of the initial population is generated by selecting k random indices from the set $\{1, 2, \dots, m\}$. In this work, we set $\mathcal{P} = 20$ which is chosen experimentally.

4.3 Fitness computation

Fitness of a chromosome represents the goodness of the solution encoded in it with respect to MCLP. The objective of MCLP is to maximize the coverage (i.e., the percentage ratio of total demands of the customers covered by some facilities

to the total demands of all customers). Hence, the coverage of the solution encoded in a chromosome is considered as the fitness of the chromosome, as shown in Eq. 1. The fitness is to be maximized.

4.4 Genetic operators

Three genetic operators, namely selection, crossover and mutation are used to create the next generation population from the current one.

4.4.1 Selection

Selection is the process of making a *mating pool* from the population. The chromosomes selected in the mating pool are capable of taking part in crossover. Here well-known *binary tournament selection* (Goldberg 1989) is used. In binary tournament selection, two chromosomes are randomly picked up from the population and the better one (with respect to the *fitness value*) is put in the mating pool. The process is repeated \mathcal{P} times (\mathcal{P} being the population size) to create a mating pool consisting of \mathcal{P} parents.

4.4.2 Crossover

Crossover operation is done for information exchange between two parents to create two offspring solutions. *Single-point crossover* operation is used here. In each crossover operation, two chromosomes are randomly chosen from the mating pool and their elements are swapped beyond a randomly chosen *crossover point* within length of the chromosomes. The crossover operation is controlled by *crossover probability* μ_c and repeated $\frac{\mathcal{P}}{2}$ times to fill up the child population with \mathcal{P} offspring solutions. The crossover probability μ_c is kept constant as 0.9 throughout all the generations.

4.4.3 Mutation

Unlike crossover, mutation refers to perturbation of a single parent. The following mutation procedure is adopted here. For each chromosome, each element is tested for a possible mutation with a *mutation probability* μ_m . If an element (a potential facility location) is to be mutated, it is replaced by a random point (customer) from the set of its first $\mathcal{N}\%$ nearest neighbors. Here, \mathcal{N} is chosen as 5 as it is found to provide good result. Note that the list of first $\mathcal{N}\%$ nearest neighbors of each point is precomputed to avoid run-time computation. The mutation probability is initialized to a small value of 0.01 and not changed for all the generations where the chromosomes undergo local refinement procedure (see Sect. 4.5). This ensures that mutation does not disrupt the chromosomes much when they proceed towards initial convergence through local refinements. Once the local refinement procedure is not

Table 1 Test problems

Problem	Number of nodes	Values of p	Values of S
G&R100 Galvão and ReVelle (1996)	100	[8, 10, 12]	[50, 65, 80]
G&R150 Galvão and ReVelle (1996), Galvão et al. (2000)	150	[8, 10, 12] and [5, 6, 7, 8, 10, 12, 14, 16, 18, 20]	[75, 80, 85, 90]
B700 Beasley (1990)	700	[20, 24, 28]	[13, 15, 20]
B900 Beasley (1990)	900	[20, 24, 28]	[10, 13, 16]
SJC324 Lorena and Pereira (2002)	324	[1, 2, 3, 4]	[800, 1200, 1600]
SJC402 Lorena and Pereira (2002)	403	[1, 2, 3, 4, 5, 6]	[800, 1200, 1600]
SJC500 Lorena and Pereira (2002)	500	[1, 2, 3, 4, 5, 6, 7, 8]	[800, 1200, 1600]
SJC708 Lorena and Pereira (2002)	708	[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11]	[800, 1200, 1600]
SJC818 Lorena and Pereira (2002)	818	[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14]	[800, 1200, 1600]
ZDS1800 Fazel Zarandi et al. (2011)	1800	[15, 20, 25]	[3.5, 3.75, 4]
ZDS2500 Fazel Zarandi et al. (2011)	2500	[15, 20, 25]	[3.5, 3.75, 4]

done any more, i.e., when the chromosomes have already converged well towards optimal solutions, mutation probability is increased to 0.8 to explore the search space around the near-optimal solutions with the expectation of obtaining some better solutions.

4.5 Local refinement

After *mutation*, each chromosome (solution) is locally updated. Local improvement is done as follows. First, the customers are clustered with respect to the facilities, i.e., each customer p_i is assigned to its nearest facility c_j . A cluster $clst_j$ is formed around each facility at c_j with the customers assigned to it (Jain et al. 1999; Mukhopadhyay et al. 2015). Thereafter, each facility at c_j encoded in a chromosome is updated with c_t such that

$$t = \operatorname{argmin}_{p_i \in clst_j} \sum_{p_l \in clst_j} f_j d(p_i, p_l). \quad (6)$$

Hence, each facility location is updated by the point which has minimum weighted sum of distances to the other points within the cluster corresponding to that facility. This ensures that the facility locations are shifted towards more centrally located points with higher demand values, which increases the overall coverage quickly. Hence, use of this refinement in the early stages helps faster convergence. Moreover, it may be noted that in most of the cases, the quality of the service deteriorates as the distance between customer and

facility increases. Therefore, this refinement strategy also tries to provide better service to the customers near to the facility compared to the customers which are far away from the facility. However, once the best *fitness function* does not change for 50 consecutive generations, the local refinement procedure is not applied any more to allow the chromosomes evolve freely.

4.6 Elitism

Elitism is employed to avoid losing good solutions due to randomness of the genetic operators. For this, the parent and child population for a particular generation are merged and then the top \mathcal{P} solutions according to the fitness values are propagated to the next generation. This ensures that the best chromosome obtained so far is not lost.

4.7 Termination criterion

The loop of fitness computation, selection, crossover, mutation, local refinement and elitism is iterated through generations. As mentioned earlier, the local improvements of the chromosomes are done until the best fitness value is saturated for 50 generations. Thereafter no local refinement is performed and the chromosomes are left to evolve freely. The loop continues until the best fitness value is saturated for the last 100 generations. The best solution (corresponding to the highest coverage value) is considered as the output of GA.

Table 2 Computational results for G&R100

n	p	S	Lorena and Pereira (2002)						GA (without Refinement)				GA (with Refinement)			
			Ref. Cov. (%)	Avg. Cov. (%)	Max. Cov. (%)	Ref. Time (s)	Avg. Time (s)	Max. Time (s)	Avg. Cov. (%)	Max. Cov. (%)	Avg. Time (s)	Max. Time (s)	Avg. Cov. (%)	Max. Cov. (%)	Avg. Time (s)	Max. Time (s)
100	8	50	69.43	69.19	70.49	51.69	0.84	0.94	69.36	69.92	7.84	12.85	70.34	72.9	5.85	9.65
100	10	50	76.23	76	76.94	62.9	1.01	1.15	77.43	79.18	11.75	17.43	77.98	79.38	6.57	9.87
100	12	50	81.61	81.42	82.27	64.26	1.22	1.37	81.53	85.84	10.27	16.34	81.64	86.43	7.23	10.23
100	8	65	87.36	87.09	87.89	53.81	0.9	1.04	85.84	87.03	7.39	11.82	87.59	88.43	5.39	7.14
100	10	65	94.33	94.77	95.57	20.4	1.05	1.16	94.32	95.07	8.86	13.59	94.67	96.39	5.68	6.79
100	12	65	99.18	99.57	100	22.03	2.02	3.13	97.63	99.14	9.23	13.27	98.57	100	6.21	12.07
100	8	80	88.46	88.57	88.92	43.56	0.87	1.1	87.42	88.53	8.44	11.79	87.83	89.49	6.58	9.21
100	10	80	96.21	95.84	96.41	20.54	1.18	1.37	93.74	94.13	9.31	14.55	94.34	96.98	5.63	9.05
100	12	80	100	99.76	100	7.41	2	3.3	98.21	100	10.45	16.73	98.42	100	8.52	13.37

Table 3 Computational results for G&R150

n	p	S	Lorena and Pereira (2002)						GA (without Refinement)				GA (with Refinement)			
			Ref. Cov. (%)	Avg. Cov. (%)	Max. Cov. (%)	Ref. Time (s)	Avg. Time (s)	Max. Time (s)	Avg. Cov. (%)	Max. Cov. (%)	Avg. Time (s)	Max. Time (s)	Avg. Cov. (%)	Max. Cov. (%)	Avg. Time (s)	Max. Time (s)
150	10	70	68.86	69.37	70.74	9	2.99	3.35	70.13	71.61	11.62	16.36	70.84	72.4	8.74	13.23
150	12	70	77.09	77.91	78.69	11	3.69	4.01	78.32	78.76	12.03	17.25	79.01	79.82	8.24	13.91
150	14	70	83.34	83.97	84.66	12	3.6	4.23	82.42	83.98	14.23	17.48	83.65	85.52	12.02	15.87
150	16	70	87.75	88.46	89.35	13	3.75	4.4	88.37	89.06	19.38	23.81	89.72	91.15	16.34	20.26
150	18	70	92.39	92.13	92.92	12	3.84	4.34	92.78	93.46	19.55	25.04	92.82	94.42	10.3	19.09
150	20	70	93.95	95.22	96.28	6	4.36	5.11	95.75	96.88	20.81	27.25	96.35	97.71	15.29	23.04
150	8	75	59.14	59.63	60.46	109.71	2.05	2.36	61.64	62.53	8.85	9.58	63.31	64.42	7.05	8.23
150	10	75	68.86	69.37	70.44	122.35	2.97	3.24	69.56	71.34	10.06	13.83	69.71	71.47	8.49	11.79
150	12	75	77.34	77.53	78.48	127.28	3.54	3.96	78.61	79.46	12.85	15.92	78.73	80.64	9.41	11.32
150	8	80	61.49	62.36	63.63	4	2.14	2.69	61.49	62.74	8.35	12.61	62.94	63.81	7.09	8.25
150	10	80	70.14	71.67	72.99	6	3.09	3.57	72.13	72.46	13.74	17.38	73.93	75.03	7.93	10.46
150	12	80	78.14	78.87	80.1	10	3.46	4.06	78.47	79.50	16.32	20.33	78.78	80.52	10.78	16.43
150	14	80	84.47	84.88	84.64	12	3.55	4.23	82.74	85.85	17.82	24.31	84.22	85.97	10.05	13.38
150	8	85	73.94	74.49	74.64	96.39	3.05	3.35	73.19	74.02	8.35	10.37	73.82	75.4	7.38	9.95
150	10	85	81.56	82.03	83.04	127.59	3.88	4.4	81.95	82.44	13.45	15.82	82.05	83.29	9.66	11.17
150	12	85	87.95	88.51	89.09	154.12	4.11	4.56	87.11	87.64	17.32	20.61	88.71	90.69	10.85	14.55
150	6	90	82.47	81.69	82.88	4	2.75	3.3	79.82	80.45	9.03	13.42	81.43	82.2	6.98	10.13
150	8	90	89.79	89.51	90.3	8	3.73	5.05	88.72	89.35	10.46	13.29	89.68	90.73	7.62	9.63
150	10	90	94.04	94.55	95.07	7	4.14	4.78	94.14	94.83	10.31	12.75	94.21	94.88	9.44	12.35
150	12	90	96.93	97.87	98.27	5	4.71	5.22	96.88	97.46	12.89	18.27	97.4	98.24	12.35	16.41
150	14	90	99.03	99.98	100	5	10.66	12.52	98.78	100	18.34	24.98	99.03	100	16.37	23.27
150	5	95	87.23	88.83	88.95	4	2.15	2.37	86.93	87.54	11.25	14.55	87.84	88.21	7.89	9.34
150	7	95	93.94	94.45	94.85	8	3.88	4.23	95.04	95.52	12.34	18.11	95.12	95.74	11.63	17.05

5 Experimental results

In this section, first the test problems on which the experiments are performed are discussed. Then, the results of the experiments are reported and discussed in detail.

5.1 Test problems

The performance of the proposed algorithm is tested with both real-world data and random data sets. The test problems considered in this article are summarized in Table 1 where p

Table 4 Computational results for B700

<i>n</i>	<i>p</i>	<i>S</i>	Lorena and Pereira (2002)			GA (without Refinement)				GA (with Refinement)						
			Ref. Cov. (%)	Avg. Cov. (%)	Max. Cov. (%)	Ref. Time (s)	Avg. Time (s)	Max. Time (s)	Avg. Cov. (%)	Max. Cov. (%)	Avg. Time (s)	Max. Time (s)	Avg. Cov. (%)	Max. Cov. (%)	Avg. Time (s)	Max. Time (s)
700	20	13	70.03	70.05	70.77	329	135.69	149.23	69.93	70.35	54.43	70.16	70.63	71.27	45.09	59.83
700	24	13	74.44	74.1	74.86	399	164.01	187.9	73.22	73.49	48.92	64.51	74.25	75.19	33.46	48.28
700	28	13	78.05	77.56	78.46	536	198.17	220.87	77.03	77.56	83.28	101.45	77.69	78.46	68.45	79.33
700	20	15	79.56	79.69	80.18	592	163.98	181.75	79.22	79.73	41.25	55.62	79.8	80.2	22.65	34.96
700	24	15	83.17	83.06	83.43	662	199.51	223.33	82.76	83.13	49.23	61.27	83.07	83.85	22.74	33.82
700	28	15	86.18	85.83	87.21	841	235.14	272.27	84.34	85.21	51.29	77.34	86.25	87.34	28.44	38.92
700	20	20	95.76	96.19	96.45	1281	547.08	652.02	95.61	96.04	50.39	74.82	96.28	96.52	20.03	29.37
700	24	20	97.01	97.39	97.57	1641	630.82	759.79	96.55	96.93	55.42	80.12	97.23	97.54	51.19	79.05
700	28	20	98.02	98.26	98.51	2076	737.77	909.94	97.76	97.95	61.73	78.29	98.12	98.33	58.83	72.06

Table 5 Computational results for B900

<i>n</i>	<i>p</i>	<i>S</i>	Lorena and Pereira (2002)			GA (without Refinement)				GA (with Refinement)						
			Ref. Cov. (%)	Avg. Cov. (%)	Max. Cov. (%)	Ref. Time (s)	Avg. Time (s)	Max. Time (s)	Avg. Cov. (%)	Max. Cov. (%)	Avg. Time (s)	Max. Time (s)	Avg. Cov. (%)	Max. Cov. (%)	Avg. Time (s)	Max. Time (s)
900	20	10	67.72	67.08	67.7	763	186.2	214.53	66.46	67.07	43.84	54.28	67.38	67.98	24.61	27.88
900	24	10	71.58	70.78	71.44	1099	244.17	264.19	70.89	71.31	47.93	76.46	71.14	71.91	36.69	54.36
900	28	10	75.12	73.73	74.55	1272	298.3	323.73	74.11	74.59	91.47	117.38	74.34	75.21	48.57	69.75
900	20	13	88.03	87.55	87.88	1272	467.75	528.43	86.33	86.68	75.32	96.74	87.43	87.91	37.14	49.23
900	24	13	90.48	89.75	90.19	1656	598.15	696.59	89.15	89.64	83.71	109.35	89.23	90.32	57.66	76.12
900	28	13	92.3	91.58	91.95	1989	771.27	852.51	90.92	91.12	103.57	129.41	91.33	92.07	49.84	62.38
900	20	16	96.73	96.64	96.82	2725	1533.13	178.155	96.04	96.24	100.18	116.73	96.34	96.91	30.76	42.37
900	24	16	97.66	97.76	97.98	3269	1873.06	2750.71	97.02	97.25	119.28	150.35	97.43	97.81	59.04	112.45
900	28	16	98.43	98.58	98.88	4244	2031.29	2575.65	97.39	97.87	147.22	170.71	98.26	98.64	68.02	101.35

Table 6 Computational results for SJC324

<i>n</i>	<i>p</i>	<i>S</i>	Lorena and Pereira (2002)		GA (without Refinement)		GA (with Refinement)	
			Cov. (%)	Time (s)	Cov. (%)	Time (s)	Cov. (%)	Time (s)
324	1	800	44.94	0.28	44.94	2.41	44.94	2.38
324	2	800	72.33	5.92	72.33	2.81	72.33	2.77
324	3	800	95.49	5.33	95.49	3.78	95.49	3.19
324	4	800	99.62	11.92	99.62	6.83	99.62	6.58
324	5	800	100	16.2	100	3.21	100	1.22
324	1	1200	81.73	0.27	81.73	2.34	81.73	2.67
324	2	1200	95.08	5.22	95.08	3.38	95.08	2.97
324	3	1200	100	9.84	100	0.22	100	0.17
324	1	1600	99.76	0.27	99.76	3.27	99.76	2.94
324	2	1600	100	15	100	0.19	100	0.16

and *S* denote the number of facilities and the service distance respectively.

The random data sets G&R100 and G&R150 were used by Galvão and ReVelle (1996) and by Galvão et al. (2000). These

two data sets correspond to two random networks of 100 and 150 vertices respectively. Here, the number of vertices indicates the number of customers. The number of arcs in these networks are determined by a pre-established *density*

Table 7 Computational results for SJC402

n	p	S	Lorena and Pereira (2002)		GA (without Refinement)		GA (with Refinement)	
			Cov. (%)	Time (s)	Cov. (%)	Time (s)	Cov. (%)	Time (s)
402	1	800	41.01	0.55	41.01	3.59	41.01	2.56
402	2	800	70.94	10.16	70.94	5.23	70.94	3.33
402	3	800	91.9	11.09	91.9	5.76	91.9	3.28
402	4	800	97.96	12.73	97.85	5.25	97.85	4.04
402	5	800	99.91	29.11	99.66	7.36	99.91	4.87
402	6	800	100	38.01	100	3.59	100	1.76
402	1	1200	66.36	0.71	66.36	2.94	66.36	2.83
402	2	1200	92.79	7.14	92.79	4.83	92.79	3.45
402	3	1200	100	13.46	100	1.22	100	0.18
402	1	1600	96.58	0.77	96.58	3.64	96.58	3.3
402	2	1600	100	11.87	100	2.6	100	2.29

Table 8 Computational results for SJC500

n	p	S	Lorena and Pereira (2002)		GA (without Refinement)		GA (with Refinement)	
			Cov. (%)	Time (s)	Cov. (%)	Time (s)	Cov. (%)	Time (s)
500	1	800	40.31	0.77	40.31	3.56	40.31	2.54
500	2	800	63.2	8.89	63.2	4.78	63.2	3.54
500	3	800	79.82	16.42	79.82	4.89	79.82	4.01
500	4	800	90.29	22.79	90.29	6.62	90.29	4.47
500	5	800	95.7	39.06	95.7	11.47	95.7	9.84
500	6	800	99.08	47.18	99.08	14.38	99.08	12.8
500	7	800	99.92	85.58	99.92	15.91	99.92	10.63
500	8	800	100	103.87	100	17.94	100	5.64
500	1	1200	54.43	1.08	54.43	3.76	54.43	2.77
500	2	1200	91.69	20.48	91.69	4.86	91.69	3.65
500	3	1200	98.41	22.9	98.41	8.83	98.41	6.62
500	4	1200	100	45.92	100	6.23	100	1.29
500	1	1600	75.12	1.15	75.12	4.65	75.12	3.26
500	2	1600	99.8	25.04	99.8	5.94	99.8	3.75
500	3	1600	100	60.74	100	1.86	100	1.28

Table 9 Computational results for SJC708

n	p	S	Lorena and Pereira (2002)		GA (without Refinement)		GA (with Refinement)	
			Cov. (%)	Time (s)	Cov. (%)	Time (s)	Cov. (%)	Time (s)
708	1	800	34.69	1.48	34.69	3.65	34.69	2.72
708	2	800	55	22.25	55	6.13	55	4.59
708	3	800	71.4	26.25	70.23	8.54	71.4	6.69
708	4	800	84.07	33.84	83.83	16.73	84.07	13.07
708	5	800	88.81	54.65	88	18.38	88.81	12.54
708	6	800	93.02	66.19	92.6	22.65	93.02	14.69
708	7	800	95.7	74.65	95.39	28.44	95.7	17.16
708	8	800	97.83	108.81	97.31	37.39	97.83	25.29
708	9	800	99.21	139.98	98.78	38.12	99.1	17.08

Table 9 continued

<i>n</i>	<i>p</i>	<i>S</i>	Lorena and Pereira (2002)		GA (without Refinement)		GA (with Refinement)	
			Cov. (%)	Time (s)	Cov. (%)	Time (s)	Cov. (%)	Time (s)
708	10	800	99.88	165.26	49.17	32.55	99.99	20.6
708	11	800	100	207.51	99.93	30.43	100	19.13
708	1	1200	48	1.98	48	5.36	48	2.98
708	2	1200	84.23	3.63	84.23	9.82	84.23	4.53
708	3	1200	92.68	30.4	92.68	12.42	92.68	5.74
708	4	1200	98.73	55.97	98.57	11.27	98.73	6.44
708	5	1200	99.79	84.69	98.73	16.56	99.66	7.24
708	6	1200	100	98.7	99.66	20.23	100	4.19
708	1	1600	69.56	2.04	69.56	5.65	69.56	3.84
708	2	1600	96.59	64.87	96.59	8.42	96.59	4.7
708	3	1600	98.74	52.73	98.59	8.18	98.59	5.38
708	4	1600	100	71.4	100	4.37	100	2.75

Table 10 Computational results for SJC818

<i>n</i>	<i>p</i>	<i>S</i>	Lorena and Pereira (2002)		GA (without Refinement)		GA (with Refinement)	
			Cov. (%)	Time (s)	Cov. (%)	Time (s)	Cov. (%)	Time (s)
818	1	800	28.77	1.48	28.77	4.47	28.77	2.86
818	2	800	45.62	29.16	45.62	7.72	45.62	5.48
818	3	800	60.02	37.02	59.81	10.54	60.02	6.32
818	4	800	73.46	43.83	73.37	14.63	73.46	7.72
818	5	800	84.1	51.03	83.05	23.38	84.1	13.52
818	6	800	88.82	73.87	87.59	31.65	88.82	17.73
818	7	800	92.34	99.8	90.45	42.33	92.34	22.03
818	8	800	95.25	129.84	95.12	45.59	95.35	18.11
818	9	800	97.2	158.02	96.31	43.76	97.36	18.78
818	10	800	98.19	197.79	97.36	40.38	98.55	22.22
818	11	800	99.48	215.36	98.36	47.62	99.74	34.52
818	12	800	99.78	283.91	99.42	62.63	99.81	20.49
818	13	800	99.92	299.89	99.68	57.93	99.98	21.78
818	14	800	100	337.02	99.83	42.41	100	31.17
818	1	1200	39.81	1.71	39.81	6.77	39.81	4.12
818	2	1200	69.56	49.16	69.56	9.04	69.56	5.57
818	3	1200	86.43	35.21	86.43	14.74	86.43	5.69
818	4	1200	92.67	25.95	91.91	18.95	92.67	8.23
818	5	1200	97.75	89.97	95.45	29.62	97.75	16.58
818	6	1200	99.89	106.61	99.31	14.58	99.59	7.74
818	7	1200	100	137.31	99.96	10.72	99.96	5.71
818	1	1600	57.69	2.64	97.69	4.97	57.69	3.78
818	2	1600	84.5	43.72	84.5	8.52	84.5	5.22
818	3	1600	94.87	53.89	94.87	10.37	94.87	5.22
818	4	1600	98.95	62.28	98.95	15.78	98.95	5.07
818	5	1600	100	110.4	100	6.17	100	3.16

Table 11 Computational results for ZDS1800

<i>n</i>	<i>p</i>	<i>S</i>	CPLEX Fazel Zarandi et al. (2011)			Meta-heuristic Fazel Zarandi et al. (2011)			GA (without Refinement)			GA (with Refinement)		
			Time (s)	Coverage (%)	Time (s)	Coverage (%)	Avg. Cov. (%)	Max. Cov. (%)	Avg. Time (s)	Max. Time (s)	Avg. Cov. (%)	Max. Cov. (%)	Avg. Time (s)	Max. Time (s)
1800	15	3.5	4	70.61	155	69.94	70.24	70.69	85.66	98.61	71.01	71.39	35.86	41.47
1800	15	3.75	5	78.39	157	77.67	77.28	77.7	99.52	97.44	79.01	79.23	32.56	39.02
1800	15	4	10	84.83	149	84.55	83.89	84.37	95.69	113.86	85.42	85.81	35.21	40.92
1800	20	3.5	10,922	85.17	176	83.16	83.72	84.68	109.58	123.97	87.01	87.65	52.34	66.74
1800	20	3.75	10,877	91.83	170	90.33	91.14	91.44	105.62	119.74	92.32	92.86	48.82	58.24
1800	20	4	10,846	96.72	166	94.94	95.17	95.67	95.83	102.67	96.67	97.02	52.45	68.92
1800	25	3.5	10,836	95.67	192	94.39	94.85	95.12	102.39	117.84	96.25	96.84	78.34	97.25
1800	25	3.75	10,962	99.17	187	97.77	97.35	98.11	103.73	124.86	99.11	99.25	81.03	104.82
1800	25	4	15	100	181	99.39	99.21	99.46	118.49	133.85	100	100	25.67	30.32

Table 12 Computational results for ZDS2500

<i>n</i>	<i>p</i>	<i>S</i>	CPLEX Fazel Zarandi et al. (2011)			Meta-heuristic Fazel Zarandi et al. (2011)			GA (without Refinement)			GA (with Refinement)		
			Time (s)	Coverage (%)	Time (s)	Coverage (%)	Avg. Cov. (%)	Max. Cov. (%)	Avg. Time (s)	Max. Time (s)	Avg. Cov. (%)	Max. Cov. (%)	Avg. Time (s)	Max. Time (s)
2500	15	3.5	15	68.32	224	68.08	68.04	68.43	121.62	134.97	69.47	69.64	63.42	65.39
2500	15	3.75	24	75.88	218	75.56	75.17	75.62	132.78	156.82	77.08	77.42	51.49	60.39
2500	15	4	33	83.2	215	81.71	83.04	83.6	178.03	187.84	83.52	83.96	56.23	69.03
2500	20	3.5	267	84.68	250	83.72	83.86	84.47	122.59	133.74	85.49	85.74	65.88	77.84
2500	20	3.75	10,800	91.44	246	89.72	89.91	90.33	128.03	142.18	91.62	91.98	66.74	82.13
2500	20	4	10,800	96.72	239	94.92	94.02	94.35	142.49	154.98	96.75	96.82	78.93	90.29
2500	25	3.5	10,800	95.84	273	94.12	93.93	94.27	150.25	162.29	95.72	96.02	95.32	118.39
2500	25	3.75	10,776	99.24	274	97.4	97.28	97.69	147.31	159.03	99.01	99.42	105.32	128.93
2500	25	4	2916	100	268	99.44	99.16	99.28	151.81	172.36	99.93	99.99	67.84	76.36

δ which is defined as the ratio of the number of arcs in the network to the number of arcs in the corresponding complete network (Galvão and ReVelle 1996). Galvão et al. considered δ as 0.10 and 0.05 for G&R100 and G&R150 respectively. They set the length of each arc for both the networks as an integer value taken randomly from a uniform distribution in the range [40, 60]. Then, Floyd's algorithm is used to obtain the corresponding distance matrices. The demand of each customer for G&R100 is sampled from a uniform distribution in the range [20, 30]. For G&R150, the demands are obtained from a normal distribution with mean and standard deviation equal to 80 and 15 respectively.

In addition, from Beasley's OR Library (Beasley 1990), the data sets *pmcd32* and *pmcd39* with 700 and 900 vertices respectively are obtained to form the distance matrices of B700 and B900 respectively. These data sets are for *p-median problem* (Hamacher and Drezner 2002; Mladenović et al. 2007). That is why these data sets do not include the demand of each customer. So the demand of each customer is sampled from a normal distribution with mean 80 and standard deviation 15 (Galvão et al. 2000).

Besides these, some real-world data are also considered to test the performance of the proposed GA. These data sets are taken from the geo-referenced database of São José dos Campos city, Brazil. The data sets SJC324, SJC402, SJC500, SJC708, and SJC818 consist of 324, 402, 500, 708, and 818 vertices, respectively. Lorena and Pereira (2002) used these data available in the following website: <http://www.lac.inpe.br/~lorena/instancias.html>.

The data sets ZDS1800 and ZDS2500 are used to check the performance of the proposed algorithm for large instances of MCLP. Fazel Zarandi et al. (2011) used these random data sets having the following specifications. ZDS1800 and ZDS2500 correspond to networks of 1800 and 2500 nodes respectively. The locations of the nodes are generated randomly from a uniform distribution in the range [0, 30], and the demand of each node is sampled from a uniform distribution in the range [0, 100].

5.2 Results and discussion

The proposed GA is programmed using MATLAB and run on Intel Core 2 duo with 2.2GHz processor and 3 GB RAM having Windows XP operating system. The computational results are compared with the data given by Lorena and Pereira (2002) and Fazel Zarandi et al. (2011). The percentage of coverage values and computational times obtained using the proposed GA without local refinement and with local refinement are reported in Tables 2, 3, 4, 5, 6, 7, 8, 9, 10, 11 and 12 for all the data sets. The better or same values obtained by the proposed GA with local refinement as compared to the other existing algorithms are marked in bold. For some instances, the proposed GA with local refinement

takes more time than that reported by Lorena and Pereira (2002) and Fazel Zarandi et al. (2011). However, these extra computational times can be afforded if percentage of coverage is improved. It may be noted that the improvement in percentage of coverage is crucial when high budgets are involved for setting up the facilities. Moreover, the facilities are installed only once and thus the locations of the facilities corresponding to the maximum coverage are always taken. Hence, higher improvement in percentage of coverage is more important than the computational cost. In this article, the results corresponding to the maximum coverage are taken for comparison. The three parameters of the problem, viz., number of customers, number of facilities and service distance are indicated by the columns *n*, *p* and *S*, respectively in Tables 2, 3, 4, 5, 6, 7, 8, 9, 10, 11 and 12.

The computational results for the networks with 100, 150, 700 and 900 vertices (customers) are shown in Tables 2, 3, 4 and 5, respectively. In these four tables, the best values of coverage reported by Galvão and ReVelle (1996) and Galvão et al. (2000) and the respective computational times are indicated in the columns *Ref. Cov. (%)* and *Ref. Time (%)* respectively. It is evident that in terms of maximum coverage the proposed GA with local refinement gives better results for all the instances of the data set G&R100 in comparison with the maximum coverage reported by Lorena and Pereira (2002) as shown in Table 2. The proposed GA with local refinement performs better in 19 instances out of total 23 instances for the data set G&R150 as shown in Table 3. The proposed GA with refinement strategy gives better or same results in 7 instances out of total 9 instances and 6 instances out of total 9 instances for data sets B700 and B900 respectively as shown in Tables 4 and 5, respectively. For better understanding of the results obtained using the proposed GA with refinement strategy, the difference between the obtained maximum percentage coverage using the proposed GA with local refinement with the reference coverage percentage and the maximum coverage percentage given in the columns 4 and 6 of Tables 2, 3, 4 and 5 are been plotted in Figs. 2, 3, 4 and 5, respectively. In Figs. 2, 3, 4 and 5, *other 1* and *other 2* refer to the reference coverage percentage and the maximum coverage percentage corresponding to the columns 4 and 6 of Tables 2, 3, 4 and 5, respectively.

The computational results for the data sets SJC324, SJC402, SJC500, SJC708 and SJC818 are shown in Tables 6, 7, 8, 9 and 10, respectively. In terms of maximum coverage compared to Lorena and Pereira (2002), the proposed GA with refinement strategy yields better or same results in all the instances of the data sets SJC324 and SJC500 as shown in Tables 6 and 8, respectively. Moreover, the proposed GA with local refinement provides better or same maximum coverage compared to Lorena and Pereira (2002) in 10, 18 and 24 instances out of 11, 21 and 26 instances of the data sets SJC402, SJC708 and SJC818 respectively as depicted in

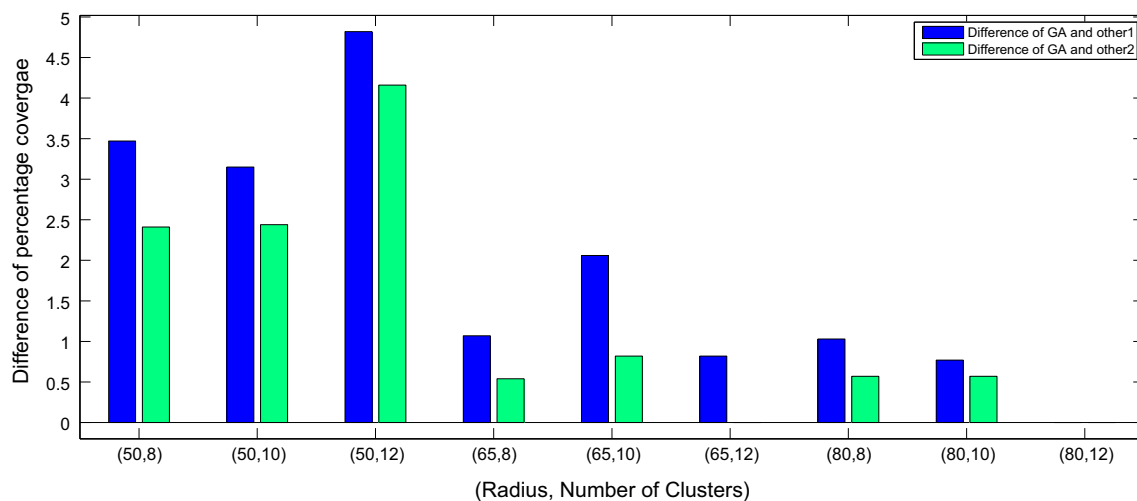


Fig. 2 Performance comparison for G&R100. GA: maximum percentage coverage using proposed GA with local refinement mentioned in column 15 of Table 2, *other1*: reference coverage percentage mentioned in column 4 of Table 2, *other2*: maximum coverage percentage mentioned in column 6 of Table 2

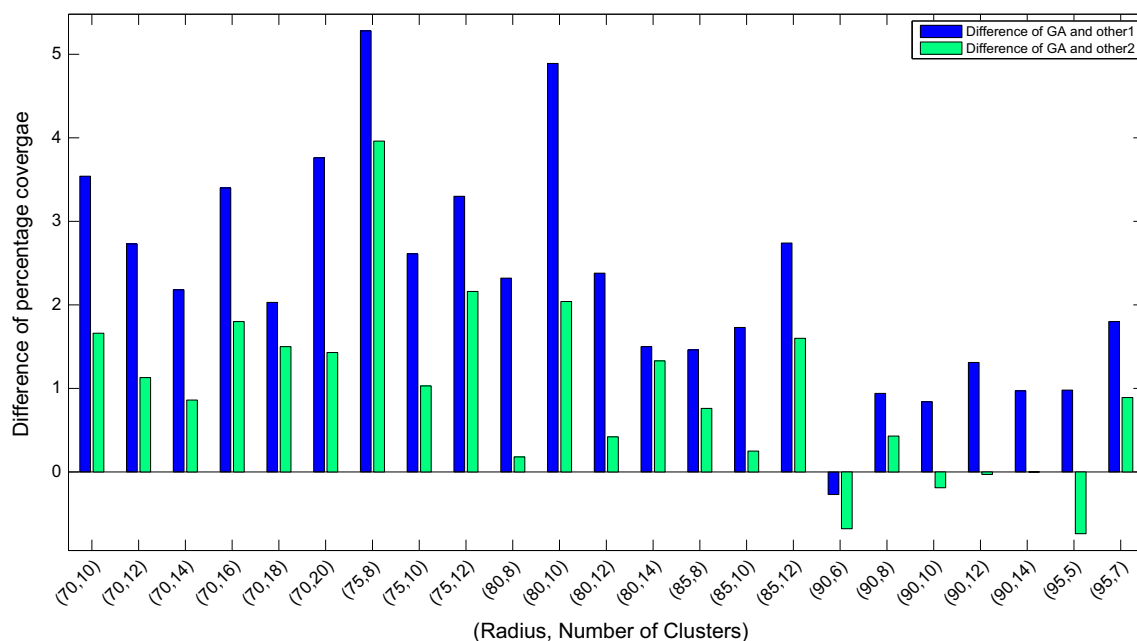


Fig. 3 Performance comparison for G&R150. GA: maximum percentage coverage using proposed GA with refinement mentioned in column 15 of Table 3, *other1*: reference coverage percentage mentioned in column 4 of Table 3, *other2*: maximum coverage percentage mentioned in column 6 of Table 3

Tables 7, 9, and 10, respectively. For visual illustration, the difference between the obtained maximum percentage coverage (given in column 8) using the proposed GA with the coverage percentage given in the column 4 of Tables 7, 9 and 10 are plotted in Figs. 6, 9 and 10, respectively. In Figs. 6, 7 and 8, *other* refers to the coverage percentage corresponding to the column 4 of Tables 7, 9 and 10 given by Lorena and Pereira (2002).

The computational results for the data sets ZDS1800 and ZDS2500 are reported in Tables 11 and 12, respectively. It appears from the tables that the proposed GA with local refinement performs better for all the instances of the data sets ZDS1800 and ZDS2500 in terms of maximum coverage given by Fazel Zarandi et al. (2011). The better performance of the proposed GA with refinement is also demonstrated in Figs. 9 and 10. In these figures, *other1* and *other2* refer to the coverage percentage obtained using CPLEX and maxi-

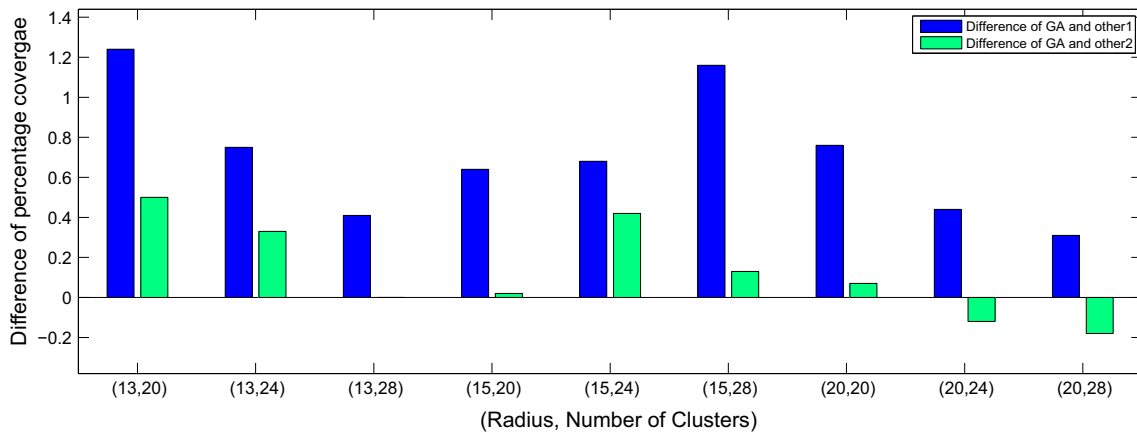


Fig. 4 Performance comparison for B700. GA: maximum percentage coverage using proposed GA with refinement mentioned in column 15 of Table 4, *other1*: reference coverage percentage mentioned in column 4 of Table 4, *other2*: maximum coverage percentage mentioned in column 6 of Table 4

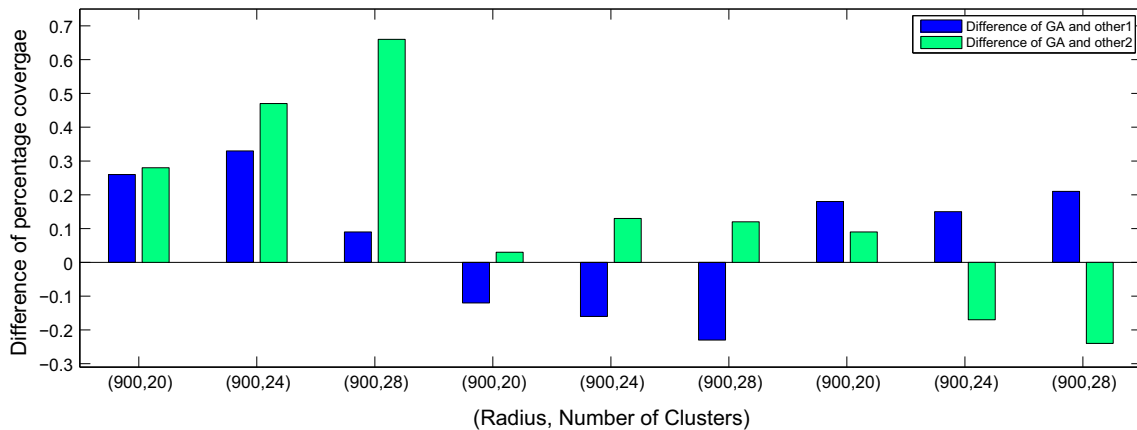


Fig. 5 Performance comparison for B900. GA: maximum percentage coverage using proposed GA with refinement mentioned in column 15 of Table 5, *other1*: reference coverage percentage mentioned in column 4 of Table 5, *other2*: maximum coverage percentage mentioned in column 6 of Table 5

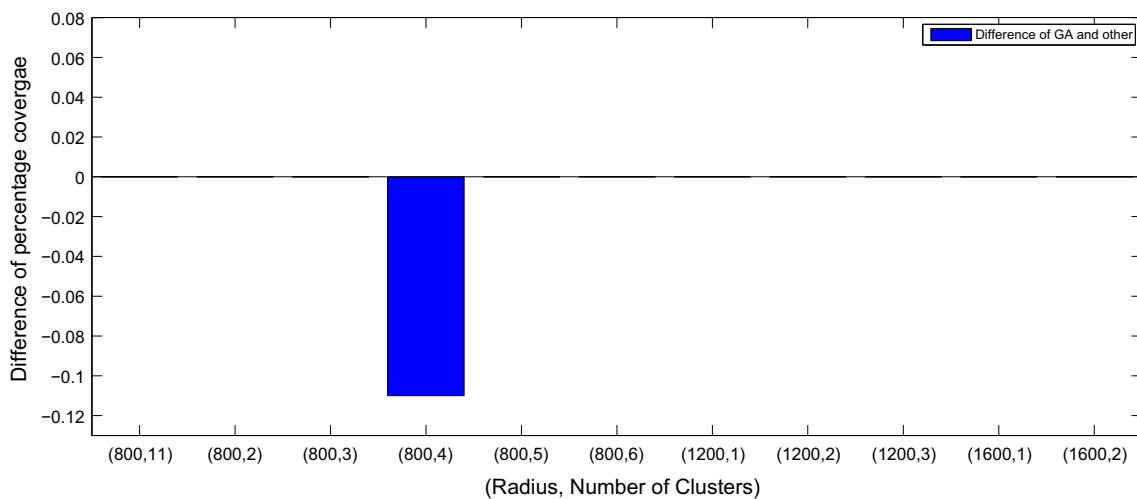


Fig. 6 Performance comparison for SJC402. GA: maximum percentage coverage using proposed GA with refinement mentioned in column 8 of Table 7, *other*: coverage percentage mentioned in column 4 of Table 7

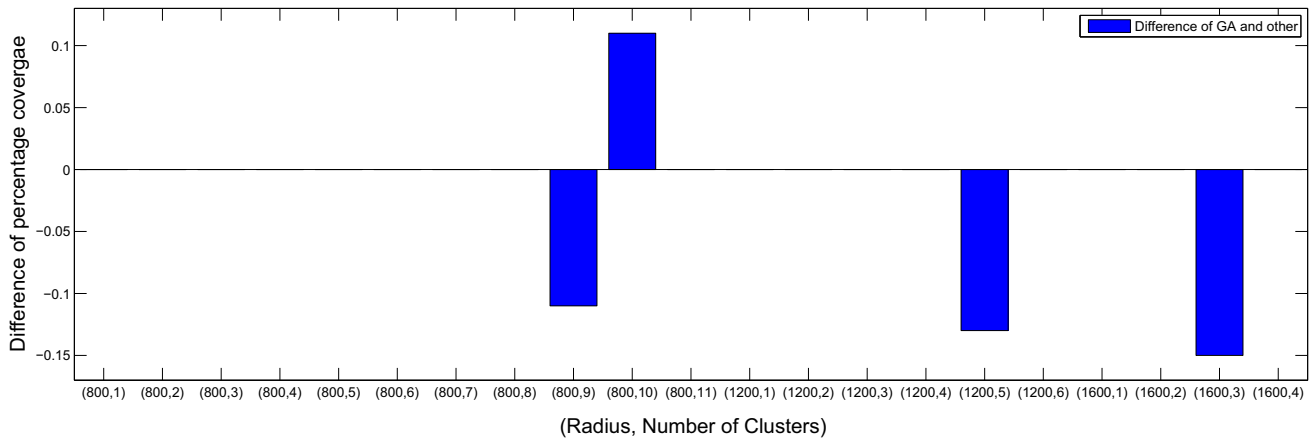


Fig. 7 Performance comparison for SJC708. GA: maximum percentage coverage using proposed GA with local refinement mentioned in column 8 of Table 9, *other*: coverage percentage mentioned in column 4 of Table 9

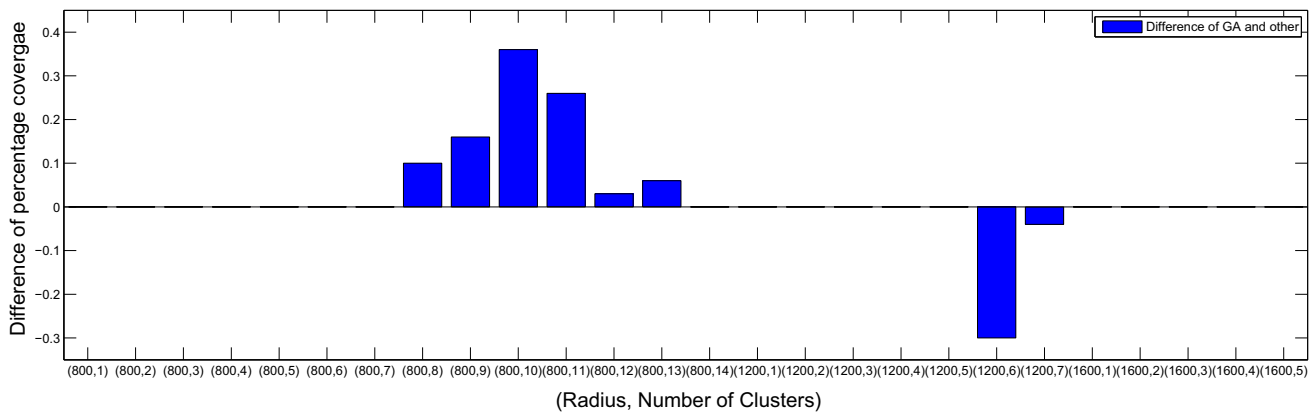


Fig. 8 Performance comparison for SJC818. GA: maximum percentage coverage using proposed GA with refinement mentioned in column 8 of Table 10, *other*: reference coverage percentage mentioned in column 4 of Table 10

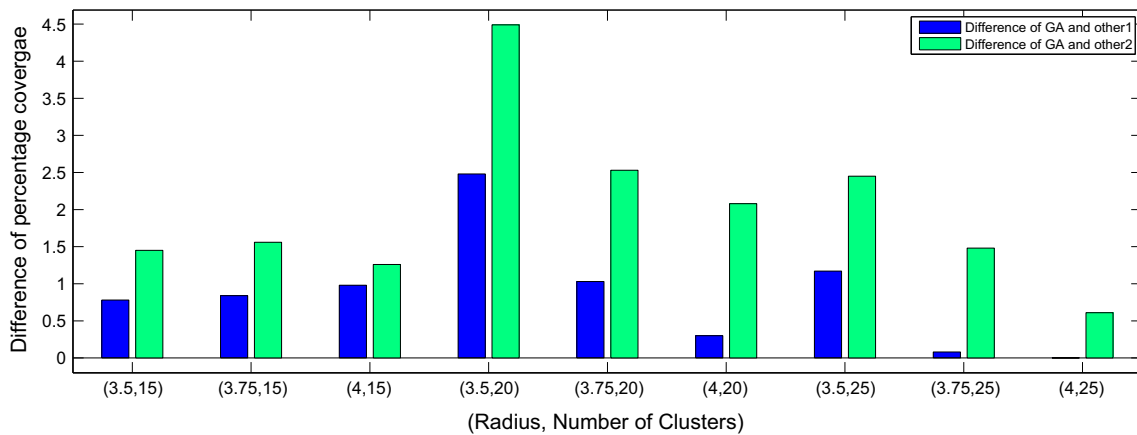


Fig. 9 Performance comparison for ZDS1800. GA: maximum percentage coverage using proposed GA with refinement mentioned in column 13 of Table 11, *other*: coverage percentage mentioned in column 5 of Table 11, *other2*: maximum coverage percentage mentioned in column 7 of Table 11

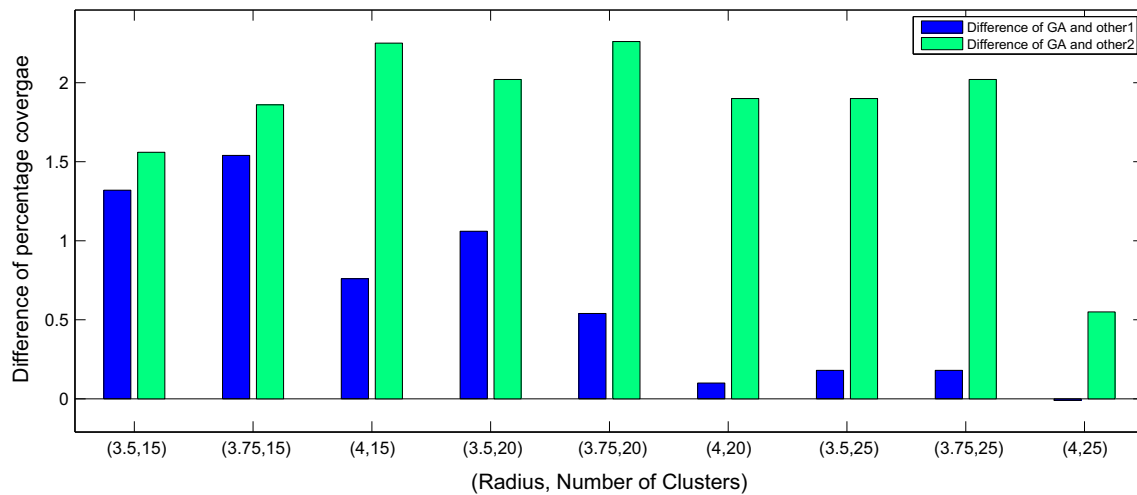


Fig. 10 Performance comparison for ZDS2500. GA: maximum percentage coverage using proposed GA with refinement mentioned in column 13 of Table 12, *other1*: coverage percentage mentioned in column 5 of Table 12, *other2*: maximum coverage percentage mentioned in column 7 of Table 12

imum coverage percentage corresponding to the columns 5 and 7 of Tables 11 and 12.

It is evident from the results reported in all the tables that the proposed GA with local refinement performs better than the proposed GA without local refinement in terms of percentage of coverage. For few instances, the proposed GA without refinement strategy gives same percentage of coverage as obtained using the GA with refinement using more computation time. This signifies the effectiveness of the local refinement strategy to the proposed GA in terms of solution quality and convergence. It is also evident from the results reported in all the tables and the figures that whenever the proposed GA with local refinement outperforms the results given by Lorena and Pereira (2002) and Fazel Zarandi et al. (2011), the improvement in terms of maximum coverage is significant. However, for the few cases, where the proposed GA with refinement strategy fails to improve the results given by Lorena and Pereira (2002), the failure is marginal. As a whole, the proposed GA with local refinement is found to provide better results in reasonable time in most of the cases for both random and real-world data sets. It has also performed equally well for both small and large instances of the problem.

6 Conclusion

In this article, we have proposed a GA-based approach for solving the maximal covering location problem. The proposed GA-based approach utilizes a local refinement strategy during initial generations to guide the search for potential facility locations. Use of this local refinement strategy improves its overall rate of convergence. The performance

of the proposed technique has been demonstrated on several benchmark data sets. Moreover, its performance has been compared with that of existing heuristic approaches and other GA-based approaches and illustrated both numerically and visually. The proposed approach has been found to outperform the existing approaches in most of the cases.

Acknowledgements This work has been partially supported by DST-PURSE scheme, Government of India at University of Kalyani.

Compliance with ethical standards

Conflict of interest This section is to certify that we have no potential conflict of interest.

References

- Adenso-Diaz B, Rodriguez F (1997) A simple search heuristic for the MCLP: application to the location of ambulance bases in a rural region. *Omega* 25(2):181–187
- Atta S, Mahapatra PRS (2013) Genetic algorithm based approach for serving maximum number of customers using limited resources. *Proced Technol* 10:492–497
- Beasley JE (1990) OR-Library: distributing test problems by electronic mail. *J Oper Res Soc* 41(11):1069–1072
- Berman O, Drezner Z, Krass D (2010) Generalized coverage: new developments in covering location models. *Comput Oper Res* 37(10):1675–1687
- Chung CH (1986) Recent applications of the maximal covering location planning (MCLP) model. *J Oper Res Soc* 37(8):735–746
- Church R, ReVelle C (1974) The maximal covering location problem. *Pap Reg Sci* 32(1):101–118
- Davari S, Zarandi MHF, Hemmati A (2011) Maximal covering location problem (MCLP) with fuzzy travel times. *Expert Syst Appl* 38(12):14535–14541
- de Assis Corrêa F, Lorena LAN, Ribeiro GM (2009) A decomposition approach for the probabilistic maximal covering location-allocation problem. *Comput Oper Res* 36(10):2729–2739

- de Berg M, Cabello S, Har-Peled S (2009) Covering many or few points with unit disks. *Theory Comput Syst* 45(3):446–469
- De Berg M, Van Kreveld M, Overmars M, Schwarzkopf OC (2000) *Computational geometry: algorithms and applications*. Springer. <https://books.google.co.in/books?id=C8zaAWuOIOcC>
- Drezner Z, Hamacher HW (2001) *Facility location: applications and theory*. Springer, Berlin
- Farahani RZ, Asgari N, Heidari N, Hosseininia M, Goh M (2012) Covering problems in facility location: a review. *Comput Ind Eng* 62(1):368–407
- Fazel Zarandi M, Davari S, Haddad Sisakht S (2011) The large scale maximal covering location problem. *Sci Iran* 18(6):1564–1570
- Galvão RD, Gonzalo Acosta Espejo L, Boffey B (2000) A comparison of Lagrangean and surrogate relaxations for the maximal covering location problem. *Eur J Oper Res* 124(2):377–389
- Galvão RD, ReVelle C (1996) A Lagrangean heuristic for the maximal covering location problem. *Eur J Oper Res* 88(1):114–123
- García-Martínez C, Lozano M (2007) Local search based on genetic algorithms. In: *Advances in metaheuristics for hard optimization*. Springer, pp 199–221
- Gary MR, Johnson DS (1979) *Computers and intractability: a guide to the theory of NP-completeness*
- Goldberg DE (1989) *Genetic algorithms in search, optimization and machine learning*. Addison-Wesley, New York
- Hamacher HW, Drezner Z (2002) *Facility location: applications and theory*. Springer, New York
- Hillsman E (1984) The p-median structure as a unified linear model for location-allocation analysis. *Environ Plan A* 16:305–318
- Jain AK, Murty MN, Flynn PJ (1999) Data clustering: a review. *ACM Comput Surv* 31(3):264–323
- Jaramillo JH, Bhadury J, Batta R (2002) On the use of genetic algorithms to solve location problems. *Comput Oper Res* 29(6):761–779. doi:10.1016/S0305-0548(01)00021-1
- Karasakal O, Karasakal EK (2004) A maximal covering location model in the presence of partial coverage. *Comput Oper Res* 31(9):1515–1526
- Karmakar A (2011) *Location problems for covering demands: algorithms and applications*. Ph.D. thesis, Indian Statistical Institute, Kolkata, India
- Lee JM, Lee YH (2010) Tabu based heuristics for the generalized hierarchical covering location problem. *Comput Ind Eng* 58(4):638–645
- Lorena LA, Pereira MA (2002) A Lagrangean/surrogate heuristic for the maximal covering location problem using Hillman's edition. *Int J Ind Eng* 9:57–67
- Mahapatra PRS, Goswami PP, Das S (2007) Covering points by isothetic unit squares. In: *CCCG*, pp 169–172
- Mahapatra PRS (2012) *Studies on variations of enclosing problem using rectangular objects*. Ph.D. thesis, University of Kalyani, Kalyani, West Bengal, India
- Mahapatra PRS, Goswami PP, Das S (2015) Placing two axis-parallel squares to maximize the number of enclosed points. *Int J Comput Geom Appl* 25(04):263–282
- Megiddo N, Zemel E, Hakimi SL (1983) The maximum coverage location problem. *SIAM J Algebraic Discrete Methods* 4(2):253–261
- Mladenović N, Brimberg J, Hansen P, Moreno-Pérez JA (2007) The p-median problem: a survey of metaheuristic approaches. *Eur J Oper Res* 179(3):927–939
- Moore GC, ReVelle C (1982) The hierarchical service location problem. *Manag Sci* 28(7):775–780
- Mukhopadhyay A, Maulik U, Bandyopadhyay S (2015) A survey of multiobjective evolutionary clustering. *ACM Comput Surv* 47(4):61:1–61:46
- Pereira MA, Coelho LC, Lorena LA, De Souza LC (2015) A hybrid method for the probabilistic maximal covering location-allocation problem. *Comput Oper Res* 57:51–59
- Preparata F, Shamos M (1993) *Computational geometry: an introduction*. Monographs in Computer Science. Springer, New York. <https://books.google.co.in/books?id=gFtvRdUY09UC>
- Resende MG (1998) Computing approximate solutions of the maximum covering problem with GRASP. *J Heuristics* 4(2):161–177
- ReVelle CS, Eiselt HA (2005) *Location analysis: a synthesis and survey*. *Eur J Oper Res* 165(1):1–19
- Rodríguez FJ, Blum C, Lozano M, García-Martínez C (2012) Iterated greedy algorithms for the maximal covering location problem. In: *European conference on evolutionary computation in combinatorial optimization*. Springer, pp 172–181
- Schilling DA, Jayaraman V, Barkhi R (1993) A review of covering problems in facility location. *Location Sci* 1(1):25–55
- Spieker H, Hagg A, Gaier A, Meilinger S, Asteroth A (2016) Multi-stage evolution of single- and multi-objective MCLP. *Soft Comput* 1–14. doi:10.1007/s00500-016-2374-9
- Weaver J, Church R (1984) A comparison of direct and indirect primal heuristic/dual bounding solution procedures for the maximal covering location problem. Unpublished paper