Yayın Geliş Tarihi (Submitted): June/Haziran-2022 | Yayın Kabul Tarihi (Accepted): September/Eylül-2023

Solution Approach To P-Median Facility Location Problem With Integer Programming And Genetic Algorithm

P-Medyan Tesis Yeri Seçim Problemine Tam Sayılı Programlama ve Genetik Algoritmasıyla Çözüm Yaklaşımı

Doç. Dr. Emre EKİN [1]

Abstract

In the study, the P-median problem was used and discussed. With this problem, in short, it is ensured that the locations for the supply of n demand points of p facilities with the lowest cost are determined and the demand points to be served are allocated to these locations. The solution of the p-median problem with the integer programming approach is in the NP (Non-Polynominal) difficult class, and the solution time increases exponentially as the size of the problem increases. For this, the use of heuristic approaches in the P-median problem significantly shortens the solution time of large-sized problems. In practice, the P-median is considered as capacity-constrained and unconstrained. Euclidean distances were calculated in Excel. The solution of mixed integer programming is obtained by using the CPLEX program. In addition to mixed integer programming, genetic algorithm, which is a meta-heuristic method, is applied to the capacity-constrained P-median problem. Different solutions were obtained by changing the population size, maximum number of iterations, crossover and mutation probability values. It has been observed that there is more than one optimum solution in repeated studies. When working with larger data sets, this algorithm will be less likely to find the global optimum. However, when compared to mixed integer programming, the genetic algorithm gave results acceptable solutions to the optimum result. The use of heuristic algorithms in real-world complex problems will both shorten the solution time considerably compared to the integer programming method and provide results that are acceptable solutions to the optimum result.

Keywords: Facility Location Problem, Integer Programming, Genetic Algorithm, P-Median.

Paper Type: Research

Öz

Çalışmada P-medyan problemi kullanılmış ve tartışılmıştır. Bu problem ile kısaca p adet tesisin n adet talep noktasının gereksinimlerine en düşük maliyet ile temini hususunda lokasyonların belirlenmesi ve servis edilecek talep noktalarının bu lokasyonlara tahsis edilmesi sağlanmaktadır. P-medyan probleminin tam sayılı programlama yaklaşımı ile çözümü NP (Non-Polinomial) zor sınıfına girmekte ve problemin boyutu arttıkça çözüm süresi de üssel olarak artmaktadır. Bunun için P-medyan probleminde sezgisel yaklaşımların kullanımı büyük boyutlu problemlerin çözüm süresini ciddi manada kısaltmaktadır. Uygulamada P-medyan, kapasite kısıtlı ve kısıtsız olarak ele alınmıştır. Öklid mesafeleri Excel'de hesaplanmıştır. Kapasite kısıtsız P- medyan problemine karışık tam sayılı programlamanın haricinde ilave olarak meta sezgisel bir metot olangenetik algoritması uygulanmıştır. Popülasyon büyüklüğü, maksimum iterasyon sayısı, çapraz geçiş ve mutasyon olasılık değerleri değiştirilerek farklı çözümler elde edilmiştir. Tekrarlı çalışmalarda birden fazlaoptimum çözüm olduğu görülmüştür. Daha büyük veri setleriyle çalışıldığında bu algoritmanın global optimum sonucu bulma olasılığı daha az olacaktır. Fakat yine de karışık tam sayılı programlama ile kıyaslandığında genetik algoritması, kabul edilebilir çözümler vermiştir. Gerçek dünyadaki karmaşık problemlerde sezgisel algoritmaların kullanılması, hem çözüm süresini tam sayılı programlama yöntemine göre fazlasıyla kısaltacak hem de kabul edilebilir çözümlerin elde edilmesini sağlayacaktır.

Anahtar Kelimeler: Tesis Yeri Seçimi, Tam Sayılı Programlama, Genetik Algoritma, P-Medyan.

Makale Türü: Araştırma

[1] Marmara University, Department of Operational Research, emrenike@hotmail.com

Atıf için (*to cite*): Ekin, E. (2024). Solution Approach To P-Median Facility Location Problem With Integer Programming And Genetic Algorithm. *Afyon Kocatepe Üniversitesi Sosyal Bilimler Dergisi, 26*(2), 547-562.

Introduction

The p-median problem deals with determining the locations of n demand points of p facilities in a way that will meet their needs with minimum cost and assigning the demand points that will receive service to these facilities. The minimum cost does not only include the monetary amount, but also the distance and time between the customer and the facility (Bastı, 2012, s. 49)

This problem was first addressed in 1964 in Hakimi's seminal work. According to Hâkimi, there is at least one optimum solution on the nodes. This problem is applicable where thedemand is regular and stable. For example, it can be used to determine the location of supermarkets and administrative offices. In the p-median problem, the solution time varies exponentially depending on the size of the problem, so the problem is in the NP-hard class (Daskin, 2008, s. 290)

Table 1. P-median solution references (CH: constructive heuristic, LS: local search, MP: mathematical programming, MH: meta-heuristic (Mladenovic, 2007, s. 50)

Type	Heuristic	References
CH	Greedy	Kuehn & Hamburger (1963), Whitaker (1983).
	Stingy	Feldman *et al.* (1966), Moreno-Pérez *et al.* (1991).
	Dual ascent	Galvão (1977, 1980).
	Hybrids	Moreno-Pérez *et al.* (1991), Captivo (1991), Pizzolato (1994).
LS	Alternate	Maranzana (1964).
	Interchange	Teitz & Bart (1968), Whitaker (1983), Hansen & Mladenović (1997), Resende & Warneck (2003), Kochetov & Alkseeva (2003).
MP	Dynamic programming	Hribara and Daskin (1997).
	Lagrangian relaxation	Cornuejols *et al.* (1977), Mulvey & Crowder (1979), Galvão (1980), Beasley (1993), Daskin (1995), Senne & Lorena (2000), Barahona & Anbil (2000), Beltran *et al.* (2004).
MH	Tabu search	Mladenovic *et al.* (1995, 1996), Voss (1996), Rolland *et al.* (1996), Ohlemüller (1997) Salhi (2002), Goncharov & Kochetov (2002).
	Variable neighborhood search	Hansen & Mladenović (1997), Hansen *et al.* (2001), García-López *et al.* (2002), Crainic *et al.* (2004).
	Genetic search	Hosage & Goodchild (1986), Dibbie & Densham (1993), Moreno-Perez *et al.* (1994), Erkut *et al.* (2001), Alp *et al.* (2003).
	Scatter search	García-López *et al.* (2003).
	Simulated Annealing	Murray & Church (1996), Chiyoshi & Galvão (2000), Levanova & Loresh (2004).
	Heuristic concentration	Rosing *et al* (1996), Rosing & ReVelle (1997), Rosing *et al.*, (1999).
	Ant colony	Levanova & Loresh (2004).
	Neural Networks	Domínguez Merino and Muñoz Pérez (2002), Domínguez Merino *et al.* (2003).
	Hybrids	Resende & Warneck (2004).
	Other	Dai & Cheung (1997), Taillard (2003), Kochetov *et al.* (2004).

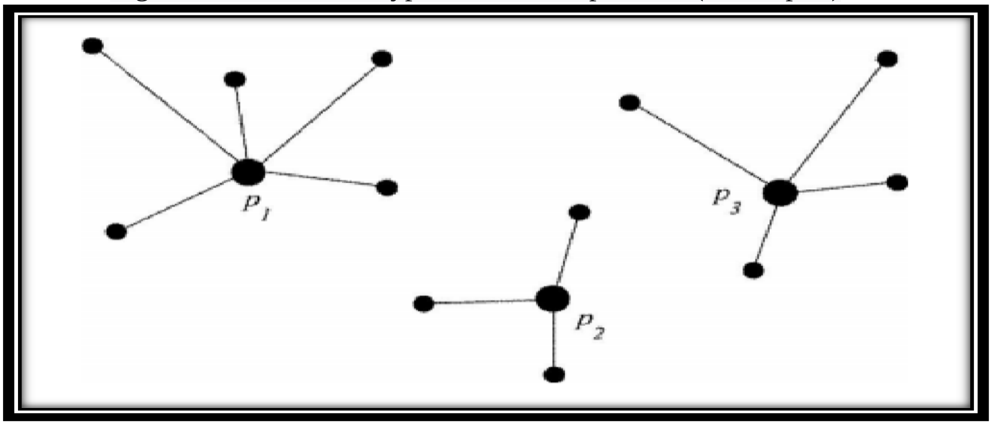
The formulation and solution of the problem with integer programming has been discussed by many in the literature (Mladenovic, 2007, s. 51). Mathematical programming and Lagrangian relaxation methods are exact solutions. Since heuristic approaches generally reach their local optimum values, the use of meta-heuristics that can also obtain global optimum solutions shortens the solution time and produces remarkable results. The most studied meta-heuristic methods in the literature are simulated annealing, genetic algorithm, variable neighborhood search and tabu search (Reese, 2005, s. 2).

1. P-Median

1.1. Features and Assumptions of P-Median

- The general features and assumptions of the p-median problem are as follows (Bastı, 2012, s. 49)
- It is a discrete (only on nodes can be opened facility) and multi-source network location selection model.
- From N candidate plants, P are selected. That is, the number of possible solutions is acombination of N with P (Hakimi, 1965, s. 456)
- Optimum solution cannot be reached in standard time (NP-hard)
- In the problem, cost and distance are directly proportional.
- The number of facilities to be opened (p) is determined.
- There is no time limit.
- There is no cost associated with opening a facility. The customer's demand is constant.
- The features of all facilities are the same (Church & ReVelle, 1976, s. 410).

Figure 1. Solution of a typical P-median problem (N=15, p=3)



1.2. P-Median Mathematical Programming

Formulation Objective function:

$$min \sum_{i=1}^{n} \sum_{j=1}^{n} a_i d_{ij} z_{ij}$$

Constraints:

- $\sum_{j=1}^{n} zij = 1 \quad \forall_i \qquad i, = 1, 2, \dots, n$ (1)
- $z_{ij} \leq y_j \; \forall_{i,j} \qquad i,\ j = 1,\ 2,\ \dots,\ n$ (2)
- $\sum_{j=1}^{n} y_j = p$ (3)
- $\sum_{i=1}^{n} a_i z_{ij} \leq c_j y_j \quad \forall_j \qquad j = 1, 2, \dots, n$ (4)
- $z_{ij},\ y_j \in \{0,\ 1\} \qquad i,\ j = 1,\ 2,\ \dots,\ n$ (5)

Variable and Parameters:

$$\sum_{j=1}^{n} z_{ij} = 1 \quad \forall i \qquad i, = 1,\ 2,\ \dots,\ n$$

$$y_j = \begin{cases} 1 & \text{if facility is established at point j} \\ 0 & \text{In another case} \end{cases}$$

a_i = demand of the customer at point i

d_{ij} = between customer i and facility j short distance (Euclidean)

p = number of facilities to be established (median)

c_j = plant capacity at point j (Durak & Yıldız, 2015, s. 49)

The variable z is the variable associated with the assignment of customer i to facility j, which takes values 0 or 1.

The variable y is the variable that takes the values 0 or 1, equal to 1 if the facility was established at point j, and 0 otherwise

a is the demand parameter of the customer at point i. d is the n x n matrix parameter giving the shortest distance between all customers and facilities. p is the median number to be selected from a certain number of candidate facilities. The objective function minimizes the weighted distance (demand x Euclidean distance) of the selected p number of facilities to the customers they will serve. Constraint 1 ensures that any demand point is served from only one facility. The second constraint ensures consistency and not assigning a demand point to the facility that will not be installed. The third constraint ensures that only p medians are selected from n points. The 4th constraint ensures that no more demand than its capacity is assigned to any of the facilities to be selected. There is no 4th constraint in the capacity less p-median problem. The 5th constraint is the constraint that allows the variables y and z to take binary (0,1) values (Kariv & Hakimi, 1979, s. 542).

2. Integer Programming

One of the basic assumptions of the linear programming model is that all the variables are continuous and the values of the decision variables are integer and fractional. In real life problems, the problem of indivisibility of inputs and outputs requires decision variables to be integer values. Number of tools, number of machines, etc. is an example of this situation. If the optimal solution values obtained by linear programming of any problem are not integers and the solution values are desired to be integers, the manager should apply to integer programming, which is another solution technique. The algorithm used to obtain the integer solution of linear programming problems is integer programming. The variables of the integer linear programming model are discrete, and some or all of these variables are integer values. In this state, it is integer programming, that is, linear programming in which some or all of the variables must be integers (Öztürk, 2002, s. 167).

In Integer Programming, the problems in which some or all of the linear and nonlinear variables are defined discretely are called integer optimization problems. Integer linear programming is a linear programming approach where some or all of its variables take positive integer values. In some cases, the values obtained from the problems can be fractional. However, there are issues where the fractional values are meaningless, or the decision variables must be n integers. e.g; The number of personnel must be an integer (Yalçıner & Can, 2019, s. 381). Integer Programming is a linear programming problem in which some or all of the variables take integer or discrete values. Despite all the research done in the last ten years, no satisfactory development has been achieved in Integer Programming calculations. Today, there is no computer software that fully solves integer programming (Taha, 2000, s. 361).

Difference between Linear Programming and Integer Linear Programming; In the linear programming model, the condition that the decision variables are zero and greater than zero is sought, while in the integer linear programming, the variable values are required to take integer values equal to and greater than zero. There are more than two types of integer Linear programming. According to the integer values that the variables will take. However, here we will focus on two types of integer programming. These; Mixed integer programming and Pure Integer programming. Problems where all variables must be integers are called "pure integer programming problems". An integer probing problem in which all variables are required to be equal to "0" or "1" is called a "0-1 integer programming problem". solution methods are; branch boundary method and cutting method (Balinski, 1965, s. 257).

2.1. Mixed Integer Programming

Mixed Integer programming, k of n decision variables have the condition of being integer, for n-k of them being positive. While the integer condition of some variables is searched, the integer condition is not searched for in some variables. All variables are real numbers. Mixed integer programming has many applications in industrial production, including job shop modeling. An important example in agricultural production planning involves determining the production yield for various crops that can share resources (eg land, labor, capital, seeds, fertilizer, etc.) (Balinski, 1965, s. 259). One possible goal is to maximize total production without exceeding available resources. In some cases, this can be expressed as a linear program, but the variables must be limited to integers. While there are purely integer problems where all variables are integers, if only some of the variables are integers, the problem is called mixed integer programming (Taha, 2000, s. 362).

3. Genetic Algorithm

Production systems have a dynamic structure. For this reason, problems related to production are often complex and difficult. In addition, decisions regarding these problems; It is the type that needs to use a large number of data to be taken within a certain time period with time constraints. Because; Problems related to production systems are mostly NP-hard (Non-

Polynomial) problems. This type of problems; It covers all problems involving algorithms with exponential time complexity functions. Many methods and mathematical models have been proposed to give optimum or near-optimal results for these problems. Among these methods, the most known and the most preferred method in practice, which gives the acceptable solutions to the optimum, is the Genetic Algorithms method.

Genetic algorithm; It is a search technique based on the coding of parameters that try to find a solution using random search techniques. Genetic algorithms are a method that gives acceptable solutions to optimum results, provided that the correct and appropriate parameters are used for a large number of problem types. The working method of the genetic algorithm is based on the natural selection principle of the scientist and researcher Darwin. The genetic algorithm focuses on the evolutionary process of living things in nature (Pullan, 2008, s. 79). The main purpose here is; It is to monitor natural systems by taking into account the harmonious operation and adaptation characteristics and to ensure the design of artificial systems. Although the genetic algorithm does not guarantee optimum results in theory, it provides solutions that can be considered unqualified and of good quality in practice. It even gives the best solution for some problems. For this reason, the genetic algorithm has been successfully applied on optimization problems (Vural, 2005, s. 2).

The genetic algorithm was first developed by the scientist Bremermann in 1958. It was Bagley who introduced the concept of genetic algorithm to the literature and made its first publication in 1967. But the first study on genetic algorithms was made by John Holland, a computer and psychology specialist at the University of Michigan. He explained the details of this theorem in his book Harmony of natural and artificial systems published in 1975. Then, with the book "Genetic Algorithm's Search Optimization and Machine Learning" published in 1989, he revealed that genetic algorithms have practical uses in various subjects (Alp, Erkut, & Drezner, 2004, s. 32).

3.1. Features of Genetic Algorithms

Genetic algorithms use probabilistic transition rules, not deterministic rules. The Genetic Algorithm works on the problem's encoded format, not on the parameter set itself. This does not apply to problems in which real-valued individuals are used. The genetic algorithm does not need derived information or other ancillary information, only the objective function and associated fitness levels affect the direction of the search (Vural, 2005, s. 3).

Optimum determination of the plant location ensures the rational operation of the production sites. Today, it has become important to react quickly to the increasing needs with the increasing competitive environment and to establish the optimum location of the supply and demand points. Genetic algorithm provides optimization solution for complex functions. It is widely used in solving multivariate optimization problems, which are seen as very difficult for classical optimization methods. Genetic algorithm is used to find specific result from a data group. With this feature, it is an ideal optimization technique (Tavakkoli & Shayan, 1998, s. 528).

3.2. Application Areas of Genetic Algorithms

Genetic algorithms; It is used to solve various problems encountered in fields such as software, engineering, operations research, medicine, numerical methods, mathematics and social sciences. Genetic algorithms have the opportunity to obtain good results from multi-dimensional and complex problems. Genetic algorithms not only provide alternative methods of problem solving, but also ensure that many of the problems occur in a way that is consistent with other traditional methods. E.g; Real-world tests in practice in obtaining optimal parameters are the hard way for traditional methods, but suitable and solution can be obtained for genetic processes. The situations in which genetic algorithms are used are as follows. In such cases, the use of genetic algorithms is appropriate. It is used in situations such as if the subject is complex and broad, if the data and information obtained about the subject is scarce, if the solution with traditional methods is unsuccessful and mathematical analysis cannot be done. The genetic algorithm is used for both problem solving and modelling. E.g; It was developed by investigating the adaptation of living things to the environment and their genetic characteristics. The parameters of genetic algorithms are crossover rate, mutation rate, population size, selection, coding, crossover and mutation type (Daskin M. , 1995, s. 105).

3.3 Process Steps of Genetic Algorithm

All possible solutions in the search space are encoded as strings. Usually, a random solution set is chosen and considered as the initial population. The fitness value is calculated for each array, the fitness values found show the solution quality of the arrays. A group of sequences is randomly selected according to a certain probability value and multiplication is performed. The fitness values of the new individuals are calculated and subjected to crossover and mutation processes. The above processes are continued for the predetermined number of generations. When the iteration reaches the specified number of generations, the process is terminated. The most suitable sequence is selected according to the objective function (Gökay & Taşkın, 2002, s. 134).

4. P-Median Solution Application

Before talking about the application part, let's talk about the originality of the study and its difference from other studies. There are various studies in the literature on heuristic approach for P-median and integer programming. However, examples comparing heuristic algorithms and integer programming methods with each other are rarely encountered. For this reason, in the study, p-median has brought a different dimension to the solution of the facility location selection problem. In this aspect of the study, it is evaluated that it differs from other studies and contributes to the literature. The data used in the application were taken from the Operation Research-Library site. Inthe application, the dataset <pmedcap1> from OR-Library was used. In this set, the number of facilities/customers (N) = 50, the median number (p) = 5, and the facility capacity (c) = 120. The solution of capacity constrained P-median Mixed Integer Programming in CPLEX is given in section 4.1 In section 4.2 the capacity-constrained P-median problem is solved with Mixed IntegerProgramming in CPLEX and Genetic Algorithm in Matlab and the results are interpreted.

Table 2. The dataset used in the application

Plant/Customer	X	Y	Demand	Capacity
1	2	62	3	120
2	80	25	14	120
3	36	88	1	120
4	57	23	14	120
5	33	17	19	120
6	76	43	2	120
7	77	85	14	120
8	94	6	6	120
9	89	11	7	120
10	59	72	6	120
11	39	82	10	120
12	87	24	18	120
13	44	76	3	120
14	2	83	6	120

15	19	43	20	120
16	5	27	4	120
17	58	72	14	120
18	14	50	11	120
19	43	18	19	120
20	87	7	15	120
21	11	56	15	120
22	31	16	4	120
23	51	94	13	120
24	55	13	13	120
25	84	57	5	120
26	12	2	16	120
27	53	33	3	120
28	53	10	7	120
29	33	32	14	120
30	69	67	17	120
31	43	5	3	120
32	10	75	3	120
33	8	26	12	120
34	3	1	14	120
35	96	22	20	120
36	6	48	13	120
37	59	22	10	120
38	66	69	9	120
39	22	50	6	120
40	75	21	18	120
41	4	81	7	120
42	41	97	20	120
43	92	34	9	120
44	12	64	1	120
45	60	84	8	120
46	35	100	5	120
47	38	2	1	120
48	9	9	7	120
49	54	59	9	120
50	1	58	2	120

4.1. Mixed Integer Programming Solution For Capacity Constrained P-Median

The mathematical programming formulation of the p-median problem is shared in the second part. In this section, the codes in CPLEX are presented and evaluations are made about the solution of the capacity constrained p-median problem.

Figures 2 and 3 show the codes written in CPLEX. Euclidean distances of customers to each other in Excel.

$d = \sqrt{(x_i - x)^2 + (y_i - y)^2}$ calculated with the formula and transferred to the data file of CPLEX.

Figure 2. Variable definitions and objective function for the P-median problem in CPLEX

```
//Data
int P = ...;
int NCustomers = ...;
int NWarehouses = ...;
range Customers = 1..NCustomers;
range Warehouses = 1..NWarehouses;
int Demand[Customers] = ...;
int Capacity[Warehouses] = ...;
float Distance[Customers][Warehouses] = ...;

//Variables
dvar boolean OpenWarehouse[Warehouses];
dvar boolean ShipToCustomer[Customers][Warehouses];

//Objective
minimize
  sum( c in Customers , w in Warehouses )
    Demand[c]*Distance[c][w]*ShipToCustomer[c][w];

```

Figure 3. Constraints for the P-median problem in CPLEX

```
//Constraints
subject to {
  forall( c in Customers )
    ctShip:
      sum( w in Warehouses )
        ShipToCustomer[c][w] == 1;

  ctOpen:
    sum( w in Warehouses )
      OpenWarehouse[w] == P;

  forall( c in Customers , w in Warehouses )
    ctShipOpen:
      ShipToCustomer[c][w] <= OpenWarehouse[w];

  forall( w in Warehouses )
    ctCapacity:
      sum( c in Customers )
        Demand[c]*ShipToCustomer[c][w] <= Capacity[w]*OpenWarehouse[w];
}
```

The optimum result obtained with the above model and the data set used is given below. The objective function value was found to be 6444.75. There are a total of 2500 z-variables and 50 y-variables. The total number of constraints is 2600.

Table 3. Solution statistics and values (capacity constrained KTP)

Statistics	**Value**
* Cplex	Solution optimal with objective 6444.75
Constraints	2601
* Variables	2551
Dual	2550
Another	1
Nonzero Coefficients	10100
* MIP	
Aim	6.444,75
Settled	6.444,75
Knots	0
The Remaining Nodes	0
Iterations	596
* Solution Pool	
Number	7
Average Purpose	11.159,34

The optimum result was achieved in 596 iterations. There are 7 solutions together with the optimum in the solution pool and the average objective value is 11159.34. The following table shows how the model reaches the optimum result with iterations. 7 cuts were applied in the solution and the optimum solution was found in 0.73 seconds.

Table 4. Improvements with iterations

	Node	Nodes Left	Objective	IInf	Best Integer	Cuts/ Best Bound	ItCnt	Gap
*	0+	0			29292.5300	0.0000	509	100.00%
*	0+	0			13061.1500	0.0000	509	100.00%
*	0+	0			9259.4600	0.0000	509	100.00%
	0	0	6330.6573	45	9259.4600	6330.6573	509	31.63%
*	0+	0			6954.1900	6330.6573	509	8.97%
	0	0	6382.0901	66	6954.1900	Cuts: 4	565	8.23%
*	0+	0			6658.0800	6382.0901	565	4.15%
*	0+	0			6445.2200	6382.0901	596	0.98%
	0	0	6390.7295	68	6445.2200	Cuts: 5	596	0.85%
*	0+	0			6444.7500	6390.7295	596	0.84%
	0	0	cutoff		6444.7500	6444.7500	596	0.00%

Elapsed time = 0.73 sec. (126.90 ticks, tree = 0.00 MB, solutions = 7)

In the optimum solution, y variable values are 1 for y10, y12, y19, y21 and y48. In otherwords, facilities will be established here

The customers who will receive service from these facilities are as follows.y10→ 7, 10, 11, 17, 23, 30, 38, 42, 45 ve 49 numbered customers

y12→ 2, 6, 8, 9, 12, 20 ,25, 35, 40 ve 43 numbered customers

y19→ 4, 5, 19, 22, 24, 27, 28, 29, 31, 37 ve 47 numbered customers

y21→ 1, 3, 13, 14, 15, 18, 21, 32, 36, 39, 41, 44, 46 ve 50 numbered customers

y48→ 16, 26, 33, 34 ve 48 numbered customers

4.2. Mixed Integer Programming and Genetics Algorithm Solution for Capacity Unconstrained P-Median

In this section, the solutions of the p-median problem without capacity constraint will be discussed with both mixed integer programming and genetic algorithm.

4.2.1 Mixed Integer Programming Solution

As mentioned earlier in the second section, there is no capacity constraint in this problem.

Again, the results obtained using the same dataset are as follows

Table 5. Solution statistics and values (capacity unconstrained KTP)

Statistics	Value
* Cplex	Solution optimal with objective 6265.53
Constraints	2551
* Variables	2551
Dual	2550
Another	1
Nonzero Coefficients	7550
* MIP	
Aim	6.265,53
Settled	6.265,53
Knots	0
The Remaining Nodes	0
Iterations	406
* Solution Pool	
Number	3
Average Purpose	14.526,58

The objective value was found to be 6265.53 at the optimum solution. A better result was obtained than the result in section 4.1 due to the lack of capacity constraints. Since the capacity constraint has been removed, there is a total of 2550 constraints and a constraint that ensures that the y,z variables are binary values. The best result was achieved in 406 iterations. There are 3 solutions in the solution pool.

Table 6. Improvements with iterations

```
        Nodes                                         Cuts/
   Node  Left     Objective  IInf  Best Integer    Best Bound    ItCnt     Gap

*     0+    0                       28896.7500        0.0000      406  100.00%
*     0+    0                        8417.4600        0.0000      406  100.00%
*     0     0      integral     0    6265.5300     6265.5300      406    0.00%
Elapsed time = 0.27 sec. (40.68 ticks, tree = 0.00 MB, solutions = 3)
```

It is concluded that y12, y17, y18, y19 ve y48 = 1 in the optimum solution. In other words, facilities will be opened at points 12, 17, 18, 19 and 48.

These facilities will meet the demand of the following customers.y12→2, 6, 8, 9, 12, 20, 35, 40 ve 43 numbered customers

y17→3, 7, 10, 11, 13, 17, 23, 25, 30, 38, 42, 45, 46 ve 49 numbered customers

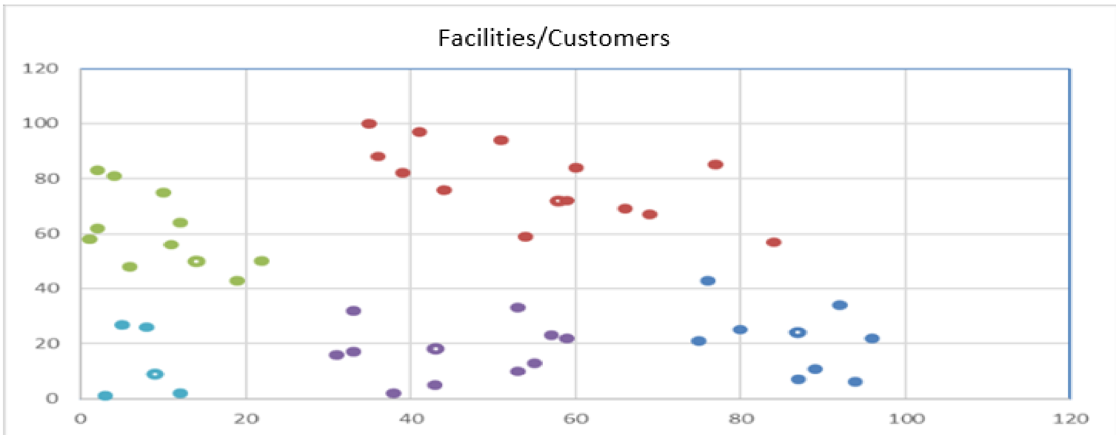
y18→1, 14, 15, 18, 21, 32, 36, 39, 41, 44 ve 50 numbered customers

y19→4, 5, 19, 22, 24, 27, 28, 29, 31, 37 ve 47 numbered customers

y48→16, 26, 33, 34 ve 48 numbered customers

The graph below shows the 5 medians obtained with the optimum result and the clustered customers who will receive service from these medians. The hollow circles (circles) represent the medians and are painted in the same color as the customers receiving service from these medians

Figure 4. Clustering of facilities and customers in problem solving



4.2.2 Genetic Algorithm Solution

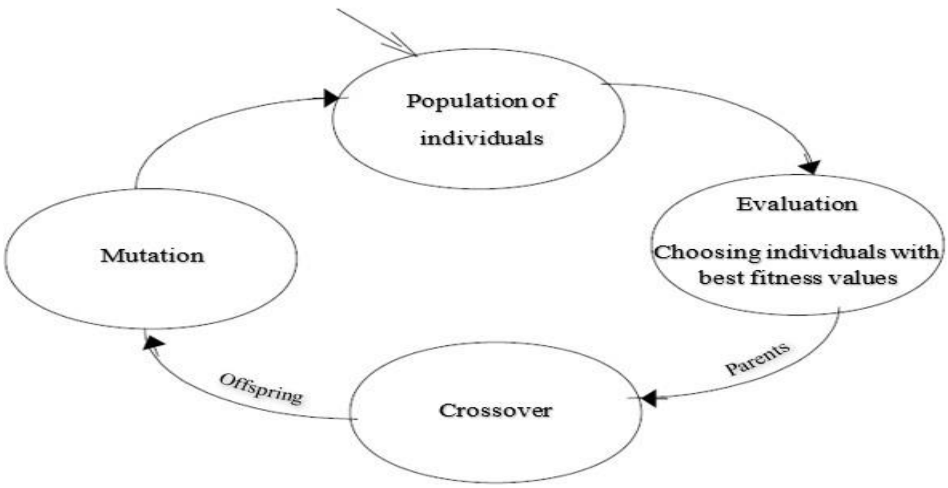
It is an adaptive and global search technique that uses the concepts of genetic algorithm, natural adaptation and selective breeding. In this algorithm, there is a population of individuals and each individual represents a possible solution to a given problem.

Considering for the p-median problem, each individual (chromosome) consists of p number of genes. The allele of each gene represents the index of one of the selected medians. The genetic algorithm creates the starting population in the current population and runs iteratively with various operators until it reaches the termination criterion.

After the initial population is randomly selected, each individual is assigned a fitness value that measures its relative quality within the population. The selection operator favors (favors) individuals with higher fitness values. The crossover operator combines the genetic codes of the parents to form the offspring (individual). The crossover process makes it possible to produce better than good solutions by providing a shaped and random exchange of genetic material between individuals. The mutation operator repairs the missing subregions of the search space by making minor changes to the offspring's genes and prevents the algorithm from converging prematurely. Genetic operators (selection, crossover, mutation) run iteratively until certain criteria are met, which stops the algorithm. In general, when a certain number of generations (occurrences) is reached, the algorithm stops and the last individual obtained is accepted as the solution. Also, if the solution cannot be improved in a certain number of iterations, the algorithm can be stopped.

The steps of the genetic algorithm are shown in the diagram below.

Figure 5. Genetic algorithm steps



After a short description of the genetic algorithm, we can move on to the Matlab application on the <pmedcap1> dataset. This genetic algorithm code for p-median problem is acquired from Code Bus website. Parameters are adapted from <pmedcap1> dataset. The initial parameters used in solving the problem are given below. The solutions were tested by playing with the population size, maximum number of iterations, crossover and mutation probability values.

Maximum number of iterations is set to 10 since more iterations doesn’t provide enough marginal improvement for global optimum solution.

Parameter pm ALLD is part of fitness function and initially set to 1. It is used for weighted distance accumulation.

Crossover probability is selected as 0.9 since we would like the offspring to have best traits combination of the parents according to genetic algorithm.

Mutation probability is set to 0.05 since this parameter is usually taking low values which prevents genetic algorithm to be stuck in suboptimal solutions and also limit genetic mutation to a certain extent.

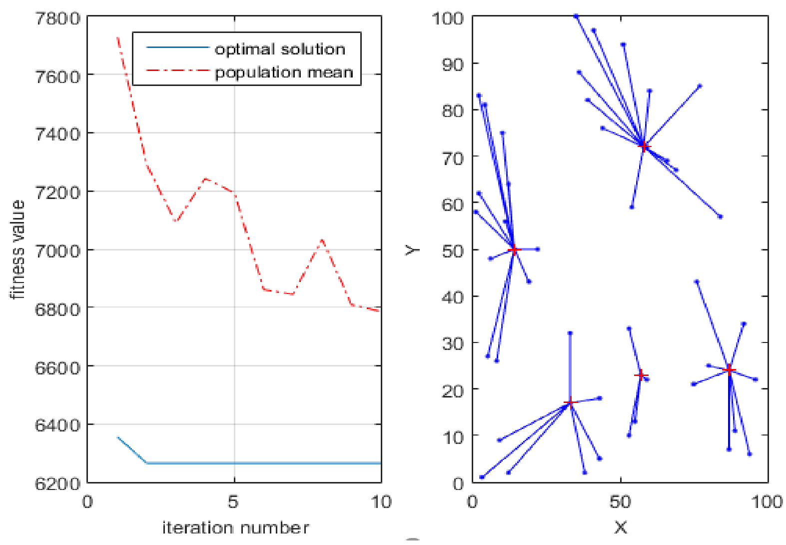
Figure 6. Genetic algorithm parameter

```
%% step 0 %% Setting parameters
pm_Chorm_size = 20;          GMAX = 10;                pm_best_val = 6265.5724;
%Population size             %Max iteration num        %Reference optimal soln
pm_p_size = 5;               pm_ALLD = 1;              gen = 0;
%p value                     %Dist. mediation param    %iteration counter
GGAP = 1;                    Pc=0.9;                   Pm=0.05;
%Generation gap              %Crossover probability    %Mutation probability
```

When the algorithm was run many times with the above settings, a solution other than 6265.5724 could not be reached. Although the optimum solution remained constant in these repeated runs, it was observed that the individuals (median points) changed. This shows that there is a result that provides more than one optimum solution. Below are some of these solutions.

[19;12;38;18;42], [38;15;40;49;42], [12;18;17;41;19], [22;40;38;18;42]

Figure 7. Genetic algorithm results



In the first graph above, it is seen that the population fitness value and the solution values decrease with iterations. The second graph represents the 5 selected medians and their assignment to customers. When the chromosome (population) size and the number of iterations are reduced, the algorithm may not find the global optimum, albeit with a low probability. When working with datasets with larger N and p values, the probability of finding the global optimum result of the algorithm will decrease even more. Some of the other solutions the algorithm found are 6453.0891, 6593.4698 and 6854.8411.

5. Conclusion

In this study, the P-median in facility location problems was mentioned, its mathematical formulation was made, and the problem was solved with mixed integer programming and genetic algorithm. Genetic algorithm, which is a meta-heuristic method, gave very close results to the integer programming method for the dataset we used. While the optimum solution value in integer programming was 6265.53, the best value was found as 6265.5724 as a result of iterative operations with the genetic algorithm. The slight difference was due to the calculation of the Euclidean distance matrix in Excel and rounding in integer programming. Genetic algorithm will give more variable and worse results in large size P-median problems. However, since the solution time will not increase exponentially as in integer programming, using this algorithm in real world problems will be a great advantage and the difference between the solutions will not be large. In addition, the weaknesses of the genetic algorithm can be compensated by increasing the number of iterations and developing hybrid algorithms by using other methods in the selection of the initial population

References

Alp, O., Erkut, E., & Drezner, Z. (2004). An Efficient Genetic Algorithm for the p-Median Problem. *Annals of Operations Research, 122*(1), s. 21-42.

Balinski, M. (1965). Integer Programming, Methods, Uses and Computation. *Manage Science, 12*(3), s. 253-313.

Bastı, D. M. (2012). P-median Facility Site Selection Problem and Solution Approaches. *Online Academic Journal of Information Technology, 3*(7), s. 49.

Church, R. L., & ReVelle, C. S. (1976). Theoretical and Computational Links between the p-Median, Location Set-covering, and the Maximal Covering Location Problem. *Geographical Analysis,, 8*(4), s. 406-415.

Daskin, M. (1995). *Network and Discrete Location: Models, Algorithms and Applications.* New York: John Wiley & Sons, Inc.

Daskin, M. S. (2008). What you should know about location modeling. *Naval Research Logistics (NRL), 55*(4), s. 283-294.

Durak, İ., & Yıldız, M. S. (2015). P- Median Facility Site Selection Problem: An Application. *International Journal of Alanya Faculty of Business, 7*(2), s. 43-64.

Gökay, E. G., & Taşkın, Ç. (2002). Bibliography genetic algorithms and their application areas. *Uludağ University journal of Economics and Administrative Sciences*, s. 129-152.

Hakimi, S. (1965). Optimum Distribution of Switching Centers in a Communication Network and Some Related Graph Theoretic Problems. *Operations Research, 13*, s. 462–475.

Kariv, O., & Hakimi, S. L. (1979). An Algorithmic Approach to Network Location Problems. II: The P-Medians. *SIAM Journal on Applied Mathematics, 37*(3), s. 539-560.

Mladenovic, N. (2007). *The p-median problem: A survey of metaheuristic approaches.* Elsevier.

Öztürk, A. (2002). *Operations Research.* Bursa: Ekin Published.

Pullan, W. (2008). A Population Based Hybrid Metaheuristic for the p-median Problem. *IEEE Congress on Evolutionary Computation,*, s. 76-82.

Reese, J. (2005). Methods for Solving the p-Median Problem: An Annotated Bibliography. *Networks Vol 48*.

Taha, H. A. (2000). *Operations Research.* İstanbul.

Tavakkoli, R., & Shayan, E. (1998). Facilities Layout Design by Genetic Algorithms. *Computers and Industrial Engineering, 35*(3), s. 527-530.

Vural, M. (2005). Mass Production Planning with Genetic Algorithm Method. *ITU Institute of Sciences*, s. 2-3.

Yalçıner, A. Y., & Can, B. (2019). Shelf Space Optimization with Integer Programming and Simulation: Application in a Packaging Company. *European Journal of Science and Technology*, s. 375-388.

ETİK ve BİLİMSEL İLKELER SORUMLULUK BEYANI

Bu çalışmanın tüm hazırlanma süreçlerinde etik kurallara ve bilimsel atıf gösterme ilkelerine riayet edildiğini yazar beyan eder. Aksi bir durumun tespiti halinde Afyon Kocatepe Üniversitesi Sosyal Bilimler Dergisi'nin hiçbir sorumluluğu olmayıp, tüm sorumluluk makale yazarına aittir.

ARAŞTIRMACININ MAKALEYE KATKI ORANI BEYANI

Yazar katkı oranı : % 100