Exploring Genetic Algorithms and Simulated Annealing for Immobile Location-Allocation Problem

Ferran TORRENT-FONTBONA [a], Víctor MUÑOZ [b] and Beatriz LÓPEZ [a]
[a] *Universitat de Girona*
[b] *Newronia S.L.*

Abstract. In this paper we introduce a new kind of immobile Location-Allocation problem that consists in determining the service each facility has to offer in order to maximize the covered demand given the positions of the customers and their service requirements and the positions of the immobile facilities. First, we provide a formalization of the problem and then we tackle the problem using two heuristic methods, genetic algorithms and simulated annealing, comparing the performance of both algorithms.

Keywords. Location-Allocation, Immobile, Genetic Algorithms, Optimization, Simulated Annealing

Introduction

Location-Allocation (LA) is a combinatorial problem that was first introduced in [1]. It consists, first, in determining the positions of k facilities over n possible locations and, second, allocating the customers to their closest facility. Depending on the features of the problem it can be classified into different types according to [2]:

- *P-median*: it consists in finding the positions of the facilities that minimize the sum of weighted distances between facilities and customers.
- *Covering*: it consists in finding the facility locations which provide customers the access to facility service within a specified distance in order to maximize the demand covering.
- *Capacitated*: it is a P-median problem that includes the capacity of the facilities in order to avoid the demand overflow problem.
- *Competitive*: it consists in finding the facility locations where there are other facilities offering the same service. This problem tries to maximize the demand taking into account the competitors.

Out of this classification, there is the immobile version of the problem [3] which consists in locating services when facilities and customers are known, so it consists in determining the service each facility offers in order to optimize a target function. The number of possible solutions of the commented problem is $\left(N_f\right)^s$, where N_f is the number of facilities and s the number of services, what demonstrates its complexity, especially when there are a lot of facilities in the scenario.

In this paper we analyse the performance of Genetic Algorithms (GA) and Simulated Annealing (SA) on solving a given LA problem. Moreover, we contribute with a new neighborhood function for the SA method that improves the performance of it. Other approaches such as Self Organizing Feature Maps (SOFM) (as in [4]) or hybrid approaches could be considered in a future.

The dataset we work with consists of 15578 facility's locations from Catalunya taken from *Páginas Amarillas*. This dataset is used for the clustering analysis exposed in this paper. In addition, the positions of the customers and their type are taken from a random simulation. For the test we generate a random number of customers between 0 and 30 for each facility and then we assigned to each customer its type according a probability function.

This paper is organized as follows. First we present some related work. Second, we formalize the problem giving its mathematical formulation. Next, we expound GA and SA; we present the results achieved with both algorithms and we make a comparison of their performance according to these results. Finally we expound the conclusions we reached and we propose some future work related to the topic of the paper.

1. Related work

LA is a widely studied problem, nevertheless it is still being studied as there have been several publications related to LA and how to solve it in the last years. For example, in [5] some decomposition strategies have been proposed in order to solve large-scale continuous LA problem. Moreover, in [4] Self-Organizing Feature Maps (SOFM) are proposed to tackle LA problem, in [6] a combination of Genetic Algorithm (GA) and Geographical Information Systems (GIS) is analysed to solve the problem and in [7] it is proposed to solve the problem of the ambulance locations using GA.

In [8] the competitive LA is analyzed and the Clonal Selection (CS) algorithm is proposed for solving it. In [9] the performance of some heuristic methods such as GA and Simulated Annealing (SA) is analysed in order to compare these heuristic methods in different LA problems. They conclude that the most suitable method depends on the topology of the problem.

In [10] the authors analyze an heuristic algorithm called Static and Transportation Facility Location Search (STFLS) to solve the problem of locating static facilities (like hospitals) and transportation facilities (like ambulances). One of the steps of the algorithm consists in doing a clustering of the dataset to reduce the search space of the problem. Nevertheless, this clustering technique cannot be applied to our problem because it does not consider different types of demand and facilities, but in [12] some clustering techniques are evaluated to analyze the suitability of a pre-clustering of the data to ease the tough search of an optimal (or nearly optimal) solution.

Finally, as said in the previous section, the most related work for us is [3] where the authors tackle the immobile LA problem. There, they consider an stochastic demand and a single type of customer and facility, then they propose to solve the problem using a GA. However, they do not consider (like the other presented related work) different types of facilities and customers, so different types of services, and they do not test with other optimization techniques like we do in this work.

2. Mathematical model

In the LA problem we want to solve there are a few types of facilities and each type can only serve the customers of the same type, i.e. *blue* customers can only be assigned to *blue* facilities. Moreover, the positions of the facilities and customers are known.

The notation required for the model is the following

N_f number of facilities.
N_c number of customers.
s number of services.
C_i capacity of the ith facility
$\mathbf{x}^q = \left\langle x_1^q, x_2^q, \cdots, x_{N_f}^q \right\rangle$ the qth solution vector. It contains the service each facility offers.
$x_i^q \in \{0, \cdots, s\}$ the service the ith facility offers in the qth solution.
$M_j \in \{0, \cdots, s\}$ the desired match of the jth customer.
z_{ij}^q a variable that is 1 when the jth customer is assigned to the ith facility.
d_{ij}^2 the square Euclidean distance between the jth customer and the ith facility.

Thus, the problem consists in finding the type of each facility that maximizes the global demand (2)

$$\max_{\mathbf{x}^q} \{D(\mathbf{x}^q)\} \tag{1}$$

$$D(\mathbf{x}^q) = \sum_{i=1}^{N_f} \sum_{j=1}^{N_c} \frac{z_{ij}^q}{1 + d_{ij}^2} \tag{2}$$

subject to

$$\forall_i \sum_{j=1}^{N_c} z_{ij}^q \leq C_i \tag{3}$$

$$\forall_j \sum_{i=1}^{N_f} z_{ij}^q \leq 1 \tag{4}$$

$$x_i^q \neq M_j \rightarrow z_{ij}^q = 0 \tag{5}$$

As Equations (1) and (2) show the problem under study is the maximization of the global demand while it is weighted by $\frac{1}{1+d_{ij}^2}$. This means that each customer will be assigned to the closest facility that satisfies the constraints of equations (3) and (5) (the facility has not reached its maximum capacity and it offers the service the customer desires). Equation (4) is a constraint that ensure that no customer will be assigned to more than one facility.

3. Genetic Algorithm Approach

GA [11] exploits the ability of the evolution operators to improve the quality of a population of solutions generation after generation in order to find the optimum solution to a given problem.

To solve the stated LA problem using GA we defined the chromosomes (solutions) as strings of length N_f. These strings contain, in each slot, the service assigned to each facility, thus in the *i*th slot there is the service assigned to the *i*th facility. The GA implementation is shown in Algorithm 1. It uses a population size of 50 chromosomes and 100 generations. However, it also includes a break function that ends the algorithm when no improvement is detected after several generations (steps 11-15).

Algorithm 1 Genetic Algorithm

Require: $N_{chr} = 50$, $MaxGenerations = 100$, $MaxUselessGenerations = 30$,$\mu_{mut} = 0.01$
1: $G = 0$; $UG = 0$; $F_{old} = -\infty$
2: Initialize: create N_{chr} new chromosomes randomly
3: Perform allocation of customers for each chromosome
4: Compute the fitness of each chromosome
5: **while** $(G < MaxGenerations)$ & $(UG < MaxUselessGenerations)$ **do**
6: Create N_{chr} new chromosomes using crossover and mutation operators
7: Perform allocation of each chromosome
8: Compute the fitness of each chromosome
9: Insert the best N_{chr} to the next generation
10: Find the chromosome with maximum fitness F_{max}
11: **if** $F_{max} < F_{old}$ **then**
12: $UG = UG + 1$
13: **else**
14: $UG = 0$; $F_{old} = F_{max}$
15: **end if**
16: $G = G + 1$
17: **end while**

In order to create new chromosomes from the existing ones, crossover and mutation operators have been used. The crossover operator consists in creating a new pair of children combining a pair of parents using a single point crossover. The parents selection is done according to the roulette selection rule that consists in creating a number of copies of each chromosome proportional to chromosome's fitness and put them all into a pool. Then, each time, a pair of copies are randomly selected from this pool to breed a new pair of children what implies that the fittest chromosomes will have more copies and therefore more chances to be selected as parents. The mutation operator consists in changing the service of a facility with a certain probability μ_{mut}. Thus, all slots of all chromosomes have a μ_{mut} probability to change their service and when the service of one slot is changed, another random service is selected to replace the current one. For our experiments we have chosen $\mu_{mut} = 0.01$.

To maintain the population size, a reinsertion operator (step 9) has been used after each new generation breeding. It consists in inserting, to the next generation, the best members of the population between the old and the new generations according the fitness (step 8) of each one. The fitness is calculated according to Equation 2, so the fitness of a solution is the global weighted demand.

Regarding the allocation process (step 7), it is needed to calculate the fitness of each chromosome. Allocation takes into account the maximum capacity of each facility so

the demand assigned to each facility never overflows its capacity. It is done by assigning each customer to the closest facility with the desired service. When a facility reaches its maximum capacity, then the customers that have another facility with the desired service closer than the others are re-allocated to this facility. If there is not another facility to re-allocate the customers, the most remote customers are deleted, so they are not assigned to any facility. Note that this criterion maximizes D as it moves the customers having a close facility as alternative what reduces the impact on D and if there is no option to re-allocate customers to other facilities, it removes the most remote customers that have the lower weight.

4. Simulated Annealing Approach

SA [11], is a heuristic method based on a metallurgy technique that heats and cools the material to move their atoms in order they can reach lower energy states. SA tries to iteratively improve an initial solution with this heating-and-cooling process.

Algorithm 2 shows our implementation of this technique where we defined a solution s as a string containing the service each facility has to offer. Additionally, s has to contain the number of customers assigned to each facility. Then, we defined the energy of a solution as its demand (see Equation (2)), thus the SA algorithm seeks the highest energy solution. To calculate D it is needed to perform the allocation of the customers; which is done following the same criterion stated in Section 3.

Algorithm 2 Simulated Annealing

Require: $T_a = 0.0001$, $T = 1$, $\delta T = 0.99$, $E_{best} = -\infty$
if $\frac{N_{customers}}{N_{facilities}} < 15$ **then**
  $\tau = 0.04$
else
  $\tau = 0.1$
end if
Select an initial random solution s
Compute the energy of s E
while $T_a < T$ **do**
  if $T < 10T_a$ **then**
    $\tau = 0.02$
  end if
  Select a neighbor solution s'
  Compute the energy E' of the new solution
  if $E - E' < 0$ **then**
    $E = E'$; $s = s'$
    if $E_{best} < E$ **then**
      $E_{best} = E$; $s_{best} = s$
    end if
  else
    Select a uniform random number x between 0 and 1
    if $x < e^{-\frac{E-E'}{T}}$ **then**
      $E = E'$; $s = s'$
    end if
  end if
  $T = \delta T \cdot T$
end while

Traditionally, simulated annealing algorithm generates a new s' solution from the current one s moving the algorithm to a neighbor point, i.e. it evaluates the target funtion on a neighbor point of the current position of the algorithm. Therefore, the algorithm needs a coordinate system where the neighbor point selection can be defined. Nevertheless, the search space of the problem we concern does not have a coordinate system. Thus, we defined a new neighborhood function that states that a neighbor solution s' is that where most of the facilities offer the same service as in s and just a few of them change the service they offer. There are several manners to select which facilities should change the service and which should not. One way to select which facilities should change their service is making a random selection but also assigning different probabilities to each facility depending in some paramenters. Here we propose and analyze four different probability functions that depend on different parameters.

1. Exponential probability with variable τ: the probability function is defined by Equation (6), where N_i is the number of customers assigned to the ith facility and C_i its capacity. Thus the probability to change the service depends on the number of customers the facilty has and the variable τ that depends on the ratio $\frac{N_{customers}}{N_{facilities}}$ (steps 1-5 Algorithm 2) and the phase where the algorithm is (steps 9-11 Algorithm 2).
2. Exponential probability with $\tau = 0.05$: the probability function is also defined by Equation (6), but τ is a constant parameter; so the probability to change the match just depends on the number of customers of the facility.
3. Uniform probability with variable τ: all facilities have the same chances to change their service since this probability function does not depend on the number of customers the facilities have assigned. So, each facility has a probability of τ to change its service. Nevertheless, τ depends on the ratio $\frac{N_{customers}}{N_{facilities}}$ (steps 1-5 Algorithm 2) and the phase where the algorithm is (steps 9-11 Algorithm 2).
4. Uniform probability with $\tau = 0.05$: all facilities have the same probability (τ) to change their match and this probability is always the same, $\tau = 0.05$.

$$P(\text{change the } i\text{th service}) = e^{-\frac{N_i/C_i}{\tau}} \tag{6}$$

Regarding the probability functions defined before, note that both exponential probability functions state that the emptier the facilities are the more chances they have to change their match. Moreover, regarding the paramater τ, we empirically found that with low $\frac{N_{customers}}{N_{facilities}}$ a lower τ works better because since all facilities would be emptier it reduces the probability of change. Additionally, we reduce τ (in cases 1 and 3) when the algorithm reaches the final iterations to force the convergence. The different values assigned to τ can be seen in Algorithm 2.

To compare the performances of the four neighborhood functions we used three measures:

- D: the value of the global weighted demand (Equation 2). The higher it is the better the perfomance of the algorithm.
- % of allocated customers: this value indicates the percentage of satisfied customers, thus the percentage of customers assigned to a facility that offers the service the require. The higher it is the better.

		Exponential prob. with variable τ			Exponential prob. with $\tau = 0.05$		
N_f	N_c	D	% allocated customers	% facil. with occup. $< 4\%$	D	% allocated customers	% facil. with occup. $< 4\%$
18	300	**217.04**	**95.33**	**0.00**	211.34	94.00	**0.00**
18	138	**104.43**	97.82	**5.56**	103.85	**98.55**	16.67
72	1403	**1223.49**	**99.43**	**0.00**	1218.94	98.93	**0.00**
72	699	616.49	99.86	**4.17**	**616.55**	**100**	**4.17**
127	2177	2010.62	**100**	**0.00**	**2013.74**	**100**	0.79
127	1066	**996.03**	**100**	9.45	994.11	**100**	**8.66**
313	5932	**5579.03**	**99.83**	**0.32**	5571.28	99.71	0.96
313	2783	**2622.78**	99.86	**6.39**	2622.36	99.89	8.95
		Uniform prob. with variable τ			Uniform prob. with $\tau = 0.05$		
N_f	N_c	D	% allocated customers	% facil. with occup. $< 4\%$	D	% allocated customers	% facil. with occup. $< 4\%$
18	300	214.45	95.00	**0.00**	216.15	93.00	**0.00**
18	138	103.04	**98.55**	11.11	104.01	96.38	16.67
72	1403	1221.93	98.93	**0.00**	1218.18	98.93	2.78
72	699	614.95	99.86	6.94	613.67	99.86	8.33
127	2177	2005.71	**100**	6.30	2007.23	**100**	10.24
127	1066	993.98	**100**	14.96	991.81	**100**	18.11
313	5932	5535.93	99.73	15.34	5531.09	99.68	13.10
313	2783	2612.07	**99.96**	28.43	2606.94	99.75	29.07

Table 1. LA results using SA with different neighborhood functions. Best results are in bold face.

- % of facilities with an occupation lower than 4%: this measure indicates the percentage of empty facilities, which are those with an occupation ($\frac{N_i}{C_i}$) lower than 4%. So, it indicates the number of underused facilities. Note that in our problem we cannot change the number of facilities and their positions, so we cannot remove this underused facilities. This measure may indicate that there is another better solution because the resources (facilities) are underused. The lower this value is the better.

Table 1 shows the results obtained with the different neighborhood functions. As it shows, the best results are generally obtained with the first neighborhood function, what means that increasing the chances a facility has to change its service when it has a low occupation, improve the solution search. Moreover, reducing the value of τ in the last iterations of the algorithm also improves the performance of the algorithm. If we compare the second and third neighborhood functions we can see that the use of an exponential function which depends on the occupation and the use of the variable τ have, approximately the same relevance, since it is not clear which of both has the best perfomance. Nevertheless it seems that when there are a lot of facilities in the scenario, it is more relevant the use of the exponential function, but when there are a few facilities, the variable τ takes more relevance.

5. Comparison

To analyse the results obtained with both algorithms, GA and SA, we also computed an additional simple method which solves individual LA where each facility decides its

N_f	N_c	D			% of allocated customers		
		Individual	GA	SA	Individual	GA	SA
8	170	76.97	**106.73**	106.27	56.47	**80.00**	**80.00**
18	361	165.32	**252.00**	250.26	54.57	95.01	**96.12**
42	859	424.46	**719.40**	736.21	54.37	99.88	**100.00**
46	707	348.48	542.36	**560.20**	55.02	97.17	**97.60**
48	921	473.54	792.45	**805.75**	55.48	99.67	**99.89**
50	968	470.02	764.65	**790.84**	53.93	98.67	**98.86**
72	1335	687.08	1168.23	**1195.90**	55.13	99.33	**99.40**
127	2196	1155.13	2003.44	**2041.05**	56.19	**100.00**	**100.00**
313	5779	2938.99	5106.05	**5407.44**	55.67	96.12	**99.78**
1495	27762	15217.71	-	**27662.54**	55.73	-	**99.97**

N_f	N_c	% of facil. with occupation $< 4\%$			Elapsed time (s)		
		Individual	GA	SA	Individual	GA	SA
8	170	**0.00**	**0.00**	**0.00**	**0.000**	0.428	0.110
18	361	**0.00**	**0.00**	**0.00**	**0.005**	2.218	0.545
42	859	**2.38**	7.14	**2.38**	**0.009**	16.358	4.09
46	707	19.57	6.52	**0.00**	**0.008**	8.160	1.826
48	921	6.25	6.25	**0.00**	**0.011**	20.967	5.564
50	968	10.00	8.00	**2.00**	**0.004**	16.311	4.142
72	1335	11.11	6.94	**0.00**	**0.020**	41.709	12.303
127	2196	12.60	7.09	**1.57**	**0.022**	77.88	44.796
313	5779	15.34	12.14	**0.96**	**0.135**	566.098	318.244
1495	27762	11.84	-	**0.33**	**3.607**	-	6130.372

Table 2. LA results. Best results are in bold face.

service by simply selecting the service that maximizes its demand without taking into account the other facilities. Table 2 presents the results obtained with the three methods using different situations with different facilities and customers but the number of services is constant: 5. Individual is the designed label to the simple method results. In addition to the previous measures ("D", "% of allocated costumers" and "% facil. with ocupation $< 4\%$") we have also analysed the Elapsed time in seconds. The computer we used to test the algorithms is an Intel(R) Core(TM) i5 760 @ 2.80GHz with 8.00 GB of RAM and Windows 7.

It can be seen in Table 2 that the individual LA is the fastest method as it does not have to perform several iterations to reach the final result, nevertheless it is the worst method as it is not able to provide service to a lot of customers what implies that its results have the lowest global weighted demand D.

Regarding GA and SA, the slowest part of these algorithms is the allocation of the customers as the algorithms have to calculate all distances between customers and facilities to find the best allocation. Between them, the fastest algorithm is SA because it has to perform a lower number of allocations (GA has to do N_{chr} at each iteration). Also, this part of the algorithm consumes a lot of memory resources limiting the size of the problem to be handled. This is the reason we do not have GA results for the problem with 1495 facilities (last row of Table 2). So, in terms of computational resources, SA is much more efficient than GA. Moreover, SA provides better results than GA as it is able to find a greater D and is able to allocate a greater number of customers in most cases. It is also able to find results with a lower number of empty facilities and so it better distributes the

demand between all the facilities. This fact is induced by the SA neighborhood function that tends to change more the services of the less occupied facilities. Additionally, we can see that GA and SA provide similar results when the problem has a low complexity, so when the problem only includes a few facilities; but when the problem becomes more complex (more facilities), the solutions found by GA have a poorer quality, only SA is able to maintain the quality of the results.

Finally, as conclusion we can say that LA results are much better when we try to maximize the global demand instead of maximizing the demand of each facility separately (individual LA). Moreover, SA is better than GA in all terms (speed, memory resources, solution quality and demand distribution) especially when the problems are more complex. This latter conclusion implies that, for this kind of problem, improving a single solution using our neighborhood function is better than improving the quality of a population of solutions using crossover and mutation operators.

6. Conclusion and Future Work

In this paper we presented two approaches for solving the Capacitated Immobile LA. Particularly we explored GA and SA methods to solve the presented problem. In so doing we defined a new SA neighborhood function that allows the algorithm to find a better solution given a certain number of iterations. Finally, we presented the results obtained after performing LA with SA and GA and we compared them in order to find the best algorithm, which has been SA. We also presented the LA results obtained after maximizing the weighted demand of each facility separately to proof this method provides the worst results (lower D and lower occupation of the facilities) than maximizing the global weighted demand.

Nevertheless, the use of SA is unfeasible for problems with a lot of facilities, thus it may be important to find more efficient algorithms for large scale problems. We are currently exploring partition methods to reduce the problem [12].

It would be also interesting to analyze the performance of some complete optimization techniques, in order to compare the solutions found by the heuristic methods presented here and the optimal solutions as well as the needed computational resources by the algorithms.

Moreover, it would be interesting to complete this study using real customers data and real distances, not the Euclidean distance between facilities and customers. So, taking into account the streets and other obstacles that increase the distances. Even, it would be interesting to use time distances instead of physical distances, or to combine both.

References

[1] L. Cooper. Location-Allocation problems, *Operations Research* (1963), 331-343.

[2] R.L. Church. Location modelling and GIS, *Geographical Information Systems* **1** (1999), 293-303.

[3] S.H.R. Pasandideh and S.T.A. Niaki, Genetic application in facility location problem with random demand within queuing framework, *Journal of Intelligent Manufacturing*, **23** 3 (2012), 651-659.

[4] K.H. Hsieh and F.C. Tien, Self-organizing feature maps for solving location-allocation problems with rectilinear distances, *Computers & Operations Research* **31** 7 (2004), 1017-1031.

[5] J. Brimberg and P. Hansen and N. Mladenović, Decomposition strtegies for large-scale continuous location-allocation problems, *IMA Journal of Management Mathematics* **17** 4 (2006), 307-316.

[6] X. Li and A.G.O. Yeh, Integration of genetic algorithms and GIS for optimal location search, *International Journal of Geographical Information Science* **19** 5 (2005), 581-601.

[7] S. Sasaki and A.J. Comber, Using genetic algorithms to optimise current and future health planning - hte example of ambulance locations, *International Journal of Health Geographics* **9** 1 (2010), 4-14.

[8] P. Xiu-Li and F. Yu-Qiang, Solving competitive facilities location problem with the clonal selection algorithm, *International Conference on Management Science and Engineering (ICMSE'06)* (2006), 413-417

[9] M.A. Arostegui and S.N. Kadipasaoglu and B.M. Khumawala, An empirical comparison of tabu search, simulated annealing, and genetic algorithms for facilities location problems. *International Journal of Production Economics* **103** 2 (2006), 742-754.

[10] W. GU and X. Wang. Studies on the perfomance of a heuristic algorithm for static and transportation facility location allocation problem. Proceeding HPCA'09 Proceedigns of the Second International Conference on High Perfomance and Applications (2010), 27-37.

[11] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, Pearson, New Jersey, 2010 (third edition).

[12] F.Torrent and V. Muñoz and B. López, An Experimental Analysis of Clustering Algorithms for Supporting Location-Allocation, *Submitted to AI 2012*.