Meta heuristic algorithms for weighted p-median problem:
A case study for locating IKEA stores

Yujiao Li
2018-05-21

Abstract
p-Median problem is a well-known discrete optimization problem aiming to locate p number of facilities that satisfies the demand of multiple places with minimum cost. There are several papers developed the solution for p-median problem based on popular algorithms. But their comparison of performance in practical data has not been studied, but only simulated data with limited size. This work applied algorithms in the big data about IKEA location, aiming to find the suboptimal solution of allocations. The studied one is Simulated Annealing, Genetic algorithm, Particle Swarm Optimization, Artificial Bee Algorithm. In addition, this work developed Artifical Fish Algorithm to solve p-median problem. The best one turns out to be Artificial Bee Algorithm in terms of convergence speed and fitness of optimal solution.

Keywords
P-median problem, Simulate Annealing; Genetic algorithm; Particle Swarm Optimization; Artificial Bee Colony algorithm; Artificial fish algorithm

1. Introduction
p-Median problem is a well-known discrete facility-location optimization problem. The goal is to select the locations of p facilities to serve n demand points so that the total distances from each demand point to its nearest facility is minimized. Usually to every demand point v_i is also assigned a weight w_i , so the problem actually asks for the minimum of weighted sum distance. P-median problem is NP-hard (Kariv and Hakimi, 1979) which is unlikely to obtain an optimal solution through polynomial time-bounded algorithms. As the problem size increases, the computation time of exact methods increases exponentially.

Some heuristic and meta-heuristic methods are proposed for the positive and negative weighted p-median problems. Heuristic algorithms have acceptable time and memory requirements but do not guarantee optimal solution. The primitive heuristic methods applied to the p-median problem is a greedy heuristic by Kuehn and Hamburger (1963). The interchange and vertex substitution method of Teits and Bart (1968). Meta-heuristic algorithms become very popular and developed to solve the discrete P-median problems with positive or negative weight case. Previous studies apply different algorithms to solve this problem: Levanova and Loresh (2004) implemented the ant systems and simulated annealing algorithm (SA); Jafar Fathali (2006) use genetic algorithm (GA); Sevkli et. al (2012) developed discrete particle swarm optimization (PSO); Jayalakshmi B. and Singh A. (2016) designed a hybrid bee colony algorithm (ABC).

Compared with SA, other new algorithms claimed to outperform SA for discrete p-median problem in previous studies. But there is no complete comparison among all of them in empirical studies. Moreover, this paper proposed an approach based on artificial fish algorithm (AFA) without knowing its power and aim to compare the performance of above-mentioned methods using one practical dataset. Their performances in solving P-median problem have not been compared completely and there is no fully developed programming package for this sake. The author thus developed integrate R source code for all of five algorithms for users to solve practical p-median problems by setting various parameters.

2. Problems and Algorithms

2.1 Problem description

p-median problem is to allocate p facilities among n demand points in such a way that the distance from the demand point to the nearest facility is minimized. The mathematical formulation for p-median problem (ReVelle and Swain, 1970) is to solve $\min \sum_{i=1}^{D} \sum_{j=1}^{N} h_i d_{ij} Y_{ij}$ which subject to $\sum_{j=1}^{N} Y_{ij} = 1, \forall i, 1 \leq i \leq D$ and $\sum_{j=1}^{N} X_j = p$ to guarantee each customer can only choose one facility and there are p facilities in total. h_i denotes the weight of the demand of the customer i, d_{ij} is the distance between customer i and facility j, X_j is the decision variable indicating the facility j is selected or not. Y_{ij} is the decision variable, indicating whether the customer i is served by facility j.

2.2 Optimization algorithms

Five algorithms are separately tested and compared: SA, GA, PSO, BEE, FISH. This section presents their pseudo code which are developed in either recent papers or this work. To reproduce the results, the author developed source R code for five algorithms[1].

SA is concerning one solution's updating and others consider the population of solutions updating. Thus, other algorithms have advantage of improving fitness faster in the beginning than SA. This work refer the neighbor

(1) Simulated Annealing (SA)

Simulated annealing (SA) is developed by Kirkpatrick et al., (1983) to deal with highly nonlinear problems by emulating the physical process of cooling energy configuration. Each step of the SA algorithm aims to update the current solution by a random solution. The measure of quality of a solution is called fitness. The new solution is accepted if its fitness is better than the original solution; otherwise, it is accepted with some probability to avoid the trap into local optimal. Intuitively, the probability depends on two factors: the worse of new solution's fitness and the longer period of retrieving, the lower probability to accept the new solution. The most common function form of the acceptance probability is $exp(-\delta/T)$ where δ is the increase in the objective function, and T is the control parameter (temperature) which decay as iteration times. α denotes the rate of decay shown in following pseudo code of SA. Note that the neighborhood solution hereby is updated by replacing one out of p facility location in original solution. The applied code is majorly based what Meng (2014) developed.

SA Algorithm Pseudo code

```
Initialize a location solution, Temperature parameter T;
Evaluate the location solution
while (!stopCondition)
        Select a new location solution from the neighborhood
        Compute fitness of the new solution
        Calculate the fitness improvement of the new one compared with the previous solution: δ
```

[1] https://github.com/liyujiao1026/pmediansolvers

if (δ >0) {Accept new location}
else {Accept with probability of $exp(-\delta/T)$}
$T = \alpha \cdot T$ $(0 < \alpha < 1)$
end while

(2) Genetic algorithm (GA)

Genetic algorithms (GAs) are search heuristic methods for solving combinatorial optimization problems by analogy of chromosomes. Essential idea is elite parent solutions could produce good child solution. Each generation of solutions are generated by crossover and mutation of parent ones. The crossover is used to produce new chromosomes from a pair of selected chromosomes. Mutations are used to promote genetic diversity.

To solve the p-median problem, the GA is implemented as follows. To eliminate bad solutions from initial population solutions, those individuals where fitness is lower than specific quantile (p_1) are removed. The updated population solutions are produced by resampling from the filtered solutions with replacement where the sampling probability is based on fitness of individuals, i.e., better fitness indicates higher probability to be selected; Then the selected individuals are used for implementing cross-over with probability of p_2 and mutation with probability of p_3. Cross-over step is to exchange one solution's several elements with another solution's counterpart elements. Mutation step for one solution is to replace one original element with another randomly selected element. If the solutions improve the optimal objective function, the global solution is updated.

GA Algorithm Pseudo code

Initialize population of location solution
Evaluate population of location solution
while (!stopCondition)
Select the best-fit individuals of location solution for reproduction with percentage of p_1;
Breed new individuals through crossover and mutation with probability p_2 and p_3
Evaluate the individual fitness of new individuals;
Update individuals depending on object function value's improvement;
end while

(3) Particle Swarm Optimization (PSO)

Sevkli et al (2012) develop a novel discrete particle swarm optimization for p-median problem. This algorithm starts with a randomly generated initial population. Each individual solution is updated simultaneously. Each solution has its personal best (pbest) as integration goes, and global best solution (gbest) is always updated based on all of personal best solution. For every iteration, each solution S_i have three options for updating. The first alternative solution S_1 is randomly searching neighbour with probability p_1; The second alternative one is using the personal best solution with probability p_2; and the third one is using the global best solution and p_3. The updated solution selects the one with the best fitness. Equation for three of them are $S_1 = w^t \oplus \eta(X_i^t)$, $w^{t+1} = w.\beta$ where w (inertia weight) is evenly decreased from 1 to 0 as iteration goes; $S_2 = c_1 \oplus \eta(P_i^t)$; $S_3 = c_2 \oplus \eta(G^t)$. The updated solution $X_i^{t+1} = best(S_1, S_2, S_3)$. Then pbest and gbest solution will be updated if the new solution is better than the orginal one.

PSO Algorithm Pseudo code

Initialize population of location solution
Calculate fitness of each location solution
Calculate pbest and gbest for each location solution

while (!stopCondition)
    for (each individual location solution)
        Create S_1: updated current solution S_i with probability p_1;
        Create S_2: updated its pbest solution $S_{i,pb}$ with probability p_2;
        Create S_3: updated current gbest S_{gb} solution with probability p_3;
        Compute their fitness of above three solutions S_1, S_2, S_3, and the best one named as S_{opt}
        Update the current solution S_i as S_{opt}
        Update pbest location $S_{i,pb}$
        Update gbest location S_{gb}
    end for
end while

---

(4) Artificial Bee Colony algorithm (ABC)

Jayalakshm (2016) developed A hybrid artificial bee colony algorithm for the p-median problem with positive/negative weights. Each solution generated by ABC algorithm is improved by an interchange based randomized local search. In addition, an interchange based exhaustive local search is applied on some of the best solutions obtained after the execution of ABC algorithm in a bid to further improve their quality.

The initial solutions for ABC algorithm are generated using the random method. The fitness of initial solutions is calculated to update the next one, i.e. better fitness should have higher probability to be selected in the updated one (roulette method). After the new solutions population generated, each of solution will search its neighbor solution. Global solution will be updated if the individual improves the original solution.

ABC Algorithm Pseudo code

Initialize population of location solution
Calculate fitness of each location solution
while (!stopCondition)
    Update each solution as its better neighbor (local greedy search)
    Calculate fitness of each solution
    Update population solutions by roulette method
    Update global best solution
    if (some individual solutions stop improving for max_trail times) {
        set these solutions as a new one randomly}
end while

---

(5) Artificial fish algorithm (AFA)

Artificial Fish Algorithm (AFA), which was presented by X. L. Li in 2002, is a new swarm intelligence optimization method by simulating fish swarm behaviour. But it has not been developed as a solution for p-median based on the Artificial fish algorithm. This work describes the steps of the algorithm in details.

AFA means fish swarm can always find food at a place where there are a lots of food, and hence more fish distributed. The essential idea is that when all fish are too close with each, some individuals should jump from current cluster for the sake of efficiency. Fishes are treated as in the same cluster if the number of mutual element in solution is ω. If the peers will swarm to this centre to form a cluster. All fish will follow to the best solution in the current cluster. In summary, in the moving process of the fish swarm, when a single fish or several fish find food, the neighbourhood partners will trail and reach the food quickly. According to this phenomenon, AFA solution for p-median problem are developed as follows:

AFA Algorithm Pseudo code
Initialize population of location solution
Calculate fitness of each location solution
while (!stopCondition)
Update each solution as its better neighbor (local greedy search)
Solutions cluster
if (the current solution has suitable number of peers (crowdedness θ))
{move to the cluster center
move to the best solution in the current cluster}
else {local greedy search}
Update global best solution
end while

3. Results

To compare the performance of five algorithms, the experiment firstly used simulation data for simplicity. Then the practical dataset about IKEA locations are applied

3.1 Simulation

Algorithms' relative percentile deviation is taken as a performance measure to compare the performance of the algorithms. Relative percentile deviation is taken as a performance measure to compare the performance of the algorithms. $\frac{f-f_{opt}}{f_{opt}}.100$ where f denotes the fitness of best solution found until each iteration by the algorithm and f_{opt} denotes the optimum value of the objective function.

In simulated data, number of number of potential facility location is N=193, required number of facility is p=5, and number of demand points are 187, the distance matrix is generated from Poisson distribution with $\lambda = 30$. Except SA has only one solution to update iteratively, other algorithms have the population size of 50. Parameters settings for each algorithm are in Table 1. Figure 1 shows that ABC algorithm outperform others in solving p-median problem. The speed of its achieving optimal solution is the fastest, and its optimal value is the best compared with other algorithms. The computing time for them are in Table 2. Considering AFA takes much time than other, it is not applied in IKEA data.

Table 1. parameters in algorithms

SA	GA	PSO	ABC	AFA
$T = 500$	$p_1 = 0.9$ $p_2 = 0.8$ $p_3 = 0.2$	$c_1 = 0.7$ $c_2 = 0.3$	$trial = 100$	$\theta = 5$ $\omega = 10$ $trial = 10$

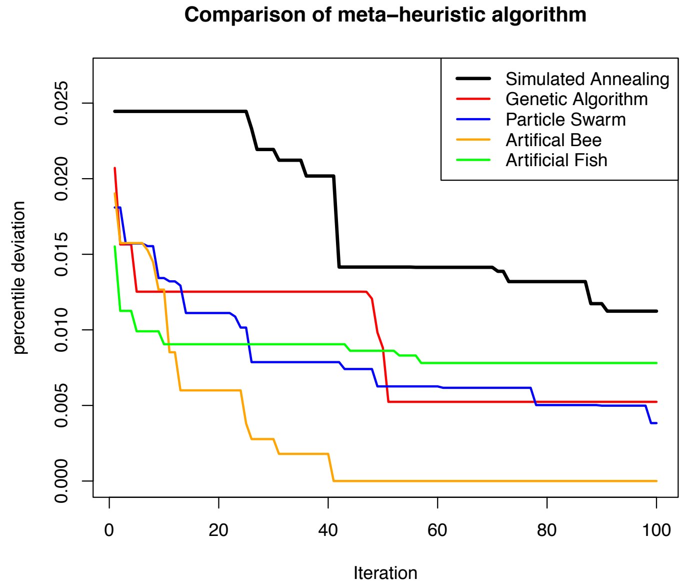

Figure 1. comparison of algorithms' performance

Table 2. computing time

Algorithm	SA	GA	PSO	BEE	AFA
Running time (seconds)	0.049	5.20	6.92	3.13	29.94

3.2 IKEA location data

Based on above comparison results, ABC and PSO achieve the better optimal value, while the computing time of SA is the shortest. In empirical data, this work will only apply above three algorithms due to their better performance than others. IKEA dataset is composed of distance matrix between every demand point and all potential facility locations. The number of number of potential facility location is N=1938 required number of facility is p=19, and number of demand points are 187679. Except SA has only one solution to update iteratively, other algorithms have the population size of 50. Iteration time is 200 times and Parameters for each algorithm are identical with settings in Table 1.

Currently, there are 19 IKEA stores in Sweden. The current total distance of all demand points to their closest IKEA stores is 284603084729. Table 2 presents the optimal distance and running time of three preferable algorithms. It concludes that three algorithms would achieve better location solution due to shorter distance cost. But Since IKEA store is a practical issue which cannot be updated or re-located. It is more meaningful to identify the next IKEA store by algorithms this work developed. The converge procedure is presented in Figure 2.

Table 2. comparison of algorithms' performance in IKEA data

Algorithms	SA	ABC	PSO
Optimal Value (meters)	272495200025	236614110898	238188838063
Time	2.218 mins	8.894 hours	15.701hours

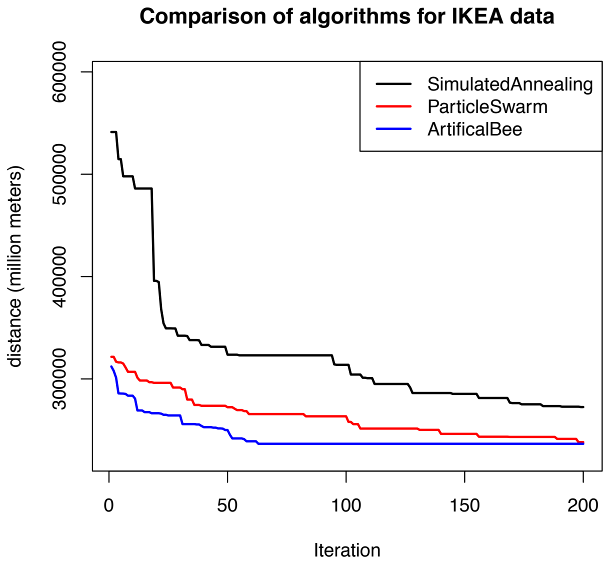

Figure 2

4. Summary

This work compared five popular algorithms performance in solving p-median problem. This problem is to allocate p facilities in a way that the total distance between n demand points and facilities is minimized. Heuristic algorithms generally have acceptable time and memory requirements, but do not guarantee optimal solution. They give a feasible solution, which is likely to be either optimal or near optimal. In our simulated data, ABC algorithm outperform others with faster convergence and better fitness. In the practical IKEA location data, ABC algorithm is still the best one. To assist researchers to implement investigated five algorithms, this work developed R source code for five algorithms.

Reference

Carling K., Meng X., 2016, On statistical bounds of heuristic solutions to location problems, J Comb Optim, 31,1518–1549

Fathali J., 2006, A genetic algorithm for the p-median problem with pos/neg weights, Applied Mathematics and Computation 183,1071–1083

Jayalakshmi B., Sing A., 2017, A hybrid artificial bee colony algorithm for the p-median problem with positive/negative weights,

Kariv, O., Hakimi, S.L., 1979, An algorithmic approach to network location problems. II. The p-medians, SIAM Journal on Applied Mathematics 37 (3), 539–560.

Kuehn, A.A., Hamburger, M.J., 1963, A heuristic program for locating warehouses, Management Science 9 (4), 643–666.

Sevklia, M., Mamedsaidovb, R., Camcic, F., 2014, A novel discrete particle swarm optimization for p-median problem, Journal of King Saud University-Engineering Sciences, 26 (1), 11-19