TECHNICAL TRANSACTIONS 1/2018
ELECTRICAL ENGINEERING
DOI: 10.4467/2353737XCT.18.008.7959
SUBMISSION OF THE FINAL VERSION: 2/1/2018

Krzysztof Schiff (kschiff@pk.edu.pl)
Department of Automatic Control and Information Technology, Faculty of Electrical Engineering, Cracow University of Technology

Ant colony optimisation algorithm for the facility localisation problem

Algorytm mrówkowy dla problemu lokalizacji punktów obsługi

Abstract

This article describes a new ant colony optimisation algorithm for the facility localisation problem with a new heuristic pattern proposed by the author, which consists of three parts: the function of the average cost of client servicing; the total minimum cost of servicing from a site, which is selected and included into the solution; the function of improving the cost of already serviced clients. In this comparison, simulations were presented, and two parameters were observed: the number of sites and the cost of client servicing. The new algorithm allowed to improve the solution in both of these parameters.

Keywords: facility localisation problem, ant colony optimisation, heuristic algorithm

Streszczenie

W artykule przedstawiono algorytm mrówkowy dla problemu lokalizacji fabryk z nową zaproponowaną heurystyką wyboru obiektów i został on porównany z innym znanym już z literatury przedmiotu algorytmem mrówkowym. Nowa heurystyka wyboru została wyrażona jako iloraz trzech funkcji pożądania wyboru, to jest funkcji określającej średni koszt obsługi klientów poprzez włączaną lokalizację do rozwiązania, funkcję określająca całkowitą minimalną sumę obsługiwania klientów z włączanej do rozwiązania lokalizacji oraz funkcję określająca maksymalną minimalizację kosztów obsługiwania klientów poprzez włączaną lokalizację, gdy ci klienci są już obsługiwani przez lokalizacje wybrane do rozwiązania. W artykule przedstawiono wyniki przeprowadzonych testów pod kątem uzyskania jak najmniejszej liczby lokalizacji i jak najmniejszego kosztu obsługiwania klientów w funkcji rozmiaru problemu i natężenia obsługiwania klientów z danej lokalizacji.

Słowa kluczowe: problem lokalizacji punków obsługi, algorytm mrówkowy, heurystyka

1. Introduction

The facility localisation problem is the most important logistic problem. It is a very important decision for managers to consider where to locate new facilities, such as retailers, warehouses or factories. This optimisation problem is the problem of locating *p* facilities relative to a set of customers in such a manner that the sum of the shortest demand weighted distance between the customers and the facilities is minimised. It is shown by Kariv and Hakimi (1979) that the facility placement problem is NP-hard. Exact methods take a lot of time in order to reach solutions [8, 9]; therefore, most often, heuristic methods are used to solve this problem [4, 6, 7, 10–13] and, among them, meta-heuristics algorithms, such as genetic algorithms [1, 14–16, 19, 20] or ant colony algorithms [2, 3, 5, 17, 18].

The facility localisation problem consists of a set of customers N where $N = \{1, 2, \dots n\}$ and a set of potential sites for the facilities M where $M = \{1, 2, \dots, m\}$. It is assumed that the cost of transporting the w_i units of product from a facility *j* to the customer *i* is *cij*; for each $i \in N$ and $j \in M$, there is no fixed cost for locating at a particular site and there is no capacity constraint on the demand supplied by the facility. The problem is to choose *p* of the *m* sites where a facility will be located in such a way that the total transportation cost is minimised.

A mathematical formulation of the problem is as follows:

$$\text{Min} \sum\nolimits_{i=1}^{n} \sum\nolimits_{i=1}^{n} c_{ij} x_{ij} \tag{1.1}$$

$$\sum\nolimits_{i=1}^{n} x_{ij} = 1 \tag{1.2}$$

$$\sum\nolimits_{i=1}^{m} y_j = p \tag{1.3}$$

$$x_{ij} \leq y_j \forall i \in n, \forall j \in M \tag{1.4}$$

$$x_{ij}, y_j \{0,1\} \forall i \in N, \forall j \in M \tag{1.5}$$

where p – the number of facilities,
y – the site for the facility to be located at
$y_{j=1}$ if a facility is located at site *j*, otherwise y j =0 $\forall\ j \in M$
and x_{ij} =1 if customer *i* is serviced by a facility at site *j*, otherwise *xij* =0 $\forall\ i \in N, j \in M$.

Constraints 1.2 guarantee that each customer is assigned to a facility. Constraints 1.3 ensure that p sites are chosen. Constraints 1.4 guarantee that a customer selects a site only from those that are chosen. Constraints 1.5 force the variables to be an integer.

2. Structure of the ACO algorithm

In ant algorithms, a colony of artificial ants is looking for a good quality solution to the problem. A pseudo-code of the ACO procedure is presents as algorithm 1. Each ant constructs an entire solution. In order to construct a solution, each ant uses common information, which is deposited is sites; this means a set of sites selected and included in the solution.

Each ant selects a new site j with probability *pj* using the pheromone trail τ_j and the attractiveness *nj* of the move (2.1). These to parameters can be intensified by power expressed by coefficients α and β. The neighbourhood *NHi* of the already selected sites *Si* consists of sites, which can be added to the already selected sites *Si*. *NHi* = *M- Si –Vex*, where *Vex* are sites, which cannot be included into set *Si,* since they service only those customers who are serviced from sites already selected and included in the set *Si*.

$$p_j = \begin{cases} \dfrac{\tau_j^{\alpha} n_j^{\beta}}{\sum j \in N_i \tau_j^{\alpha} \mu_j^{\beta}}, & \mathit{for}\ j \in NH_i \\ 0 & \mathit{for}\ j \notin NH_i \end{cases} \qquad (2.1)$$

Algorithm 1

ACO algorithm

```
while (exist cycle)
begin
    while (exist any ant, which has not worked)
    begin
        while (a solution has not been completed)
        begin
            choose a next site to be added to the already selected sites Si with a probability pj
            update NHi – a neighbourhood of the set Si
        end
        update a best solution if a better solution has been found
    end
    update a global best solution if a better solution has been found
    update a pheromone trails $\tau(i) = \tau(i) + \Delta\tau$
end
```

After a solution has been found, each ant deposits a pheromone with a quantity *Δτ* on all sites, which constitute the solution, in accordance with the pattern (2.2). Thus, those sites, which were included in the solution, have received an additional quantity of the pheromone and can be chosen to a solution with a higher probability than other sites.

$$\tau = \tau + \Delta\tau \qquad (2.2)$$

The quantity of the deposited pheromones *Δτ* is expressed by formula 2.3, where: *z* – the current solution, z_{best} – the best solution.

$$\Delta\tau = f(Q) = \frac{1}{1 + \dfrac{Z_{best} - Z}{Z_{best}}} \qquad (2.3)$$

Thus, those sites, which were included in the solution, have received an additional quantity of pheromones and can be selected afterwards with a higher probability than other objects.

An evaporation mechanism is incorporated into the ant algorithms in order to avoid too rapid a convergence to a suboptimal solution. The intensity of evaporation is controlled by the parameter ρ. The quantity of a pheromone on each object is updated at the end of each cycle in accordance with the pattern:

$$\tau=\rho\tau,\ \rho\in(0,1] \tag{2.4}$$

3. Improved Hybrid ACO algorithm

The algorithm presented in this paper, called the Improved Hybrid ACO algorithm (IHACO), is a modified version of the ant algorithm described in [5], which would be called the Hybrid ACO algorithm (HACO). The new algorithm is called the Improved Hybrid ACO algorithm since a desirability function is presented in a new form in comparison with the Hybrid ACO algorithm. In both of these algorithms, site *j* is selected to be in set S_i with probability

$$p(j)=\frac{\tau(j)n(j)}{\sum k\in NH_i\left(\tau(k)n(k)\right)},NH_i\in M \tag{3.1}$$

where:
NH_i – a neighbourhood of the set S_i
$NH_i = M\text{-}S_i - V_{ex}$
V_{ex} – these are sites, which cannot be included into the set S_i since they service customers who are serviced from sites already included in set *Si*
$\tau(j)$ – the quantity of pheromone deposited on site *j,*
$n\ (j)$ – the desirability of selecting site *j.*

In both algorithms, a following site that should be added to a partial solution is selected with a probability that depends on the pheromone trail, heuristic information and transition rule. The m differences between the elaborated algorithm and the algorithm, which is presented in paper [5], concern the heuristic information. The heuristic pattern, which has been presented in paper [5], is described by

$$n(j)=\frac{c}{\sum_{j=1}^{n}w_j},j\in NH_i \tag{3.2}$$

and a new heuristic formula presented in this paper by $n(j)=n1(j)n2(j)n3(j)$; this is the main contribution presented in this paper, where

$$n1(j)=\frac{nc}{\sum_{j=1}^{n}w_j},j\in NH_i \tag{3.3}$$

$$n2\left(j\right)=\sum_{j=1}^{n}\left(w_{\max}-w_{jk}\right), j\in NH_{i}, k\in N_{n}, N_{n}=N-N_{s} \tag{3.4}$$

$$n3\left(j\right)=\sum_{j=1}^{n}\left(w_{jk}-w_{ik}\right), i\in S_{i}, k\in N_{S}, j\in NH_{i} \tag{3.5}$$

where:
nc – the number of customers, which can be potentially serviced from site *j*
w_{max} – the maximal cost taken from all costs of customer servicing
w_{ij} – the cost of servicing customer *j* from site *i*
N_n – customers not yet serviced by any site already included into the set *Si*
N_s – customers serviced by sites already included into the set *Si*
S_i – already selected sites

The desirability function $n(j)$ of selecting site *j* consists of: $n1(j)$ – this is the inverse of the cost attributable to client servicing from site *j*, $n2(j)$ – this allows to choose site *j* at a minimum cost for all clients who are serviced from site *j* and who are not serviced already, $n3(j)$ – this allow to choose site j, which can lower the cost of servicing clients who are already serviced by sites included in solution S_i. The quantity of pheromone Δτ is deposited by ants during one cycle of the algorithm on all sites of the set S_i

$$\Delta\tau=\frac{1}{1-\frac{c_{best}-c}{c_{best}}} \tag{3.6}$$

where:
c_{best} – the minimum global cost of customer servicing from all cycles,
c – the actual cost of customers from the current cycle.

4. Results of experiments

The first is the HACO algorithm, which was described in [5], and the second is the IHACO algorithm, which is described in this paper and presents a new desirability function. The two main parameters of the facility placement problem under observation are: the average cost and the average number of localisation sites, which were received as a result of 50 measures. The number of cycles, the number of ants and the evaporation rate were taken from experience of dealing with ant algorithms, and under the conducted investigation, these parameters are not so important since we will see that the quality of the solution of the IHACO algorithm is better than the HACO algorithm. The number of cycles and the number of ants cause time constraints, so under these time constraints, the IHACO algorithm provides a better solution than the HACO algorithm. When we change the value of the evaporation rate, we cannot see any relevant influence on the obtained solutions, so these results were omitted and were not shown here. In this paper, only the most important results of experiments were shown.

During the first experiment, both algorithms were studied for a bipartite graph with 100x100 vertices and for different graph densities *q*, which were generated randomly. One set of the vertices represents clients and the second set of vertices represents the potential sites. The weight assigned to the edge represents the cost of servicing. Any edge links a client with a site. These graphs with a density *q* belong to the particular group of graphs, since each edge exists with probability *q* and each vertex has an almost equal degree, meaning that it almost has an equal number of edges. The graph density *q* is a model of the proportion of all clients, which can be serviced by a site, so these proportions are equal for all sites.

The average costs for the 50 measures were presented in Table 1 and in Figure 1. The average number of localisation sites for the 50 measures was presented in Table 2 and in Figure 2.

These algorithms operate with the following parameters: the evaporation rate was set to *r* = 0.995, the number of ants was set to *lm* = 30 and the number of cycles was set to *lc* = 100.

Table 1. Average minimum cost of localisation for ***n*** = 100, ***lc*** = 100, lm = 30, ***r*** = 0.995

q	**0.1**	**0.3**	**0.5**	**0.7**	**0.9**
HACO	2251.62	1852.19	1788.58	1800.06	2161.2
IHACO	1880.48	1672.22	1660.11	1714.59	2088.8

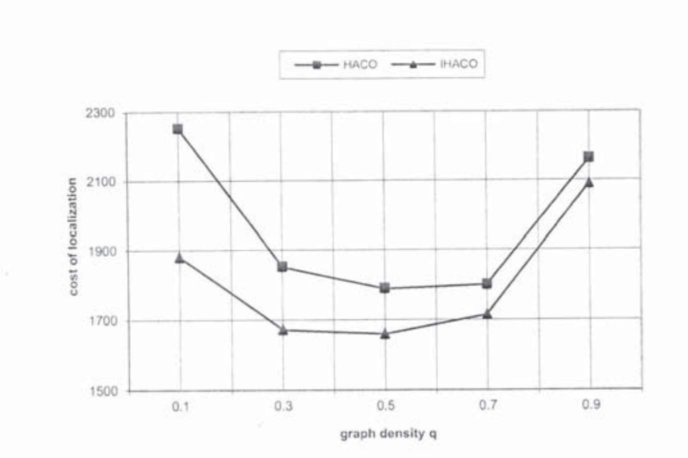

Fig. 1. Average minimum cost of localisation for *n* = 100, *lc* = 100, *lm* = 30, *r* = 0.995

Table 2. Average number of localisation sites ***n*** = 100, ***lc*** = 100, ***lm*** = 30, ***r*** = 0.995

q	**0.1**	**0.3**	**0.5**	**0.7**	**0.9**
HACO	26.98	11.74	7.2	4.95	3
IHACO	25.42	11.04	6.98	4.88	3

During the second experiment, both algorithms were studied for a constant graph density *q* = 0.5 and for a different graph size *n* = {50, 100, 150, 200, 250}. The average costs for the 50 measures were presented in Table 3 and in Figure 3.

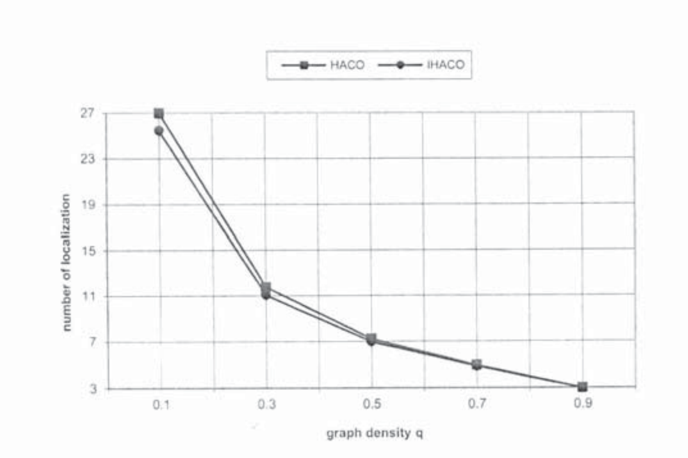

Fig. 2. Average number of localisation sites $n = 100$, $lc = 100$, $lm = 30$, $r = 0.995$.

The average number of localisation sites for the 50 measures was presented in Table 4 and in Figure 4.

These algorithms operate with the following parameters: the evaporation rate was set to $r = 0.995$, the number of ants was set to $lm = 30$ and the number of cycles was set to $lc = 100$.

Table 3. Average minimum cost of localisation for $q = 0{,}5$, $lc = 100$, $lm = 30$, $r = 0.995$.

n	50	100	150	200	250
HACO	925.48	1788.58	2635.98	3405.41	4232.29
IHACO	834.4	1660.11	2490.35	3303.22	4118.43

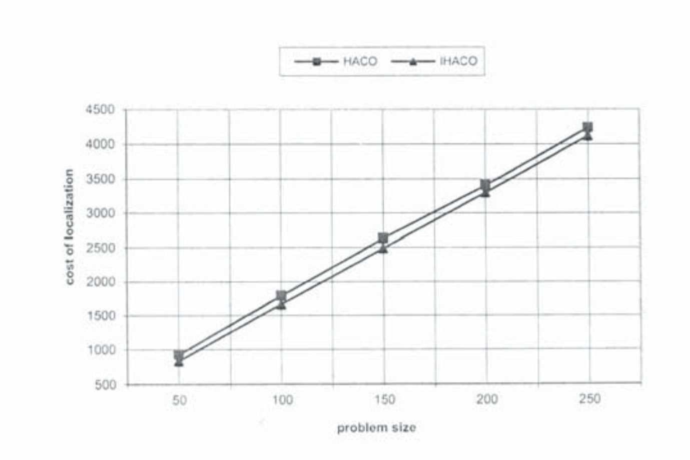

Fig. 3. Average minimum cost of localisation for $q = 0{,}5$, $lc = 100$, $lm = 30$, $r = 0.995$.

There is an improvement in the quality of the solution when the IHACO algorithm is used instead of the HACO algorithm, since there are lower average numbers of localisation sites and costs for the investigated graph density $q = \{0.1, 0.3, 0.5, 0.7, 0.9\}$ and the size of the problem $n = \{50, 100, 150, 200, 250\}$.

Table 4. Average number of localisation sites for $q = 0{,}5$, $lc = 100$, $lm = 30$, $r = 0.995$.

n	50	100	150	200	250
HACO	5.93	7.2	7.97	8.57	8.99
IHACO	5.87	6.98	7.73	8.39	8.65

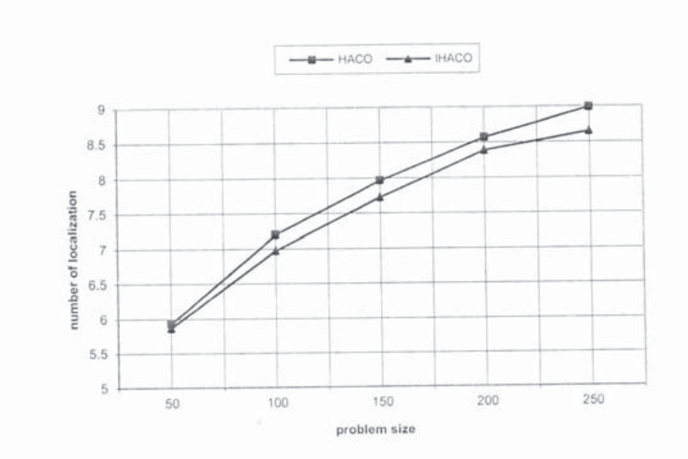

Fig. 4. Average number of localisation sites $q = 0{,}5$, $lc = 100$, $lm = 30$, $r = 0.995$

5. Conclusion

This paper presents a new ant algorithm and, primarily, a new desirability function in the probabilistic formula of site selection. This improvement makes the algorithm more efficient in searching for localisation sites at a minimum cost. Ants in the new elaborated algorithm take into consideration not only new unselected sites from set M, which service as many customers as possible from set N at a minimum cost, but also the already serviced costumers from set N, which can be serviced by these new sites from the M at a lower cost. This strategy makes this new elaborated ant algorithm more efficient than others in searching for a minimum number of facility sites at a minimum cost.

References

[1] Pasricha R., Wadhwa V., *Optimizing facility location problem using genetic algorithm, comparison between ACO algorithms for the set covering problem*, International Journal of Computer Science and Telecommunication, Vol. 2, Issue 3, 2011, 13–15.

[2] Varnamkhasti M.J., *Overview of algorithms for solving the p-median facility location problems, comparison between ACO algorithms*, Advanced studies in biology, Vol. 4, No. 2, 2012, 49–55.

[3] Lessing L., Dumitrescu I., Stutze T., *A comparison between ACO algorithms for the set covering problem*, LNCS, Vol. 3172, 2004, 1–12.

[4] Lghoseiri K., Ghannadpour S.F., *An efficient heuristic method for capacitated p-median problem*, International Journal of Management Science, Vol. 4, 2009, No. 1, 72–80.

[5] Crawford B., Soto R., Momfroy E., Paredes F., Palma W., *A hybrid Ant algorithm for the set covering problem*, International Journal of the Physical Sciences, Vol. 6/19, 2011, 4667–4673.

[6] Antomoshkin A.N., Kazakovtsev L.A., *Random search algorithm for the p-median problem*, Informatica, Vol. 37, 2013, 267–278.

[7] Sevkli M., Mamedsaidov R., Camci F., *A novel discrete particle swarm optimization for p-median problem*, Journal of King Saud University – Engineering Sciences, Vol. 26, 2014, 11–19.

[8] Balas E., Carrera M.C., *A dynamic subgradient-based branch-and-bound procedure for set covering*, Operations Research, Vol. 44/6, 1996, 875–890.

[9] Fisher M.L., Kedia P., *Optimal solution of set covering/partitioning problems using dual heuristics*, Management Science, Vol. 36/6, 1990, 674–688.

[10] Chvatal V., *A greedy heuristic for the set-covering problem*, Oper. Research, Vol. 4/3, 1979, 233–235.

[11] Lan G., Depuy G.W., *On the effectiveness of incorporating randomness into a multi-star -heuristic with application to the set covering problem*, Computer Industrial Engineer., Vol. 51, No. 3, 2006, 362–374.

[12] Cerias S., Nobili P., Sassano A., *A Lagrangian-based heuristic for large-scale set covering problems*, Mathematical Programming, Vol. 81, 1998, 215–228.

[13] Caprara A., Fischetti M., Toth P., *A heuristic method for the set covering problem*, Operations Research, Vol. 47/5, 1999, 730–743.

[14] Chiou Y., Lan L.W., *Genetic clustering algorithms*, European Journal of Operational Research, Vol. 2, No. 132, 2001, 413–427.

[15] Bozkaya B., Zhang J., Erkut E., *An efficient genetic algorithm for the p-median problem*, in Facility Location: Applications and Theory Operations Research, Berlin, Springer, 2002, 179–205.

[16] Alp O., ErkutE., Drezner Z., *An efficient genetic algorithm for the p-median problem*, Annals of Operations Research, Vol. 122, 2003, 21–42.

[17] Lessing L., Dumitrescu I., Stutzle T., *A comparison between ACO algorithms for the set covering problem*, [in:] *Ant Colony Optimization and Swarm Intelligence*, LNCS, Vol. 3172, Springer Verlag, Berlin, 2004, 1–12.

[18] Crawford B., Castro C., *Integrating look ahead and post processing procedures with ACO for solving set partitioning and covering problems*, [in:] *Proc. of the 8th international conference on Artificial Intelligence and Soft Computing*, Springer–Verlag, Berlin, Heidelberg, 2006, 1082–1090.

[19] Lim A., Hu Z., *A fixed-length subset genetic algorithm for the p-median problem*, LNCS, Vol. 2724, Springer Verlag, Berlin, 2003, 1596–1597.

[20] Lorena A.N., Furtado J.C., *A Constructive genetic algorithm for clustering problems*, Evolutionary Computation, Vol. 3, No. 9, 2001, 309–328.