

Mesa, Armacheska; Castromayor, Kris; Garillos-Manliguez, Cinmayii; Calag, Vicente

## Article

# Cuckoo search via Levy flights applied to uncapacitated facility location problem

Journal of Industrial Engineering International

## Provided in Cooperation with:

Islamic Azad University (IAU), Tehran

*Suggested Citation:* Mesa, Armacheska; Castromayor, Kris; Garillos-Manliguez, Cinmayii; Calag, Vicente (2018) : Cuckoo search via Levy flights applied to uncapacitated facility location problem, Journal of Industrial Engineering International, ISSN 2251-712X, Springer, Heidelberg, Vol. 14, Iss. 3, pp. 585-592,  
<https://doi.org/10.1007/s40092-017-0248-0>

This Version is available at:

<https://hdl.handle.net/10419/195626>

### Standard-Nutzungsbedingungen:

Die Dokumente auf EconStor dürfen zu eigenen wissenschaftlichen Zwecken und zum Privatgebrauch gespeichert und kopiert werden.

Sie dürfen die Dokumente nicht für öffentliche oder kommerzielle Zwecke vervielfältigen, öffentlich ausstellen, öffentlich zugänglich machen, vertreiben oder anderweitig nutzen.

Sofern die Verfasser die Dokumente unter Open-Content-Lizenzen (insbesondere CC-Lizenzen) zur Verfügung gestellt haben sollten, gelten abweichend von diesen Nutzungsbedingungen die in der dort genannten Lizenz gewährten Nutzungsrechte.

### Terms of use:

*Documents in EconStor may be saved and copied for your personal and scholarly purposes.*

*You are not to copy documents for public or commercial purposes, to exhibit the documents publicly, to make them publicly available on the internet, or to distribute or otherwise use the documents in public.*

*If the documents have been made available under an Open Content Licence (especially Creative Commons Licences), you may exercise further usage rights as specified in the indicated licence.*



<https://creativecommons.org/licenses/by/4.0/>

# Cuckoo search via Levy flights applied to uncapacitated facility location problem

Armacheska Mesa<sup>1</sup> · Kris Castromayor<sup>1</sup> · Cinmayii Garillos-Manliguez<sup>1</sup> · Vicente Calag<sup>1</sup>

Received: 22 August 2017 / Accepted: 8 November 2017 / Published online: 18 November 2017  
© The Author(s) 2017. This article is an open access publication

**Abstract** Facility location problem (FLP) is a mathematical way to optimally locate facilities within a set of candidates to satisfy the requirements of a given set of clients. This study addressed the uncapacitated FLP as it assures that the capacity of every selected facility is finite. Thus, even if the demand is not known, which often is the case, in reality, organizations may still be able to take strategic decisions such as locating the facilities. There are different approaches relevant to the uncapacitated FLP. Here, the cuckoo search via Lévy flight (CS-LF) was used to solve the problem. Though hybrid methods produce better results, this study employed CS-LF to determine first its potential in finding solutions for the problem, particularly when applied to a real-world problem. The method was applied to the data set obtained from a department store in Davao City, Philippines. Results showed that applying CS-LF yielded better facility locations compared to particle swarm optimization and other existing algorithms. Although these results showed that CS-LF is a promising method to solve this particular problem, further studies on other FLP are recommended to establish a strong foundation of the capability of CS-LF in solving FLP.

**Keywords** FLP · CS-FLP · Optimization problem · Metaheuristics · Hybrid · Algorithms

## Introduction

Facility location problem (FLP) is a technique that may aid any organization's decision making process. It is a mathematical way to locate the optimal location of facilities to satisfy the clients' demands. In reality, however, it is seldom possible to predict the quantity of demand when organizations take strategic decisions such as locating facilities (Silva and Serrab 2005). For this situation, the problem can be modeled as uncapacitated FLP (UFLP) which this study was addressed to. In the UFLP, aside from the assumption that the total cost has to be minimized, the problem has two components, namely the set-up cost of operating a facility and the transportation cost of satisfying the clients' requirements from a facility. It also assumes that the capacity of every facility is infinite (Guner and Sevkli 2008). This implies that even with an unknown demand, it is still possible to locate facilities since it is assumed that one facility is able to satisfy all the demands of the clients.

Cuckoo search (CS) is a metaheuristic algorithm developed by Yang and Deb in 2009, which is based on the behavior of breeding parasitic cuckoos (Yang 2010). It was inspired by the obligate brood parasitism of some cuckoo species by laying their eggs in the nests of other host birds. Some host birds can engage direct conflict with the intruding cuckoos. For example, if a host bird discovers the eggs are not their own, it will either throw these alien eggs away or simply abandon its nest and build a new nest elsewhere. Some cuckoo species such as the new world brood-parasitic *Tapera* have evolved in such a way that

✉ Armacheska Mesa  
armesa1@up.edu.ph

Kris Castromayor  
kriscastromayor@gmail.com

Cinmayii Garillos-Manliguez  
cgmanliguez@up.edu.ph

Vicente Calag  
vbcalag@up.edu.ph

<sup>1</sup> University of the Philippines Mindanao, Davao City, Philippines

female parasitic cuckoos are often very specialized in the mimicry in colors and pattern of the eggs of a few chosen host species (Payne et al. 2005).

In simple representation of the cuckoo search, Yang and Deb (2009) formulated the following rules. First, each cuckoo lays one egg at a time and dumps its egg in randomly chosen nest. Second, the best nests with high quality of eggs will be carried over to the next generation. Third, the number of available host nests is fixed, and the egg laid by a cuckoo is discovered by the host bird with a probability  $P_a \in (0, 1)$ . In this case, the host bird can either get rid of the egg or simply abandon the nest and build a completely new nest. The solutions are the eggs in the nest and the cuckoo egg is the best and new solution among the solutions in the nest.

The study of Reynolds and Fyre (2007) on Lévy flights states that most animals and insects explore their landscapes using a series of straight flight paths punctuated by a sudden 90° turn. This type of behavior is called the Lévy-flights-style intermittent scale-free search pattern and has been applied to optimization and produces optimal preliminary search results (Yang 2009).

Zaw and Mon (2013) used cuckoo search via Lévy flight in web document clustering. Their study considered locating the optimal centroids of a cluster. The cuckoo search via Lévy flight appeared to produce better results compared to particle swarm optimization. Cuckoo search was also used as an approach to solve nurse scheduling problem in Tein and Ramli's (2010) study. Their paper aimed to use cuckoo search to optimize time scheduling of their duties in the hospital. Results showed that cuckoo search was able to produce better solutions. In engineering optimization problems such as the welded beam, cuckoo search algorithm was able to obtain a better solution than the other algorithms (Ragsdell and Phillips 1976, Cagnina et al. 2008). Santillan et al. (2017) also explored the capability of cuckoo search with Lévy flight for solving the capacitated vehicle routing problem. The study demonstrated the potential of the hybrid algorithm in acquiring solutions for routing problems.

Based on the literature, CS exhibits better performance compared to other evolutionary techniques when applied to optimization problems. Yang (2010) stated that CS is potentially far more efficient than particle swarm optimization and genetic algorithms. Yang and Deb (2009) used benchmark test functions such as De Jong's first function, Easom, Shuber bivariate, Griewank, Ackley, Rosenbrock, Schwefel, Rastrigin, and Michalewicz to test the performance of optimization algorithms such as particle swarm optimization, genetic algorithm, and CS. It appears that CS has 100% success rate of finding the global optima with the average mean of 927 with a standard deviation of 105. Cuckoo metaheuristic algorithm was also used to

optimize warranty periods in a heterogeneous customer population (Roozitalla and Asgharizadeh 2013). The algorithm was used to estimate the optimal warranty duration and it was observed in the study that there was a dramatic increase in manufacture's revenue.

This study aimed to use CS-LF to find approximations to the solutions of the UFLP, specifically, to evaluate the performance of CS-LF using several combinations of parameter settings in terms of best solution and running time;

The cuckoo search via Lévy flights technique served as another method in solving uncapacitated facility location problem. This study determines if the technique was able to generate the facility location and then compares it to the existing locations of the facilities.

The results of this study may benefit organizations with the same nature of data used in this study. This study focused only on one type of FLP, the uncapacitated FLP, which is restricted on the data obtained from a certain company. This study also serves as an additional contribution in UFLP and may lead to another research in the future. Opening or closing of facilities was not considered in this study. The parameters for CS-LF were based on Yang's study (2010).

## Materials and methods

### Data for uncapacitated FLP

The data, specifically, its layouts were taken from a department store. The department store is divided into different sections. These were the basis for the set of demand points for every section. It should be noted that the layout presented in this study is a constricted version of the actual layout in this study.

On the other hand, the counters were considered as the set of facilities. However, the current locations of these existing facilities in the department store were disregarded as this study aimed to minimize the total cost by optimally locating the facilities in the search space and not with the opening and closing of them.

The search space has a dimension of 125.6129 m by 50.245161-m area. This was represented in the Cartesian plane with a scale ration of 1 mm. That is, in the layout, the search space is 295 × 118 mm area. This was achieved by getting the location of each point which is the center of each section in the department store. For those sections in regular forms, the center is easily determined; for those in irregular forms, the center is determined by adapting the closest regular form of the irregular regions.

For the set of facilities, their locations should be within the search space. However, since this study is applied to

real data, there were solutions obtained which are not possible in reality. These are the shaded regions in Fig. 1. From the same figure, the set of demand points is also illustrated.

Mathematically, let  $I$  be the fixed set of  $m$  demand points with element  $d_i$ , where  $i = 1, 2, \dots, m$ . From Table 1, their locations are known. Each demand point  $d_i \in I$  was represented as a coordinate  $(d_{xi}, d_{yi})$  in the Cartesian plane. Thus,  $I$  was illustrated as a  $2 \times m$  matrix where the  $i$ th column represents the coordinate of the  $i$ th demand point, that is,

$$I = \begin{pmatrix} d_{x1}d_{x2} \dots d_{xm} \\ d_{y1}d_{y2} \dots d_{ym} \end{pmatrix} \tag{1}$$

Similarly, let  $J$  be the set of  $n$  facility with element  $f_j$ , where  $j = 1, 2, \dots, n$ . Each facility  $f_j \in J$  was also represented as a coordinate  $(f_{xi}, f_{yi})$  in the Cartesian plane. A  $2 \times n$  matrix was also used to illustrate  $J$  where the  $j$ th column represents the coordinate of the  $j$ th facility, that is,

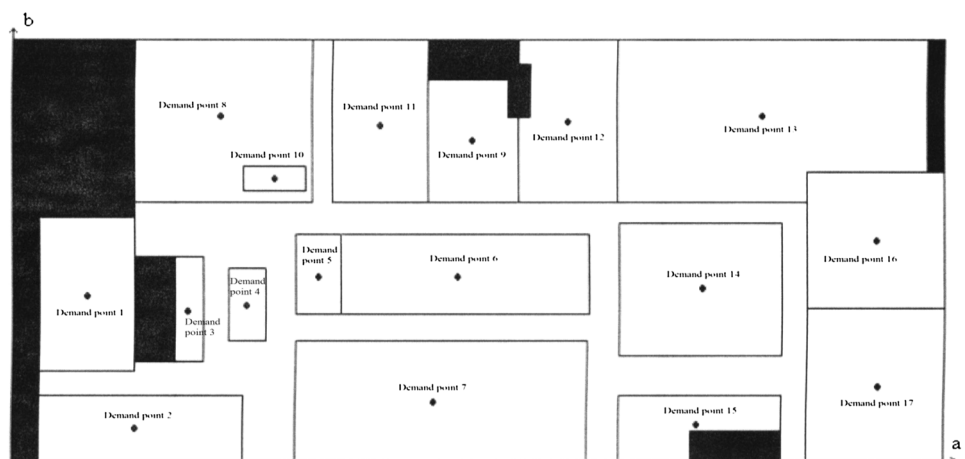
$$J = \begin{pmatrix} f_{x1}f_{x2} \dots f_{xn} \\ f_{y1}f_{y2} \dots f_{yn} \end{pmatrix} \tag{2}$$

To apply CS algorithm for the UFLP, a mapping between the facility in the host nests and the problem were established.

Since each cuckoo egg is a possible solution to the problem and since the UFLP aims to find the optimal locations of the facilities to meet its objective, then each egg consists of  $n$  locations of the  $n$  facilities. Let  $s$  be the set of  $s$  cuckoo eggs in the nest with the elements denoted by the  $2 \times n$  matrix  $C_l$  where  $l = 1, 2, \dots, s$ . Each  $C_l \in s$  consists of the locations of the  $n$  facilities. For  $l = 1, 2, \dots, s$

$$C_l = \begin{pmatrix} f_{x1}f_{x2} \dots f_{xn} \\ f_{y1}f_{y2} \dots f_{yn} \end{pmatrix} \tag{3}$$

**Fig. 1** The search space indicating the set of demand points and the restricted (shaded) regions



**Table 1** Coordinates of the demand points

Demand point	Coordinates	Demand point	Coordinates
1	(26.5, 45.5)	10	(84.5, 77.0)
2	(39.0, 09.5)	11	(144.0, 89.0)
3	(57.0, 41.0)	12	(176.0, 95.5)
4	(72.5, 41.0)	13	(231.0, 94.5)
5	(92.0, 50.0)	14	(216.5, 43.0)
6	(137.5, 50.0)	15	(216.5, 09.5)
7	(131.5, 17.0)	16	(274.5, 64.0)
8	(66.0, 94.5)	17	(274.0, 21.0)
9	(113.5, 94.5)		

Suppose  $I = \{d_1, d_1, \dots, d_m\}$  and  $J = \{f_1, f_2, \dots, f_n\}$ . Then, for  $i = 1, 2, \dots, m$  and  $j = 1, 2, \dots, n$ , the relationship between  $I$  and  $J$  can be expressed as where  $a_{ij}$

$$A = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \dots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{pmatrix} \tag{4}$$

represents the distance between  $d_i \in I$  and  $f_j \in I$ . The distance between  $d_i$  and  $f_j$  is computed by

$$a_{ij} = \|f_j - d_i\| = \sqrt{(f_{yj} - d_{yi})^2 + (f_{xj} - d_{xi})^2} \tag{5}$$

From the definition of the UFLP, for each  $d_i \in I$ ,  $d_i$  is assigned to the  $f_j$  where  $a_{ij}$  is at minimum.

For CS, the fitness evaluation was based on the main objective of the UFLP subject to the following constraints:

1. The facility should not be in the location of the demand point;
2. The location of the facility must be within the  $a \times b$  area of the search space;
3. There should be no two facilities with the same location;

4. All the facilities must be used; and
5. Facilities should not be located in the restricted regions.

The fitness value for each cuckoo egg is computed using the fitness function represented in Eq. (6).

For each violated constraint, there is an associated penalty to the fitness function. The purpose of this penalty is to worsen the fitness value, thus making it less possible to be the best fitness. The penalty weights for the violated constraints may differ depending on the preference of the organization. In this study, however, the penalty weights were the same.

The fitness function measures the performance of each cuckoo egg as a solution to the UFLP. Since the objective is to minimize the total cost incurred by establishing the facilities, then lower the numerical value of the fitness, better the quality of the solution. Consequently, the cuckoo egg with the lowest fitness becomes the optimal solution.

### CS-LF for the UFLP

The area of the department store was considered as the search space where the cuckoo searched for the solution. Each nest consisted of a set of facilities and was represented as coordinates.

Given a fitness function, each cuckoo searched for the optimal locations of its facilities in the search space by comparing the new egg to the older egg that was found. Updating each egg in the nest was based on its memory about its current location, its best location so far ( $F_j$ ), and location of the cuckoo's most successful egg so far ( $F_j$ ).

### Fitness evaluation

The fitness evaluation for cuckoo was based on the main objectives of the UFLP and on the constraints identified. In this study, the fitness function measured the performance of each egg as a solution to the UFLP. The lower the numerical value of fitness the better the solution. Note that for each violated constraint, there is an associated penalty to the fitness function.

These constraints were denoted by  $k$  where  $k = 1, 2, 3, 4, 5$ , respectively. The fitness value for each cuckoo egg  $C_l$ ,  $l = 1, 2, \dots, s$ , was computed using the fitness function represented in the study of Galvão (2004)

$$\text{fitness}(C_l) = \sum_{i=j}^m \sum_{j=1}^n t_{ijl} x_{ijl} + \sum_{k=1}^5 m_{kl} u_{kl} v_{kl}, \quad (6)$$

where  $m_{kl}$  represents the weight associated to each constraints identified in  $C_l$ ;  $u_{kl}$  represents the number of facilities violating constraints  $k$  in  $C_l$ ;  $v_{kl}$  indicates whether constraint  $k$  is violated ( $v_k = 1$ ) or not ( $v_k = 0$ ) in  $C_l$ ;  $t_{kl}$

represents the transport cost from facility  $j$  to demand point  $I$  in  $C_l$  and  $x_{kl}$  indicates whether the demand point  $i$  is assigned to facility  $j$  ( $x_{ijl} = 1$ ) or not ( $x_{ijl} = 0$ ) in  $C_l$ .

Observe that the first term of the fitness function is simply the objective function of the UFLP. Due to the assumption that the facilities to be established are considered “open,” the second term of Eq. [1] is disregarded so as to lessen functional evaluations.

The objective of the UFLP is to minimize the total cost incurred by setting up  $n$  facilities. Hence, the egg with the best fitness is the one with the lowest total cost incurred, and therefore, the location of the facility.

### Parameters and terminating condition

The main parameters needed for the implementation of the CS-LF are the population size ( $n$ ) and probability to discover ( $P_a$ ). In this study, there were 11 parameter settings. This is to determine the best set of parameter values for the CS-LF algorithm to solve the UFLP.

*MaxGen* is the maximum number of generation iterations allowed. The parameter settings and *MaxGen* were based on Yang and Deb's (2010) study. Due to the probabilistic nature of the algorithm, results may differ from another, thus the algorithm was run 30 times for each parameter setting where results for the 30 times running were average. The terminating condition for each run was when *MaxGen* is reached.

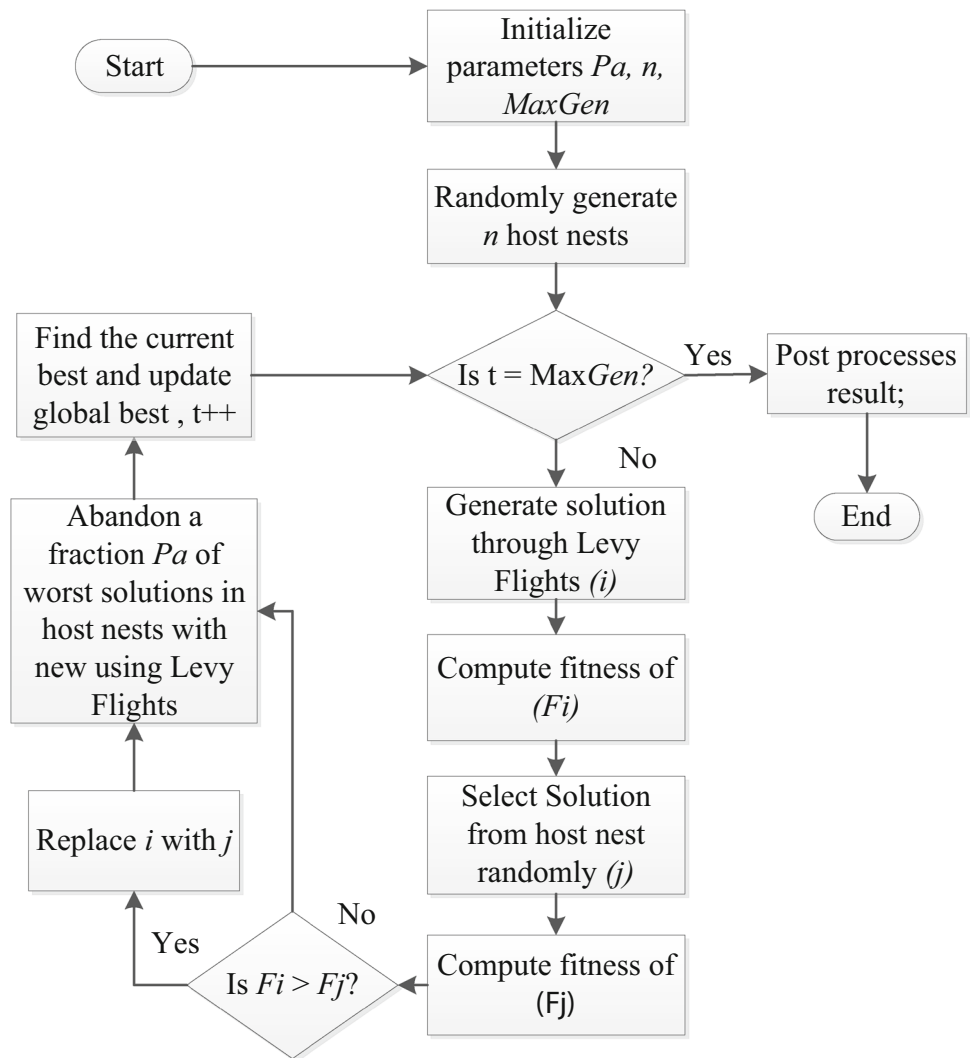
### Algorithm of CS-LF for the UFLP

The flowchart and the pseudocode of the CS-LF algorithm for the UFLP are shown in its full form in Fig. 2. So, the algorithm started with the initialization of parameters. Set counter  $t = 0$ , population size ( $n$ ) = (5–500) and  $P_a = (0, 1)$  were based on the parameter settings.

The LF randomly generated host nests which should be within  $a \times b$  area of the search space. The initial location of nests was equal to the initial location of eggs. The next step was, if the counter  $t = \text{MaxGen}$ , then searching was stopped; if not, then generate another solution [cuckoo egg ( $i$ )] using Lévy flights then get fitness of ( $F_i$ ) using the fitness function. Then choose random nest in initial population ( $j$ ) then get fitness of the egg located in the nest ( $F_j$ ). Compare  $F_i$  and  $F_j$ , if  $F_i$  was better than ( $F_j$ ) then replace the egg in  $j$  with egg  $i$ . Then sort eggs in the current population based on fitness value, and then abandon a fraction of worse nests or eggs. Build new ones and new locations via LF and find current best. The process was repeated from the start.



**Fig. 2** Flowchart of the CS-LF—UFLP



**Implementation**

This phase is the actual coding for the CS-LF. This phase also included researches and intensive data gathering. The program had undergone several processes to get the solutions to the UFLP.

**Results and discussion**

**Implementation of the CS-LF Algorithm**

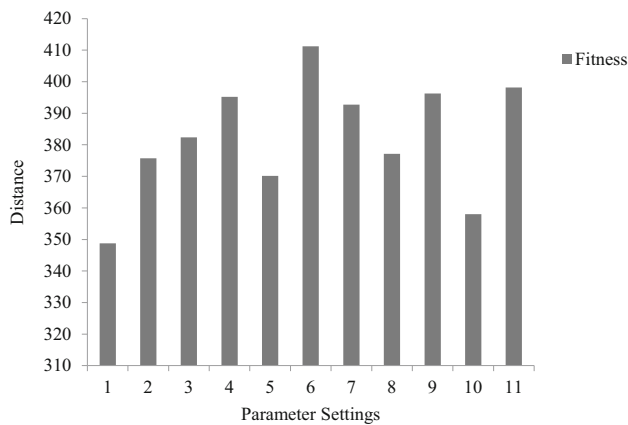
The implementation of the CS-LF algorithm for finding the minimum fitness uses a two-dimensional array. The program flow started with generating an initial solution for the facilities of size 10 and extracting data from demand points of size 17. After obtaining the input array, get the distance from a facility to a demand point (minimum distance) and saved its distance in the float *MinSOD*. In getting the

distance, get the coordinates of the initial population (*x*, *y*) and coordinates of the demand points (*pointX*, *pointY*). For the next process, compute and compare each distance from facility to demand point and saved it in *Dist* and get the sum of all minimum distance which it was the goal of the implementation and saved it in the variable *ArrayList MinSODs*.

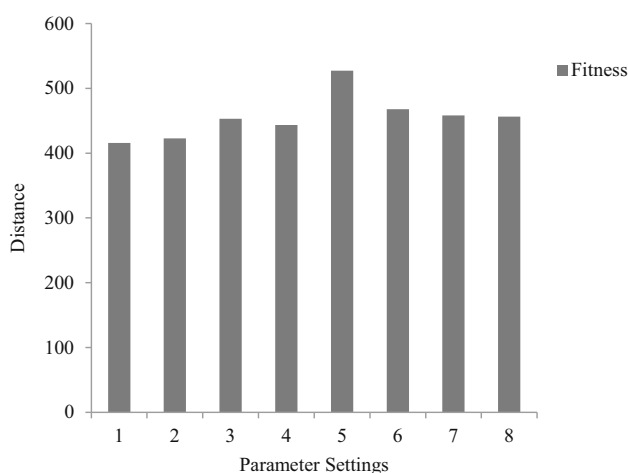
From the trend of the performance of the parameter settings, based on the results, it was observed that the lesser the value assigned to parameters *n* and *Pa*, the better the fitness value. It was noticed that the ranking of the parameter settings based on the best fitness follows this observation. It was also observed that the lesser the value assigned to parameter *n*, the solution was obtained faster.

Figures 3 and 4, respectively, the best of PSO results (Palma 2010) and the best of CS-LF results are shown. As can be seen, the CS-LF obtained better solutions compared to PSO.





**Fig. 3** CS-LF results for an uncapacitated FLP



**Fig. 4** PSO results for an uncapacitated FLP (Palma 2010)

As for the performance of the parameter settings based on solution time, no definite conclusion can be made other than the fact that for all 11 parameter settings, it can be found between parameter settings 1–6. As can be seen, parameter setting 1 obtained better solution in terms of solution time with the value of 312 mm since it has lesser fitness value and the value of ( $n$ ) that is assigned to this parameter setting is lesser.

As stated earlier, the early convergence method was due to the initialization of the locations of the facilities bounded within the search space. Moreover, it can be acknowledged that the values assigned to parameters  $n$  and  $P_a$ , were too large for the search space. For this reason, for every updating of locations of facilities, the locations obtained were outside the search space which greatly affects the fitness due to the penalties acquired. Thus, it might be better if smaller values were to be assigned for parameters  $n$  and  $P_a$  for this particular problem.

To avoid extremities if the best fitness and solution were the basis in determining the best solutions in terms of best

fitness and solution time, this study considered the average fitness and solution time instead. It was concluded that parameter setting 1 obtained the best solution based on best fitness while parameter setting 1 yielded the best solution based on solution time.

Furthermore, it is concluded that parameter setting 1 is the best parameter setting for the CS-LF algorithm to the uncapacitated FLP of the company as it resulted to the best fitness among the 11 parameter settings applied. The coordinates of the facility locations of the best solution obtained here are shown in Table 2.

Using the fitness function in Eq. (6), the fitness value obtained for the existing data is 352.50561, for the PSO is 415.65406 which is greater than the best fitness obtained when CS-LF algorithm is applied which is 348.79346. This means that the application of CS-LF to the uncapacitated FLP of the company yielded a better solution. In fact, no constraints were violated on the best solution obtained when CS-LF algorithm was used. In getting the fitness value, we get the summation of the distance between each demand point and its corresponding assigned facility.

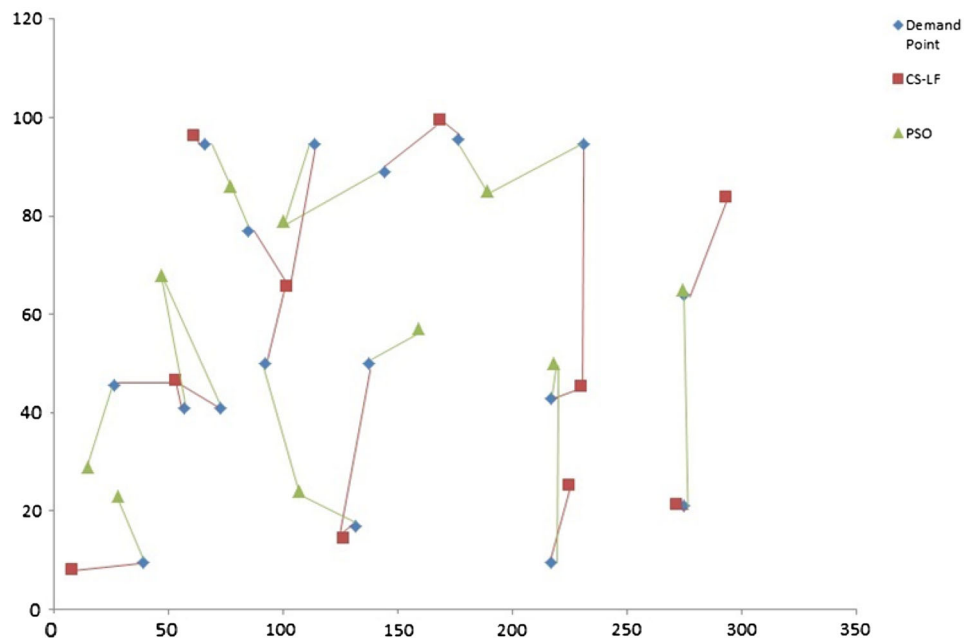
In the implementation of PSO (Palma 2010), facility 8 has not been assigned a demand point that causes a penalty to the fitness function which is all facilities must be used. From this, it may be deduced that the company may have established that facility not to be open all the time but rather, only when it is peak season. If this constraint in the fitness function was disregarded due to the mentioned reason, then the distances were the main basis for determining the fitness. Without this constraint, it is found that the fitness value from the CS-LF algorithm which yielded to 348.79346 is better than the fitness value from the original data which is 352.50561 and from the PSO which is 415.65406 when applied to the problem.

**Table 2** Coordinates of the facility locations based on the PSO best solution (Palma 2010), and CS-LF best solution

Facility	PSO (Palma, 2010)	CS-LF
1	(28.00, 23.00)	(229.95, 45.43)
2	(218.00, 50.00)	(168.13, 99.55)
3	(189.00, 85.00)	(07.84, 8.21)
4	(47.00, 68.00)	(53.25, 46.53)
5	(77.00, 86.00)	(61.23, 96.28)
6	(159.00, 57.00)	(224.60, 25.37)
7	(100.00, 79.00)	(101.76, 65.70)
8	(107.00, 24.00)	(271.39, 21.31)
9	(15.00, 29.00)	(293.07, 83.71)
10	(274.00, 65.00)	(126.12, 14.40)



**Fig. 5** Plot of the coordinates of the facility locations under CS-LF and PSO



In Fig. 5, the CS-LF and PSO results of the coordinates of the facility location are shown. With these, comparison with the visualization of coordinates was done.

However, since it is assumed in this study that all the established facilities were considered open, results obtained by applying the CS-LF algorithm to the problem yielded better facility locations. All the facilities that were established are all used.

## Summary and conclusion

Facility location problem (FLP) is a mathematical way to optimally locate facilities within a set of candidates to satisfy the requirements of a given set of clients. This study addressed the uncapacitated FLP as it assures that the capacity of every selected facility is finite. Thus, even if the demand is not known, which often is the case, in reality, organizations may still be able to take strategic decisions such as locating the facilities.

There are different approaches relevant to the uncapacitated FLP. Here, the cuckoo search via Lévy flight (CS-LF) was used to solve the problem. Though hybrid methods produce better results, this study employed CS-LF to determine first its potential in finding solutions for the problem, particularly when applied to a real-world problem.

A successful application of CS-LF to the problem was done as the mapping between the cuckoo egg in the host nests and the problem was established. An uncapacitated FLP program was designed using the CS-LF algorithm as

its framework. The program was implemented using eleven parameter settings.

Due to the probabilistic nature of the algorithm, results may differ from another, thus the algorithm was run thirty times for each of the eleven parameter settings where the best fitness based on the distance and convergence speed was recorded. Results showed that parameter setting 1 with a value of 348.79346 converged to be the best solution. Parameter setting 1, on the other hand, was the fastest among 11 parameter settings in getting its optimal solution. Moreover, parameter setting 1 was concluded as the best set of parameter values for the algorithm to solve the problem with the value of the parameter ( $n$ ) = 5, and the value of the parameter ( $P_a$ ) = 0.0. The parameter  $n$  is the size of the population and the parameter  $P_a$  is the fraction of eggs to be abandoned each generation.

Though results showed that applying CS-LF for the problem produced better facility locations when compared to the existing facility locations from the data and from the PSO program, it cannot be concluded that CS-LF is the most effective approach for the problem. It is also recommended to apply this algorithm to other data sets to check if the algorithm could arrive at better results. Future study can also be done about the performance of the algorithm to other facility location problem.

**Open Access** This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.



## References

- Cagnina LC, Esquivel SC, Coello CA (2008) Solving engineering optimization problems with the simple constrained particle swarm optimizer. *Informatica*. 32: 319–326
- Galvão RD (2004) Uncapacitated facility location problems: contributions. *Pesquisa Operacional* 24(1):7–38
- Guner AR, Sevklı M (2008) A discrete particle swarm optimization algorithm for uncapacitated facility location problem. Dept. of Industrial and Manufacturing Engineering, College of Engineering, Wayne State Univ., Detroit. p 9
- Palma HP (2010) Particle swarm optimization for an uncapacitated facility location. Undergraduate special problem. University of the Philippines Mindanao, Davao
- Payne RB, Sorenson MD, Kiltz K (2005) *The cuckoos*. Oxford University Press Inc. New York
- Ragsdell K, Phillips D (1976) Optimal design of a class of welded structure using geometric programming. *J Eng Ind* 98:1021–1025
- Reynolds AM, Frye MA (2007) Free-flight odor tracking in *Drosophila* is consistent with an optimal intermittent scale-free search. *PLoS ONE* 2:e354
- Roostitala A, Asgharizadeh E (2013) Optimizing the warranty period by cuckoo meta-heuristic algorithm in heterogeneous customers' population. *J Ind Eng Int* 9:27. <https://doi.org/10.1186/2251-712X-9-27>
- Santillan JH, Tapucar S, Manliguez C, Calag V (2017) Cuckoo search via Levy flights for the capacitated vehicle routing problem. *J Ind Eng Int*. <https://doi.org/10.1007/s40092-017-0227-5>
- Silva MF, Serrab D (2005) The queue length capacitated facility location problem. GREL, IET, Universitat Pompeu Fabra, Ramon Trias Fargas, Barcelona, p 13
- Tein LH, Ramli R (2010) Recent advancements of nurse scheduling models and a potential path. In: Proc. 6th IMT-GT Conference on Mathematics, Statistics and its Applications (ICMSA). p 395–409
- Yang XS (2009) Harmony search as a metaheuristic algorithm. In: Geem ZW (ed) *Music-inspired harmony search algorithm. Studies in computational intelligence*, vol 191. Springer, Berlin, Heidelberg
- Yang XS (2010) *Nature-inspired metaheuristics algorithms*, 2nd edn. Luniver Press, United Kingdom
- Yang XS, Deb S (2009) Cuckoo search via Lévy flights. In: *Proceeding of World Congress on Nature & Biologically Inspired Computing*. p 210–214
- Yang XS, Deb S (2010) Engineering optimization by cuckoo search. *Int J Math Model Numer Optim* 1:330–343
- Zaw M, Mon E (2013) Web document clustering using cuckoo search clustering algorithm based on Lévy flights. *Int J Innov Appl Stud*. 4(1):182–188 (ISSN 2028-9324)

