# A modified squirrel search algorithm for solving facility layout problems

**Najla Akram Al-Saati[1], Zakariya Yahya Algamal[2]**
[1]Software Department, College of Computer Science and Mathematics, University of Mosul, Mosul, Iraq
[2]Department of Statistics & Informatics, College of Computer Science and Mathematics, University of Mosul, Mosul, Iraq

**Article Info**

***Article history:***

Received Sep 27, 2021
Revised Feb 21, 2022
Accepted Mar 01, 2022

***Keywords:***

Combinatorial optimization
Facility layout planning
Squirrel search algorithm
Swarm intelligence

**ABSTRACT**

With the huge advance in artificial intelligence and the rapid development of intelligent swarm algorithms, the exploration of facility layout problem (FLP) with its non-deterministic polynomial-time (NP-Hard) nature has gained much more attention. The squirrel search algorithm is one of the swarm algorithms that is known for its effective gliding feature that provides cheap exploration of lengthy distances. In this work, Msqrl algorithm is presented as a modification of squirrel search algorithm to be capable of handling permutation-specific FLP. The modification is done by introducing two new operators: Msqrl-Exchange and Msqrl-Winter. It is used to investigate the effectiveness in finding acceptable solutions to variable-size, single-row FLPs in a fast and efficient manner. Tests included small and large benchmark instances for comparisons. Outcomes show that Msqrl was able to improved quite a few previously found solutions by acting efficiently and converging rapidly to solutions. It outperformed both semidefinite programming and cuckoo optimization in finding optimal solutions in an acceptable number of iterations and relatively small population sizes.



***Corresponding Author:***

Najla Akram Al-Saati
Software Department, College of Computer Science and Mathematics, University of Mosul
Mosul, Iraq
Email: dr.najla_alsaati@uomosul.edu.iq

## 1. INTRODUCTION

Facility location or facility layout planning problems (FLP) outline a vital subdivision of optimization problems, which are responsible of optimally positioning facilities to minimalize an objective function based either on effort, cost, time, or distance. Facilities are elements, machines or workstations that usually enable activity functioning, and their layouts resemble configurations essential for performing certain operations. In the framework of production systems, the main purpose is minimizing overall costs for material handling involved in the movement of materials among sections. It has been found that nearly up to 20-50% of the entire operational manufacturing cost goes to materials handling. FLP has many practical application areas such as planning layouts for gardens, stores, hospitals, offices, and plants [1].

In view of the high demand in manufacturing industries for reduced costs, efficient facility layout designs are sought. This problem is a recognized combinatorial optimization problem, with its non-deterministic polynomial-time (NP-Hard) complexity, it is considered to be difficult to solve [2]. Finding the exact solution for a certain FLP problem means enumerating all possible arrangements of layouts, with n facilities the number of permutations is $n!$. That is why only small problems can be treated using exact algorithms. Such exact methods usually require excessive memory are considered be expensive in terms of computations [2], [3]. That is way, other intelligent methods are being investigated such as metaheuristics and swarm intelligence. These methods can develop fast approximated near-optimal solutions for larger problems. One of the recently developed swarm algorithms is the squirrel search algorithm, it replicates the behavior of southern flying squirrels with their economical way of movement (locomotion) that gives them the ability to modify lift and drag. As squirrels fly in the forest trying to find food sources, they glide changing locations on trees in search of nuts to eat (acorn) and nuts for winter storage (hickory). Researchers have several objectives to think about when dealing with FLPs, such as reducing the cost of material handling, decreasing material flow, dropping down the overall distance traveled for the material, and even refining the entire adjacency among facilities.

Through the last decades, authors have addressed various types of this problem, as in 2005, Anjos *et al.* constructed a relaxed semidefinite programming (SDP) method to find lower bounds on the optimal values by using the natural symmetry of the problem [2]. In 2008, Khilwani *et al.* used Psycho-Clonal algorithm in relation to the dynamic system characteristics and operational constraints to solve the FLP, it was found that the suggested algorithm outperformed others [4]. In 2009, Amaral suggested a new lower bound for the single-row FLP through the optimization of a linear program over the partial description at hand with cutting planes [5]. In 2010, Ohmori *et al*. proposed solving FLP using swarm intelligence employing particle swarm optimization (PSO) [6]. In 2011, Datta *et al.* used genetic algorithms (GA) that is based on permutations to solve single-row FLPs [7]. Prasad *et al.* in 2014, designed manufacturing plant layouts using computerized relative allocation of facilities (CRAFT) and developed a JAVA program to design the optimum layout using STEP file as input [8]. In 2015, Matai used a modified simulated annealing to solve the multi objective facility layout problem [9]. Goncalves and Resende presented an encoding of unequal area FLPs using GA grounded on biased randomized number [10]. In 2016, Anjos and Vieira initiated a framework to solve unequal area layouts using a combination of two models based on mathematical optimization [11]. Park and Seo suggested a two-step heuristic algorithm in 2017, the first step uses a construction method which sequentially places facilities, and the second step enhances that of the first step. Results indicated that the proposed algorithm yielded equivalent quality of layouts within much shorter time when compared with former work [12]. Zhao *et al.* incorporated GA with Levy Flight for improving mutations, they also used an island model genetic algorithm with varying approaches of decoding every island [13]. In 2018, Kalita and Datta studied the constrained single-row facility layout problem using GA with large problem instances [14]. Moslemipour *et al.* used a hybridization among ant colony, clonal selection and simulated annealing for large FLPs [15]. Feng and Chae, made the effort of solving fixed-shape rectangular layouts through the use of integer linear programming, they employed new constraints to maximize flow of material between the closed facilities [16]. In 2019, Lin and Yingjie presented survival signature for layout problems along with optimizing solutions using GA [17]. Also in 2019, Kim and Chae employed the monarch butterfly optimization with a slicing tree to denote the layout [18]. Kalita and Datta in 2020, attempted to solve the constrained Single-row FLP using GA with repair procedures, the method was applied to a set of different-size FLP instances [19]. Tongur *et al.* employed migrating bird optimization, tabu search and simulated annealing to find layouts for a big scaled university hospital [20]. Moreover in 2020, Meskar and Eshghi a used a lower bound to solve the generalized single-row FLP using cuckoo optimization algorithm [21]. Lately in 2021, Ingole and Singh used biogeography-based optimization to find solutions for unequal area (fixed/flexible) FLPs [22].

In this work, the squirrel search algorithm is to be modified for tackling the problem of varied length, single-row facility layout problems. The modified algorithm (Msqrl) is investigated using benchmark flow instances less than 30 [7], and others between 60 and 80 [2], two operators are introduced to modify the algorithm (Msqrl-Exchange and Msqrl-Winter). Results are compared with pervious work done using the same data instances to show the effectiveness of the modifications carried out in this work.

## 2. RESEARCH METHOD

### 2.1. Problem formulation for facility layout planning

Facility layout planning represents a very interesting real-world class of problems, a variety of observational studies can be found in the literature. Going through the literature, the definition of FLP is given as “a common industrial problem in which the objective is to configure facilities, so as to minimize the cost of transporting materials between them” [23]. As for optimization problems, definite optimal solutions are sought, so there will be a need for sufficient methods capable of recognizing optimal solutions [24].

Layouts for facilities can be stated using an arrangement of entities necessary to carry out certain responsibilities such as producing profits or services delivery. Each facility represents a unit object for the implementation and execution of any task. This ranges from machine shops to departments or warehouses [23].

For optimization problems, definite optimal solutions are sought. So, there will be a need for some sufficient methods capable of recognizing at least one optimal solution [24]. Mathematically, the objective is minimizing the distance-based measure given in (1).

$$Min \sum_i \sum_j (flow_{ij} \cdot d_{ij}) \cdot c_{ij} \tag{1}$$

where $flow_{ij}$ is the flow quantity between department $i$ and $j$, $d_{ij}$ is the direct straight distance between them, $c_{ij}$ is the cost of moving one unit from $i$ to $j$.

Given two dissimilar facilities $f_i$ and $f_j$, and a permutation $p$. The value of $D_P(i, j)$ is the summation of the lengths for facilities from the center of $f_i$ to the center of $f_j$ (taking into consideration if the facilities are placed on the right or left side of the other facility). Solving the problem requires finding a permutation that gives the minimized sum of distances between all facilities pairs. Mathematically [2]:

$$\underset{p \in P}{Min} \sum_{i<j} flow_{ij} \cdot D_p(i,j) \tag{2}$$

In this work, a special case of facility layout planning is studied for comparison reasons. It is called the linear single-row facility layout problem. This problem involves placing facilities of various lengths on a straight line.

### 2.2. Squirrel search algorithm

Squirrels fly among trees for foraging and fulfilling their needs. They glide from one tree to another seeking resources of food in autumn, during this time, they vary their location to discover diverse areas of the forest. When weather temperatures get sufficiently hot, squirrels can find their everyday energy requirements fast enough by instantly consuming acorns whenever found. Having completed their daily supply, they begin a new quest for a food source that is best for winter, that is hickory nuts. Storing nuts can aid in sustaining energy needs in severe meteorological conditions and cuts the expenses of foraging journeys. In cold seasons, forest trees drop their leaf cover exposing squirrels to predators, so they tend to be less energetic with no hibernation. Such a process keeps repeating itself during the life of a squirrel, establishing the basis of the algorithm. Squirrel search algorithm presumes ($n$) flying squirrels, each is found on a separate tree in a forest and separately looks for nuts using accessible food resources optimally with some behavior of dynamic foraging. In the forest, three types of trees exist: normal, oak (acorns) and hickory trees, assuming that the forest includes three oak and only one hickory tree [25].

#### 2.2.1. The basic algorithm

Starting with a randomly originated vector $S$ containing locations for $N$ squirrels, they begin gliding in the search space to modify their locations. Where $S_{i,j}^t$ denotes iteration $t$ with the $i^{th}$ squirrel in the $j^{th}$ dimension. Next, is the $S$ matrix showing the representation of squirrels:

$$S = \begin{bmatrix} S_{1,1} & S_{1,2} \cdots & S_{1,d} \\ S_{2,1} & S_{2,2} & S_{2,d} \\ \vdots & \vdots & \vdots \\ S_{N,1} & S_{N,2} \cdots & S_{N,d} \end{bmatrix}$$

A fitness function $f\,(S_{i,j}^t)$ is used to evaluate the locations of squirrels, the squirrel with the minimum fitness value is said to be on the hickory tree (storing food) and denoted as $(S_{i,j}^t)_h$ , the subsequent three best fitness values are for squirrels presumed to be on acorn trees (consuming normal food) and are denoted as $(S_{i,j}^t)_a$. The rest of flying squirrels are on normal trees $(S_{i,j}^t)_n$ with no resource of food. As they change locations, a number of the flying squirrels move in the direction of the hickory tree having achieved their daily needs, others move to acorn trees to get their daily requirements.

The mechanism of gliding is defined by the lift $L$ computed in (3) and drag $D$ force calculated in (4). Flying squirrels glide at steady speed and drop at angle ($\varphi$) horizontally with lift-to-drag ratio. The descending angle of gliding is shown in (5) [25].

$$L = 1 \,/\, 2\rho CLV2S \tag{3}$$

where $\rho$ is the air density and equals 1.204 $kgm^{-3}$. $C_L$ is the lift coefficient usually set in the range $0.675 \le C_L \le 1.5$, $V$ is the speed and equals 5.25 $ms^{-1}$. $S$ is the body surface area and equals 154 $cm^2$.

$$D = 1 \,/\, 2\, \rho V2SCD \tag{4}$$

where $C_D$ is the frictional drag coefficient, usually set to 0.6.

$$\varphi = arctan\ (\frac{D}{L}) \tag{5}$$

by substitution, we get:

$$\varphi = arctan\ (\frac{1/2\ \rho V^2 S C_D}{1/2 \rho C_L V^2 S}) \tag{6}$$

$$\varphi = arctan\ (\frac{C_D}{C_L}) \tag{7}$$

Meaning that the only $C_D$ and $C_L$ are controlling ($\varphi$) and are both constants. Now, the constant of gliding distance $G_d$ can be computed as in (8).

$$G_d = H_{loss} / (ScF\ *\ tan\ (\varphi)) \tag{8}$$

where $H_{loss}$ is the height loss after gliding set to 8 m, *ScF* is a scaling factor set between 16 and 37.

Flying squirrels update their locations based upon three factors. These are their previous location, gliding distance, and probability ($P_{pd}$) that a predator is close. They move according to three circumstances: [25]

- Flying squirrels on acorn trees can head to hickory tree to preserve optimal food resources by (9).

$$(S_{i,j}^{t+1})_a = \begin{cases} if\ rand \geq P_{pd}\ (S_{i,j}^t)_a + G_d \times C \times \left((S_{i,j}^t)_h - (S_{i,j}^t)_a\right) \\ \qquad Otherwise \qquad Random\ location \end{cases} \tag{9}$$

where $C$ is a constant of gliding to governor the traversing of the search space, *rand* is a randomly generated number in [0, 1].

- Squirrels residing on normal trees are able to glide in the direction of acorn nut trees to satisfy daily consumptions. The resulting location of squirrels can be updated as in (10).

$$(S_{i,j}^{t+1})_n = \begin{cases} if\ rand \geq P_{pd}\ (S_{i,j}^t)_n + G_d \times C \times \left((S_{i,j}^t)_a - (S_{i,j}^t)_n\right) \\ \qquad Otherwise \qquad Random\ location \end{cases} \tag{10}$$

- For squirrels that has consumed acorn nuts and are now on normal trees, they will possibly fly gliding for the hickory tree to start storing food for winter. Hence, updated locations can be obtained from (11).

$$(S_{i,j}^{t+1})_n = \begin{cases} if\ rand \geq P_{pd} (S_{i,j}^t)_n + G_d \times C \times \left((S_{i,j}^t)_h - (S_{i,j}^t)_n\right) \\ \qquad Otherwise \qquad Random\ location \end{cases} \tag{11}$$

**2.2.2. Observing seasons**

In the cold winter season, flying squirrels lose substantial amount of heat. Owing to their raised temperature and small body size, searching for food becomes costly and also risky because of predators. Unlike autumn, winter influences and reduces their activity. Due to its importance, a procedure for observing and controlling the season check is add to avoid getting stuck in local optima. The season constant $S_c$ is calculate using (12).

$$S_c^i = \sqrt{\sum_{k=1}^{d} \left(S_{a,k}^i - S_{h,k}\right)^2} \tag{12}$$

where $i$ is the number of squirrels on acorn trees, $d$ is the dimension length of a squirrel.

Next, seasonal check is carried out ($S_c^t < S_{min}$), here $S_{min}$ is the minimum value of a season constant calculated as [25]:

$$S_{min} = \frac{10E^{-6}}{365^{t/(tm/2.5)}} \tag{13}$$

where $t$ is the current iteration, $tm$ is the maximum iteration.

Higher values of $S_{min}$ encourage exploration, whereas lesser values improve the ability of exploitation for an algorithm. There should always be a suitable stability between them. If checking for seasons becomes true, meaning winter has ended, flying squirrels that were unable to discover optimal food for winter are repositioned in a random fashion. Figure 1 gives the detailed flowchart of the squirrel search algorithm and the process of finding optimal solutions.

### 2.3. Modification of squirrel search

Due to the nature of the FLP, a variation is carried out to modify the squirrel searching methodology. This modification addresses the main movement steps of squirrels along with other adaptations. The process is mainly grounded on discovering good-enough solutions through some certain tactics to renew locations of squirrels, thus producing new populations targeting for global optimums. Each squirrel represents a sequence vector of *N* facilities numbered form 1 to *N*, every single one of these sequences is a suggested layout solution.

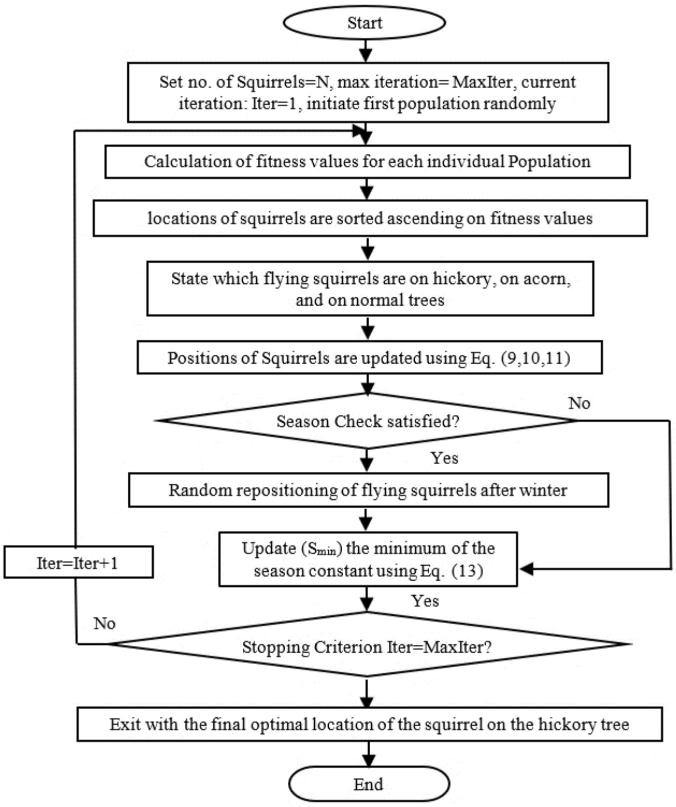


Figure 1. Flowchart for squirrel search algorithm

The modifications suggested in this work involve three issues. These are: correctly creating initial solutions, updating locations of squirrels, and monitoring season check. They are conducted as follows:

#### 2.3.1. Initial solution

Initializing the individuals of the first population of flying squirrels is performed randomly. Here the permutation characteristic of FLP should be kept in mind. Accordingly, the first population is randomly generated from the sequence of facilities ranging from 1 to number of facilities *N*.

#### 2.3.2. Locations update for squirrels

In the original algorithm, squirrels update their location in three ways:

- Squirrels on acorn trees head for the hickory nut tree,
- Squirrels on normal trees head for direction of acorn trees, and
- Squirrels on normal trees head for the hickory nut tree.

This is done using (9), (10), and (11), these equations yield incorrect solutions when applied to FLP. In order to preserve the correctness of the produced solutions a new operation Msqrl-Exchange (⊖) is introduced to replace subtraction and gliding. This operation uses a mechanism to find a set of location pairs if exchanged, new acceptable sequences will be obtained as solutions to FLP.

The Msqrl-Exchange operation takes two squirrel sequences as input and performs a subtraction like process to imitate the movement of squirrel from one type of trees to the other. This is done by finding pairs of locations to exchange in the second sequence without repetition of locations (locations that are already been chosen for exchange are neglected). The process is explained with a fully detailed example in Figure 2.

As for the two constants Gd and C used in the original work. Both constants have no impact on updating locations of squirrels using the Msqrl-Exchange operation. They can be omitted without affecting the procedure of the algorithm.

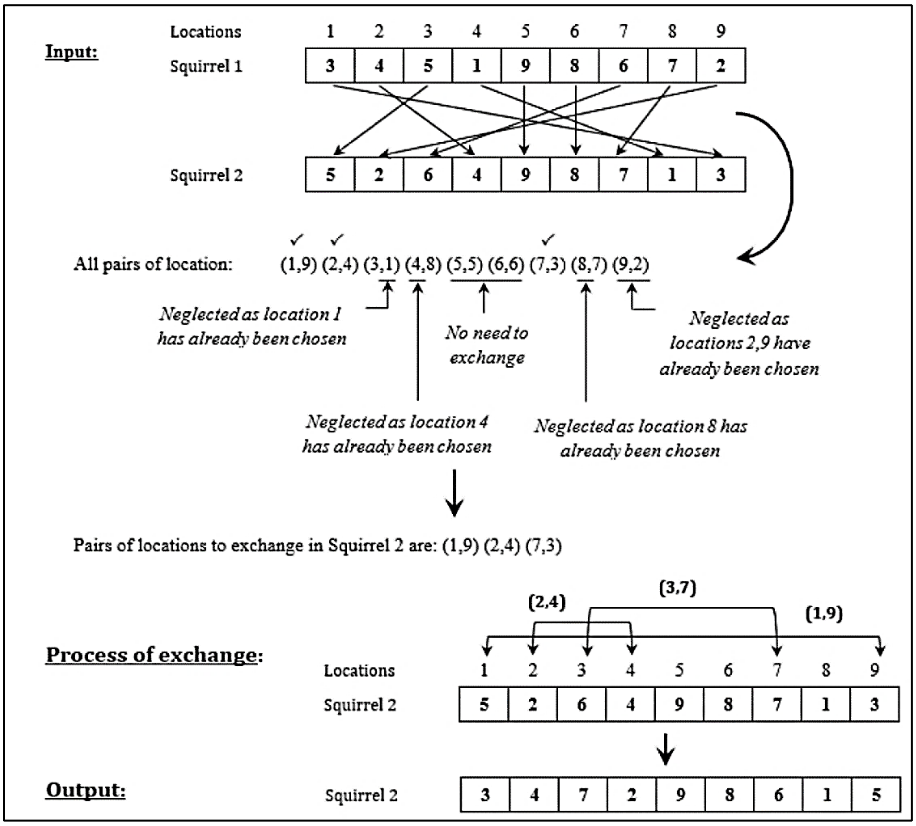


Figure 2. An illustration of the proposed Msqrl-Exchange process with an example

In achieving all modifications mentioned so far, (9), (10), and (11), become:

$$(S_{i,j}^{t+1})_a = \begin{cases} if\ rand \geq P_{pd}\ (S_{i,j}^{t})_a\ + \left((S_{i,j}^{t})_h\ \ominus\ (S_{i,j}^{t})_a\right) \\ Otherwise \qquad\qquad Random\ location \end{cases} \tag{14}$$

$$(S_{i,j}^{t+1})_n = \begin{cases} if\ rand \geq P_{pd}\ (S_{i,j}^{t})_n\ + \left((S_{i,j}^{t})_a\ \ominus\ (S_{i,j}^{t})_n\right) \\ Otherwise \qquad\qquad Random\ location \end{cases} \tag{15}$$

$$(S_{i,j}^{t+1})_n = \begin{cases} if\ rand \geq P_{pd}\ (S_{i,j}^{t})_n\ + \left((S_{i,j}^{t})_h\ \ominus\ (S_{i,j}^{t})_n\right) \\ Otherwise \qquad\qquad Random\ location \end{cases} \tag{16}$$

#### 2.3.3. Adjustment for winter season

The season constant *Sc* measures the closeness of squirrels on acorn trees to the hickory tree (best achieved solution in the population). It should be possible to accomplish the same effect using another new adjustment operator called Msqrl-Winter (⊟) instead of subtraction. So, having two squirrels with *n* facilities each, only locations with the same value are counted, this is illustrated as follows:

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | **Squirrel 1**: | 5 | 2 | 6 | 4 | 7 | 8 | 1 | 9 | 3 |
| ⊟ | | | | | | | | | | |
| | **Squirrel 2**: | 5 | 2 | 6 | 7 | 8 | 9 | 1 | 4 | 3 |
| | | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 1 = 5 |

That is, squirrels 1 and 2 have five similar locations out of nine. Hence, (12) can be rewritten as in (17). This way, we have a correct measure for the season constant ready for use with FLP.

$$S_c = \sum_{i=1}^{ac} \frac{\sum_{k=1}^{d}(S_{a,k}^{i} \boxminus S_{h,k})}{d} * C1 \tag{17}$$

where *ac* is the no. of acorn trees, *d* is the dimension, *C1* is *a* constant set to 0.01.

## 3. RESULTS AND DISCUSSION

This section involves testing the effectiveness of the modified squirrel search algorithm Msqrl in solving FLPs. Flow matrix instances with lengths of facilities are taken from the literature, small problem instances (≤30) [7] and larger numbers between (60 and 80) [2] are considered for comparisons. The distance matrix is calculated for each generated layout in the course of the methodology. Solutions are compared with semidefinite programming (SDP) [2], permutation-based GA (PGA) [7], and cuckoo optimization algorithm (COA) [21].

### 3.1. Parameter setting

For small numbers of facilities, setting the parameter involves taking (50) squirrels as the population size taken in the original work [25]. For food resources, (1) tree is for hickory and (3) trees are for acorn nuts, and the rest (46) trees are normal trees without nuts. As for larger facility numbers, (100, 150, and 200) squirrels are assigned: (1) hickory tree, (6, 9, and 12) acorn trees, and the remaining are normal trees. The probability of a predator appearing $P_{pd}$ is (0.1). The number of facilities, maximum iteration and the total number of possibilities $n$! are shown in Table 1.

Table 1. Parameters for tests along with the number of possibilities

| | No. of facilities | No. of squirrels | No. of iterations | Possibilities *n!* |
|---|---|---|---|---|
| Small problems | 5 | 50 | 5 | 120 |
| | 8 | 50 | 5 | 40,320 |
| | 10 | 50 | 10 | 3,628,800 |
| | 11 | 50 | 10 | 39,916,800 |
| | 20 | 50 | 30 | 2.432902e+18 |
| | 30 | 50 | 100 | 2.6525286e+32 |
| Large problems | 60 | 100 | 500 | 8.3209871e+81 |
| | 70 | 150 | 1000 | 1.197857e+100 |
| | 75 | 150 | 2500 | 2.480914e+109 |
| | 80 | 200 | 4000 | 7.156946e+118 |

### 3.2. Tests and comparisons

In order to inspect the proficiency of the modified algorithm, two tests are carried out. The first one uses six traditional instances with identified optimal solutions, these problems have no clearance between the facilities [2]. The obtained results in Table 2 indicate the capability of Msqrl algorithm in achieving the same results with a small number of iterations and squirrels.

Table 2. Comparison of the Msqrl using facility numbers ≤ 30 (Test1)

| No. of facilities | Methods | | |
|---|---|---|---|
| | SDP [2] | PGA [7] | Msqrl |
| 5 | 151.0 | 151.0 | 151.0 |
| 8 | 2324.5 | 2324.5 | 2324.5 |
| 10 | 2781.5 | 2781.5 | 2781.5 |
| 11 | 6933.5 | 6933.5 | 6933.5 |
| 20 | 15549.0 | 15549.0 | 15549.0 |
| 30 | 44965.0 | 44965.0 | 44965.0 |

On the other hand, the second test uses 20 instances for larger problems having no optimal solution. These instances include 60, 70, 75 and 80 facilities, and are divided into 4 groups of 5 instances each. These were originally created in random by [2] and are frequently used for comparisons in the literature. Table 3 illustrates the results gained by former methods and the modified algorithm Msqrl. As can be seen from the results, the Msqrl algorithm was capable of enhancing 5 solutions out of 20 (shown in bold along with the layout sequence of the enhanced solutions). As for the remaining 15 instances, results are the same with those gained by PGA which are the best results among other.

Results of Test1 together with Test2 signify the outstanding performance of Msqrl. The algorithm was successful in surpassing both SDP and COA and finding better solutions. This was achieved using a relatively small number of iterations and population sizes of squirrels.

Table 3. Comparison of Msqrl algorithm using facility numbers 60-80 (Test2)

| No. of facilities | SDP [2] | COA [21] | PGA [7] | Msqrl | Best layout (for improved solutions only) |
|---|---|---|---|---|---|
| 60 | 1,493,704 | 1,477,834 | 1,477,834 | 1,477,834 | |
| | 843,644 | 841,814 | 841,792 | 841,790 | 53-10-13-41-29-50-30-9-15-42-8-14-34-32-40-33-28-21-56-37-25-5-24-1-54-39-17-46-45-18-36-52-43-6-59-51-26-55-58-27-16-60-38-47-7-49-31-3-12-48-23-2-4-19-22-57-35-44-11-20 |
| | 656,272.5 | 648,651.5 | 648,337.5 | 648,337 | |
| | 405,433 | 398,481 | 398,468 | 398,424 | 13-20-31-19-55-57-10-39-7-44-24-22-36-33-56-11-21-6-23-17-46-40-59-53-5-3-45-50-2-35-14-27-15-18-58-8-16-60-29-26-42-37-54-47-30-51-32-48-52-9-49-38-28-1-4-41-34-25-43-12 |
| | 319,501 | 318,855 | 318,805 | 318,805 | |
| 70 | 1,543,098 | 1,528,760 | 1,528,621 | 1,528,621 | |
| | 1,494,182 | 1,441,504 | 1,441,028 | 1,441,028 | |
| | 1,524,171.5 | 1,519,578.5 | 1,518,993.5 | 1,518,993.5 | |
| | 974,856 | 968,796 | 968,796 | 968,796 | |
| | 4,230,912.5 | 4,218,017.5 | 4,218,017.5 | 4,218,003.5 | 67-28-17-44-64-32-69-33-18-13-52-47-25-57-48-45-55-15-59-42-66-3-46-60-34-29-7-11-53-21-4-30-36-5-12-70-20-35-14-24-56-43-41-61-26-8-49-27-63-65-16-54-6-10-39-22-50-37-58-40-2-51-31-62-23-68-1-9-38-19 |
| 75 | 2,399,583.5 | 23,958,845 | 2,393,456.5 | 2,393,456.5 | |
| | 4,348,544 | 4,321,381 | 4,321,190 | 4,321,190 | |
| | 1,295,085 | 1,248,664 | 1,248,537 | 1,248,537 | |
| | 3,949,276.5 | 3,942,749.5 | 3,941,891.5 | 3,941,816.5 | 21-17-9-49-72-38-69-68-27-25-6-51-61-74-11-31-4-71-43-54-59-58-23-26-66-63-56-48-55-19-2-67-1-32-13-45-24-18-34-73-44-53-28-52-16-65-35-64-3-12-46-39-33-40-29-41-20-57-22-47-30-70-8-37-62-42-10-75-7-50-15-14-5-60-36 |
| | 1,816,455 | 1,792,038 | 1,791,408 | 1,791,408 | |
| 80 | 2,138,083.5 | 2,069,192.5 | 2,069,097.5 | 2,069,097.5 | |
| | 1,939,938 | 1,921,202 | 1,921,177 | 1,921,177 | |
| | 3,332,421 | 3,251,435 | 3,251,368 | 3,251,368 | |
| | 3,773,429 | 3,747,548 | 3,746,515 | 3,746,515 | |
| | 1,611,495 | 1,588,963 | 1,588,901 | 1,588,892 | 25-17-48-41-58-60-69-78-13-49-42-72-12-29-16-51-14-64-26-28-24-15-56-5-57-50-46-23-63-40-45-20-80-4-18-77-39-32-3-59-65-55-66-71-35-53-43-6-44-54-75-79-34-30-62-33-36-76-9-19-31-8-70-1-22-11-74-37-10-68-67-21-73-61-27-38-52-47-7-2 |

## 4. CONCLUSION

In this work, the squirrel search algorithm was successfully modified with a newly introduced operators (Msqrl-Exchange and Msqrl-Winter) to be applied correctly for solving single-row facility layout problems. Small and large facility numbers were considered in the testing process and comparisons were made with instances taken from the literature to show the efficiency of the modified algorithm Msqrl. In view of all the tests and results achieved in this work, the modified algorithm has a fast convergence to solutions, it was able to find better solutions to 5 of the used instances and it and outperformed both SDP and COA.

## ACKNOWLEDGEMENTS

The authors are grateful to the University of Mosul for their provided facilities.

## REFERENCES

[1] S. Ohmori and K. Yoshimoto, “A numerical stability analysis for application of particle swarm optimization to facility layout problem,” *International Journal on Computer and Engineering*, vol. 10, no. 7, pp. 164-170, Jul. 2018. [Online]. Available: http://www.enggjournals.com/ijcse/doc/IJCSE18-10-07-021.pdf

[2] M. F. Anjos, A. Kennings, and A. Vannelli, “A semidefinite optimization approach for the single-row layout problem with unequal dimensions,” *Discrete Optimization*, vol. 2, no. 2, pp. 113–122, Jun. 2005, doi: 10.1016/j.disopt.2005.03.001.

[3] P. Mladenović, *Combinatorics: A problem-based approach*. Springer Nature Switzerland AG, 2019, doi: 10.1007/978-3-030-00831-4.

[4] N. Khilwani, R. Shankar, and M. K. Tiwari, “Facility layout problem: An approach based on a group decision-making system and psychoclonal algorithm,” *International Journal of Production Research*, vol. 46, no.4, Feb. 2008, doi: 10.1080/00207540600943993.

[5] A. R. S. Amaral, “A new lower bound for the single row facility layout problem,” *Discrete Applied Mathematics*, vol. 157, no. 1, pp. 183-190, Jan 2009, doi: 10.1016/j.dam.2008.06.002.

[6] S. Ohmori, K. Yoshimoto, and K. Ogawa, “Solving Facility Layout Problem via Particle Swarm Optimization,” *2010 Third International Joint Conference on Computational Science and Optimization*, 2010, pp. 409-413, doi: 10.1109/CSO.2010.242.

[7] D. Datta, A. R. S. Amaral, and J. R. Figueira, “Single row facility layout problem using a permutation-based genetic algorithm,” *European Journal of Operational Research*, vol. 213, no. 2, pp. 388-394, Sep. 2011, doi: 10.1016/j.ejor.2011.03.034.

[8] N. H. Prasad, G. Rajyalakshmi, and A. S. Reddy, "A typical manufacturing plant layout design using CRAFT algorithm," *Procedia Engineering*, vol. 97, pp. 1808-1814, 2014, doi: 10.1016/j.proeng.2014.12.334.

[9] R. Matai, "Solving multi objective facility layout problem by modified simulated annealing," *Applied Mathematics and Computation*, vol. 261, pp. 302-311, Jun. 2015, doi: 10.1016/j.amc.2015.03.107.

[10] J. S. Goncalves and M. G. C. Resende, "A biased random-key genetic algorithm for the unequal area facility layout problem," *European Journal of Operational Research*, vol. 246, no. 1, pp.86–107, Oct. 2015, doi: 10.1016/j.ejor.2015.04.029.

[11] M. F. Anjos and M. V. C. Vieira, "An improved two-stage optimization-based framework for unequal-areas facility layout," *Optimization Letters*, vol. 10, no. 7, pp.1379–1392, 2016, doi: 10.1007/s11590-016-1008-6.

[12] H. Park and Y. Seo, "An efficient algorithm for unequal area facilities layout planning with input and output points," *INFOR: Information Systems and Operational Research*, vol. 57, no.1, pp. 56-74, 2017, doi: 10.1080/03155986.2017.1396709.

[13] D. Zhao, C. Aranha, and H. Kanoh, "Solving real-world facility layout problems using GA with Levy Flights and multi-decoding," *2017 18th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD)*, 2017, pp. 333-338, doi: 10.1109/SNPD.2017.8022742.

[14] Z. Kalita and D. Datta, "A constrained single-row facility layout problem," *International Journal of Advanced Manufacturing Technology*, vol. 98, no. 4, pp. 2173–2184. 2018, doi: 10.1007/s00170-018-2370-6.

[15] G. Moslemipour, T. S. Lee, and Y. T. Loong, "Solving stochastic dynamic facility layout problems using proposed hybrid AC-CS-SA meta-heuristic algorithm," *International Journal of Industrial and Systems Engineering*, vol. 28, no. 1, pp. 1–31, Jan. 2018, doi: 10.1504/IJISE.2018.088561.

[16] J. Feng and A. Che, "Novel integer linear programming models for the facility layout problem with fixed-size rectangular departments," *Computer and Operations Research*, vol. 95, pp.163–171, Jul. 2018, doi: 10.1016/j.cor.2018.03.013.

[17] Z. Lin and Z. Yingjie, "Solving the Facility Layout Problem with Genetic Algorithm," *2019 IEEE 6th International Conference on Industrial Engineering and Applications (ICIEA)*, 2019, pp. 164-168, doi: 10.1109/IEA.2019.8715148.

[18] M. Kim and J. Chae, "Monarch butterfly optimization for facility layout design based on a single loop material handling path," *Mathematics*, vol. 7, no. 2, p. 154. 2019, doi: 10.3390/math7020154.

[19] Z. Kalita and D. Datta, "The constrained single-row facility layout problem with repairing mechanisms," *Nature-Inspired Methods for Metaheuristics Optimization. Part of the Modeling & Optimization in Science and Technologies book series*, Springer, 2020, doi: 10.1007/978-3-030-26458-1_20.

[20] V. Tongur, M. Hacibeyoglu, and E. Ulker, "Solving a big-scaled hospital facility layout problem with meta-heuristics algorithms," *Engineering Science and Technology, an International Journal*, vol. 23, no. 4. pp: 951–959, Aug. 2020, doi: 10.1016/j.jestch.2019.10.006.

[21] M. Meskar, and K. Eshghi, "An algorithm for solving generalized single row facility layout problem," *American Journal of Operations Research*, vol. 10, no. 6, pp. 299-320, Jan. 2020, doi: 10.4236/ajor.2020.106017.

[22] S. Ingole, and D. Singh, "Fixed and flexible shape facility layout problems using biogeography-based optimization algorithm," *International Journal of Industrial and Systems*, vol. 37, no.1, pp. 84-118. 2021, doi: 10.1504/IJISE.2021.112469.

[23] A. Drira, H. Pierreval, and S. Hajri-Gabouj, "Facility layout problem: a survey," *Annual Reviews in Control*, vol. 31, no. 2, pp. 255–267, 2007, doi: 10.1016/j.arcontrol.2007.04.001.

[24] S. S. Heragu, *Facilities Design*. Taylor and Francis Group, 2011, doi: 10.1201/9781420066272.

[25] M. Jain, V. Singh, and A. Rani, "A novel nature-inspired algorithm for optimization: squirrel search algorithm," *Swarm Evolutionary Computation*, vol. 44, pp. 148-175, Feb. 2019, doi: 10.1016/j.swevo.2018.02.013.


## BIOGRAPHIES OF AUTHORS

**Najla Akram Al-Saati** received the BSc. degree in Computer Science from University of Mosul, Mosul, Iraq, in 1995. She received the M.Sc. and the Ph. D. degrees in Computer Science from the University of Mosul, Mosul, Iraq, in 1998 and 2005, respectively. She is currently an Assistant Professor at the Department of Software Engineering, University of Mosul. Her current research interests include: Artificial Intelligence, Swarm Algorithms, Bioinformatics, Optimization, and Software engineering. She can be contacted at email: dr.najla_alsaati@uomosul.edu.iq.

**Zakariya Yahya Algamal** received the B.S. degree in Statistics from University of Mosul, Mosul, Iraq, in 2001, the M.S. degree in Statistics from University of Mosul, Mosul, Iraq, in 2004, the PhD degree in Mathematical Science/Statistics from Universiti Teknologi Malaysia (UTM), Malaysia, in 2016, and Post. Doctorate in Mathematical Science/Statistics from Universiti Teknologi Malaysia (UTM), Malaysia, in 2017. He is a professor in Statistics in University of Mosul. His research interest includes the development of high dimensional data, generalized linear models, bioinformatics, chemoinformatics, and optimization algorithms. He can be contacted at email: zakariya.algamal@uomosul.edu.iq.