Engineering Science and Technology, an International Journal 23 (2020) 951–959

HOSTED BY

ELSEVIER

Contents lists available at ScienceDirect

Engineering Science and Technology, an International Journal

journal homepage: www.elsevier.com/locate/jestch

Full Length Article

Solving a big-scaled hospital facility layout problem with meta-heuristics algorithms

Vahit Tongur [a], Mehmet Hacibeyoglu [a,*], Erkan Ulker [b]

[a] Necmettin Erbakan University, Dere Aşıklar Mah. Demeç Sok, Konya 42140, Turkey
[b] Konya Teknik University, Ardıçlı Mah. Rauf Orbay Cad, Konya 42250, Turkey

ARTICLE INFO

Article history:
Received 8 May 2019
Revised 13 September 2019
Accepted 21 October 2019
Available online 6 November 2019

Keywords:
Hospital facility layout problem
Migrating birds optimization
Optimization problem
Tabu search
Simulated annealing

ABSTRACT

The main objective of the hospital facility layout problem is to place the polyclinics, laboratories and radiology units within the predefined boundaries in such way that minimize the movement cost of patients and healthcare staff. Especially in big-scaled hospitals including several different specialized departments, it is important in terms of hospital efficiency that interacting units are placed closely. Nowadays meta-heuristic algorithms are often used to solve optimization problems such as facility layout. In this study; polyclinic, laboratory and radiology units' layout of a big-scaled university hospital was organized using three meta-heuristic algorithms which are migrating bird optimization (MBO), tabu search (TS) and simulated annealing (SA). The results were compared with the existing clinic layout. Consequently MBO and SA meta-heuristic algorithms have given the same best results improving the existing clinic layout efficiency approximately by 58%.

1. Introduction

Placing facilities in a plant area, often referred to as a facility layout problem, has a significant impact on production costs, process operation, lead times and productivity [1]. The facility layout problems have gained importance after significant contributions of some research studies categorizing facility layout problems either as single floor or multi-floor layout problems [2–4]. The problem of hospital facility layout has received less attention in the literature compared to other facilities because many aspects of patient selection cannot be controlled by health managers [5] and demand is very uncertain [6]. Therefore, in addition to the complexity of the algorithms used to solve the problem of hospital placement, establishing the problem definition, providing the right parameters and obtaining reliable data are other critical points for obtaining the best solution. Hospital facility layout problem is an NP-Hard problem mathematically [3,7,8]. There are three commonly used types of solution approaches for facility layout problem in the literature, whether it is a single- or multi-floor settlement problem. The first is the quadratic assignment problem (QAP) which assumes equal areas for each department and known locations [9]. The second is the heuristic approaches which are efficient in solving several layout problems [10]. The third is the mixed-integer programming (MIP) which uses a distance-based objective for facility layout [11] for departments with equal and unequal areas [12]. Facility layout problem defined by Azadivar and Wang as placing of predetermined given number of facilities in feasible sizes and neighborhoods [13]. Lee and Lee placed facilities in different areas to improve facility efficiency in a boundary area [14]. Shayan and Chittilappilly considered the facility layout problem as an optimization problem. They have achieved optimal facility layout, taking into account interfacility interactions and material handling costs [15]. Recently, meta-heuristic based and swarm intelligence based methods have been used for solving facility layout problems. Sahin and Turkbey proposed a new hybrid meta-heuristic algorithm for the solution of multi objective facility layout problems, which is based upon simulated annealing (SA) and supported by taboo list [16]. Cheng and Lien presented a hybrid algorithm for multi-facility layout problems by integrating bee algorithm (BA) global search ability with the local search advantages of particle swarm optimization (PSO) [17]. Luo et al. represented a new model for emergency medical facilities layout planning in city by using ant colony optimization (ACO) [18]. Ileri used ACO to design the layout of polyclinics, laboratories and radiology units of a hospital especially for outpatients to minimize movement costs [19]. Huyen et al. investigated hospital-cost analysis. They used autoregressive integrated moving average (ARIMA) to estimate the number of

* Corresponding author at: Necmettin Erbakan University, Dere Aşıklar Mah. Demeç Sok, Konya 42140, Turkey.
E-mail address: hacibeyoglu@konya.edu.tr (M. Hacibeyoglu).
Peer review under responsibility of Karabuk University.

https://doi.org/10.1016/j.jestch.2019.10.006
2215-0986/

patients which coming to the hospital. Also, geographic information system is used to demonstrate the current fact or status through the distribution of patients and the hospital costs [20]. Aydin and Fogarty proposed a new method that distributed evolutionary SA for solving classical job-shop scheduling problem and the incapacitated facility location problem [21]. Kaveh and Sharafi used charged system search (CSS) optimization algorithm that acts on the basis of the interaction between charged particles to solve facility layout problem on a network [22]. Kaveh et al. proposed an adapted harmony search algorithm to find an optimal facility arrangement in an existing layout [23]. Chan et al. investigated the fine-tuning mutation rate of the genetic algorithm and the neighborhood function of the SA for solving the incapacitated facility location problem. They used variable mutation rate instead of constant mutation rate or randomly selected genes [24]. Yang et al. compared the performance of some meta-heuristics algorithms in terms of mathematical model for base station location planning problem which was modeled as a p-median problem [25]. As given above, several studies in the literature focused on layout problems of different facilities, but there are only a few studies on hospital layout problems. The objective of this study is to develop a multi-floor facility layout for hospitals to minimize the movement cost of patients. A cost reduction can be achieved by minimizing travel among entities. Our optimal hospital facility layout problem seeks the best possible arrangement of departments, within the predefined area to reduce distance between departments having close relation to each other. We developed a model for optimal placement of hospital departments at minimal cost. Various alternative layouts were also developed from the proposed strategies and the best possible one was chosen. As a novelty of the paper the migrating bird optimization (MBO) algorithm is used for the first time for solving facility layout problem of a hospital. Local search of MBO algorithm is strong. In order to measure the performance of MBO algorithm in real-world problem, results of the MBO is compared with the results obtained from the Tabu Search (TS) and SA algorithms. SA algorithm has been chosen for its simplicity and good localization. TS algorithm prevents local optimum and global search is effective. The results showed that MBO algorithm enhanced good performance for solving facility layout problem of a hospital.

Rest of the paper is organized as follows. The facility layout problem definition for hospital and statistical information of the problem are given at Section 2. MBO, TS and SA meta-heuristic algorithms are explained in detail at Section 3. Implementation of the used meta-heuristic algorithms of the problem is addressed at Section 4. The parameter setting of the used meta-heuristic algorithms is described at Section 5. At Section 6, the computational results are given and the paper is finalized with conclusions and discussions.

2. Model

As presented by Ileri [19], the hospital facility layout design should incorporate all essential requirements such as modelling of entities, movement difficulty, motion in the vertical direction, and the arrangement of departments within the predefined area in a way that reduces the distance between departments having high interaction and satisfies the increase in demand. To achieve efficient placement of the departments, the number of consultations between departments should be taken into consideration. This depends on the travel frequency between departments. The travel frequency determines the relationship factor and the departments having high traffic between them should be placed closer to each other than the ones with less traffic.

We analyzed these factors with the aim of minimizing the total movement cost.

1-. The interaction between departments, which depends on the traffic intensity between the two departments were calculated to make sure that the departments having more interaction were placed closer than the ones having lesser interaction.
2-. The frequency of visits of outpatients to each department were calculated to be able to place departments to which patient visits are more frequent closer to hospitals' main entrance and patient wards.
3-. The movement cost was calculated directly proportional to the distance, travel frequency, trip difficulty rating, and baseline travel cost and thus, varying any of the attributes changed the movement cost.
4-. The optimal solution was obtained by running simulations with the aim of altering one or some of these attributes.

For solving this problem, we formulated the objective function according to the physical structure of the hospital, the number of polyclinics to be placed with required size and the monthly average data for a year taken from hospital information system including number of inpatients and number of consultations between each department.

The physical structure of hospital shown in Fig. 1 consists of three blocks, thirty one feasible polyclinic areas with a size of 800 m2 and one hospital entrance. There are twenty six polyclinics to be placed with different sizes in the hospital which are given in Table 1.

In Table 2, a part of consultation numbers for polyclinics brain surgery, pediatric, medical oncology and otorhinolaryngology is given because showing all consultation data between polyclinics will take too much space.

The size and the distance to hospital entrance of the areas to be placed are given in Table 3.

According to the inputs in Tables 1–3 and the modeling entities explained above, the objective function of the model is formulated as in Eq. (1).

$$min\left(\sum_{i=0}^{m} x_i y_i + \sum_{j=0}^{n}\sum_{k=0}^{m} z_{jk} t_{jk}\right) \tag{1}$$

where, n is the number of polyclinic, m is the number of areas that polyclinics to be placed, x_i is the distance of the feasible areas to the hospital entrance, y_i is the monthly number of patients in polyclinic that placed to ith area, z_{jk} is the number of the patients who sent from polyclinic that placed to jth area to polyclinic that placed to kth area. t_{jk} is the distance between jth and kth areas. z_{jk} matrix can be considered as flow matrix in QAP problems. This matrix con-

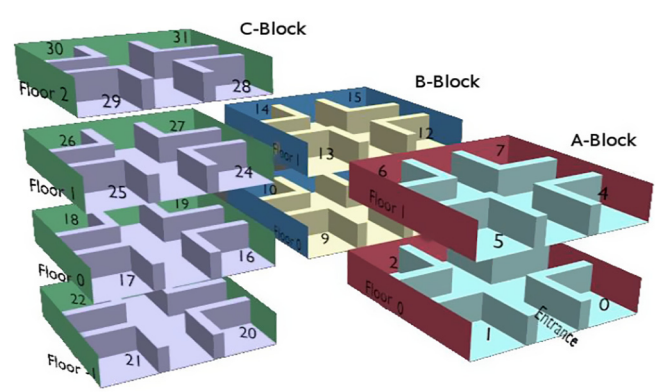

Fig. 1. Physical structure of the hospital.

Table 1
Number of patients coming to the polyclinics and required size for each polyclinic.

Polyclinic Code	Polyclinic Name	Average number of inpatients per month	Required size (m^2)
0–3	Laboratories	–	3200
4	Family Practice	25,609	800
5	Brain Surgery	14,609	800
6–7	Pediatrics	55,406	1600
8	Dermatology	28,540	800
9	Infection	8442	800
10	Psychiatry	10,527	800
11	General surgery	15,135	800
12	Chest diseases	15,659	800
13–14	Eye diseases	38,668	1600
15	Hematology	6772	800
16	Rheumatology	10,613	800
17	Nephrology	5855	800
18	Gastroenterology	8221	800
19	Endocrinology	19,638	800
20	Medical oncology	7483	800
21	Internal diseases	1232	800
22	Gynecology	22,510	800
23	Plastic surgery	11,524	800
24	Cardiology	10,684	800
25–26	Otorhinolaryngology	27,421	1600
27	Neurology	11,578	800
28	Nuclear medicine	140	800
29	Orthopedics	28,155	800
30	Medical genetics	2869	800
31	Urology	13,647	800

Table 2
The patient consultations between the polyclinics.

Sending Polyclinic	Accepting Polyclinic	Average number of consultations per month
Brain Surgery	Laboratory	70
Brain Surgery	Neurology	5
Brain Surgery	Orthopedics	5
Pediatric	Laboratory	1486
Pediatric	Brain Surgery	4
Pediatric	Otorhinolaryngology	13
Pediatric	Orthopedics	15
Medical oncology	Laboratory	364
Medical oncology	Infection	4
Medical oncology	Chest diseases	6
Medical oncology	Cardiology	12
Medical oncology	Otorhinolaryngology	4
Medical oncology	Neurology	2
Otorhinolaryngology	Laboratory	190
Otorhinolaryngology	Pediatric	9
Otorhinolaryngology	Infection	3
Otorhinolaryngology	Chest diseases	7
Otorhinolaryngology	Endocrinology	5

sist of consultation values that are number of intra-movements in hospital.

Our study had two main restrictions. First of all, the hospital facility layout must consider the cost associated with patient movements and the movements of the accompanying people but we did not consider the number of accompanying people as it was impossible to reach valid data. Secondly, doctors, medical and non-medical staff were not considered since their movements are directly dependent on the needs of the primary travel entities.

3. Methods

3.1. Migrating birds optimization algorithm

MBO firstly proposed by Duman et al. in 2012 [26]. MBO is inspired from energy saving of migrating birds in *V* formation flying. This algorithm is designed for discrete problems and tested on QAP problems which are based on real life problems.

Table 3
The size and the distance to hospital entrance of the areas to be placed.

Area Code	Distance to entrance	Size
0	10	800
1	10	800
2	20	800
3	20	800
4	60	800
5	60	800
6	70	800
7	70	800
8	110	800
9	110	800
10	120	800
11	120	800
12	160	800
13	160	800
14	170	800
15	170	800
16	210	800
17	220	800
18	220	800
19	210	800
20	240	800
21	250	800
22	250	800
23	240	800
24	260	800
25	270	800
26	270	800
27	260	800
28	310	800
29	320	800
30	320	800
31	310	800

Most common flying style of the migrating birds is *V* formation to be able to fly longer distances. It is known that basic instinct of this formation is energy saving. Leader bird is the one which spends the most energy in *V* formation. Other birds can fly longer by the wind energy created by wing moves of bird that in front of.

The MBO algorithm has been used for solving many problems such as flow shop problem [27–32], credit card fraud detection problem [33], knapsack problem [34], travelling salesman problem [35,36], two-way number partitioning problem [37], robotic U-shaped assembly line balancing problem [38] in the literature.

The MBO algorithm is started with randomly generated initial solutions and these solutions are tried to improve at every step. Swap, insert and inverse operations are used generate candidate solutions for neighbor to improve the current solutions. Then, the generated candidate neighbor solutions is shared with the solution that following it. Each current solutions are compared to the best neighbor solution. If the neighbor solution is better than the current solution, the neighbor solution replaces with the current solution. In addition, a leader replacement is carried out at certain times to transfer better solutions to both sides of the flock. The MBO algorithm consists of the initial population generation, candidate solution generation, neighbor sharing process and leader replacement stages.

3.1.1. Initial population generation

The MBO algorithm is started with initial solutions that randomly generated and try to improve these solutions. One of these solutions is assigned as leader bird and rest solutions are placed on left and right side of the leader bird. Thus, the initial population is generated like Fig. 2.

In the initial population each bird represents a solution.

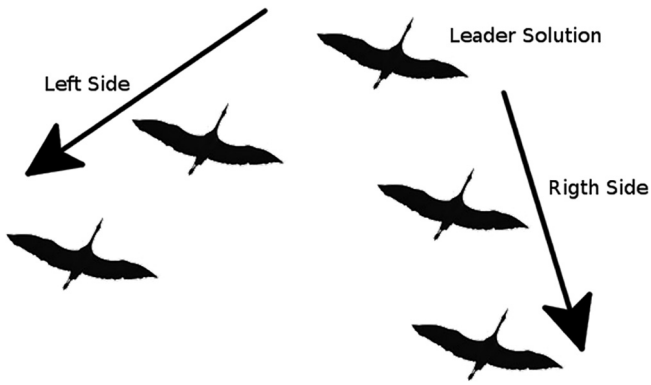

Fig. 2. Initial population of the MBO.

3.1.2. Candidate solution generation

The MBO algorithm uses swap, insert and inverse operations to generate candidate neighbors for local search. In this study, candidate neighbor solutions are generated with swap operation of selected two positions randomly from the current solution. Number of candidate neighbor solutions for leader bird and number of candidate solutions for rest birds are different because of the structure of the MBO algorithm. Number of candidate neighbor solutions are calculated according to the Eqs. (2)–(4) [36].

$$k \in N^{+}; k \geq 3; \quad k = \{3, 5, 7, 9, \cdots\} \tag{2}$$

$$p \in N^{+}; 1 \leq p \leq (k-1)/2 \tag{3}$$

$$r = k - p \tag{4}$$

where, k is number of neighbor solutions of the leader bird, r is number of neighbor solutions of other birds which except leader bird, p is number of neighbor sharing.

An example of generating candidate neighbor solution with swap operation is shown in Fig. 3 [36].

3.1.3. Neighbor sharing process

Neighbor sharing process is a specific feature of the MBO algorithm that distinguish the MBO from other meta-heuristic algorithms. This feature provides interaction of all birds in the flock. Neighbor solutions of the leader bird are shared to left and right side. Neighbor solutions of the other birds are shared to only own side. According to Eq. (2), k neighbor solutions are generated for leader bird. These k neighbor solutions are evaluated according to objective function and they are sorted from the best solution to the worst solution. The best solution is compared with the current solution of the leader bird. If the neighbor solution is better than the current solution, the neighbor solution replaces the current solution. Remaining p neighbor solutions are transferred to the bird which is following on left side of the leader bird. After these processes, the remaining neighbor solutions (if any) are discarded. r neighbor solutions are generated for the bird at the left rear position of the leader and added p solutions which are coming from the leader. These $p + r$ solutions are evaluated and sorted from best to worst. The best neighbor solution is compared with the current solution of this bird. If the neighbor solution is better than the current solution, the neighbor solution replaces the current solution. Remaining p solutions are transferred to the following bird and other neighbor solutions are discarded. This process is applied to the right side of the flock too. Neighbor sharing is repeated until the end of the flock. Neighbor sharing is shown in Fig. 4 [36].

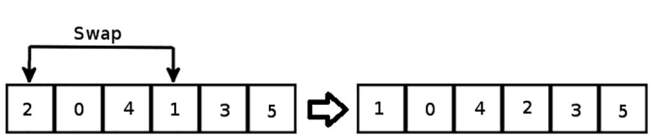

Fig. 3. Neighbor solution generation.

3.1.4. Leader replacement

In the MBO algorithm, a flap parameter (f) is used to keep individuals in the same sequence for a certain period of time. Then, leader replace processing is applied. First replacement is applied to the left side of the flock. While the leader is sent behind the left side of the flock, another bird that follows the leader is replaced by the leader. Thus, a new sequence is created and parameter m is reset. The next replacement is applied to the right side of the flock. This process is continued until the algorithm is terminated. Leader replacement process is shown in Fig. 5 [36].

The pseudo code of the MBO algorithm is given in Algorithm 1 as follows:

Algorithm 1. The pseudo code of the MBO algorithm

```
Generate random initial population
Repeat
    Repeat
        Generate k neighbors for leader
        Generate r neighbors for other solutions
        Share best p neighbors to rear solution
        If (best neighbor solution < current solution)
            Current solution = neighbor solution
    Until flap value
    Replace leader
Until termination criterion
Return best solution in the flock
```

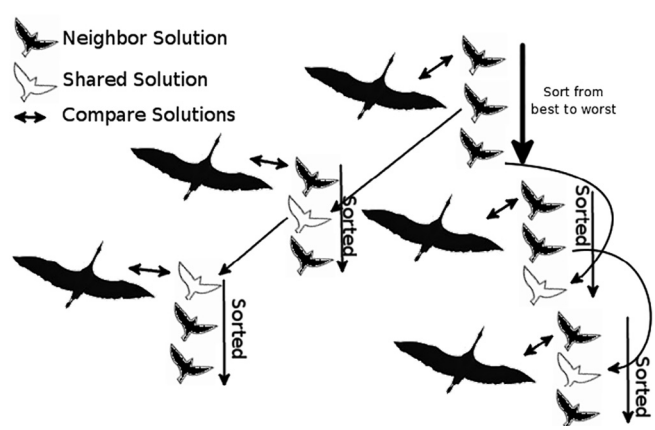

Fig. 4. Neighbor sharing for $k = 3$ and $p = 1$.

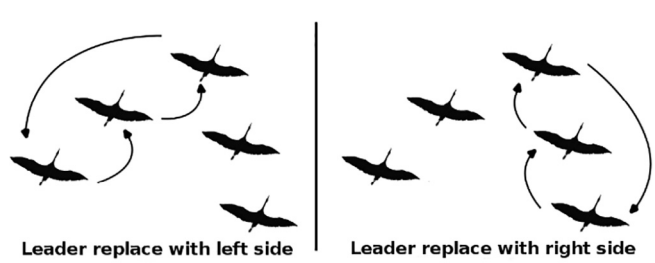

Fig. 5. The leader replacement.

3.2. Tabu search algorithm

The TS algorithm was first proposed by Glover for combinatorial optimization problems in 1989 [39]. In the literature, the TS algorithm has been used in many different optimization problems such as vehicle routing problem [40], flow shop problem [41], traveling salesman problem [42].

In the TS algorithm, a single solution is generated and tried to find the global optimal solution by applying local search techniques like swap and insertion operations. There are usually two different memory structures in the TS algorithm. These are short-term memory and long-term memory structures. The short-term memory allows the selection of the best possible move to generate a new solution and prevents some solutions from being generated which are called tabu. In this way, the TS algorithm avoids from local optimum and finds global optimum. When the predetermined iteration value is reached, the specified tabu is broken and previously found solution can be selected again as a solution. Throughout the search process, the best solution found by the TS algorithm is taken into the long-term memory and all the solutions generated are compared with the best solution. If there is a better solution than the best solution, the long-term memory is updated.

The pseudo code of the TS algorithm is given in Algorithm 2 as follows:

Algorithm 2. The pseudo code of the TS algorithm

```
Generate randomly initial solution. Set this solution as the
  current solution and the best solution.
Repeat
    Find neighboring solutions using local search techniques.
    Select a neighboring solution that non-tabu or meets tabu
  break criteria, even if it is tabu.
    If the new solution is better than the current solution, set the
  new solution as tabu.
    If the new solution is better than the best solution, set the
  new solution as the best solution.
Until termination criterion
```

3.3. Simulated annealing algorithm

SA algorithm is proposed by Kirkpatrick et al. for the solution of combinatorial problems [43]. The SA algorithm is designed to find global optimum by avoiding local solutions for functions with multiple variables. This algorithm takes this name due to the fact that it exemplifies the perfect arrangement of atoms and minimizes potential energy while cooling solid objects. The process of heating the solid to the melting point and then cooling it slowly until it crystallizes in the perfect lattice structure is called annealing. Annealing is achieved in three stages: heating the material to a certain temperature (heating), keeping this temperature for a suitable period (waiting) and reducing the temperature in a controlled manner (cooling). During heating, the particles in the solid object are randomly transformed into liquid form, and when properly cooled, the crystal particles of regular structure are formed.

The physical annealing process is based on the method of Monte Carlo by Metropolis et al. [44] At a given temperature T, the probability distribution of the system energies is determined according to Eq. (5).

$$P(E) = e^{-E/(kT)} \tag{5}$$

where, E is the system energy and k is the Boltzmann constant.

In case of a slight change in the state of the system, the new energy of the system is calculated according to the Metropolis algorithm. The current state of the solid with E_1 energy is mechanically changed by displacement of a randomly selected small part and the energy level E_2 is switched to the other state. If the energy is reduced ($\Delta E = (E_2\text{-}E_1) < 0$), the system changes to this new state. If the energy is increased ($\Delta E > 0$), it is decided whether or not to accept the energy state E_1 according to Eq. (6). A uniform number is generated ($\gamma \in [0,1]$). If the condition given in Eq. (6) is provided, the new solution is accepted as the existing solution. Otherwise, the existing solution is not changed.

$$\gamma \leqslant e^{\Delta E/T} \tag{6}$$

where, ΔE is the difference between the energy levels of the two states. This acceptance criterion is known as the Metropolis criterion. According to Eq. (5), for all energy states at high temperatures, P(E) converges to 1. With a small probability, the system may have a high energy level, even at low temperatures.

Simulated annealing algorithm has provided successful results in vehicle routing [45–47], feature selection [48] traveling salesman problem [49], facility layout problems [50] and job shop scheduling [51].

The pseudo code of the SA algorithm is as follows:

Algorithm 3. The pseudo code of the SA algorithm

```
The pseudo code of the SA algorithm as follow.
Generate randomly initial solution Si
Repeat
    Generate a neighbor solution Sn
    if (Sn better than Si)
      Set the Sn as existing solution
    else if (random(0,1) < eΔE/T)
      Set the Sn as existing solution
    Update T
Until termination criterion
```

4. Implementation of meta-heuristics to the hospital facility layout problem

The problem addressed in this study is a discrete problem. TS, SA and MBO meta-heuristics are designed for discrete problems. Therefore, there is no change in the structure of algorithms to solve the hospital facility layout problem. In the solution of this problem, the polyclinic codes given in Table 1 are arrayed randomly. The size of this array is equal to the number of fields (32) which is shown in Fig. 1. The index of array shows the number of the area where the polyclinic is placed. The value in the array is the polyclinic number. An example arrangement of polyclinics are shown in Fig. 6.

In Fig. 6, the polyclinic 2 is placed in area 0. After all polyclinics are placed in the available areas, the fitness of the solution is evaluated according to Eq. (1). The solution is evaluated that considering the distance from the area where the polyclinics placed are distance to entrance of the hospital, number of patients coming to polyclinics, patient consultation between polyclinics and distance between polyclinics.

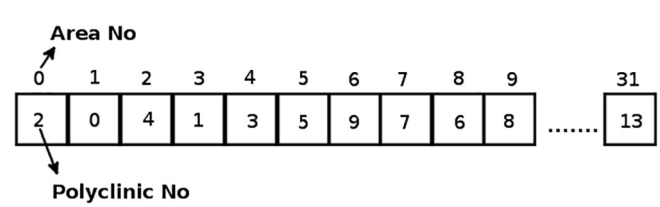

Fig. 6. Permutation coding for a solution.

Example. Assume that four polyclinics (A, B, C, D) will be placed in five equal areas. In the Tables 4–6 the number of patients coming to the polyclinics, the patient consultations between the polyclinics, the areas to be placed and the distances to the hospital entrance are given respectively.

In Table 4, polyclinic B is shown as two separate polyclinics because it has double the amount of the area to be placed. The number of patients in Table 4 represents the y variable in Eq. (1).

In Table 5, columns 2 and 3 represent the same polyclinics (B) and need to place in the adjacent area. Therefore, the number of consultations among themselves is very large (500). Table 5 represents matrix z in the Eq. (1).

According to the sample solution, the polyclinic 4, 2, 5, 1 and 3 are placed in the 1st, 2nd, 3rd, 4th and 5th areas respectively.

Table 7 represents matrix t in the Eq. (1). A sample solution is as follows.

4	2	5	1	3

5. Parameter settings

In order to obtain the best solution from the polyclinic placement problem, parameter settings are performed for both algorithms. In parameter settings, the TS and the MBO algorithms are run 30 times independently and the parameters are selected according to the average of the best results.

Table 4
Number of patients coming to the polyclinics.

Polyclinic No (Name)	Number of patients (y)	Required size (m^2)
1 (A)	14,609	800
2 (B)	55,406	1600
3 (B)		
4 (C)	27,421	800
5 (D)	10,684	800

Table 5
The number of patient consultations between the polyclinics.

		Polyclinics sending consultations				
	No	1	2	3	4	5
Polyclinics accepting consultations	1	0	75	75	16	85
	2	14	0	500	50	41
	3	14	500	0	50	41
	4	40	9	9	0	63
	5	32	3	3	4	0

Table 6
Size of area and distances to the hospital entrance.

Area No	Size	Distance to the entrance (x)
1	800	10
2	800	10
3	800	20
4	800	20
5	800	30

Table 7
The distance between the areas.

	1	**2**	**3**	**4**	**5**
1	0	5	10	10	20
2	5	0	10	10	20
3	10	10	0	5	10
4	10	10	5	0	10
5	20	20	10	10	0

5.1. MBO parameter setting

Four parameters of the MBO algorithm is examined. These parameters are population, flapping, neighborhood, and sharing.

In Table 8, the population parameter is evaluated with values of 31, 41, 51 and 61, respectively. Flap, neighborhood and sharing parameters are fixed to 20, 3, and 1 respectively. As can be seen from Table 8, the best average value is obtained at population = 61. As can be seen from Table 9, the best average value is obtained at flap = 20.

In Table 9, the flap parameter is evaluated with values of 20, 30, 40 and 50, respectively. Population, neighborhood and sharing parameters are fixed to 61, 3, and 1 respectively.

In Table 10, the neighbor parameter is evaluated with values of 3, 5, 7 and 9, respectively. Population, flapping and sharing parameters are fixed to 61, 20, and 1 respectively.

As can be seen from Table 10, the best average value is obtained at neighbor = 3. Due to the structure of the MBO algorithm, the minimum value of the neighbor parameter can be assigned as 3. According to Eq. (3), if $k = 3$, sharing parameter (p) can be only 1. Therefore, the sharing parameter is not examined and its value is assigned as 1.

According to the results obtained from parameter examining, the best parameters of the MBO algorithm are given in Table 11.

5.2. TS parameter setting

Three parameters of the TS algorithm is examined. These parameters are tabu length, penalizing long term and long term length.

Table 8
Parameter setting of population.

Fixed Parameters		Population Value	Fitness Average
Parameter	Value		
Flap	20	31	10447906,36
Neighbor	3	41	10381691,10
Share	1	51	10340931,30
		61	10290411,95

Table 9
Parameter setting of flap.

Fixed Parameters		Flap Value	Fitness Average
Parameter	Value		
Population	61	20	10237226,00
Neighbor	3	30	10276514,80
Share	1	40	10302385,16
		50	10256843,93

Table 10
Parameter setting of neighbor.

Fixed Parameters		Neighbor Value	Fitness Average
Parameter	Value		
Population	61	3	10224274,63
Flap	20	5	10275422,16
Share	1	7	10305902,26
		9	10269521,46

Table 11
Best parameters for the MBO.

Parameters	Value
Population	61
Flap	20
Neighbor	3
Share	1

Table 12
Parameter setting of tabu length.

Fixed Parameters		Tabu length Value	Fitness Average
Parameter	Value		
Penalizing long term	10	10	11324841,17
		20	11162676,17
long term length	100	30	11000655,80
		40	11293772,77

In Table 12, the tabu length parameter is evaluated with values of 10, 20, 30 and 40, respectively. Penalizing long term and long term length parameters are fixed to 10 and100 respectively. As can be seen from Table 12, the best average value is obtained at tabu length = 30.

In Table 13, the penalizing long term parameter is evaluated with values of 5, 10, 15 and 20, respectively. Tabu length and long term length parameters are fixed to 30 and 100 respectively. As can be seen from Table 13, the best average value is obtained at penalizing long term = 20.

In Table 14, the long term length parameter is evaluated with values of 80, 90, 100 and 110, respectively. Tabu length and penalizing long term parameters are fixed to 30 and 20 respectively. As can be seen from Table 14, the best average value is obtained at long term length = 80.

According to the results obtained from parameter examining, the best parameters of the TS algorithm are given in Table 15.

Table 13
Parameter setting of penalizing long term.

Fixed Parameters		Penalizing long term	Fitness
Parameter	Value	Value	Average
tabu length	30	5	11163703,20
		10	11391734,70
long term length	100	15	11268983,43
		20	11133762,73

Table 14
Parameter setting of long term length.

Fixed Parameters		long term length	Fitness
Parameter	Value	Value	Average
tabu length	30	80	10979237,50
		90	10991142,00
penalizing long term	20	100	11203672,33
		110	11203930,57

Table 15
Best parameters for the TS.

Parameters	Value
tabu length	30
penalizing long term	20
long term length	80

6. Experimental results

It is not possible to compare the results of this problem with the results of benchmark problems in literature because this is a specific and special problem. Therefore, this problem is solved with not only the MBO but also the TS and SA to ensure accuracy of the results. Experiments are run with Intel(R) Core(TM) i5-3330 CPU @ 3.00 GHz processor, 4 GB RAM and Linux Ubuntu 14.04 (64-bit) operating system. All algorithms are coded with QT Creator 3.0.1 gcc compiler and C++ language. Algorithms are independently run 30 times in order to make sure the results. Runtime of each experiment is 120 s. 120 s are considered to be sufficient for both algorithms according to parameter settings. Polyclinic layouts obtained from the experiments of the MBO, TS and SA are given in Table 16.

According to Table 16, the best result of the MBO and SA algorithms are 10142848. On the other hand, according to Table 16, the best result of the TS algorithm is 10254893. This result is about 1.1% worse than the MBO and SA algorithms.

The existing polyclinic layout and polyclinic layout obtained best result from the MBO and SA algorithms are given in Table 17.

According to Table 17, polyclinics 0–3 are placed to the 0th floor of C-Block. According to the number of consultation, it seems that there is flow from each polyclinic to laboratories. When examined consultation numbers of the polyclinics and distance to the laboratories, the laboratories are placed to the 0th floor of C-Block may be considered reasonable. Polyclinics 6 and 7 are placed to 0th floor of B-Block, polyclinics 13 and 14 are placed to 0th floor of A-Block and polyclinics 25 and 26 are placed to 1st floor of A-Block. As seen in Table 17, the proposed polyclinic layout and the existing polyclinic layout are almost completely different except policlinics 10 and 17. They are placed to 1st floor of B-Block in both cases. In addition, the fitness values of the proposed layout and the existing layout are given in Table 17. According to the proposed layout, the current layout has been improved by about 58%.

Also, convergence curves of the MBO, TS and SA algorithms are given Figs. 7, 8 and 9, respectively.

According to Figs. 7, 8 and 9, convergence rate of the MBO and TS algorithms are close to each other and more speed than SA algorithm. However, obtained results from the SA algorithm are better than the MBO and TS algorithms according to the average and worst results.

Table 16
The results obtained from the MBO, TS and SA.

	Algorithms		
	MBO	TS	SA
Min.	**10142848,0**	10254893,0	**10142848,0**
Avg.	10236612,3	11054017,2	**10158816,4**
Worst	10652953,0	11872804,0	**10331538,0**

Table 17
Existing and proposed polyclinic layout.

Block	Floor	Proposed Polyclinic Layout	Existing Polyclinic Layout
A-Block	0	13, 14, 29, 8	29, 23, 30, 22
	1	26, 25, 5, 12	8, 16, 24, 31
B-Block	0	4, 6, 7, 11	15, 18, 19, 28
	1	10, 27, 17, 23	10, 11, 17, 21
C-Block	−1	19, 16, 31, 22	12, 20, 4, 5
	0	2, 0, 3, 1	6, 7, 9, 27
	1	18, 9, 15, 24	13, 14, 25, 26
	2	21, 28, 30, 20	0, 1, 2, 3
Fitness Value		10, 142, 848	17, 433, 012

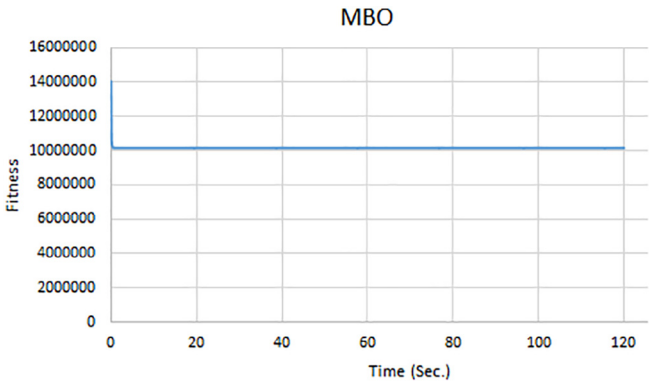

Fig. 7. Convergence curves of the MBO.

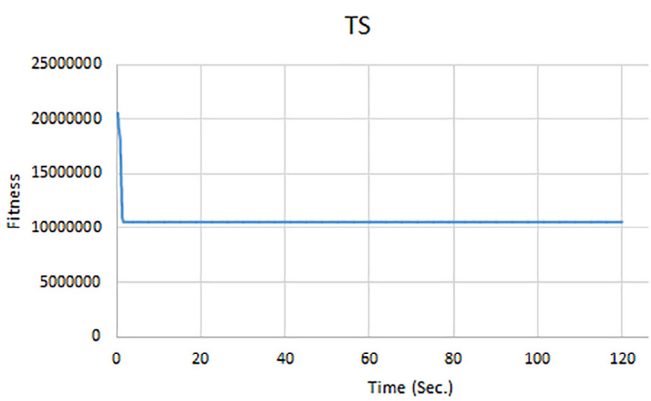

Fig. 8. Convergence curves of the TS.

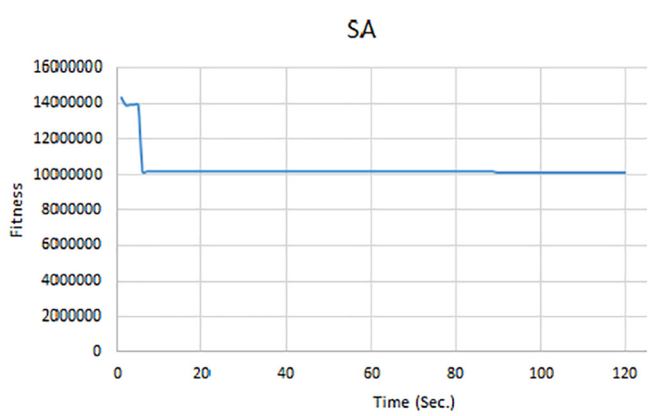

Fig. 9. Convergence curves of the SA.

The results obtained from all three meta-heuristics are examined statistically. Wilcoxon test is used in this study to test whether there is a significant difference between the methods used [52]. There is no significant difference between the two methods compared according to the H_0 hypothesis. There is a significant difference between the two methods compared according to the alternative hypothesis H_1. Acceptance rate of H_0 hypothesis is determined as 95%. That is, if the difference between the two methods is less than 5% ($\alpha = 0{,}05$), there is a significant difference between the two methods. The results of the methods at 5% significance level are given in Table 18.

Table 18
p-values obtained from Wilcoxon test.

p-values		
MBO&TS	MBO&SA	SA&TS
0	0,149	**0**

According to Table 18, there is no significant difference between the MBO and SA algorithms (0,149 > 0,05). On the other hand, it is seen that there is a significant difference between the MBO and TS algorithms and the TS and SA algorithms (0 < 0,05).

7. Conclusion and discussion

Optimization problems are real life problems which we can face in several areas in our daily life. Nowadays, generally meta-heuristic algorithms are mostly used in solution of these optimization problems. Placing the facilities in suitable areas in the facility layout problem increases operational efficiency by reducing production costs, processing process and delivery times. The way the polyclinics are placed within the hospital has an important role in determining the transportation times of the patients and the hospital staff. It is the time of transportation for a patient is to come directly to the polyclinic for treatment or to be served from one polyclinic to another polyclinic. The prolongation of these transportation times leads to a long hospital stay and a decrease in hospital efficiency. Therefore, correct placement of these outpatient clinics ensures the best use of the medical workforce, minimizing the unnecessary distance between patients, staff and visitors and increasing hospital productivity. In the study, we used the MBO, SA and the TS meta-heuristics for the polyclinic layout problem of a big scaled hospital. According to the obtained results of the experiments, success of the MBO, SA and the TS algorithms on facility layout problems are tested on a real life problem and it is seen that total cost of the MBO and SA algorithms much better than the TS algorithm. When the fitness value obtained from experimental results is compared with the existing fitness value, it is seen that the efficiency is 58% better. As a result, meta-heuristics can be used in polyclinic layout problem for hospitals. For future work, some assumptions and constraints can be added to the problem. Such as an inpatient has several appointments in several polyclinic or some inpatients requires assistance staff for their transfer with wheelchairs and stretchers. Discharge operations of inpatients also can be considered.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] A. Drira, H. Pierreval, S. Hajri-Gabouj, Facility layout problems: a survey, Ann. Rev. Control 31 (2) (2007) 255–267.
[2] J. Mecklenburg, Process Plant Layout, Institute of Chemical Engineers, London, 1985.
[3] A. Kusiak, S. Heragu, The facility layout problem, Eur. J. Oper. Res. 29 (1987) 229–251.
[4] R. Meller, K. Gau, The facility layout problem: recent and emerging trends and perspectives, J. Manuf. Syst. 15 (1996) 351–366.
[5] J. Tompkins, J. White, W. Bozer, J. Tanchoco, Facilities Planning, John Wiley & Sons, New York, USA, 2003.
[6] Sarma, Sisira, Demand for outpatient healthcare, Appl. Health Econ. Health Policy 7 (4) (2009) 265–277.
[7] R.S. Amaral, A new lower bound for the single row facility layout problem, Discrete Appl. Math. 157 (2009) 183–190.

[8] M. Mohammadi, K. Forghani, A novel approach for considering layout problem in cellular manufacturing systems with alternative processing routings and subcontracting approach, Appl. Math. Model. 38 (14) (2014) 3624–3640.
[9] A. Barbosa-Povoa, R. Mateus, A. Novais, Optimal two-dimensional layout of industrial facilities, Int. J. Prod. Res. 39 (12) (2001) 2567–2593.
[10] S.P. Singh, R.R. Sharma, A review of different approaches to the facility layout problems, Int. J. Adv. Manuf. Technol. 30 (5–6) (2006) 425–433.
[11] B. Montreuil, A modeling framework for integrating layout design and flow network design, in: Proceedings of the Material Handling Research Colloquium, 1990, pp. 43–58.
[12] T. Lacksonen, Pre-processing for static and dynamic facility layout problems, Int. J. Prod. Res. 35 (1997) 1095–1106.
[13] F. Azadivar, J. Wang, Facility layout optimization using simulation and genetic algorithms, Int. J. Prod. Res. 38 (17) (2000) 4369–4383.
[14] Y.H. Lee, M.H. Lee, A shape-based block layout approach to facility layout problems using hybrid genetic algorithm, Comput. Ind. Eng. 42 (2) (2002) 237–248.
[15] E. Shayan, A. Chittilappilly, Genetic algorithm for facilities layout problems based on slicing tree structure, Int. J. Prod. Res. 42 (2) (2002) 237–248.
[16] R. Sahin, O. Turkbey, A new hybrid heuristic algorithm for the multi objective facility layout problem, J. Faculty Eng. Archit. Gazi Univ. 25 (1) (2010) 119–130.
[17] M.Y. Cheng, L.C. Lien, A hybrid AI-based particle bee algorithm for facility layout optimization, Eng. Comput. 28 (1) (2012) 57–69.
[18] Q. Luo, Q. Su, J. Le, L. Lu, in: 10th International Conference on Service Systems and Service Management, IEEE, 2013, pp. 224–227.
[19] Y.Y. Ileri, The Importance of Layout Organization in Hospital Management Efficiency: A Model in S.U. Medical Faculty Hospital, Selcuk University, 2013.
[20] D.T.T. Huyen, N.T. Binh, T.M. Tuan, T.Q. Trung, N.G. Nhu, N. Dey, L.H. Son, Analyzing trends in hospital-cost payments of patients using ARIMA and GIS: case study at the hanoi medical university hospital, Vietnam, J. Med. Imaging Health Inf. 7 (2) (2017) 421–429.
[21] M.E. Aydin, T.C. Fogarty, A distributed evolutionary simulated annealing algorithm for combinatorial optimisation problems, J. Heuristics 10 (3) (2004) 269–292.
[22] A. Kaveh, P. Sharafi, Charged system search algorithm for minimax and minisum facility layout problems, Asian J. Civ. Eng. 6 (12) (2011) 703–718.
[23] A. Kaveh, M.A.A. Shakouri, M.S. Zolfaghari, An adapted harmony search based algorithm for facility layout optimization, Int. J. Civ. Eng. 1 (10) (2012) 37–42.
[24] K.Y. Chan, M.E. Aydin, T.C. Fogarty, Main effect fine-tuning of the mutation operator and the neighbourhood function for uncapacitated facility location problems, Soft. Comput. 10 (11) (2006) 1075–1090.
[25] J. Yang, M.E. Aydin, J. Zhang, C. Maple, UMTS base station location planning: a mathematical model and heuristic optimisation algorithms, IET Commun. 1 (5) (2007) 1007–1014.
[26] E. Duman, M. Uysal, A.F. Alkaya, Migrating birds optimization: a new metaheuristic approach and its performance on quadratic assignment problem, Inf. Sci. 217 (2012) 65–77.
[27] V. Tongur, E. Ülker, Migrating birds optimization for flow shop sequencing problem, J. Comput. Commun. 2 (04) (2014) 142.
[28] A. Sioud, C. Gagné, Enhanced migrating birds optimization algorithm for the permutation flow shop problem with sequence dependent setup times, Eur. J. Oper. Res. 264 (1) (2018) 66–73.
[29] B. Zhang, Q.K. Pan, L. Gao, X.L. Zhang, H.Y. Sang, J.Q. Li, An effective modified migrating birds optimization for hybrid flowshop scheduling problem with lot streaming, Appl. Soft Comput. 52 (2017) 14–27.
[30] T. Meng, Q.K. Pan, J.Q. Li, H.Y. Sang, An improved migrating birds optimization for an integrated lot-streaming flow shop scheduling problem, Swarm Evol. Comput. 38 (2018) 64–78.
[31] Q.K. Pan, Y. Dong, An improved migrating birds optimisation for a hybrid flowshop scheduling with total flowtime minimisation, Inf. Sci. 277 (2014) 643–655.
[32] K.Z. Gao, P.N. Suganthan, T.J. Chua, in: An Enhanced Migrating Birds Optimization Algorithm for No-Wait Flow Shop Scheduling Problem, IEEE, 2013, pp. 9–13.
[33] E. Duman, I. Elikucuk, Solving credit card fraud detection problem by the new metaheuristics migrating birds optimization62–71, International Work-Conference on Artificial Neural Networks, Springer, Berlin, Heidelberg, 2013.
[34] E. Ulker, V. Tongur, Migrating birds optimization (MBO) algorithm to solve knapsack problem, Proc. Comput. Sci. 111 (2017) 71–76.
[35] V. Tongur, E. Ülker, The analysis of migrating birds optimization algorithm with neighborhood operator on traveling salesman problem, in: Intelligent and Evolutionary Systems, Springer, Cham, 2016, pp. 227–237.
[36] V. Tongur, E. Ülker, PSO-based improved multi-flocks migrating birds optimization (IMFMBO) algorithm for solution of discrete problems, Soft. Comput. 23 (14) (2019) 5469–5484.
[37] M. Hacibeyoglu, K. Alaykiran, A.M. Acilar, V. Tongur, E. Ulker, A Comparative Analysis of metaheuristic approaches for multidimensional two-way number partitioning problem, Arabian J. Sci. Eng. 43 (12) (2018) 7499–7520.
[38] Z. Li, M.N. Janardhanan, A.S. Ashour, N. Dey, Mathematical models and migrating birds optimization for robotic U-shaped assembly line balancing problem, Neural Comput. Appl. (2019) 1–17.
[39] F. Glover, Tabu search part I., ORSA J. Comput. 1 (3) (1989) 190–206.
[40] F.A.T. Montané, D.G. Roberto, A tabu search algorithm for the vehicle routing problem with simultaneous pick-up and delivery service, Comput. Oper. Res. 33 (3) (2006) 595–619.
[41] J. Grabowski, W. Mieczyslaw, A very fast tabu search algorithm for the permutation flow shop problem with makespan criterion, Comput. Oper. Res. 31 (11) (2004) 1891–1909.
[42] C.N. Fiechter, A parallel tabu search algorithm for large traveling salesman problems, Discrete Appl. Math. 51 (3) (1994) 243–267.
[43] S. Kirkpatrick, C.D. Gelatt, M.P. Vecchi, Optimization by simulated annealing, Am. Assoc. Adv. Sci. 220 (4598) (1983) 671–680.
[44] N. Metropolis, A.W. Rosenbluth, M.N. Rosenbluth, A.H. Teller, E. Teller, Equation of state calculations by fast computing machines, J. Chem. Phys. 21 (6) (1953) 1087–1092.
[45] H. Bagherlou, A. Ghaffari, A routing protocol for vehicular ad hoc networks using simulated annealing algorithm and neural networks, J. Supercomput. 74 (6) (2018) 2528–2552.
[46] L. Wei, Z. Zhang, D. Zhang, S.C. Leung, A simulated annealing algorithm for the capacitated vehicle routing problem with two-dimensional loading constraints, Eur. J. Oper. Res. 265 (3) (2018) 843–859.
[47] K. Karagul, Y. Sahin, E. Aydemir, A. Oral, A simulated annealing algorithm based solution method for a green vehicle routing problem with fuel consumption, in: Lean and Green Supply Chain Management, Springer, Cham, 2019, pp. 161–187.
[48] M.M. Mafarja, S. Mirjalili, Hybrid whale optimization algorithm with simulated annealing for feature selection, Neurocomputing 260 (2017) 302–312.
[49] S.H. Zhan, J. Lin, Z.J. Zhang, Y.W. Zhong, List-based simulated annealing algorithm for traveling salesman problem, Comput. Intell. Neurosci. 2016 (2016) 8.
[50] S. Kulturel-Konak, A. Konak, A large-scale hybrid simulated annealing algorithm for cyclic facility layout problems, Eng. Optim. 47 (7) (2015) 963–978.
[51] N. Shivasankaran, P.S. Kumar, K.V. Raja, Hybrid sorting immune simulated annealing algorithm for flexible job shop scheduling, Int. J. Comput. Intell. Syst. 8 (3) (2015) 455–466.
[52] S. García, D. Molina, M. Lozano, F. Herrera, A study on the use of non-parametric tests for analyzing the evolutionary algorithms' behaviour: a case study on the CEC'2005 special session on real parameter optimization, J. Heuristics 15 (6) (2009) 617.