Expert Systems with Applications 42 (2015) 6586–6597

Contents lists available at ScienceDirect

Expert Systems with Applications

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

Modified migrating birds optimization algorithm for closed loop layout with exact distances in flexible manufacturing systems

Sadegh Niroomand [a,*], Abdollah Hadi-Vencheh [b], Ramazan Şahin [c], Béla Vizvári [a]

[a] Department of Industrial Engineering, Eastern Mediterranean University, Famagusta, North Cyprus, Turkey
[b] Department of Mathematics, Khorasgan (Isfahan) Branch, Islamic Azad University, Isfahan, Iran
[c] Department of Industrial Engineering, Gazi University, Ankara, Turkey

ARTICLE INFO

Article history:
Available online 1 May 2015

Keywords:
Flexible manufacturing system
Facility layout problem
Closed loop layout
Migrating birds optimization
Meta-heuristic

ABSTRACT

This paper addresses a new meta-heuristic algorithm to solve a closed loop layout problem. The proposed algorithm applies a modified version of the recently invented migrating birds optimization method. The computational experiments show that in most of the benchmark problems the results obtained from the proposed migrating birds optimization method is better than those obtained by other methods which are published in the literature.

1. Introduction

Facility layout problems (FLPs) determine the placement of facilities in order to obtain an efficient arrangement based on some given criteria. The common criterion considered in most of FLPs is minimization of total material handling cost between facilities. Material handling cost plays a very important and critical role while calculating the costs of a manufacturing system. Tompkins et al. (1996) showed that approximately 20–50% of the total cost incurred by a manufacturing system comes from material handling. Obviously, material handling cost of a manufacturing system depends on its layout type and the way its material handling paths are determined. Therefore, in order to reduce the material handling cost, an efficient layout of facilities is necessary.

A classification of FLPs was given by Chae and Peters (2006) and Niroomand and Vizvári (2013) where they mentioned that there are two types of layout problems such as (i) general facility layout problem and (ii) machine layout planning. General facility layout problem locates some departments considering their general area (mostly rectangular departments). Machine layout planning uses the specific shape of machines or departments for designing their related layout e.g. cell formation problem that determines the layout of machines in a manufacturing cell (Javadi, Jolai, Slomp, Rabbani, & Tavakkoli-Moghaddam, 2013). Schematically, FLPs are classified in four well-known categories, namely, open-field, closed loop, single row and ladder layout as are illustrated in Fig. 1. These categories are distinguished by the shape of their material handling path. Das (1993) and Rajasekharan, Peters, and Yang (1998) (also Cong et al., 2012; Niroomand, Takacs, & Vizvari, 2011) discussed an open-field layout in details while Chae and Peters (2006) and Tavakkoli-Moghaddam and Panahi (2007) as well as Niroomand and Vizvári (2013) focused on closed loop layout problems. Single row layout problems were also discussed by many other authors e.g. Kothari and Ghosh (2013), Ou-Yang and Utamima (2013), Amaral (2009), Anjos, Kennings, and Vannelli (2005) and Ficko, Brezocnik, and Balic (2004).

In open-field layout problems, unlimited space is considered to locate the manufacturing cells on the ground. The most prominent limitation of designing an open-field layout is non-overlapping constraints of the model that forces the cells to lie on the ground without any overlapping. Some other constraints are also needed to determine the pick-up points of cells and to measure distances between the cells. Das (1993) introduced one such mathematical model and used a four-step heuristic method to solve it. Rajasekharan et al. (1998) used genetic algorithm to propose a new solution to Das' model. Kim and Kim (2000) considered cells with different input and output points (pick-up and drop-off points) in open-field layout problems.

The literature of closed loop layout is not as rich as other types of layout problems. Just three studies focused on arrangement of cells on a physical closed loop as mentioned before. Tavakkoli-Moghaddam and Panahi (2007) introduced a mathematical model to locate cells just outside of a closed loop. They used

* Corresponding author. Tel.: +90 392 630 1103.
E-mail addresses: sadegh.niroomand@cc.emu.edu.tr, niroomand59@gmail.com (S. Niroomand).

http://dx.doi.org/10.1016/j.eswa.2015.04.040
0957-4174/

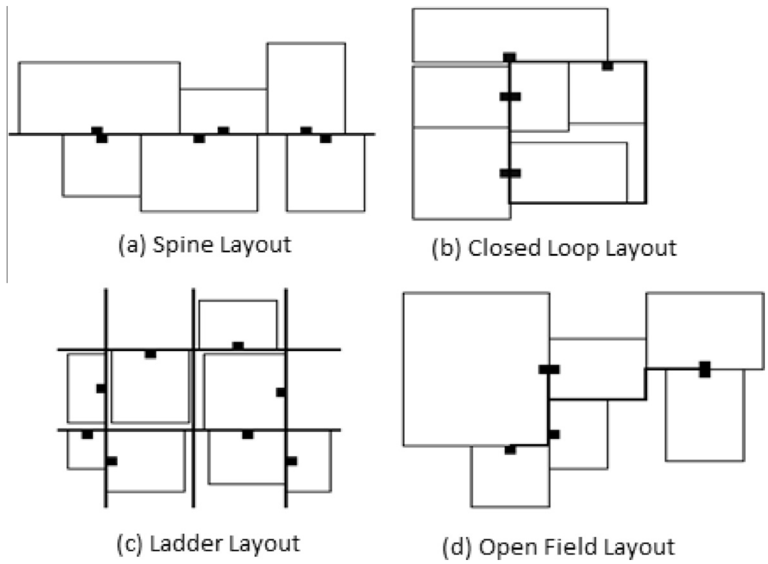

Fig. 1. Different patterns for arrangement of facilities on floor (Niroomand & Vizvári, 2013).

Lingo software and some meta-heuristics to solve their model. Chae and Peters (2006) benefited from Das' model (open-field layout model) and used simulated annealing method to arrange the cells around a given closed loop of material handling path. They located cells on both inside and outside of a closed loop. It should be mentioned that no mathematical model for closed loop layout introduced by Chae and Peters (2006). The most recent study on closed loop layout was done by Niroomand and Vizvári (2013) which introduced an exact mixed integer linear programming (MILP) model that locates cells on both sides of a closed loop. They used Xpress software to solve their model. While the studies of Das (1993) and Rajasekharan et al. (1998) (open-field layout) and Chae and Peters (2006) consider an approximation of distances of cells (Manhattan (rectilinear) distance) in the obtained solution, the model of Niroomand and Vizvári (2013) measures the exact distances between cells. These distances will be explained in next section explicitly.

FLPs tend to be of Nondeterministic Polynomial-time hard (NP-hard) type problems (Garey & Johnson, 1979). In practice, applying exact solution methods to NP-hard problems is time consuming (Ou-Yang & Utamima, 2013). Meaning that when the problem size increases, the problem cannot be solved optimally in a polynomial run time (see Bénabès, Poirson, & Bennis, 2013). Such difficulty motivates a researcher of FLP to focus on developing efficient meta-heuristic algorithms. In most cases, these algorithms solve FLPs in shorter running time in comparison with exact methods. Some well-known meta-heuristic and decision making algorithms applied to FLPs are genetic algorithms, simulated annealing, tabu search, ant colony, etc. (see Aiello, Enea, & Galante, 2006; Brintup, Ramsden, & Tiwari, 2007; Garcia-Hernandez et al., 2013; Hadi-Vencheh & Mohamadghasemi, 2013; Islier, 1998; Kaveh, Majazi Dalfard, & Amiri, 2013; McKendall & Shang, 2006; McKendall, Shang, & Kuppusamy, 2006; Naderi & Azab, 2014; Pierreval, Caux, Paris, & Viguier, 2003; Sahin, Ertogral, & Turkbey, 2010; Solimanpur, Vrat, & Shankar, 2005; Wang, Hu, & Ku, 2005).

Recently, a new meta-heuristic algorithm named migrating birds optimization (MBO) was proposed by Duman, Uysal, and Alkaya (2012). They applied their algorithm to quadratic assignment problems and proved its efficiency. This paper introduces a modification of the MBO algorithm to the closed loop layout model with exact distances which was recently introduced by Niroomand and Vizvári (2013). Taguchi experimental design (Taguchi, 1986) is used to find the best level of parameters of the introduced algorithm. To show applicability of the proposed method the results are compared with those of the MBO algorithm, simulated annealing (SA) algorithm (Kirkpatrick, Gelatt, & Vecchi, 1983; Niroomand & Vizvári, 2014) as well as Xpress software in the design of closed loop layout.

The rest of this paper is organized as follow. Section 2 discusses differences between approximate open-field and closed loop layouts and the exact closed loop layout. The MBO algorithm designed for closed loop layout with exact distances is proposed in Section 3. The proposed modified MBO algorithm is introduced in Section 4. A detailed computational experiment is done in Section 5. The paper ends with a conclusion in Section 6.

2. Problem statement: closed loop layout with exact distances

In this study the closed loop layout model which was explicitly presented in Niroomand and Vizvári (2013) is tackled. The model and its brief literature is conceptually presented in this section.

As aforementioned, Das (1993) introduced a general mathematical model for the open-field layout problem. In that model the objective function is the sum of Manhattan distances of any pair of cells weighted by the flow value between them. The Manhattan distance of a pair of cells is calculated as sum of absolute differences of Cartesian coordinates of their pick-up points as shown by Fig. 2. As closed loop layout is a special case of open-field layout, the concepts of Das' model were used by Chae and Peters (2006) to arrange cells around a rectangular closed loop material handling path meta-heuristically. In both studies by Das (1993) and Chae and Peters (2006), the approximation of material handling cost was evaluated by the objective function of the model because Manhattan distances may not be correct in some cases. In the case of open-field layout the Manhattan distance of a pair of cells is not exact if there is at least one cell laying between that pair of cells (see Fig. 2). Neither in a closed loop formation, the Manhattan distance of a pair of cells yield an exact distance when the cells are placed on two opposite sides of a rectangular closed loop as shown in Fig. 2.

Niroomand and Vizvári (2013) introduced a new MILP model for closed loop layout problems. The model includes the basic open-field model of Das (1993) and some additional constraints.

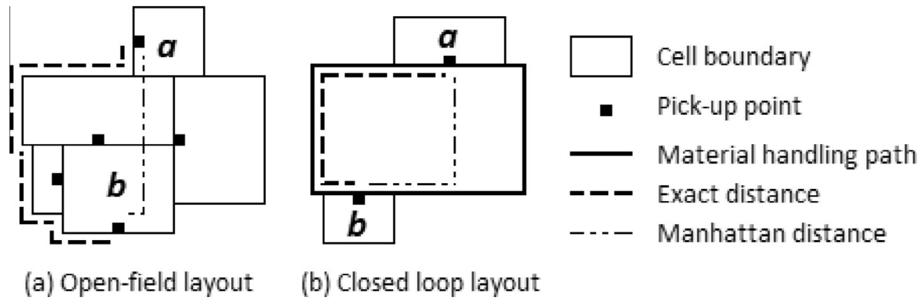

Fig. 2. Exact distance vs. Manhattan distance in closed loop and open-field layout.

Another basic assumption of the model is that the pick-up point of each cell is placed in the middle of one of its edges (Like cell *a* in Fig. 2). The additional sets of constraints of the model of closed loop layout with exact distances are as follow:

- A set of constraints to allow any manufacturing cell to lie inside or outside of a closed loop.
- A set of constraints to force pick-up point of any cell to be placed on only one edge of the closed loop.
- A set of constraints to describe the position of any pair of cells (e.g. "cell *a* on upper side and cell *b* on lower side" in Fig. 2(b)).
- A set of constraints to measure the exact distance of any pair of cells (Fig. 2 (b)).

Niroomand and Vizvári (2013), used Xpress to solve the benchmark problems that are given by Das (1993) and Chae and Peters (2006). A sample arrangement of a sequence of cells that can be an output of the model is shown in Fig. 3.

The next section of the paper applies a recently introduced meta-heuristic algorithm and its modified version to solve the model introduced by Niroomand and Vizvári (2013).

3. A new solution methodology for closed loop layout with exact distances

The experiments of Niroomand and Vizvári (2013) using Xpress show that the model of closed loop layout with exact distances is an NP-hard model. Their results proved that when the size of the problem increases, the model cannot be optimally solved in a deterministic polynomial time. Therefore, in this section a meta-heuristic algorithm is proposed to solve the model.

In the proposed meta-heuristic method any sequence of cells locating around a predefined size of a closed loop forms a solution. Hence, a first-fit principle is applied as a placement strategy to arrange a sequence of manufacturing cells (as a solution) around a predefined size of closed loop. Afterwards, standard migrating birds optimization (MBO) technique together with its modification is used to improve the solution by generation of some neighboring solutions. MBO algorithm is a new meta-heuristic method, proposed by Duman et al. (2012), to solve combinatorial optimization problems. In the next parts of this section, the proposed algorithm will be explained in details.

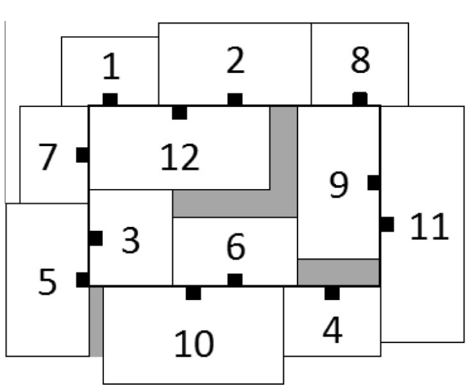

Fig. 3. A sample solution of the model introduced by Niroomand and Vizvári (2013) for 10 cells. Gray areas are dead spaces that con not be used for any other cell.

3.1. First-fit principle as placement strategy of cells on a given size of closed loop

The placement strategy applied in this part is a modification of placement strategy that was used by Chae and Peters (2006). Some changes were made in the procedure in order to decrease the cost of the layout.

A given sequence of cells located on the sides of a closed loop will be the output of this placement strategy. A closed loop material handling path has four sides. The upper, right, below and left sides of the closed loop are named side1, side 2, side 3 and side 4, respectively (see Fig. 4). To start locating the sequence of cells, the first cell of the sequence is placed on the intersection of sides 1 and 4, making sure it remains outside side 4 in a way that its pick-up point is placed exactly on the corner point. Then, any other cell of the given sequence of cells may lay inside or outside of any side of the closed loop. For any other cell of the sequence the ordering priorities are as follows: outside side 1, inside side 1, outside side 2, inside side 2, outside side 3, inside side 3, outside side 4 and inside side 4. Clearly, for each cell the first priority (side) which has enough free space for the length of the cell is selected then the free space of that side is updated to be used for other cells of the sequence.

An example of placement strategy for random sequence of cells, e.g. $(1, 2, 3, 4, 5, 6, 7, 8, 9, 10)$ on a given size of a closed loop is shown in Fig. 5. The procedure starts with cell 1 which is placed on the corner of sides 1 and 4 and outside of the given closed loop.

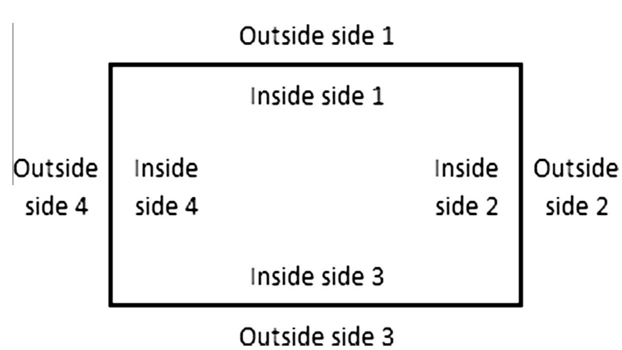

Fig. 4. Different sides of a closed loop.

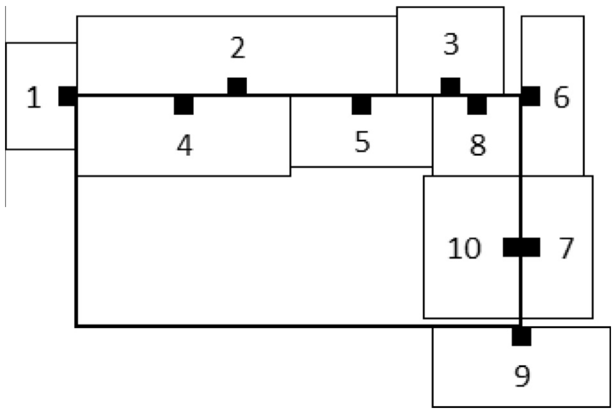

Fig. 5. A solution obtained from first-fit principle for a given sequence of cells.

Then, for cell 2 the first priority (outside of side 1) is selected because its free space is greater than the length of cell 2. The other cells are located based on their first fitted priority. For example, for locating cell 6, the first and second priorities, i.e. outside and inside of side 1, are not fitted because its required length is greater than the free space available outside and inside side 1, as such the third priority, i.e. outside side 3 is suitable to cell 6. Then, because it is the first cell lying outside side 2, it is placed on the corner of sides 1 and 2. Using the same reasoning cell 7 is located outside side 2, but since the length of cell 8 is equal to the free space available inside side 1 (second priority), it is placed on side 1. The same also happens for cells 9 and 10.

3.2. Natural migration of birds

Although a group of birds may migrate in different formations, one of the most popular birds' migrating formations is the V formation. In this formation of migrating birds, some parameters like wing-tip spacing (WTS), wing span (b), angle of the V formation (α), depth, and maximum width of the wing (w) are important to form an effective V formation. A V formation is shown in Fig. 6.

In order to minimize the flying consumed energy by a group of birds, the values of WTS and depth are useful parameters to contemplate. Although Lissaman and Shollenberger (1970) and Badgerow and Hainsworth (1981) applied some experimental studies to find the best value of WTS, Hummel and Beukenberg (1989) stated that,

$$\mathrm{WTS}_{opt} = -0.05\mathrm{b} \tag{1}$$

where b shows the wing span of a bird.

Furthermore, as mentioned by Rayner (1979), the optimum value for depth can be obtained using:

$$\mathrm{Depth}_{opt} = 2\mathrm{w} \tag{2}$$

where w shows maximum width of the wing of a bird.

In a flock of migrating birds, most of the energy is spent by the leader which flies at the front of the flock, so it gets tired faster than the others. Then, usually after some time the leader goes to the end of the flock and one of its followers will be the new leader of the flock. Further details regarding the migration of birds can be found in Duman et al. (2012).

3.3. Migrating birds optimization method

Duman et al. (2012) applied the concepts of V formation of migrating birds to develop a new meta-heuristic method called migrating birds optimization (MBO) algorithm for solving quadratic assignment problems (QAP).

The MBO algorithm was introduced based on the neighboring search technique. Similar to birds of a V shape migration, some initial solutions (also called main solution) organize a V formation including one leading solution and some followers. In the flock of solutions a limited number of neighboring solutions for each main solution are generated. The neighboring solutions of each main solution are evaluated and if there are any improvements among them, that main solution is replaced by the solution provided by the most improved neighbor. Then, each main solution can further try to be improved by the help of some neighbors of the solution in its front. This means that each solution, will share some of its unused neighbors (that were not the best neighbor of that solution) to the next (behind) main solution. Therefore, except the leading solution, the other main solutions of the flock have chance to be improved by one of the neighbors of the main solution in front of them. The procedure is repeated a number of times (tours). Then, the leading solution moves to the end of the flock and one of its followers become the new leader. The same procedure is done and repeated for the new flock. The algorithm continues until a number of iteration is reached (total number of generated neighboring solutions in the flock). Finally, the best solution of the flock is introduced as the solution of the MBO algorithm.

The notations used in the MBO algorithm are as follows,

n: number of initial (main) solutions of the flock.
k: number of neighboring solutions generated for each initial (main) solution.
x: number of neighboring solutions shared with the next solution.
m: number of tours.
K: number of iterations (total number of generated neighbor solutions).

The similarities between the parameters of the MBO algorithm and real migration of birds in V formation based on the details of

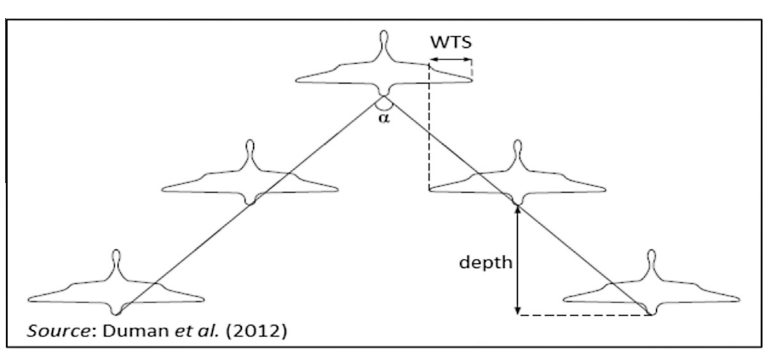

Fig. 6. V formation of migrating birds.

the study provided in Duman et al. (2012) are summarized in Table 1.

Unlike some other meta-heuristic methods e.g. simulated annealing, the MBO algorithm starts with several solutions at the same time. Hence more feasible solutions of the given problem are covered. This also provides the algorithm with the chance to evaluate more neighboring solutions in different directions at the same time. Even if a solution cannot improve itself by its neighbors, it has chance to be improved using the neighbors of other solutions that may be obtained from a different direction of the solution space of the problem.

3.4. Migrating birds optimization algorithm for closed loop layout

In the MBO algorithm for closed loop layout, any randomly generated sequence of cells placed around a closed loop with given size forms a random solution. The above-mentioned first-fit principle is used as a placement strategy for the sequence of cells around the given closed loop. Then the solution is evaluated by calculating its cost based on the exact distances as discussed in Niroomand and Vizvári (2013). Swapping procedure is used to generate the neighbors of a main solution. In the sequence of cells of a main solution two cells are selected randomly and the sequence of cells for the neighbor solution is obtained by interchanging the place of selected cells in the sequence. The size of closed loop for a solution and its neighbor is the same. As was the case with the sequence of the main solution, the neighbor's sequence is placed around the same closed loop and is evaluated in the same way. For example, if in the solution given in Fig. 4, the cells 4 and 8 are randomly selected, the neighbor sequence will be $(1,2,3,8,5,6,7,4,9,10)$ and the related neighbor solution by applying the above-mentioned placement strategy with the same closed loop size is shown in Fig. 7. Because enough free space is not available for the length of cells 4 and 9 outside and inside side 2, they are placed outside side 3 which is the next priority as explained before.

The MBO algorithm for closed loop layout starts with an initial size of closed loop such as $v = h$ (v is the vertical length and h is the horizontal length of the closed loop). An acceptable estimation for initial v and h is $v = h = \left(\sum_i^N l_i\right)/2$, where l_i is the length of the side of cell i that contains its pick-up point while N shows the number of cells (size of the problem) that should be located around the closed loop. In the initial size of the closed loop, the MBO procedure explained in Section 3.3 is performed; whereas the generation and evaluation of the solutions and the method to generate their neighboring solutions are explained at the beginning of this section. Afterwards the size of the closed loop is decreased by a unit and the last flock of solutions obtained from the MBO procedure of initial size of the closed loop is moved to the new size of the closed loop and the MBO procedure is repeated again based on the new size of closed loop. The algorithm continues its iterations until the last feasible size of closed loop is reached ($v_f \times h_f$, the size which is physically enough for locating the cells based on the placement strategy e.g. $v_f = h_f$). In any of the considered

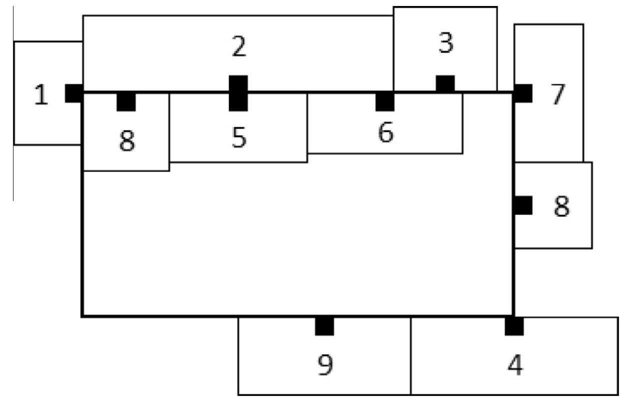

Fig. 7. Neighbor solution of solution shown in Fig. 4 when cells 4 and 8 are swapped.

sizes of the closed loop, the best solution is saved and finally the best solution among all saved solutions is introduced as the best solution found by the MBO algorithm.

The flowchart of the MBO algorithm for closed loop layout based on the aforementioned explanations is shown in Fig. 8.

3.5. Modified migrating birds optimization (MMBO) method for closed loop layout

In the MBO algorithm any main solution may be improved using some unused neighbors in front of the main solution. In this way some unused neighbors of any solution is replaced by the neighbors of the main solution at its front.

As mentioned above, in the V shape migration of birds the value of WTS lets any bird be covered by a part of the body of the bird in front. This means that the impact of any bird gives benefit to the bird behind it to spend less power. By exploiting this fact, the proposed MMBO tries to improve the neighbors of each main solution by impact of the neighbors of the other main solutions of the flock. In fact the neighbors do not move between main solutions directly but they are used to generate some new neighbors for other main solutions.

In the proposed MMBO, any neighbor of each main solution (one process prior to process A of MBO flowchart) is regenerated by the same neighbor of the main solution in front or beside it using crossover and mutation operators. Later the regenerated neighbor is considered instead of the original one if it has a less objective function value. After regeneration of all neighbors of the flock, each main solution will be tried to be replaced by new neighbor which has better cost, if such a neighbor exists.

Crossover and mutation operators are widely applied in genetic algorithm (GA) based meta-heuristic methods. These operations are performed on current solutions, named as 'parents', to generate a new solution from them, called 'offspring'. Two different types of crossovers are used in this part of the paper to generate new neighbors from an existing flock of solutions.

3.5.1. Type 1 crossover

In this type of crossover two parents are selected (a neighbor of a main solution and the same neighbor of the main solution in front of it) and by applying Partial Mapped Crossover (PMX) (see Eiben & Smith, 2003; Garcia-Hernandez et al., 2013) a single offspring is generated. This method guarantees that no element is repeated in the final offspring. Then with probability of α a mutation operation is performed on the new neighbor obtained by PMX. The PMX and mutation used in crossover type 1 is shown in Figs. 9 and 10. In the pseudo code of crossover type 1, n is the number of

Table 1
Similarities of MBO algorithm and V shape natural migration of birds.

Parameter of the MBO algorithm	Similar concept in real migration of birds in V formation
n	Birds
k	The induced power required which is inversely proportional to the speed
x	WTS
m	The number of wing flaps before a change in the leading bird

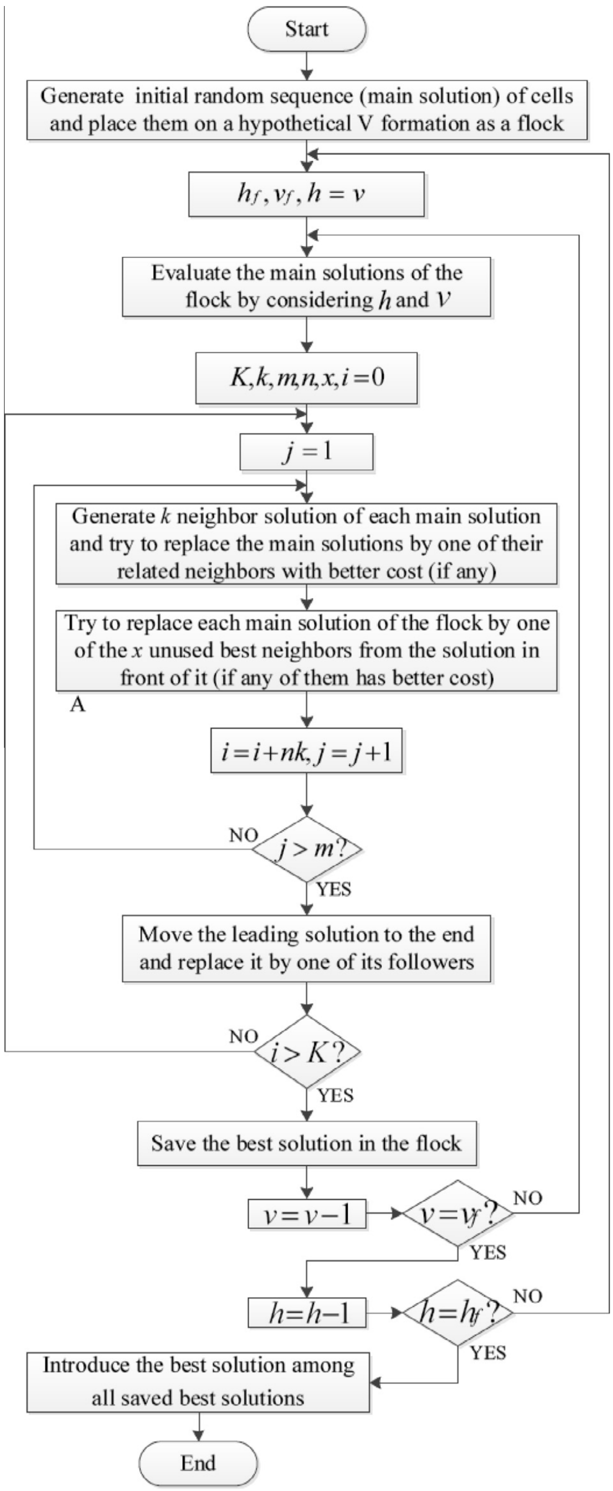

Fig. 8. Flowchart of MBO algorithm for closed loop layout.

main solutions (birds) of the flock and k shows the number of neighbors generated for each main solution. This procedure occurs at process A of the MBO flowchart (instead of movement of neighbors to the next solution).

3.5.2. Type 2 crossover

In a crossover of type 2, a procedure similar to the crossover of type 1 is used. The difference is that two of the same neighbors from two main solutions beside each other are selected as two parents. The PMX and mutation operators result in two different offsprings which may be used instead of their parents if their cost is less. Fig. 10 shows the crossover of type 2 and its flowchart can be seen in Fig. 11. The same happens at process A of the MBO flowchart.

The schematic example of the above-mentioned crossover operations is shown in Fig. 12 (the mutation operator is not included in the figure) where the flock includes three main solutions and each main solution includes three neighbors. Each solution shows the sequence of cells to be placed around a closed loop.

4. Computational experiments

The MMBO and MBO algorithms for closed loop layout was coded in Matlab and was run on a computer with an Intel

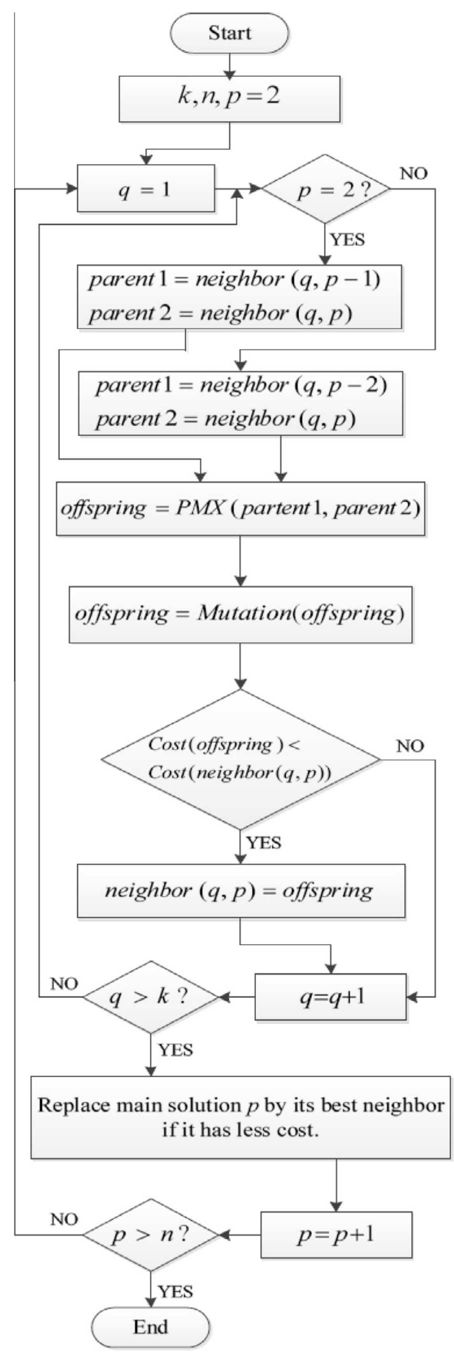

Fig. 9. Flowchart of crossover type 1.

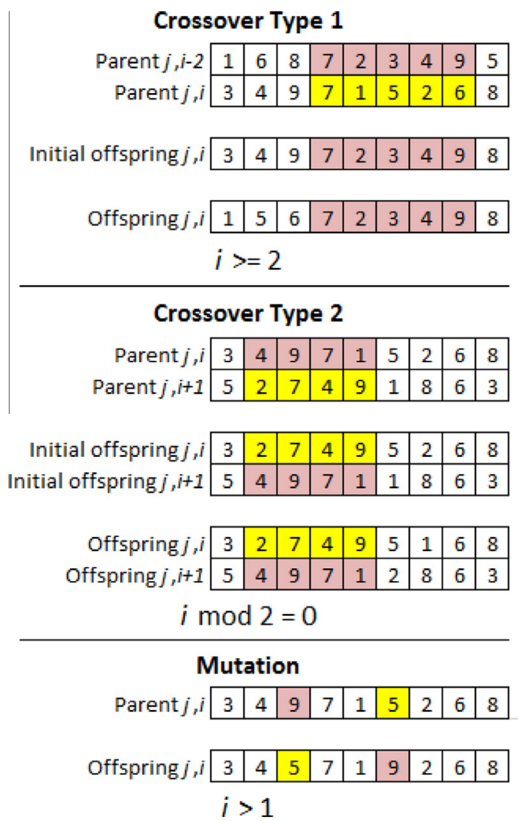

Fig. 10. Crossover and mutation operators used for neighbor regeneration.

Pentium 4 processor, 3.00 GHz processor and 2.00 GB RAM. In order to compare the performance of these algorithms, the SA algorithm (Chae & Peters, 2006) was also developed using Matlab. The same benchmark problems from the literature of closed loop problem were used to carry out the experiments. This set of problems consists of 8 problems of different sizes including 4, 6, 8, 10, 12, 14, 16 and 18-cell problems. In addition, three more benchmark problems with size of 20, 25 and 30-cellwere generated uniformly to perform more experiments.

The objective function value obtained by the MBO algorithm has the same concept with the objective function value named "closed loop" in Niroomand and Vizvári (2013). This means that the constant inter-cell transportation cost is not included in the objective function value of the MBO algorithm. The reason is that the assumption of the MBO algorithm and the closed loop layout model of Niroomand and Vizvári (2013) locate the pick-up point of each cell on one of its edges and the exact distance of any pair of cells is calculated from their pick-up point.

Like other meta-heuristic methods, the performance of the MBO method is also sensitive to the level of its parameters. A set of experiments for selecting the level of parameters of MBO method were performed by Duman et al. (2012) and the effect of each parameter was explained therein. In this part of the paper first 30-cells problem from the set of benchmark problems is selected for carrying out initial experiments to find the appropriate level of each parameter of the MBO algorithm. Then the selected level of each factor is applied for final experiments on all benchmark problems.

4.1. Taguchi experimental design for parameter setting

Obviously, the behavior of parameters is important to obtain better results in meta-heuristic algorithms. When the number of parameters and their number of levels increase, it is difficult to study the effect of all possible combination of parameter levels. In this study the levels of independent factors of the MMBO, MBO and SA algorithms are determined based on Table 2. The remaining factors of the MMBO and MBO algorithms including x (the number of neighboring solutions shared with the next solution) and K (number of iterations) are not considered as independent parameters by the following relations, $x = k$ and $K = knm$ meaning that in the MBO algorithm all neighbors of any solution is moved to the next solution. In this way the number of regenerated neighbors in MMBO and the number of transferred neighbors in MBO will be equal.

Full factorial designs are extensively applied to set the parameter levels of meta-heuristic methods (Al-Aomar & Al-Okaily, 2006; Kim, Kim, & Jang, 2003; Tsai, Ho, Liu, & Chou, 2007). These methods help the experimenter in finding the best level of each parameter by using a reduced number of experiments covering all predetermined levels of parameters.

The Taguchi experimental design is a method introduced by Taguchi as a robust method to set the level of parameters (Taguchi, 1986). This method is widely applied to optimization problems for setting the level of parameters (see Hsu, 2013; Mahmoodi-Rad, Molla-Alizadeh-Zavardehi, Dehghan, Sanei, & Niroomand, 2013; Molla-Alizadeh-Zavardehi, Hajiaghaei-Keshteli, & Tavakkoli-Moghaddam, 2011; Naderi, Zandieh, Ghoshe Balagh, & Roshanaei, 2009). Taguchi design method clusters the parameters into two groups of controllable and uncontrollable parameters. The level of controllable parameter is fixed during the process, while the level of uncontrollable parameter changes during the process. This method tries to minimize the impact of uncontrollable parameters and to find the best level of effective controllable parameters. In order to achieve this purpose the Taguchi method uses orthogonal arrays to design the experiments that reduce the number of total experiments to be run. Finally, the output of each experiment is transformed to a signal-to-noise ratio (S/N ratio) which calculates the amount of variation of the response parameter. The objective of Taguchi method is to minimize the amount of S/N ratio. As mentioned in Naderi et al. (2009) the S/N ratio of minimization type problems is obtained by:

$$\mathrm{S/N} = -10\log_{10}\,(\text{objective function})^2 \tag{3}$$

In order to find the appropriate orthogonal array for any problem, the total degree of freedom of its parameters is needed. For the proposed MMBO algorithm, there are four parameters of three levels, so two degrees of freedom is considered for each level and parameter. The only parameter with two levels also needs one degree of freedom. By considering one degree of freedom for the total mean there will be totally ten degrees of freedom ($4 \times 2 + 1 + 1 = 10$). Therefore the appropriate orthogonal array for the MMBO algorithm consists of at least 10 experiments. The orthogonal array considered for experiments of the MMBO algorithm is L18 (3^5). In this array there are five parameters with three levels for each, but in our case the fifth parameter has only two levels, therefore the rows related to the third level of the fifth parameter are filled by its first and second levels arbitrarily. The orthogonal array of Taguchi design for the MMBO algorithm is shown in Table 3.

Similarly the degree of freedom of the MBO and SA algorithms is calculated and their appropriate orthogonal arrays are designed and shown in Table 4.

The benchmark problem with the highest size (the 30-cell problem) was selected to perform the experiments designed by the Taguchi method. To obtain more reliable results, each experiment was run three times and the average was used in the calculations. The best level of parameters introduced by Taguchi method is shown in Table 5. The effect of factors can be seen in Figs. 13

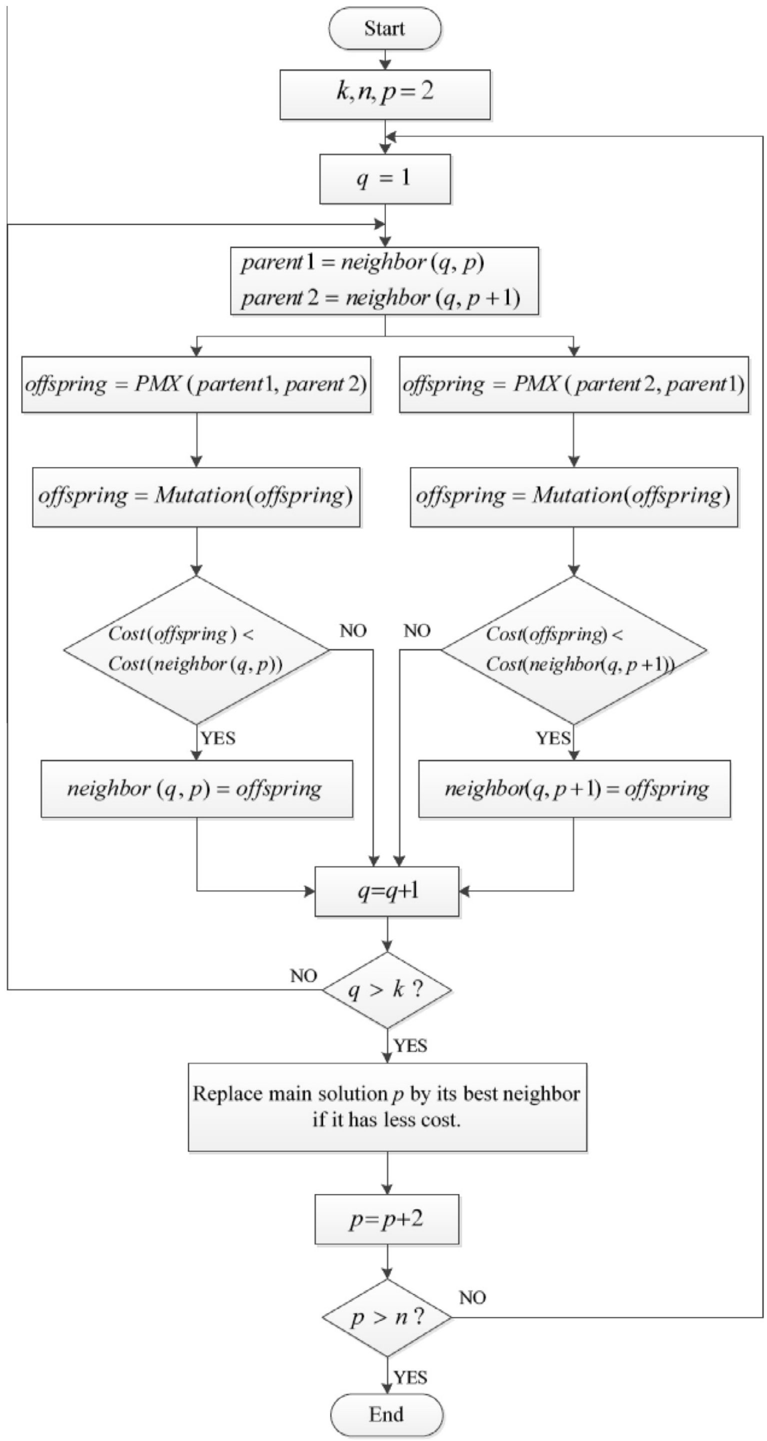

Fig. 11. Flowchart of crossover type 2.

and 14 as well. These are best levels of parameters obtained for the 30-cell problem which is the largest benchmark problem. Therefore these levels of parameters are also used in other benchmark problems as their sizes are smaller than 30-cell problem, thus, these levels of parameters seem to be sufficient.

4.2. Final experiments on the benchmark problems

The above-obtained levels of parameters were applied to the MMBO, MBO and SA algorithms of all benchmark problems of literature of the closed loop layout problem. For all algorithms the same run time of $2000 + 0.5N^3$ milliseconds for each closed loop size was considered (where, N is the number of cells of each benchmark). The constant value of 2000 ms is considered to give possibility to the small size benchmarks to have at least two seconds run time. Based on the size of the benchmark problems, the run times varied from several minutes to one hour. Each problem was run five times and the results are summarized in Table 6.

As can be seen in Table 6, all the meta-heuristic methods are able to find the optimal solution of the 4-cell problem which are proven by Xpress but their performance is not as good as Xpress for the 6 and 8-cell problems. The results also indicate that the

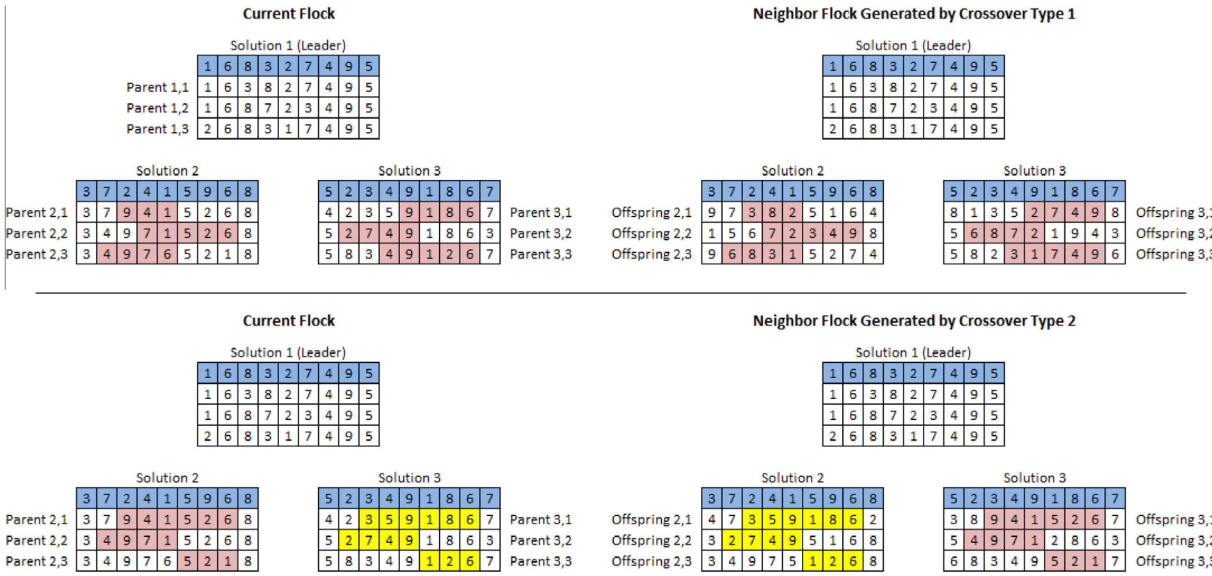

Fig. 12. Representation of crossover operations on a given flock.

Table 2
Initial levels of parameters for proposed algorithms.

MMBO		MBO		SA	
Parameter	Levels	Parameter	Levels	Parameter	Levels
(A) Number of solutions (n)	21, 39, 51	(A) Number of solutions (n)	21, 39, 51	(A) Initial temperature	100, 200, 350
(B) Number of neighbors (k)	20, 30, 45	(B) Number of neighbors (k)	20, 30, 45	(B) Cooling ratio	0.8, 0.9, 0.99
(C) Number of tours (m)	1, 2, 3	(C) Number of tours (m)	1, 2, 3	(C) Number of replications	50, 130, 200
(D) Mutation probability	0.3, 0.65, 1				
(E) Crossover type	Type 1, Type 2				
Possible number of combinations	$3^4 \times 2 = 162$	Possible number of combinations	$3^3 = 27$	Possible number of combinations	$3^3 = 27$

Table 3
Modified L18 orthogonal array of Taguchi design for MMBO algorithm.

Experiment	Number of solutions	Number of neighbors	Number of tours	Mutation probability	Crossover type
1	21	20	1	0.3	Type 1
2	21	30	3	1	Type 1
3	21	45	2	1	Type 2
4	21	30	2	0.3	Type 1
5	21	45	1	0.65	Type 2
6	21	20	3	0.65	Type 2
7	39	30	2	0.65	Type 2
8	39	45	1	0.3	Type 2
9	39	20	3	0.3	Type 1
10	39	45	3	0.65	Type 1
11	39	20	2	1	Type 1
12	39	30	1	1	Type 2
13	51	45	3	1	Type 1
14	51	20	2	0.65	Type 2
15	51	30	1	0.65	Type 1
16	51	20	1	1	Type 2
17	51	30	3	0.3	Type 2
18	51	45	2	0.3	Type 1

meta-heuristics showed better performance in the case of other benchmark problems comparing to Xpress software's results. In the case of small size problems (4 to 14-cell problems) the output of all meta-heuristics is approximately the same while both MBO and MMBO methods give better results than the SA algorithm in the case of problems with larger than 14-cell size except the 18-cell problem. It is seen that in all problems that have a cell size larger than 14, the performance of the MMBO algorithm is better than the MBO algorithm.

4.3. Number of explored solutions

In most of researches, meta-heuristic methods are compared from the "running time" point of view. Another criterion that can be used for comparing meta-heuristic methods is that of the "number of explored solutions" that is equivalent to the number of times that the algorithm tries to make improvement in the objective function value. Although running time is an important property in any meta-heuristic method, it is strongly dependent on the structure of the method. A meta-heuristic may have better performance even if it is able find a better solution in a more limited number of explored solutions.

In the MBO algorithm any neighbor, after being generated, is used once more to improve another leading solution so one more chance is given to any generated neighbor to become a leading solution, hence, it is counted twice. Thus, the method for calculation of explored solutions of the MMBO and MBO algorithms are the same. Actually for the case of the MBO algorithm the number of explored solution is the number of times that the MBO algorithm tries to make improvement in the objective function value. Based on the final experiments on the benchmark problems, the number of explored solutions in any size of the closed loop material handling path is shown in Table 7. Also some characteristics of the MMBO, MBO and SA algorithms are compared in Table 8. As the table shows, both the MMBO and MBO algorithms start with a multiple initial solution that considers more directions of solution area comparing to the SA algorithm that uses a single initial solution. The multiple initial solution also causes multiple search directions that increases the probability of finding better solutions. On

Table 4
L9 orthogonal arrays of Taguchi design for MBO and SA algorithms.

Experiment	MBO			SA		
	Number of solutions	Number of neighbors	Number of tours	Initial temperature	Cooling ratio	Number of replications
1	21	20	1	100	0.8	50
2	21	30	2	100	0.9	130
3	21	45	3	100	0.99	200
4	39	20	2	200	0.8	130
5	39	30	3	200	0.9	200
6	39	45	1	200	0.99	50
7	51	20	3	350	0.8	200
8	51	30	1	350	0.9	50
9	51	45	2	350	0.99	130

Table 5
Best level of parameters obtained by Taguchi method for MMBO, MBO and SA algorithms.

MMBO		MBO		SA	
Parameter	Level obtained by Taguchi method	Parameter	Level obtained by Taguchi method	Parameter	Level obtained by Taguchi method
Number of solutions (n)	51	Number of solutions (n)	51	Initial temperature	350
Number of neighbors (k)	45	Number of neighbors (k)	45	Cooling ratio	0.9
Number of tours (m)	2	Number of Tours (m)	3	Number of replications	200
Mutation probability	0.3				
Crossover type	Type 2				

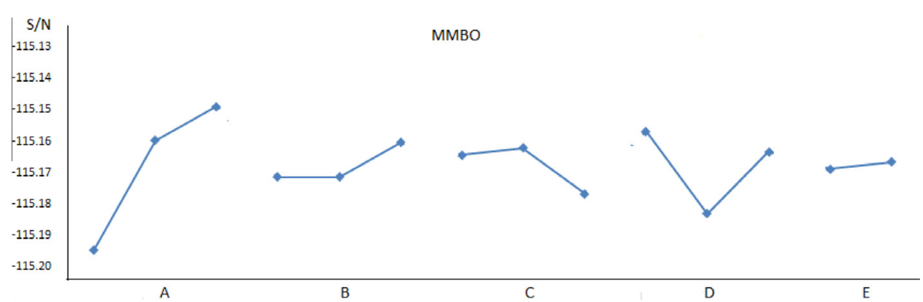

Fig. 13. The effect of different levels of MMBO algorithm based on S/N ratio.

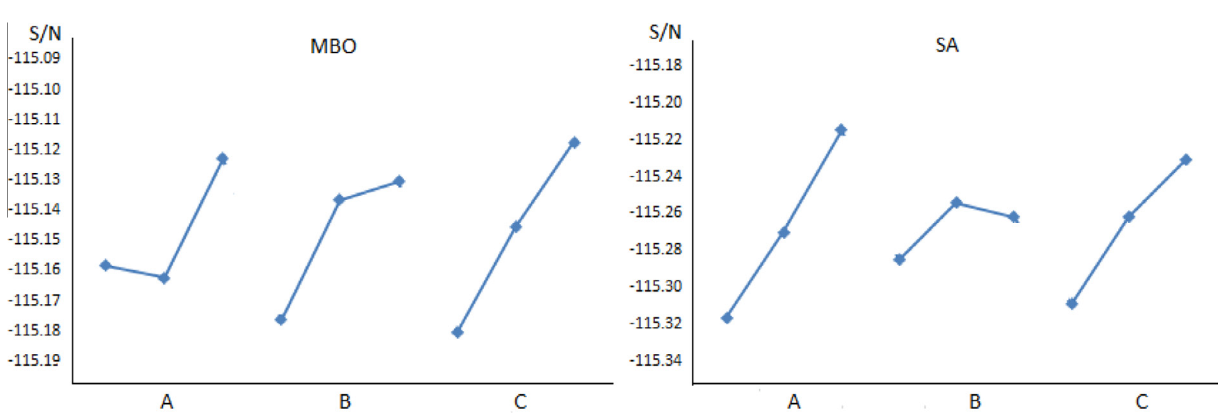

Fig. 14. The effect of different levels of MBO and SA algorithms based on S/N ratio.

the other hand, the MMBO algorithm can be able to explore better solutions than the MBO and SA algorithms as it combines two solutions from different directions to construct a new neighbor that depended on its objective function value may be considered as a new search direction (main solution or even leader). Also the MMBO algorithm has opportunity to perform better than the MBO and SA methods as it uses two different neighborhood search operators consecutively.

The CPU run time to explore the number of solutions mentioned in Table 7 for the SA, MBO and MMBO algorithms for the largest benchmark (30-cell problem) is about 12.45, 14.12 and 15.05 s, respectively, which are approximately in the same range. The

Table 6
The final results obtained from proposed algorithms for closed loop layout problem.

Problem No.	Size	Xpress result[a]	Minimum obtained solution			Average of runs		
			SA	MBO	MMBO	SA	MBO	MMBO
1	4	**547.5**[b]	**547.5**	**547.5**	**547.5**	**547.5**	**547.5**	**547.5**
2	6	**1601.5**[b]	1659.0	1659.0	1659.0	1659.0	1659.0	1659.0
3	8	**5943.5**[b]	6354.0	6354.0	6354.0	6376.7	6354.0	6354.0
4	10	13,417.0	**12747.0**	**12747.0**	**12747.0**	**12747.0**	**12747.0**	**12747.0**
5	12	37,281.5	**34333.0**	**34333.0**	**34333.0**	34650.2	**34333.0**	**34333.0**
6	14	45,402.5	**44407.0**	**44407.0**	**44407.0**	44613.2	**44407.0**	**44407.0**
7	16	69,337.0	57903.0	57913.0	**57675.0**	58126.0	57980.4	**57828.2**
8	18	88,807.5	**75933.0**	**75933.0**	**75933.0**	76069.8	76101.0	**76044.8**
9	20	–	126820.0	126640.0	**126530.0**	127972.0	126726.0	**126648.0**
10	25	–	311250.0	307938.0	**307152.0**	313440.0	309320.2	**308520.8**
11	30	–	571190.0	569220.0	**569020.0**	576310.0	570406.0	**570208.0**

The best output among the algorithms for each problem in the categories of minimum solution and average of runs is bolded.
[a] Closed loop layout cost obtained by Xpress (Niroomand & Vizvári, 2013).
[b] Optimality was proved by Xpress.

Table 7
Number of generated solutions in each closed loop size by proposed algorithms.

Method	Formula for number of explored solutions	Number of explored solutions
SA	[(no.of temperatures)(no of replications)]	8200
MBO	$[(nk) + ((n-1)k)]m$	13,635
MMBO	$[(nk) + ((n-1)k)]m$	9090

Table 8
Comparison of some characteristics of MMBO, MBO and SA algorithms.

Characteristic	Algorithm		
	MMBO	MBO	SA
Initial solution	Multiple	Multiple	Single
Search directions	Multiple	Multiple	Single
Neighbors obtained from	Two different search directions	Single search direction	Single search direction
Number of operators	2	1	1

differences may arise from the coding structure of the neighborhood operators of the algorithms.

5. Discussion and concluding remarks

Based on the limitation of the floor and material handling path, different layout patterns may be considered for FMSs. In the cases that there is not enough space for straight line layout (single row), open-field and closed loop layouts may be useful. The closed loop layout pattern is even more useful where Automated Guided Vehicles (AGVs) are used to move the material.

An exact MILP model of closed loop layout from the literature was tackled in this study. The NP-hardness of the model is the main limitation of previous study (Niroomand & Vizvári, 2013) and this study which make the model difficult to give a good feasible solution especially for the case of large size problems.

A recently introduced meta-heuristic algorithm (MBO) was modified (MMBO) in this paper to solve the closed loop layout model. The proposed MMBO algorithm uses some natural and logical rules to solve the model. The algorithm mixes the logics of geometry and human thinking to arrange the cells around a rectangular closed loop without any overlap by use of logical functions in computer programming. On the other hand, the natural behavior of birds when migrating, is applied in a computer program to construct the proposed algorithm.

To test the performance of the proposed MMBO algorithm, the standard form of the MBO and SA algorithms were also simulated and used in the computational experiments part. The results obtained by Xpress solver from the literature also was used in the comparisons. The computational experiments proved that in the first eight benchmarks out of all eleven benchmarks of the study (as the Xpress results from the literature is available only for these eight benchmarks) all the meta-heuristic algorithms are efficient in finding improved solutions as compared to Xpress solver in five out of the eight benchmark problems. For the remaining three benchmarks of the first eight benchmarks, in only one case the algorithms obtain the optimal solution of Xpress solver. Finally, considering all eleven benchmarks, the proposed MMBO algorithm has better performance in comparison to the MBO and SA algorithms in most of the benchmarks, especially large size benchmarks.

Nowadays, researchers try to combine standard form of two or more meta-heuristic algorithms in order to construct a hybrid meta-heuristic algorithm to obtain a better performance in problem solving. As a future study, MBO algorithm can be hybridized by combining standard MBO and other meta-heuristics e.g. simulated annealing, genetic algorithm, tabu search, etc. Modified version of some other meta-heuristics which are common in computational computer programming (e.g. artificial bee colony algorithm, sheep flock heredity algorithm, etc.) also may be applied on the exact closed loop layout model. On the other hand, heuristic algorithms e.g. Bender's decomposition, Lagrangian relaxation, etc. which are bridges between artificial intelligence and exact optimization methods may also be used to solve the exact closed loop layout model.

Acknowledgements

The authors are indebted to the editors and the referees of the journal for their helpful and constructive comments that improved the quality of the paper.

References

Aiello, G., Enea, M., & Galante, G. (2006). A multi-objective approach to facility layout problem by genetic search algorithm and Electre method. *Robotics and Computer-Integrated Manufacturing, 22*, 447–455.
Al-Aomar, R., & Al-Okaily, A. (2006). A GA-based parameter design for single machine turning process with high-volume production. *Computers and Industrial Engineering, 50*, 317–337.
Amaral, A. (2009). A new lower bound for the single row facility layout problem. *Discrete Applied Mathematics, 157*, 183–190.
Anjos, M. F., Kennings, A., & Vannelli, A. (2005). A semidefinite optimization approach for the single-row layout problem with unequal dimensions. *Discrete Optimization, 2*, 113–122.
Badgerow, J. P., & Hainsworth, F. R. (1981). Energy savings through formation flight? A re-examination of the vee formation. *Journal of Theoretical Biology, 93*, 41–52.

Bénabès, J., Poirson, E., & Bennis, F. (2013). Integrated and interactive method for solving layout optimization problems. *Expert Systems with Applications, 40*, 5796–5803.
Brintup, A. M., Ramsden, J., & Tiwari, A. (2007). An interactive genetic algorithm-based framework for handling qualitative criteria in design optimization. *Computers in Industry, 58*, 279–291.
Chae, J., & Peters, B. A. (2006). A simulated annealing algorithm based on a closed loop layout for facility layout design in flexible manufacturing systems. *International Journal of Production Research, 44*, 2561–2572.
Cong, X. C., Yang, S. L., Cao, S. Q., Chen, Z. L., Dai, M. X., & Peng, S. T. (2012). Effect of aggregate stockpile configuration and layout on dust emissions in an open yard. *Applied Mathematical Modelling, 36*(11), 5482–5491.
Das, S. K. (1993). A facility layout method for flexible manufacturing systems. *International Journal of Production Research, 31*(2), 279–297.
Duman, A., Uysal, M., & Alkaya, A. F. (2012). Migrating birds optimization: A new meta-heuristic approach and its performance on quadratic assignment problem. *Information Sciences, 217*, 65–77.
Eiben, A., & Smith, J. (2003). *Introduction to evolutionary computing*. Springer-Verlag.
Ficko, M., Brezocnik, M., & Balic, J. (2004). Designing the layout of single- and multiple-rows flexible manufacturing system by genetic algorithms. *Journal of Material Processing Technology*, 150–158.
Garcia-Hernandeza, L., Pierrevalb, H., Salas-Moreraa, L., & Arauzo-Azofraa, A. (2013). Handling qualitative aspects in unequal area facility layout problem: An interactive genetic algorithm. *Applied Soft Computing, 13*, 1718–1727.
Garey, M. R., & Johnson, D. S. (1979). *Computers and intractability: A guide to the theory of NP-completeness*. New York: WH Freeman.
Hadi-Vencheh, A., & Mohamadghasemi, A. (2013). An integrated AHP–NLP methodology for facility layout design. *Journal of Manufacturing Systems, 32*(1), 40–45.
Hsu, C. (2013). Application of SVR, Taguchi loss function, and the artificial bee colony algorithm to resolve multiresponse parameter design problems: A case study on optimizing the design of a TIR lens. *Neural Computing and Applications*. http://dx.doi.org/10.1007/s00521-013-1357-3.
Hummel, D., & Beukenberg, M. (1989). Aerodynamsiche Interferenseffekte beim formationsflug von vögeln. *Journal of Ornithology, 130*, 15–24.
Islier, A. A. (1998). A genetic algorithm approach for multiple criteria facility layout design. *International Journal of Production Research, 36*(6), 1549–1569.
Javadi, B., Jolai, F., Slomp, J., Rabbani, M., & Tavakkoli-Moghaddam, R. (2013). An integrated approach for the cell formation and layout design in cellular manufacturing systems. *International Journal of Production Research*. http://dx.doi.org/10.1080/00207543.2013.791755.
Kaveh, M., Majazi Dalfard, V., & Amiri, S. (2013). A new intelligent algorithm for dynamic facility layout problem in state of fuzzy constraints. *Neural Computing and Applications*. http://dx.doi.org/10.1007/s00521-013-1339-5.
Kim, J.-G., & Kim, Y.-D. (2000). Layout planning for facilities with fixed shapes and input and output points. *International Journal of Production Research, 38*, 4635–4653.
Kim, S. J., Kim, K. S., & Jang, H. (2003). Optimization of manufacturing parameters for a brake lining using Taguchi method. *Journal of Material Processing Technology, 136*, 202–208.
Kirkpatrick, S., Gelatt, C. D., Jr., & Vecchi, M. P. (1983). Optimization by simulated annealing. *Science, 220*, 671–680.
Kothari, R., & Ghosh, D. (2013). Tabu search for the single row facility layout problem using exhaustive 2-opt and insertion neighborhoods. *European Journal of Operational Research, 224*, 93–100.
Lissaman, P. B. S., & Shollenberger, C. A. (1970). Formation flight of birds. *Science, 168*, 1003–1005.
Mahmoodi-Rad, A., Molla-Alizadeh-Zavardehi, S., Dehghan, R., Sanei, M., & Niroomand, S. (2013). Genetic and differential evolution algorithms for the allocation of customers to potential distribution centers in a fuzzy environment. *International Journal Advanced Manufacturing Technology*. http://dx.doi.org/10.1007/s00170-013-5383-1.
McKendall, A. R., & Shang, J. (2006). Hybrid ant systems for the dynamic facility layout problem. *Computers & Operations Research, 33*(3), 790–803.
McKendall, A. R., Shang, J., & Kuppusamy, S. (2006). Simulated annealing heuristics for the dynamic facility layout problem. *Computers & Operations Research, 33*(8), 2431–2444.
Molla-Alizadeh-Zavardehi, S., Hajiaghaei-Keshteli, M., & Tavakkoli-Moghaddam, R. (2011). Solving a capacitated fixed-charge transportation problem by artificial immune and genetic algorithms with a Prüfer number representation. *Expert Systems with Applications, 38*, 10462–10474.
Naderi, B., & Azab, A. (2014). Modeling and heuristics for scheduling of distributed job shops. *Expert Systems with Applications, 41*(17), 7754–7763.
Naderi, B., Zandieh, M., Ghoshe Balagh, A. K., & Roshanaei, V. (2009). An improved simulated annealing for hybrid flow shops with sequence-dependent setup and transportation times to minimize total completion time and total tardiness. *Expert Systems with Applications, 36*, 9625–9633.
Niroomand, S., Takacs, S., & Vizvari, B. (2011). To lay out or not to lay out? *Annals of Operations Research, 191*, 183–193.
Niroomand, S., & Vizvári, B. (2013). A mixed integer linear programming formulation of closed loop layout with exact distances. *Journal of Industrial and Production Engineering, 30*(3), 190–201.
Niroomand, S., & Vizvári, B. (2014). Exact mathematical formulations and metaheuristic algorithms for production cost minimization: A case study of the cable industry. *International Transactions in Operational Research*. http://dx.doi.org/10.1111/itor.12096.
Ou-Yang, C., & Utamima, A. (2013). Hybrid estimation of distribution algorithm for solving single row facility layout problem. *Computers & Industrial Engineering, 66*(1), 95–103.
Pierreval, H., Caux, C., Paris, J. L., & Viguier, F. (2003). Evolutionary approaches to the design and organization of manufacturing systems. *Computers & Industrial Engineering, 44*(3), 339–364.
Rajasekharan, M., Peters, B. A., & Yang, T. (1998). A genetic algorithm for facility layout design in flexible manufacturing systems. *International Journal of Production Research, 36*, 95–110.
Rayner, J. M. V. (1979). A new approach to animal flight mechanics. *Journal of Experimental Biology, 80*, 17–54.
Sahin, R., Ertogral, K., & Turkbey, O. (2010). A simulated annealing heuristic for the dynamic layout problem with budget constraint. *Computers & Industrial Engineering, 59*, 308–313.
Solimanpur, M., Vrat, P., & Shankar, R. (2005). An ant algorithm for the single row layout problem in flexible manufacturing systems. *Computers & Operations Research, 32*(3), 583–598.
Taguchi, G., (1986) Introduction to quality engineering. White Plains: Asian Productivity Organization/UNIPUB.
Tavakkoli-Moghaddam, R., & Panahi, H., (2007) Solving anew mathematical model of a closed-loop layout problem with unequal-sized facilities by a genetic algorithm. In *Proceedings of the 2007 IEEE IEEM* (pp. 327–331). Singapore.
Tompkins, J. A., White, J. A., Bozer, Y. A., Frazelle, E. H., Tanchoco, J. M. A., & Trevino, J. (1996). *Facilities planning* (2nd ed.). New York: Wiley.
Tsai, J. T., Ho, W. H., Liu, T. K., & Chou, J. H. (2007). Improved immune algorithm for global numerical optimization and job shop scheduling problems. *Applied Mathematics and Computation, 194*, 406–424.
Wang, M. J., Hu, M. H., & Ku, M. H. (2005). A solution to the unequal area facilities layout problem by genetic algorithm. *Computers in Industry, 56*(2), 207–220.