International Journal of Production Research
Publication details, including instructions for authors and subscription information:
http://www.tandfonline.com/loi/tprs20

A multi-objective particle swarm optimisation algorithm for unequal sized dynamic facility layout problem with pickup/drop-off locations

Fariborz Jolai [a], Reza Tavakkoli-Moghaddam [a] & Mohammad Taghipour [a]
[a] Department of Industrial Engineering, College of Engineering, University of Tehran, Tehran, Iran

Available online: 26 Oct 2011

International Journal of Production Research
2011, 1–15, iFirst

Taylor & Francis
Taylor & Francis Group

A multi-objective particle swarm optimisation algorithm for unequal sized dynamic facility layout problem with pickup/drop-off locations

Fariborz Jolai*, Reza Tavakkoli-Moghaddam and Mohammad Taghipour

Department of Industrial Engineering, College of Engineering, University of Tehran, Tehran, Iran

(*Received 27 November 2010; final version received 29 July 2011*)

This paper deals with a multi-objective unequal sized dynamic facility layout problem (DFLP) with pickup/drop-off locations. First, a mathematical model to obtain optimal solutions for small size instances of the problem is developed. Then, a multi-objective particle swarm optimisation (MOPSO) algorithm is implemented to find near optimal solutions. Two new heuristics to prevent overlapping of the departments and to reduce 'unused gaps' between the departments are introduced. The performance of the MOPSO is examined using some sets of available test problems in the literature and various random test problems in small, medium, and large sizes. The percentage of improvements on the initial solutions is calculated for small, medium and large size instances. Also, the generation metric and the space metric for non-dominated solutions are examined. These experiments show the good performance of the developed MOPSO and sensitivity analysis show the robustness of the obtained solutions.

Keywords: dynamic facility layout problem; pickup/drop-off locations; MOPSO; department overlapping

1. Introduction

Facility layout has an important role in increasing competitive capability of manufacturing organisations. Most of the relevant researches in the literature focused on material handling costs as an objective function and neglected other important costs conflicting with this one. Also, focusing on dynamic layout is now more and more important because of rapid science developments and product changes.

In this paper we consider a multi-objective dynamic facility layout problem (DFLP) with unequal fixed size departments and pick up/drop off locations. Our objectives are minimising material handling and rearrangement costs and maximising total adjacency and distance requests.

The static multi-objective facility layout problem (FLP) has been considered by many researchers. Aiello *et al.* (2006) proposed a multi-objective model that considered material handling cost, satisfaction of weighted adjacency, satisfaction of distance requests and satisfaction of aspect ratio requests simultaneously. They employed a constrained multi-objective genetic algorithm to approximate the Pareto-optimal frontier. Khilwani *et al.* (2008) modelled a multi-person, multi-criteria and multi-preference scenario for FLP. They used psycho-clonal evolutionary algorithm with weighted averaging operator for the aggregation of the decision maker's information. Singh and Singh (2010) proposed a multi-objective model considering work flow, closeness rating, material handling time and hazardous movement. They used a heuristic approach that considered various weighting methods for making the decision process completely designer independent. Chen and Sha (2005) proposed five heuristics to solve multi-objective FLP using a multi-pass halving and doubling procedure. Zhou *et al.* (2006) formulated a multiple-objective nonlinear mixed-integer model for FLP with aisles, minimising material handling costs and maximising adjacent requirement between resources. They developed a multi-objective genetic algorithm (MOGA) with local search to solve the problem. Şahin and Türkbey (2009) considered a multi-objective FLP, minimising total material handling cost and maximising total closeness rating. They proposed a simulated annealing (SA) algorithm to find the Pareto optimal solution.

Recently, the single objective dynamic facility layout problem (DFLP) with unequal size departments has attracted some researchers. Meller *et al.* (2007) proposed a new formulation for the facility layout problem with unequal-area based on the sequence-pair representation. Chan and Malmborg (2010) described a procedure that uses Monte Carlo simulation with minimum material handling cost objective across stochastic demand scenarios for

*Corresponding author. Email: fjolai@ut.ac.ir

ISSN 0020–7543 print/ISSN 1366–588X online

http://dx.doi.org/10.1080/00207543.2011.613863
http://www.tandfonline.com

unequal sized facility layout problem. McKendall and Hakobyan (2010) presented a dynamic model with unequal-area departments where the layout for each period is generated on the continuous plant floor. They developed a heuristic boundary search technique, which places departments along the boundaries of already placed departments. They also improved the obtained solutions using a tabu search heuristic. Paul *et al.* (2006) proposed a particle swarm optimisation (PSO) algorithm with a heuristic method for creating a relationship between the facilities and passages for solving unequal area facility layout problems with inner walls and passages.

The above review shows many researchers proposed meta-heuristic approaches to solve various types of FLP. In this paragraph we briefly review the rest of the relevant applications of the meta-heuristic methods. El-Baz (2004), Ramkumar *et al.* (2009) and Nourelfath *et al.* (2007) presented GA, modified iterated fast local search algorithm (IFLS) and a hybrid ant colony optimisation (ACO) algorithm to solve static FLP, respectively. Ye and Zhou (2006) and Logendran and Kriausakul (2006) developed tabu search based methods for the same problem. Ulutaş and Işlier (2007) proposed a clonal selection algorithm (part of an artificial immune system (AIS)) for FLPs, minimising material handling costs. The algorithms based on the SA approach are proposed by Chwif *et al.* (1998) and Deb and Bhattacharyya (2005) for FLP considering pick-up/drop-off points. McKendall and Shang (2006) proposed a hybrid ant system for solving DFLP. The ant colony optimisation (ACO) was also proposed by Baykasoglu *et al.* (2006) for DFLP with budget constraints. Sahin and Turkbey (2009) presented a hybrid algorithm based on the simulated annealing (SA) method and tabu search (TS) for the DFLP. They also compared their method with SA and TS, separately.

In this paper, we develop a non-linear mixed integer mathematical model and a multi-objective PSO (MOPSO) algorithm to solve unequal size DFLP with pick-up/drop-off points. To implement MOPSO, we present a heuristic approach to prevent the department overlapping and another heuristic to reduce the 'unused gaps' between departments.

The rest of the paper is as follows: Section 2 presents DFLP mathematical model. In Section 3, MOPSO algorithm is described. Computational and sensitivity analysis are presented in Section 4 and Section 5, respectively. Finally Section 6 concludes the paper.

2. Problem Formulation

First, we present some notations to describe our mathematical model.

Indices:

i, j	Indices for departments where $i, j = 1, \ldots, N$, $N =$ number of departments.
t	Index for periods where $t = 1, \ldots, T$, $T =$ number of periods.

Parameters:

C_{tij}	Cost of material handling (a unit distance) between department i and department j in period t.
RC_{ti}	Rearrangement cost of moving or reorienting department i in the beginning of period t.
AV_{tij}	Adjacency value (0 to 5) between department i and department j in the beginning of period t.
DR_{tij}	Distance Rating between department i and department j in the beginning of period t.
Sh_{ti}	Shorter edge length of department i in period t.
Lg_{ti}	Longer edge length of department i in period t.
Ort_{it}	Forced orientation of department i in period t (0: Free, 1: Horizontal, -1: Vertical).
Lgh	Length of factory floor.
Wth	Width of factory floor.

Variables:

x_{ti}, y_{ti}	Centroid coordinate of department i in period t
l_{ti}, w_{ti}	Length and width of department i in period t
x'_{tij}, y'_{tij}	Horizontal and vertical distance between centroid of department i and department j in period t
o_{ti}	Orientation of department i in period t (0: Vertical, 1:Horizontal)
r_{ti}	$\begin{cases} 1 \text{ if department } i \text{ is rearranged at the beginning of period } t \\ 0 \text{ otherwise} \end{cases}$
af_{tij}	Adjacency factor between department i and department j in the beginning of period t

dp_{ti} Distance between pick-up/drop-off point and centroid coordinate of department i in period t

p'_{ti} $\begin{cases} 1 \text{ if pick} - \text{up/drop} - \text{off point is in longer edge of department } i \text{ in period } t \\ 0 \text{ otherwise} \end{cases}$

p'_{ti} $\begin{cases} 1 \text{ if pick} - \text{up/drop} - \text{off point is in north} - \text{west edges of department } i \text{ in period} \\ 0 \text{ if pick} - \text{up/drop} - \text{off point is in south} - \text{east edges of department } i \text{ in period } t \end{cases}$

sgn_{ti} A sign variable (see table 1)

Objectives:

$$\text{(1)} \quad \text{Minimise}: \sum_{t=1}^{T}\sum_{i=1}^{N}\sum_{j=1}^{N} C_{tij}\Big((x'_{tij} + y'_{tij}) + (\text{sgn}_{ti} * dp_{ti} + \text{sgn}_{tj} * dp_{tj})\Big) \quad i \neq j \tag{1}$$

$$\text{(2)} \quad \text{Minimise}: \sum_{t=2}^{T}\sum_{i=1}^{N} RC_{ti} * r_{ti} \tag{2}$$

$$\text{(3)} \quad \text{Maximise}: \sum_{t=1}^{T}\sum_{i=1}^{N}\sum_{j=1}^{N} AV_{tij} * af_{tij} \quad i \neq j \tag{3}$$

$$\text{(4)} \quad \text{Maximise}: \sum_{t=1}^{T}\sum_{i=1}^{N}\sum_{j=1}^{N} DR_{tij}(x'_{tij} + y'_{tij}) \quad i \neq j \tag{4}$$

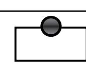
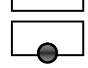
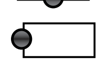
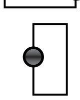
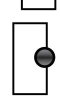
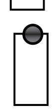
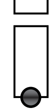
Constraints:

$$x_{ti} - 0.5l_{ti} \geq 0 \quad \forall t, i \tag{5}$$

$$x_{ti} + 0.5l_{ti} \leq Lgh \quad \forall t, i \tag{6}$$

$$x_{ti} - 0.5w_{ti} \geq 0 \quad \forall t, i \tag{7}$$

Table 1. The values of dp_{ti} and sgn_{ti} variables.

Orientation	Longer/Shorter	North-west/South-east	Shape	dp_{ti}	Relative location between dep$i$&j	sgn_{ti}
$o_{ti}=1$	$p'_{ti}=1$	$p''_{ti}=1$		l_{ti}	If $y_{ti} < y_{tj}$ If $y_{ti} > y_{tj}$	$-$ $+$
		$p''_{ti}=0$		$-l_{ti}$	If $y_{ti} < y_{tj}$ If $y_{ti} > y_{tj}$	$-$ $+$
	$p'_{ti}=0$	$p''_{ti}=1$		$-w_{ti}$	If $x_{ti} < x_{tj}$ If $x_{ti} > x_{tj}$	$-$ $+$
		$p''_{ti}=0$		w_{ti}	If $x_{ti} < x_{tj}$ If $x_{ti} > x_{tj}$	$-$ $+$
$o_{ti}=0$	$p'_{ti}=1$	$p''_{ti}=1$		$-w_{ti}$	If $x_{ti} < x_{tj}$ If $x_{ti} > x_{tj}$	$-$ $+$
		$p''_{ti}=0$		w_{ti}	If $x_{ti} < x_{tj}$ If $x_{ti} > x_{tj}$	$-$ $+$
	$p'_{ti}=0$	$p''_{ti}=1$		l_{ti}	If $y_{ti} < y_{tj}$ If $y_{ti} > y_{tj}$	$-$ $+$
		$p''_{ti}=0$		$-l_{ti}$	If $y_{ti} < y_{tj}$ If $y_{ti} > y_{tj}$	$-$ $+$

$$x_{ti} + 0.5w_{ti} \le Wth \quad \forall t, i \tag{8}$$

$$x'_{tij} = \left| x_{ti} - x_{tj} \right| \quad \forall t, i, j \quad i \ne j \tag{9}$$

$$y'_{tij} = \left| y_{ti} - y_{tj} \right| \quad \forall t, i, j \quad i \ne j \tag{10}$$

$$l_{ti} = Lg_{ti} o_{ti} + Sh_{ti}(1 - o_{ti}) \quad \forall t, i \tag{11}$$

$$w_{ti} = Lg_{ti}(1 - o_{ti}) + Sh_{ti} o_{ti} \quad \forall t, i \tag{12}$$

$$o_{ti} = 1 \text{ If } Ort_{ti} = 1 \quad \forall t, i \tag{13}$$

$$o_{ti} = 0 \text{ If } Ort_{ti} = -1 \quad \forall t, i \tag{14}$$

$$r_{ti} = \begin{cases} 1 \text{ If } x_{t-1,i} \ne x_{ti} \text{ or } y_{t-1,i} \ne y_{ti} \text{ or } o_{t-1,i} \ne o_{ti} & \forall t, i \\ 0 \quad \text{Otherwise} \end{cases} \tag{15}$$

$$x_{ij}, y_{ij}, x'_{tij}, y'_{tij}, l_{ti}, w_{ti} \ge 0 \quad \forall t, i, j \tag{16}$$

$$o_{ti}, r_{ti}, p'_{ti}, p''_{ti} \in \{0, 1\} \quad \forall t, i, j \tag{17}$$

The objective (1) minimises the total material handling costs. Variable dp_{ti} represents the distance between the centroid and the pickup/drop-off location of department i. The value of this variable depends on o_{ti}, p'_{ti} and p''_{ti} and can be derived from Table 1. According to the relative location of the departments i and j, dp_{ti} value is either positive or negative, that is indicated by 'sign' variable in Table 1. Objective (2) minimises the total rearrangement costs. Objective (3) maximises the total adjacency value. As Table 2 shows the adjacency factor value, af_{tij}, depends on the distance between two departments and is given by Table 2. Also, the most common adjacency values, AV_{tij}, found in the literature are given in Table 3. Finally objective (4) maximises the distance requests that the designer forced on the layout.

Constraints (5) to (8) assure the departments are not going out of the factory floor. Constraints (9) and (10) calculate the distance between the centroids of the departments i and j. These constraints are similar to four constraints presented by McKendall and Hakobyan (2010), but we summarised them by adding the absolute sign. Constraints (11) and (12) calculate the length and width of the departments according to their orientations. Constraints (13) and (14) ensure the forced orientations are respected. Constraint (15) shows that any change in departments leads to a rearrangement cost. Finally constraints (16) and (17) show the restrictions of the decision variables.

Table 2. Range of adjacency factor.

Adjacency factor af_{tij}	Relationship condition
1.0	$0 < x'_{tij} + y'_{tij} <$ (width + length)/6
0.8	(width + length)/6 $< x'_{tij} + y'_{tij} <$ (width + length)/3
0.6	(width + length)/3 $< x'_{tij} + y'_{tij} <$ (width + length)/2
0.4	(width + length)/2 $< x'_{tij} + y'_{tij} <$ 2*(width + length)/3
0.2	2*(width + length)/3 $< x'_{tij} + y'_{tij} <$ 5*(width + length)/6
0.0	5*(width + length)/6 $< x'_{tij} + y'_{tij} <$ (width + length)

Table 3. Adjacency values correspond to adjacency requirement.

Adjacency requirement	AV_{tij}	Relationship
A	5	Extremely desirable
E	4	Very desirable
I	3	Desirable
O	2	So-so
U	1	Unimportant
X	0	Undesirable

3. Solution procedure

PSO, introduced by Kennedy and Eberhart (1995) as an optimisation tool, is a population-based search method in which solutions, called particles, change their positions continuously. In PSO, particles fly around in a multidimensional search space, and during flight, each particle adjusts its position according to its own past, and the experience of neighbour particles. Since the solutions (particles) in PSO naturally move in continuous space and because of the simplicity and convergence speed of PSO, we selected this method to solve our problem.

Suppose we have an n-dimensional feasible solution space and there are m particles in the swarm. The position and velocity of particle i are denoted by $X_i = \{x_{i1}, x_{i2}, \ldots, x_{in}\}$ and $V_i = \{v_{i1}, v_{i2}, \ldots, v_{in}\}$ for $i = 1, 2, \ldots, m$, respectively. Each particle uses its best position (pbest) and the swarm best position (gbest) to move in each iteration of the algorithm. The new position and velocity of particles can be calculated from Equations (18) and (19).

$$V_t = w * V_{t-1} + c1 * \text{rand}_1 * (\text{pbest} - X_{t-1}) + c2 * \text{rand}_2 * (\text{gbest} - X_{t-1}) \tag{18}$$

$$X_t = X_{t-1} + V_t \tag{19}$$

where w, named as the inertial weight, is used for preventing the particles from falling into local optimum traps or helping them for better local searches. The c1 is selfish factor and shows the reliance on own particle power. The c2 is social factor and shows the reliance on the whole swarm. Generally c1 and c2 are set equal to 2. Also, rand_1 and rand_2 are two uniform random numbers between 0 and 1.

To implement MOPSO, the pbest and gbest must be redefined. Also, the algorithm must restrict the particles from going out of the solution space, formed by the constraints, or draws them back to the solution space. The following sub-sections explain how we handle these issues.

3.1 *Finding pbest*

At the first step of PSO, we consider the first solution as the pbest. In the next iterations, if the current position of particle i dominates the pbest, the current position will replace it. If the current position is dominated by the pbest, pbest remains unchanged. If there is no dominance relation between them, we calculate a penalty point (sub-section 3.3) for both of them and the solution with the lower penalty will be selected as the pbest. If the penalty points are equals, we randomly select one of them as the pbest.

3.2 *Finding gbest*

By increasing the number of objectives, the size of non-dominated solutions becomes very large and finding gbest among them will be a difficult task. To find gbest, we use the preference order ranking method proposed by Wang and Yang (2009) with two modifications: the first one, we consider a weight (given by decision maker) for each objectives subsets and the second one, we use the roulette wheel method for selecting the gbest among the non-dominated solutions. The complete procedure is as follows:

(1) Find all non-dominated solutions (particles) in each iteration ($X^i_{ND}, i = 1, 2, \ldots, m$) where m is the number of particles.
(2) Consider all subsets of the objectives.
(3) Build a $[m + 1 \times m + 1]$ matrix in which first m rows and columns represent m non-dominated solutions and the last ones represent gbest. The elements of this matrix are the dominance relationships between solutions.
(4) Calculate the dominance score (Wang and Yang 2009) for each solution according to the weight of subsets.
(5) Employ the roulette wheel method on the $m + 1$ dominance score values and select its output as the gbest.

A simple example is given to illustrate the proposed ranking procedure. Suppose we have three objectives O1, O2 and O3 with the weights of 2, 3 and 1, respectively. The subsets of these objectives will be $\{$O1, O2, O3, $\{$O1, O2$\}$, $\{$O1, O3$\}$, $\{$O2, O3), $\{$O1, O2, O3$\}\}$. Suppose, among the n particles, we have five non-dominated solutions. The corresponding matrix is given by Table 4.

This table shows that regarding objectives subsets $\{O_1, O_2\}$ and O_2, solution X5 dominates X4. Also considering O_2, solution X2 is dominated by X3 and gbest. The dominance score of solution X5 is calculated as 23 by summation of the weights of the objectives. After calculating all of the dominance scores, we use the roulette wheel method and select one of the solutions as gbest.

Table 4. The dominance relation between all non-dominated solutions and the last gbest.

	X1	X2	X3	X4	X5	gbest
X1	–	*	*	O_2	O_1	*
X2	O_1, O_3	–	O_2, $\{O_2, O_3\}$	O_1	*	O_1
X3	O_2	O_2	–	O_2	*	*
X4	*	*	O_3	–	O_1	*
X5	$\{O_1, O_2\}$, $\{O_1, O_3\}$,	O_2, O_3	O_1	O_2, $\{O_1, O_2\}$	–	
gbest	O_1, O_2, $\{O_2, O_3\}$	O_2	O_1, O_2, O_3	O_2	O_1	–

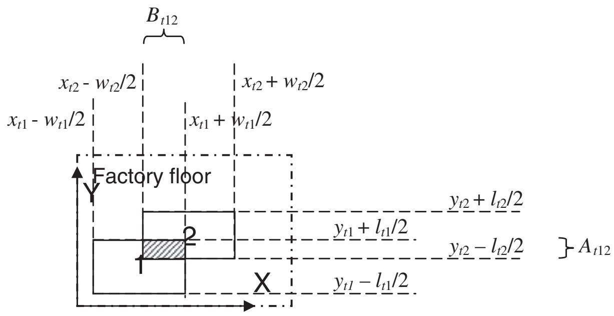


Figure 1. The general formation for the overlapped departments.

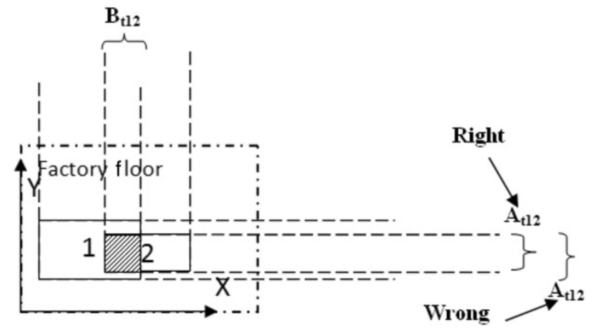


Figure 2. A special case of the overlapped departments.

3.3 *Defining penalty*

The department overlapping is not permitted. Chwif *et al.* (1998) proposed to consider a penalty for violent solutions. This penalty is calculated by sum of A_{ij} and B_{ij} values given by the Equations (20) and (21) that are the values of overlapping in X and Y co-ordinates, respectively. Here, if both A_{ij} and B_{ij} are greater than zero, it means that departments i and j are overlapped.

$$A_{ij} = \frac{w_i + w_j}{2} - |x_i - x_j| \tag{20}$$

$$B_{ij} = \frac{l_i + l_j}{2} - |y_i - y_j| \tag{21}$$

We rewrite these equations below and draw them in Figure 1.

$$A_{tij} = \begin{cases} (w_{ti} + w_{tj}) - \left| \left(x_{ti} - \frac{w_{ti}}{2} \right) - \left(x_{tj} + \frac{w_{tj}}{2} \right) \right| & \text{If } x_{ti} < x_{tj} \\ (w_{ti} + w_{tj}) - \left| \left(x_{tj} - \frac{w_{tj}}{2} \right) - \left(x_{ti} + \frac{w_{ti}}{2} \right) \right| & \text{If } x_{ti} > x_{tj} \end{cases} \tag{22}$$

$$B_{tij} = \begin{cases} (l_{ti} + l_{tj}) - \left| \left(y_{ti} - \frac{l_{ti}}{2} \right) - \left(y_{tj} + \frac{l_{tj}}{2} \right) \right| & \text{If } y_{ti} < y_{tj} \\ (l_{ti} + l_{tj}) - \left| \left(y_{tj} - \frac{l_{tj}}{2} \right) - \left(y_{ti} + \frac{l_{ti}}{2} \right) \right| & \text{If } y_{ti} > y_{tj} \end{cases} \tag{23}$$

Figure 2 shows a case that A_{ij} and B_{ij} do not present real overlapping values. To correct this error, we use Equations (24) and (25) instead of (22) and (23).

$$A_{tij} = \begin{cases} w_{tj} & \text{If } \left(x_{ti} + \frac{w_{ti}}{2} \right) > \left(x_{tj} + \frac{w_{tj}}{2} \right) \text{ and } \left(x_{ti} - \frac{w_{ti}}{2} \right) < \left(x_{tj} - \frac{w_{tj}}{2} \right) \\ w_{ti} & \text{If } \left(x_{ti} + \frac{w_{ti}}{2} \right) < \left(x_{tj} + \frac{w_{tj}}{2} \right) \text{ and } \left(x_{ti} - \frac{w_{ti}}{2} \right) > \left(x_{tj} - \frac{w_{tj}}{2} \right) \end{cases} \tag{24}$$

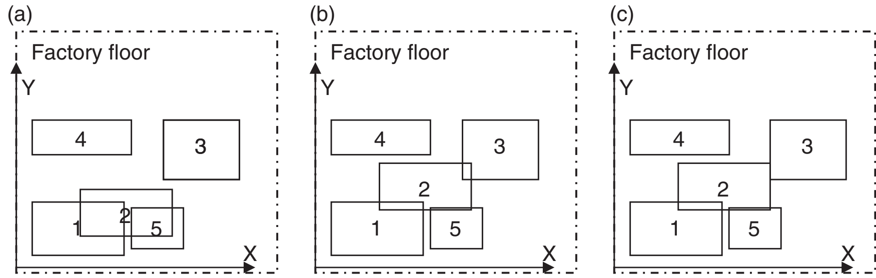


Figure 3. The illustrative example for shifting procedure.

$$B_{tij} = \begin{cases} l_{tj} & \text{If } \left(y_{ti} + \frac{l_{ti}}{2}\right) > \left(y_{tj} + \frac{l_{tj}}{2}\right) \text{ and } \left(y_{ti} - \frac{l_{ti}}{2}\right) < \left(y_{tj} - \frac{l_{tj}}{2}\right) \\ l_{ti} & \text{If } \left(y_{ti} + \frac{l_{ti}}{2}\right) < \left(y_{tj} + \frac{l_{tj}}{2}\right) \text{ and } \left(y_{ti} - \frac{l_{ti}}{2}\right) > \left(y_{tj} - \frac{l_{tj}}{2}\right) \end{cases} \tag{25}$$

If both of these values are greater than zero, departments i and j are overlapped. Instead of summation, we multiply these numbers and add the results to the objective function. So the overall penalty (with a suitable coefficient of α) is:

$$\text{Total Penalty} = \alpha * \sum_{t=1}^{T} \sum_{i=1}^{N} \sum_{j=1}^{N} A_{tij} * B_{tij} \quad \text{If } A_{tij} > 0 \text{ and } B_{tij} > 0 \text{ for } i \neq j \tag{26}$$

Adding this penalty will prevent over-closeness of departments but will not completely prevent overlapping of the departments. So, we propose a heuristic approach to correct the solutions with overlapped departments in the next sub section.

3.4 *Shifting the overlapped departments*

As mentioned by evaluating A_{tij} and B_{tij} we can find overlapped departments. Suppose departments i and j in period t are overlapped. First we check whether A_{tij} is greater than B_{tij} or vice versa. If A_{tij} is greater, the department with greater x_{ti} will be shifted to the right by A_{tij} value. If B_{tij} is greater, the department with greater y_{ti} will be shifted to the up by B_{tij} value. It is possible a shifting leads to another overlapping. Thus, after each shifting, we recalculate A_{tij} and B_{tij} values for the shifted departments. This procedure is repeated until there are no overlapped departments. As we shift the departments to the north-east points, after a finite number of steps, all of the overlapping cases will be solved.

We illustrate this heuristic by an example. Suppose after some iteration, we have a solution as Figure 3, part (a). Departments 1 and 2 and departments 2 and 5 are overlapped. Because we check the departments in order of their indexes, first the overlap of departments 1 and 2 must be considered. B_{t12} is greater than A_{t12}, so the department with greater y_{ti} (department 2) must be shifted upward. The resulting layout is shown in part (b) of Figure 3. As you can see, this shift leads to another overlap between departments 2 and 3. We continue the procedure and solve the overlapping of departments 2 and 3. Figure 3, part (c) shows the final layout.

3.5 *Condensing of solutions*

As the PSO deals with continuous variables; it is possible to have some unusable free spaces between departments. For example, in Figure 3, part (c), if we reduce the gap between departments 2 and 4 (or departments 5 and 2), we may reduce material handling costs. So we propose a condensing heuristic procedure to reduce such gaps.

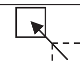
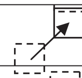
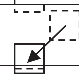
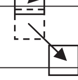
State	Pointer	Sequence of movements
North-west		Up-left-up-left …
		Left-up-left-up …
North-east		Up-right-up-right …
		Right-up-right-up …
South-west		Down-left-down-left …
		Left-down-left-down …
South-east		Down-right-down-right …
		Right-down-right-down

Figure 4. The vector and sequence for condensing of the departments.

First we calculate material handling costs of all departments by Equation (27). Then we select the department with the largest material handling cost and name it the base department.

$$MHC_{ti} = \sum_{j=1}^{N} C_{tij} \quad \text{For } i = 1, 2, \ldots N \text{ and } t = 1, 2, \ldots T \tag{27}$$

By connecting the centroid of the base department to the centroid of the nearest department to it, we may encounter four possible states, as Figure 4. In this figure, the candidate department for movement is dashed (opposite to the base department). In each part of the movement (moving towards up, down, left or right) we shift the candidate department in its direction until it reaches another department or reaches the centroid of the base department. After each movement, we calculate the improvement in the material handling costs and select the best sequence of movement (or none of them, if there are no improvements). We continue this procedure for the rest of the non-base departments until we could not move any department with improvement in the material handling cost.

This procedure can be explained with an example. Suppose we have a layout as Figure 5, part (a). We suppose every movement leads to a decrease in material handling costs. The base department is department 2 and its nearest department is department 4. The vector points to south-east, so we must use one of the 'down-right…' or 'right-down…' procedures. Both of these sequences lead to a new layout as Figure 5, part (b). The next nearest department is department 1. We could shift this department to the left with no obstacle until it is aligned with the base department. Departments 3 and 5 are also shifted near to the base department. The final layout after condensing is shown in Figure 5, part (e). The complete pseudo code of the procedure is shown in Figure 6.

4. Numerical experiments

Based on our knowledge, the proposed model has not been considered so far, and we have not found another solution method to compare its performance with that of the MOPSO algorithm. To evaluate the performance of the MOPSO algorithm, we have done some computational experiments discussed in the following sub-sections. In these experiments, the parameters of MOPSO were set as $c1 = 2$, $c2 = 2$ and $w = 0.05$.

4.1 *Single objective test problems*

There are two meta-heuristic methods (Dunker *et al.* 2005, McKendall and Hakobyan 2010) for the single objective DFLP. To compare the attained results of MOPSO with those of HGA (Dunker *et al.* 2005) and TS\BSH (McKendall and Hakobyan 2010) for two available test problems (Yang and Peters 1998), we consider the sum of material handling cost and rearrangement cost as a single objective and relax other objectives and their related constraints in our problem. We also removed the pickup/drop-off locations assumption.

We must mention that for applying the proposed method for single objective case, a pseudo particle is generated. At each iteration of PSO, the current solution and the pbest of each particle are compared period by period. The pseudo particle consists of the layout with minimum material handling costs at each period. Then we try to improve this pseudo particle through the following procedure.

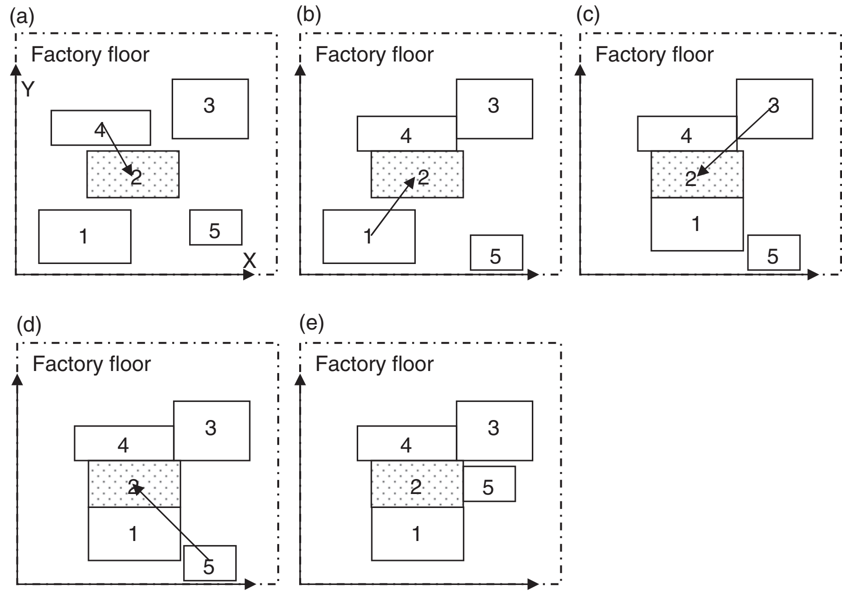


Figure 5. An example for condensing of the departments.

```
Do
 For each particle
     Initialise particle
     Correct the orientation of departments according to forced orientation.
     Correct the position of departments if they go out of the factory floor.
     Calculate the penalty
     Correct the overlapped departments.
Condense the system
     Set the other parameters of the problem.
     Calculate the fitness value of each objective.
     Check for dominance and set the pbest
   End
   Use the PO ranking method and set the gbest
   For each particle
     Calculate particle velocity
     Update particle position
   End
While maximum iterations or minimum error criteria is not attained
```

Figure 6. The complete pseudo code for implementing the MOPSO.

For a given pseudo particle **do**:
For $i = 2$ **to** T
  Replace the layout of period i with that of period $i - 1$
  Calculate overall objective value (material handling costs + rearrangement costs)
  If the objective value is improved, accept this change.
End

After implementation of these changes, if the pseudo particle has a better overall objective value than the pbest, we change it with pbest.

The two test problems are a six-department with six-period problem (P6-6) and a 12-department with four-period problem (P12-4). We also use 50 units and 200 units for rearrangement costs for the first and second test problems respectively as in Yang and Peters (1998).

Table 5 shows the convergence of the PSO algorithm. The PSO reaches a good solution after 200 iterations.

The best solutions and running times of three algorithms are given in Table 6. The results of applying our MOPSO to single objective DFLP have been uniformly attractive in terms of both solution quality and computation times. Although our method is designed for a completely general form of DFLP with pick up/drop off locations.

Table 5. The PSO convergence for two available test problems.

	Initial solution		Iteration 50		Iteration 100		Iteration 200		Iteration 500	
	Cost	RTime	Cost	RTime	Cost	RTime	Cost	RTime	Cost	RTime
P6-6	7495.2	0	6932.1	63 sec	6791.3	127 sec	6659.4	251 sec	6659.4	508 sec
P12-4	30594.3	0	28268.3	303 sec	28142.3	610 sec	28115.5	1225 sec	28115.5	3024 sec

Table 6. The comparison of three algorithms for test problems P6-6 and P12-4.

	HGA		TS/BSH		PSO	
	Cost	RTime	Cost	RTime	Cost	RTime
P6-6	6569.0	29.4 min	6648.3	27.77 min	6659.4	8.4 min
P12-6	27748.0	160.0 min	26845.5	82.00 min	28115.5	50.5 min

Table 7. The comparison results for three generated test problems.

	VIP-PLANOPT		PSO	
No. departments	Costs	RTime	Costs	RTime
20	1157.00	Instant	1239.30	16.5 min
28	6447.25	Instant	6846.55	24.6 min
50	78224.68	Instant	79566.25	73.8 min

We expect as the size of departments becomes more and more varied, our proposed algorithm becomes better than the competitors, because it does not rely on the shape of departments and uses different heuristics to form a layout. For further inspecting of the proposed algorithm, three test problems from static FLP benchmarks of VIP-PLANOPT software were selected and their optimum solutions were compared with those of our method. The results are shown in Table 7. It is clear that the proposed algorithm generated better results where the test problem has the more various-size departments. This approves our expectation about the better performance of the algorithm in solving the problems with various size departments.

4.2 *Randomly generated multi-objective test problems*

We randomly generated 12 instances of the considered problem with small (instances 1 to 4), medium (instances 5 to 8) and large (instances 8 to 12) sizes. The distribution functions and the values of the parameters used to generate these instances are given in Table 8. These values are inspired from research works such as Meller *et al*. (1998), Lee and Lee (2002), Chen and Sha (2005), Deb and Bhattacharyya (2005), Aiello *et al*. (2006), Ye and Zhou (2006), Xie and Sahinidis (2008), Ramkumar *et al*. (2009) and Singh and Singh (2010).

These 12 instances were solved by MOPSO starting with a random feasible initial solution. As mentioned before, for calculating the dominance score of each solution, the objectives are weighted. Naturally these weights should be proposed by decision makers considering the real world conditions. In our experiments, the weights of material handling costs, rearrangement costs, adjacency requests and distance requests are 6, 2, 3 and 1 respectively.

Table 9 shows the objective values of the initial solutions and the final solution of MOPSO attained after 200 iterations. Also, the percentage of improvement is reported. Note that the negative percentage for the third and fourth objectives indicates an improvement, since we want to maximise these two objectives.

On average, the percentage of improvements on all objectives is 14%. The average improvement for closeness rating is 29% on 12 solved problems and is 2% for adjacency value criteria.

4.3 *Performance metrics*

Several performance metrics are available for testing the quality of multi-objective solutions. Most of these metrics are concentrated on two issues: minimising the distance between Pareto frontier generated by an algorithm and actual Pareto frontier, Generational distance (GD) metric, and secondly maximising the smoothness of solutions distribution, Spacing (SP) metric.

Table 8. General specifications of 12 test problems.

Department number	Small size: Uniform (5, 10) Medium size: Uniform (15, 20) Large size: Uniform (25, 30)
Period number	Small size: Uniform (2, 4) Medium size: Uniform (4, 6) Large size: Uniform (6, 8)
Material handling cost	Uniform (0, 20)
Rearrangement cost	Uniform (20, 50)
Adjacency value	Uniform (0, 5)
Distance rating	Uniform (0, 10)
Shorter length of departments	Uniform (3, 8)
Longer length of departments	Shorter length + Uniform (0, 2)
Forced orientation	Uniform (−1, 1)
Length of factory floor	$\sqrt{\text{Total departments area} * 2} \pm$ Threshold[1]
Width of factory floor	(Total departments area ∗ 2)/Length

Note: [1]Threshold = 0.3 * $\sqrt{\text{Total departments area} * 2}$.

Table 9. The comparison results for 12 test problems.

		Material handling cost			Re-arrangement cost			Adjacency value			Closeness rating		
		Initial	Iter 200	%	Initial	Iter 200	%	Initial	Iter 200	%	Initial	Iter 200	%
Small	P1	53,641	41,068	23.44	714	714	0.00	420	425	−1.19	13,211	14,740	−11.57
	P2	95,857	71,997	24.89	1131	945	16.45	635	653	−2.77	19,864	26,149	−31.64
	P3	29,138	24,395	16.28	886	886	0.00	289	298	−3.04	7692	8348	−8.52
	P4	61,004	52,364	14.16	1124	1,013	9.88	535	549	−2.58	15,378	17,626	−14.62
Medium	P1	1,320,618	953,524	27.80	3982	3982	0.00	5040	5156	−2.31	2,50,839	337,529	−34.56
	P2	511,114	359,921	29.58	2080	2080	0.00	2285	2343	−2.52	1,01,508	136,479	−34.45
	P3	804,796	560,678	30.33	3456	3665	−6.05	3529	3606	−2.19	1,42,167	208,401	−46.59
	P4	911,122	731,348	19.73	3474	3474	0.00	3985	4074	−2.25	177,893	222,937	−25.32
Large	P1	5,830,121	4,292,566	26.37	8047	7804	3.02	16,161	16,427	−1.65	1,060,739	1,451,951	−36.88
	P2	2,946,141	2,106,281	28.51	5226	4804	8.08	8692	8852	−1.85	524,057	720,980	−37.58
	P3	5,890,602	4,443,666	24.56	7466	7466	0.00	15,104	15,406	−1.99	1,119,889	1,431,425	−27.82
	P4	5,169,247	3,605,151	30.26	7283	7024	3.56	13,981	14,230	−1.79	901,716	1,265,444	−40.34

4.3.1 *Generational distance*

This metric was introduced by Van Veldhuisen and Lamont (1998). It evaluates the distance between the obtained non-dominated solutions by an algorithm and the actual Pareto optimal solutions (assuming we know these solutions). This distance is calculated by Equation (28).

$$GD = \frac{\sqrt{\sum_{i=1}^{n} d_i^2}}{n} \tag{28}$$

where n is the number of non-dominated solutions and d_i is the Euclidian distance (that depends on the number and actual value of objectives) between each non-dominated solution and the nearest one in the Pareto optimal set. It is clear that $GD = 0$ means all solutions are in the Pareto frontier. The values greater than zero indicate the relative distance between the found solutions and the actual Pareto frontier.

The mathematical model presented in Section 2 was coded with GAMS software using the WSM method. Also the MOPSO ran 500 times. Table 10 shows the resulting GD values for four random small size instances. This table shows that the distance between the MOPSO results and the Pareto frontier is satisfying.

Table 10. Summary of the results for the generational distance metric for the small sized test problems.

Test problem	Objective	Mean	SD	Max	GD value
1	1	9345.344	622.510300	10959.26	78.7790
	2	120.000	000.000000	120.00	
	3	106.024	2.823177	112.80	
	4	1715.040	85.556360	2030.17	
2	1	52802.110	2044.629000	58066.77	481.3953
	2	1015.770	12.748420	1125.00	
	3	503.492	3.682162	511.00	
	4	16065.200	714.949900	18447.80	
3	1	13510.070	1485.750000	17422.00	150.7744
	2	720.570	4.989393	740.00	
	3	194.814	1.735722	198.80	
	4	4606.160	266.831600	5254.26	
4	1	98277.000	3379.225000	109429.80	506.3619
	2	1395.110	20.506660	1420.00	
	3	884.306	4.728974	897.80	
	4	26067.600	1020.803000	28511.60	

Note: SD, standard deviation.

Table 11. The spacing value for random generated test problems.

Size	Index	Spacing value	Size	Index	Spacing value	Size	Index	Spacing value
Small	1	58.63	Medium	5	1164.1	Large	9	7945.2
	2	386.65		6	10368.1		10	36988.3
	3	135.42		7	4846.3		11	7338.5
	4	406.95		8	3119.2		12	18824.2

4.3.2 *Spacing*

This metric, introduced by Schot (1995), is a tool to measure the uniformity of the spread of solutions. Equation (29) measures the distance variance of each point in the current solution set to its closest neighbour.

$$SP = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(\bar{d} - d_i)^2} \quad \text{where } d_i = \min_j\left(\sum_{o=1}^{O}\left|Z_o^i - Z_o^j\right|\right), i,j = 1,\ldots n \tag{29}$$

where n is the number of non-dominated solutions obtained by MOPSO algorithm, O is the number of objectives and d is the mean value of all d_i. Note that $SP = 0$ means that all non-dominated solutions are spaced equally from each other.

Also for calculating the spacing metric 500, 250 and 100 runs were carried out for small, medium and large size test problems, respectively. Table 11 shows the acquired spacing values for the test problems. These values show good uniform distribution of the results.

5. Sensitivity analysis of PSO parameters

PSO has three main parameters: c1 (known as selfish factor), c2 (known as social factor) and w (weighting factor). The values of these parameters could affect the quality of the solutions. In this section we verify the change in the results by changing these values. Five values for each 'c1' and 'c2' and four values for 'w' are considered that leads us to 100 combinations. For each combination, three random instances, with small, medium and large size, are generated and solved by MOPSO. In the previous sections, the presented computational results of PSO were obtained by setting $c1 = 2$, $c2 = 2$ and $w = 0.05$. In this section these values are used as the base of comparison.

Table 12. Sensitivity analysis results on PSO parameters.

	c2						c2				
c1	0	1	2	3	4	c1	0	1	2	3	4
			$w=0.01$						$w=0.05$		
0	0.1165	0.0245	0.0322	0.0160	0.0442	0	0.1079	0.0467	0.0203	0.0276	0.0524
1	0.0733	0.0003	0.0039	0.0028	0.0389	1	0.0403	−0.0164	−0.0121	0.0125	0.0228
2	0.0534	−0.0138	0.0135	0.0134	0.0272	2	0.0668	0.0044	0.0000	0.0159	0.0520
3	0.0612	0.0059	0.0174	0.0518	0.0334	3	0.0641	0.0104	0.0169	0.0355	0.0559
4	0.0531	0.0324	0.0363	0.0394	0.0682	4	0.0241	0.0209	0.0234	0.0401	0.0507
			$w=0.1$						$w=0.2$		
0	0.0863	0.0274	0.0315	0.0112	0.0214	0	0.0688	0.0108	0.0080	0.0131	0.0334
1	0.0807	0.0086	−0.0084	0.0093	0.0249	1	0.0798	−0.0037	0.0004	0.0050	0.0295
2	0.0531	−0.0038	0.0112	0.0209	0.0455	2	0.0435	−0.0150	0.0080	0.0228	0.0468
3	0.0497	0.0164	0.0269	0.0330	0.0455	3	0.0437	−0.0117	0.0085	0.0324	0.0526
4	0.0667	0.0262	0.0180	0.0316	0.0517	4	0.0447	0.0241	0.0320	0.0445	0.0382

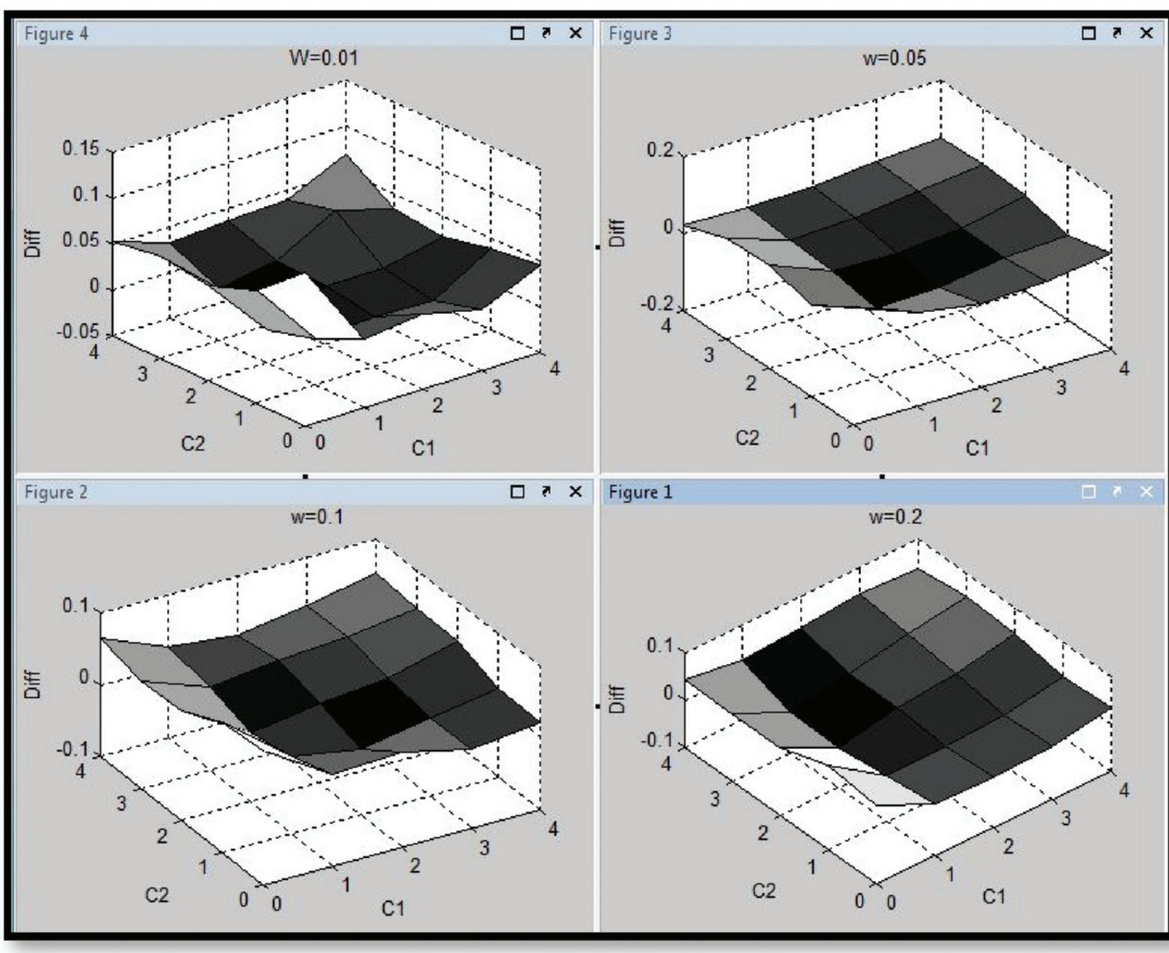


Figure 7. Results of sensitivity analysis on PSO parameters.

Equation (30) calculates the change in material handling cost value as the most important objective for a given combination.

$$\sigma = \frac{Fitness(C1, C2, w) - Fitness(2, 2, 0.05)}{Fitness(2, 2, 0.05)} \tag{30}$$

Table 12 shows the average σ on three instances. Also the results are plotted in Figure 7. As can be seen the changes are not significant.

6. Conclusion

In this paper we considered a multi-objective dynamic facility layout problem with unequal-size departments and pick-up/drop-off locations. Minimising material-handling cost, re-arrangement cost and maximising adjacency factor and distance requests are considered as four distinct objectives. First a mathematical model is presented for this problem. Then a multi-objective particle swarm optimisation (MOPSO) algorithm with a new heuristic approach for handling of departments overlapping is implemented to solve the proposed model. Also another heuristic approach is introduced to reduce the possible unused gaps between departments.

The performance of the proposed method was evaluated by different computational experiments. The first two sets of available test problems in the literature for the single objective case of the problem were selected and solved by a modified version of the proposed MOPSO and two existent methods in the literature. MOPSO was capable of finding the solutions with quality near to those of the other algorithms with less computation time. Then three benchmark problems of PLANOPT software were solved by the proposed method. This experiment showed the good performance of the MOPSO to solve the problems with unequal size departments. Also, 12 random instances were generated and the percentage of improvements on the initial solutions of MOPSO was calculated. The average improvements were in the range of 2 to 24% for four objectives. The calculated generational distance and spacing multi-objective metrics confirmed again the good performance of MOPSO to solve the considered problem. Finally, the results of the sensitivity analysis proved the robustness of the developed MOPSO.

Acknowledgements

The first author would like to express his appreciation to the Iranian National Science Foundation (grant number 89001465) for the financial support of this study. The authors are also grateful to the respected reviewers for their valuable comments in preparation of the revised manuscript.

References

Aiello, G., Enea, M., and Galante, G., 2006. A multi-objective approach to facility layout problem by genetic search algorithm and Electre method. *Robotics and Computer-Integrated Manufacturing*, 22 (5–6), 447–455.

Baykasoglu, A., Dereli, T., and Sabuncu, I., 2006. An ant colony algorithm for solving budget constrained and unconstrained dynamic facility layout problems. *Omega*, 34 (4), 385–396.

Chan, W.K. and Malmborg, C.J., 2010. A Monte Carlo simulation based heuristic procedure for solving dynamic line layout problems for facilities using conventional material handling devices. *International Journal of Production Research*, 48 (10), 2937–2956.

Chen, C.-W. and Sha, D.Y., 2005. Heuristic approach for solving the multi-objective facility layout problem. *International Journal of Production Research*, 43 (21), 4493–4507.

Chwif, L., Barretto, M.R.P., and Moscato, L.A., 1998. A solution to the facility layout problem using simulated annealing. *Computers in Industry*, 36 (1–2), 125–132.

Deb, S.K. and Bhattacharyya, B., 2005. Solution of facility layout problems with pickup/drop-off locations using random search techniques. *International Journal of Production Research*, 43 (22), 4787–4812.

Dunker, T., Radons, G., and Westkamper, E., 2005. Combining evolutionary computation and dynamic programming for solving a dynamic facility layout problem. *European Journal of Operational Research*, 165 (1), 55–69.

El-Baz, M.A., 2004. A genetic algorithm for facility layout problems of different manufacturing environments. *Computers & Industrial Engineering*, 47 (2–3), 233–246.

Kennedy, J. and Eberhart, R., 1995. *Particle swarm optimization. IEEE international conference on neural networks*, November/December 1995, Perth, Australia. Piscataway, NJ: IEEE Service Center, IV, 1942– 1948. .

Khilwani, N., Shankar, R., and Tiwari, M.K., 2008. Facility layout problem: an approach based on a group decision-making system and psychoclonal algorithm. *International Journal of Production Research*, 46 (4), 895–927.

Lee, Y.H. and Lee, M.H., 2002. A shape-based block layout approach to facility layout problems using hybrid genetic algorithm. *Computers & Industrial Engineering*, 42 (2–4), 237–248.

Logendran, R. and Kriausakul, T., 2006. A methodology for solving the unequal area facility layout problem using distance and shape-based measures. *International Journal of Production Research*, 44 (7), 1243–1272.

McKendall Jr, A.R. and Hakobyan, A., 2010. Heuristics for the dynamic facility layout problem with unequal-area departments. *European Journal of Operational Research*, 201 (1), 171–182.

McKendall Jr, A.R. and Shang, J., 2006. Hybrid ant systems for the dynamic facility layout problem. *Computers & Operations Research*, 33 (3), 790–803.

Meller, R.D., Chen, W., and Sherali, H.D., 2007. Applying the sequence-pair representation to optimal facility layout designs. *Operations Research Letters*, 35 (5), 651–659.

Meller, R.D., Narayanan, V., and Vance, P.H., 1998. Optimal facility layout design. *Operations Research Letters*, 23 (3–5), 117–127.

Nourelfath, M., Nahas, N., and Montreuil, B., 2007. Coupling ant colony optimization and the extended great deluge algorithm for the discrete facility layout problem. *Engineering Optimization*, 39 (8), 953–968.

Paul, R.C., Asokan, P., and Prabhakar, V.I., 2006. A solution to the facility layout problem having passages and inner structure walls using particle swarm optimization. *International Journal of Advanced Manufacturing Technology*, 29 (7–8), 766–771.

Ramkumar, A.S., Ponnambalam, S.G., and Jawahar, N., 2009. A new iterated fast local search heuristic for solving QAP formulation in facility layout design. *Robotics and Computer-Integrated Manufacturing*, 25 (3), 620–629.

Ramkumar, A.S., Ponnambalam, S.G., and Jawahar, N., 2009. A new iterated fast local search heuristic for solving QAP formulation in facility layout design. *Robotics and Computer-Integrated Manufacturing*, 25 (3), 620–629.

Şahin, R. and Türkbey, O., 2009. A new hybrid tabu-simulated annealing heuristic for the dynamic facility layout problem. *International Journal of Production Research*, 47 (24), 6855–6873.

Şahin, R. and Türkbey, O., 2009. A simulated annealing algorithm to find approximate Pareto optimal solutions for the multi-objective facility layout problem. *International Journal of Advanced Manufacturing Technology*, 41 (9–10), 1006–1018.

Schott, J.R., 1995. *Fault tolerant design using single and multicriteria genetic algorithm optimisation*. Thesis (MS). Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, Cambridge, MA.

Singh, S.P. and Singh, V.K., 2010. An improved heuristic approach for multi-objective facility layout problem. *International Journal of Production Research*, 48 (4), 1171–1194.

Ulutaş, B.H. and Işlier A.A., 2007. Parameter setting for clonal selection algorithm in facility layout problems. Lecture Notes in Computer Science, Computational Science and Its Applications – ICCSA 2007, Part I, 886–899.

Van Veldhuisen, D.A. and Lamont, G.B., 1998. Multiobjective evolutionary algorithm research: A history and analysis. Department of Electronics and Computer Engineering, Graduate School of Engineering, Air Force Institute of Technology, Wright-Patterson, AFB, OH. Technical Report TR-98-03.

Wang, Y. and Yang, Y., 2009. Particle swarm optimisation with preference order ranking for multi-objective optimisation. *Information Sciences*, 179 (12), 1944–1959.

Xie, W. and Sahinidis, N.V., 2008. A branch-and-bound algorithm for the continuous facility layout problem. *Computers & Chemical Engineering*, 32 (4–5), 1016–1028.

Yang, T. and Peters, B.A., 1998. Flexible machine layout design for dynamic and uncertain production environments. *European Journal of Operational Research*, 108 (1), 49–64.

Ye, M. and Zhou, G., 2006. A local genetic approach to multi-objective, facility layout problems with fixed aisles. *International Journal of Production Research*, 45 (22), 5243–5264.

Zhou, G., Ye, M., Cao, Z. and Ye, F., 2006. A genetic algorithm approach on a facility layout design problem with aisles. Lecture Notes in Computer Science, Computational Intelligence, 1008–1013.