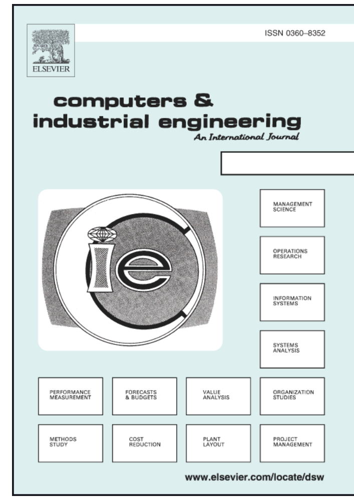
Accepted Manuscript



Designing cellular manufacturing systems considering S-shaped layout

Mohammad Mohammadi, Kamran Forghani

PII: S0360-8352(16)30194-2
DOI: http://dx.doi.org/10.1016/j.cie.2016.05.041
Reference: CAIE 4366

To appear in: *Computers & Industrial Engineering*

Received Date: 24 May 2015
Revised Date: 23 January 2016
Accepted Date: 30 May 2016

Please cite this article as: Mohammadi, M., Forghani, K., Designing cellular manufacturing systems considering S-shaped layout, *Computers & Industrial Engineering* (2016), doi: http://dx.doi.org/10.1016/j.cie.2016.05.041

This is a PDF file of an unedited manuscript that has been accepted for publication. As a service to our customers we are providing this early version of the manuscript. The manuscript will undergo copyediting, typesetting, and review of the resulting proof before it is published in its final form. Please note that during the production process errors may be discovered which could affect the content, and all legal disclaimers that apply to the journal pertain.

ACCEPTED MANUSCRIPT

Designing cellular manufacturing systems considering S-shaped layout

Mohammad Mohammadi[*,1] and Kamran Forghani[1]

[1] *Department of Industrial Engineering, Faculty of Engineering, Kharazmi University, Tehran, Iran*

Email addresses: mohammadi@khu.ac.ir (M. Mohammadi) and kamran21f@gmail.com (K. Forghani)

Abstract

In this paper, we present a new layout framework, called S-shaped layout, for the layout of cellular manufacturing systems. Based on the proposed S-shaped layout, we formulate an integrated bi-objective cell formation and layout problem, considering parameters such as part demands, operation sequences, machine dimensions and aisle widths. The first objective is to minimize the total inter- and intra-cell material handling costs. Also, the second objective is to maximize the total similarity between machines. A normalized weighted sum method is suggested to unify these objectives. As the problem is NP-hard, an efficient hybrid solution method combining simulated annealing and dynamic programming is developed to solve large-sized problems in a reasonable computational time. Finally, by solving numerical examples from the literature the suggested approach is compared with two recently developed approaches.

Keywords: Cellular manufacturing, cell formation, bi-objective optimization, layout problem, dynamic programming, simulated annealing

1. Introduction

In the last two decades, Cellular Manufacturing system (CMS) has emerged as an innovative and successful strategy in manufacturing systems producing a medium-volume and medium-variety productions. It derives from group technology concept and joins together the advantages of both flexible and mass production approaches. The main advantages of production using CMS are: reduction in setup times, work-in-process inventories, lead times, tool requirements and material handling costs. It could also cause remarkable improvement in product quality, productivity and production control (Wemmerlöv & Hyer, 1986; Mungwattana, 2000; Solimanpur et al., 2004). One of the crucial steps in the CMS design process is the cell formation (CF) problem, which has been extensively studied in the literature. It involves grouping parts with similar design features or processing requirements into part families and grouping machines into machine cells on the basis of operations required by the part families. The material flow between manufacturing cells as a result of exceptional elements (EEs) is a major obstacle in achieving the benefits of CMS (Arıkan & Güngör, 2009). An EE is a part that needs to be processed in more than one cell. So, in the CF problem, one common objective is to minimize the number of EEs. Other common objectives include minimization of inter-cell movement costs, minimization of number of inter-cell material flows, maximization of grouping efficiency, and maximization of grouping efficacy.

Facilities layout is a key factor in manufacturing systems and has a direct impact on the operational performance, as measured by manufacturing lead time, throughput rate and work-in-process (Benjaafar, 2002). Tompkins et al. (2003) estimated that 20–50% of manufacturing costs is related to the handling of parts. They also stated that an efficient facility layout may reduce them for 10–30%. Although minimizing the number of EEs or optimizing other common objectives mentioned above may reduce the flows between the cells, they do not necessarily lead to a minimum material handling cost. Because, the parameters related to the facility layout problem are ignored in the calculation of these objectives. This brings the attention towards the need for

[*] Corresponding author: Tel. & Fax: +98 21 88830891

ACCEPTED MANUSCRIPT

incorporating the facility layout problem into the CMS design process. In a CMS, the material handling cost usually consists of two cost components, inter- and intra-cell material handling costs. The inter-cell material handling arises from the movements of parts between the cells, and the intra-cell material handling cost arises from the movements of parts within the cells. Therefore, the inter-cell layout involves the placement of the cells on the shop floor, and the intra-cell layout involves the arrangement of machines within the cells.

Most approaches in the area of facility layout and CF problems, usually consider one of the inter- and intra-cell layouts in the CMS design problem. For simplicity, these approaches aim at minimizing the number of inter-cell movements or intra-cell movements or both, instead of minimizing the material handling cost. Moreover, those approaches that aim at minimizing the material handling cost usually apply unrealistic assumptions such as fixed cell locations and equal-sized machines in the layout problem. Consequently, the resulting layout may be inefficient. To overcome these drawbacks, this paper presents a new layout framework, called S-shaped layout, for designing CMS layout. Based on this S-shaped layout, we formulate an integrated bi-objective CF and layout problem, considering parameters such as part demands, operation sequences, machine dimensions and aisle widths. The first objective is to minimize the total inter- and intra-cell material handling costs, and the second one is to maximize the total similarity between machines. A normalized weighted sum method is proposed to unify these objectives. Due to computational complexity of the problem, a hybrid solution method combining Simulated Annealing (SA) and Dynamic Programming (DP) is developed to solve large-sized problems in a reasonable computational time. Finally, by solving numerical examples from the literature the suggested approach is compared with two recently developed approaches.

The remainder of this paper is organized as follows: Section 2 reviews relevant literature, Section 3 presents the description of the problem and Section 4 presents the hybrid method. Computational experiments and comparisons are carried out in Section 5. Finally, conclusions and directions for future research are given in Section 6.

2. Literature review

In recent years, there have been some studies that have applied simultaneous or sequential approaches to solve the CF and layout problems. As both of these problems are NP-hard (Garey & Johnson, 1979; Papaioannou & Wilson, 2010), using heuristic and meta-heuristic algorithms is very popular among researchers.

In the context of sequential approaches, Heragu and Kakuturi (1997) attempted to integrate the machine grouping problem with the layout problem. The machine cells were first formed by a heuristic algorithm, and then a hybrid SA algorithm was employed to construct near-optimal inter- and intra-cell layouts. Chan et al. (2006) proposed a two-stage solution approach based on Genetic Algorithm (GA) for solving the CF problem as well as the cell layout problem. The first stage was to identify machine cells and part families. Also, the second stage was to obtain the layout sequence of machine cells (linear inter-cell layout) such that the total inter-cell material handling cost is minimized. In their approach, the Quadratic Assignment Problem (QAP) was used to represent the inter-cell layout. Leung et al. (2008) first attempted to find the number of each machine type by knowing part demands, the processing times and machine capacities. Then, they applied various space filling curves to create initial layouts (it was assumed that each machine occupies almost the same area, i.e., one unit block of floor). In the next step, each of these initial layouts was improved by the CRAFT software. Finally, the layout design with the lowest cost was selected. Krishnan et al. (2012) developed a method based on the flow between machines to obtain machine cells. A modified grouping efficiency measure was used to determine the efficiency of grouping. They employed a GA-based machine placement procedure for the placement of machines in a facility layout matrix (grid) as a QAP. Chang et al. (2013) formulated a two-stage mathematical programming model to integrate the CF and layout problems. They employed a Tabu Search (TS) algorithm to solve the problem. In their approach, the CF and inter-cell layout problems were simultaneously solved in the first stage. Then, the intra-cell machine sequence was determined on the basis of the solution obtained in the first stage. In their research, the linear single- and double-row layouts were considered as two alternatives for the inter-cell layout. Javadi et al. (2014) presented a mathematical model for the inter- and intra-cell layout

ACCEPTED MANUSCRIPT

problems in a dynamic environment. The objective was to minimize the total costs of rearrangement as well as the total inter- and intra-cell material flows. They assumed that the CF phase has been already done and the locations for placing machines and cells are known in advance. A hybrid algorithm combining electromagnetism-like algorithm and GA was used to solve the problem. Ghosh et al. (2015) used the QAP to represent the cell layout problem, and developed an Immune GA to solve it.

In connection with simultaneous approaches, Aktürk and Turkcan (2000) proposed a solution methodology to simultaneously solve the CF and intra-cell layout problems. A holistic approach was used to maximize the profit of not only the overall system but also individual cells. Lee and Chiang (2001) addressed the joint problem of CF and its layout assignment to minimize the inter-cell material flow cost. It was assumed that the cell locations are approximately equally spaced, and the machine cells are located along a bi-directional linear layout. They proposed a new graphic approach based on a multi-terminal cut tree network model to form machine cells. A partition procedure was developed to separate the cut tree into a number of sub-graphs (cells) and assigns the location sequence of each cell by comparing the cut capacities. Chiang and Lee (2004) employed a SA algorithm combined with DP for solving the same problem presented in (Lee and Chiang, 2001). In their approach, the configuration of a solution is comprised of a string of integer values, where each value is associated with a machine. The DP algorithm is applied to partition each string into several segments (cells) such that the total inter-cell flow cost is minimized. Yin et al. (2005) incorporated part demands, sequence data and alternative process routings into a nonlinear mathematical model. They aimed to minimize a weighted sum of both inter-cell and intra-cell movements in which the weights are based on the actual unit costs of inter- and intra-cell movements. A heuristic methodology was also developed for solving such a nonlinear problem. Wu et al. (2007) developed a GA for solving an integrated CF and layout problem considering sequence data, work load, machine capacities, part demands, batch sizes, and layout type. Mahdavi et al. (2008) presented a heuristic approach based on a modified flow matrix for solving an integrated CF and intra-cell layout problem. The objective was to minimize the number of inter- and intra-cell movements as well as EEs and voids. Paydar et al. (2010) formulated the integrated CF and intra-cell layout problem as a multiple departures single destination multiple traveling salesman problem, and proposed a SA algorithm to solve it. Jolai et al. (2011) proposed a mathematical model for the inter- and intra-cell layout problem in CMS. They assumed that both machines and cells are assigned to pre-specified locations. A binary particle swarm optimization algorithm was implemented to minimize the total material handling cost. Jolai et al. (2012) presented a modified version of the proposed model by Wu et al. (2007) considering parameters such as forward and backward transportation, different batch sizes for parts and sequence data. They developed an Electromagnetism-like algorithm with a heuristic local search to minimize the total material handling cost and the number of EEs. Mohammadi and Forghani (2014) developed a GA for solving an integrated CF and layout problem. To increase the accuracy of the inter- and intra-cell layouts, they calculated the material handling cost on the basis of the actual location of machines on the shop floor and regarding machine dimensions and aisle widths. In their approach, machines assigned to a same cell are arranged along a line. Also, the machine cells are placed on the shop floor from bottom to top as a multi-row layout. Forghani et al. (2014) combined the QAP with the two-dimensional facility layout problem, and formulated an integrated CF and layout problem. The QAP was used to represent the intra-cell layout, and the inter-cell layout was represented by the continuous layout problem. It was assumed that cells are rectangular in shape and machines are equal in size. They employed a GA to solve the formulated problem.

3. Problem description and mathematical model

In this section, we present an integrated bi-objective problem to solve the CF, intra- and inter- cell layout problems, simultaneously. Different parameters such as part demands, operation sequences, machine dimensions, and aisle widths are taken into consideration. The first objective is to minimize the total material handling cost (including the inter- and intra-cell material handling costs), and the second one is to maximize the total similarity between machines. A new framework, called S-shaped layout, is developed for the layout of CMS. This layout framework is a modified version of the multi-row layout proposed by Mohammadi and Forghani (2014). In their approach, once the allocation of machines to the cells is specified, machines assigned to a same cell are arranged along a line, from left to right, by considering their length and the horizontal aisle

ACCEPTED MANUSCRIPT

width. The machine cells are also placed on the shop floor from bottom to top by taking into account the width of the cells and the vertical aisle width. In their approach, the maximum width of the shop floor was not taken into consideration. To address this parameter in the CMS layout problem and also to increase the flexibility of the multi-row layout, we present a new framework based on an S-shaped layout. It is assumed that the width of the shop floor is known in advance. Machines are laid out in an S-shaped form according to their assignment to the cells and by considering their dimensions, the width of the shop floor, the horizontal and vertical aisle widths. To do so, we start placing machines on the shop floor from the bottom-left corner to the bottom-right corner as far as possible (i.e., the constraint on the width of the shop floor is not violated). Once the first row is completed, this process is repeated on the second row (above the first row) from right to left. Again in the third row, machines are placed from left to right. This procedure is repeated until all machines are placed on the shop floor. It should be noted that in each row the entire layout is horizontally aligned to the middle of the shop floor. Also, in each row, machines are vertically aligned to the center of the row. The width of each row is equal to the maximum width of machines in that row. The proposed S-shaped layout has been illustrated in Fig. 1. According to Fig. 1(a), the sequence of machines to be placed on the shop floor is (7, 10, 6, 12, 9, 3, 2, 4, 8, 11, 5). Let's assume that machines 7, 10 and 6 form cell 1, machines 12, 9 and 3 form cell 2, machines 2, 4, 8 and 11 form cell 3, and machines 1 and 5 form cell 4. The width of the shop floor is 7.8 units. The horizontal and vertical aisle widths are also 0.6 and 1 units, respectively. The dimension of each machine is shown in the parenthesis in Fig. 1(b). For instance, the length and the width of machine 6 are 1.6 and 1.4 units, respectively. Based on the given layout sequence (permutation), machines 7, 10, 6 and 12 are placed in the first row from left to right. Machine 9 cannot be placed at the end of the first row, because the width of the shop floor is limited. So, it is placed in the second row. In the second row, we start placing machines 9, 3, 2 and 4 from right to left. Finally, in the third row machines 8, 11, 1 and 5 are placed from left to right. In the first row, machines 6 and 7 have the maximum width (1.4 units), so the width of the first row is 1.4 units. Similarly, the width of the second and third rows is obtained as 1.6 and 1.4 units, respectively.

Based on this S-shaped layout, we can calculate the coordinates of machines. For instance, in Fig. 1(b), the horizontal and vertical coordinates of machine 3 is 4.8 and 3.2, respectively. Once the coordinates of machines is determined, the material handling cost can be calculated according to the center-to-center distance between machines.

[Please insert Fig.1 here]

3.1. Notations

To formulate the problem the following notations are defined.

Indices

i	index of parts (i = 1, …, P) where P is the number of parts
k, k'	index of machines (k, k' = 1, …, M) where M is the number of machines
l	index of cells (l = 1, …, L) where L is the number of cells to be formed (L is a decision variable)

Parameters

D_i	demand of part i
$c^A_{i,k,k'}$	unit intra-cell material handling cost for transporting part i from machine k to machine k' per unit distance
$c^E_{i,k,k'}$	unit inter-cell material handling cost for transporting part i from machine k to machine k' per unit distance
w_k	width of machine k
h_k	length of machine k
$S_{k,k'}$	similarity coefficient between machines k and k'
L^X	horizontal aisle width
L^Y	vertical aisle width

ACCEPTED MANUSCRIPT

W width of the shop floor

$f_{i,k,k'}$ number of times that an operation at machine k immediately follows an operation at machine k' or vice versa for part i

NM maximum number of machines permissible in a cell

$C^{\max}$ maximum number of cells allowed

Π set of possible permutation of machines

$\boldsymbol{\pi}$ sequence (permutation) of machines to be laid out in the S-shaped layout $\boldsymbol{\pi} = \{\pi(1),\pi(2), \ldots,\pi(M)\}$, where $\pi(k)$ represents the machine index placed in kth order

$x_k(\boldsymbol{\pi})$ horizontal coordinate of the centroid of machine k in permutation $\boldsymbol{\pi}$

$y_k(\boldsymbol{\pi})$ vertical coordinate of the centroid of machine k in permutation $\boldsymbol{\pi}$

b_l, b_l' index of breaking node for cell l, where $\pi(b_l)$ is the last machine on the sequence to be included in cell l

Note that the coordinates of machines, $(x_k(\boldsymbol{\pi}), y_k(\boldsymbol{\pi}))$, for a given permutation $\boldsymbol{\pi}$ can be calculated by the procedure given in Appendix A.

In relation to the similarity coefficient between machines, Yin and Yasuda (2005) carried out a comparative study to evaluate the performance of twenty well-known similarity coefficients in the literature. These similarity coefficients are given in Table 1. They concluded that three similarity coefficients: Jaccard, Sorenson, and Sokal and Sneath 2 perform best among the tested similarity coefficients. They also did not recommend four similarity coefficients: Hamann, Simple matching, Rogers and Tanimoto, and Sokal and Sneath for CF applications due to their inefficient performance in test problems. Minimization of the total material handling cost may result in poor independence in the clustered cells. To overcome this difficulty, a similarity coefficient that covers negative values can be applied in the second objective. For this purpose, we apply Yule's similarity coefficient to calculate $S_{k,k'}$.

[Please insert Table 1 here]

3.2. Mathematical model

To simplify the formulation of the problem, we first introduce two auxiliary variables $F^A_{k,k'}$ and $F^E_{k,k'}$. These variables are calculated by Eqs. (1) and (2).

$$F^A_{k,k'} = \sum_{i=1}^{P} D_i c^A_{i,k,k} f_{i,,k,k} \left(\left|x_k(\boldsymbol{\pi}) - x_{k'}(\boldsymbol{\pi})\right| + \left|y_k(\boldsymbol{\pi}) - y_{k'}(\boldsymbol{\pi})\right|\right), \forall k, k', \tag{1}$$

$$F^E_{k,k'} = \sum_{i=1}^{P} D_i c^E_{i,k,k} f_{i,k,k} \left(\left|x_k(\boldsymbol{\pi}) - x_{k'}(\boldsymbol{\pi})\right| + \left|y_k(\boldsymbol{\pi}) - y_{k'}(\boldsymbol{\pi})\right|\right), \forall k, k'. \tag{2}$$

Note that in Eqs. (1) and (2) we applied the rectilinear norm to calculate the distance between two machines. However, without loss of generality, other norms (e.g., Euclidian norm and square Euclidian norm) can be used instead.

Now, the integrated CF and layout problem can be formulated as the following integer programming model:

$$\min_{\boldsymbol{\pi}\in\Pi} TH(\boldsymbol{\pi}) = \min_{\boldsymbol{\pi}\in\Pi} \left\{ \sum_{l=1}^{L} \left(\sum_{k=b_{l-1}+1}^{b_l - 1} \sum_{k'=k+1}^{b_l} F^A_{\pi(k),\pi(k')} + \sum_{k=b_{l-1}+1}^{b_l} \sum_{k'=1}^{b_{l-1}} F^E_{\pi(k),\pi(k')} \right) \right\}, \tag{3}$$

$$\max_{\boldsymbol{\pi}\in\Pi} TS(\boldsymbol{\pi}) = \max_{\boldsymbol{\pi}\in\Pi} \left\{ \sum_{l=1}^{L} \sum_{k=b_{l-1}+1}^{b_l - 1} \sum_{k'=k+1}^{b_l} S_{\pi(k),\pi(k')} \right\}. \tag{4}$$

Subject to:

$$b_0 = 0 \text{ and } 1 \le b_1 < \ldots < b_L = M, \tag{5}$$

ACCEPTED MANUSCRIPT

$$b_l - b_{l-1} \le NM, \forall l = 1, \ldots L, \tag{6}$$

$$L \le C^{\max}. \tag{7}$$

Objective function (3) minimizes the total material handling cost. In this function, the first term is associated with the intra-cell material handling cost, and the second one corresponds to the inter-cell material handling cost. Objective function (4) maximizes the total similarity between machines. Constraint (5) ensures that each formed cell includes at least one machine, and also guarantees that all machines are assigned to the cells. Constraint (6) represents that each cell can contain at most *NM* machines. Finally, constraint (7) prevents the formation of more than $C^{\max}$ cells.

To simplify $TH(\boldsymbol{\pi})$ in objective function (3), we can rewrite it as follows:

$$TH(\boldsymbol{\pi}) = \sum_{l=1}^{L}\left(\sum_{k=b_{l-1}+1}^{b_l-1} \sum_{k'=k+1}^{b_l} F^{A}_{\pi(k),\pi(k')} + \sum_{k=b_{l-1}+1}^{b_l} \sum_{k'=1}^{b_{l-1}} F^{E}_{\pi(k),\pi(k')} + \sum_{k=b_{l-1}+1}^{b_l-1} \sum_{k'=k+1}^{b_l} F^{E}_{\pi(k),\pi(k')} - \sum_{k=b_{l-1}+1}^{b_l-1} \sum_{k'=k+1}^{b_l} F^{E}_{\pi(k),\pi(k')} \right). \tag{8}$$

Now, Eq. (8) is rearranged as follows:

$$TH(\boldsymbol{\pi}) = \sum_{l=1}^{L} \sum_{k=b_{l-1}+1}^{b_l} \left(\sum_{k'=1}^{b_{l-1}} F^{E}_{\pi(k),\pi(k')} + \sum_{k'=k+1}^{b_l} F^{E}_{\pi(k),\pi(k')} \right) + \sum_{l=1}^{L} \sum_{k=b_{l-1}+1}^{b_l-1} \sum_{k'=k+1}^{b_l} \left(F^{A}_{\pi(k),\pi(k')} - F^{E}_{\pi(k),\pi(k')} \right). \tag{9}$$

Next, we introduce $\overline{TH}(\boldsymbol{\pi})$ and $TH'(\boldsymbol{\pi})$ to respectively replace the first and second terms of Eq. (9). In the following, we show that the first term in Eq. (9), i.e., $\overline{TH}(\boldsymbol{\pi})$ is constant.

$$\begin{aligned}
\overline{TH}(\boldsymbol{\pi}) &= \sum_{l=1}^{L} \sum_{k=b_{l-1}+1}^{b_l} \left(\sum_{k'=1}^{b_{l-1}} F^{E}_{\pi(k),\pi(k')} + \sum_{k'=k+1}^{b_l} F^{E}_{\pi(k),\pi(k')} \right) \\
&= \sum_{k=1}^{b_1} \sum_{k'=1}^{0} F^{E}_{\pi(k),\pi(k')} + \sum_{k=1}^{b_1} \sum_{k'=k+1}^{b_1} F^{E}_{\pi(k),\pi(k')} + \ldots + \sum_{k=b_{L-1}+1}^{M} \sum_{k'=1}^{b_{L-1}} F^{E}_{\pi(k),\pi(k')} + \sum_{k=b_{l-1}+1}^{M} \sum_{k'=k+1}^{M} F^{E}_{\pi(k),\pi(k')} \\
&= \sum_{k=1}^{M-1} \sum_{k'=k+1}^{M} F^{E}_{\pi(k),\pi(k')}.
\end{aligned} \tag{10}$$

As it can be seen in Eq. (10), $\overline{TH}(\boldsymbol{\pi})$ is independent of b_l. Hence, $\overline{TH}(\boldsymbol{\pi})$ is constant and $TH'(\boldsymbol{\pi})$ in objective function (3) can be expressed as:

$$\begin{aligned}
TH(\boldsymbol{\pi}) &= \overline{TH}(\boldsymbol{\pi}) + TH'(\boldsymbol{\pi}) \\
&= \sum_{k=1}^{M-1} \sum_{k'=k+1}^{M} F^{E}_{\pi(k),\pi(k')} + \sum_{l=1}^{L} \sum_{k=b_{l-1}+1}^{b_l-1} \sum_{k'=k+1}^{b_l} \left(F^{A}_{\pi(k),\pi(k')} - F^{E}_{\pi(k),\pi(k')} \right).
\end{aligned} \tag{11}$$

3.3. Unifying objective functions

In this section, a normalized weighted sum function, $TC(\boldsymbol{\pi})$, is suggested to unify objective functions (3) and (4). Let Ω denotes the set of constraints (5)–(7). Thus, the problem can be reformulated as follows:

$$\min_{\substack{\boldsymbol{\pi}\in\Pi \\ \Omega}} TC(\boldsymbol{\pi}) = \min_{\substack{\boldsymbol{\pi}\in\Pi \\ \Omega}} \left\{ \alpha \left(\frac{TH(\boldsymbol{\pi}) - TH_L}{TH_U - TH_L} \right) + (1-\alpha) \left(\frac{TS_U - TS(\boldsymbol{\pi})}{TS_U - TS_L} \right) \right\}, \tag{12}$$

where α ($0 \le \alpha \le 1$) is a weighting factor which presents the relative importance between the corresponding normalized objective functions. TH_L and TH_U are the lower and upper bounds of the total material handling cost,

ACCEPTED MANUSCRIPT

respectively. Also, TS_L and TS_U are the lower and upper bounds of the total similarity, respectively. These parameters are specified by the designer. To do so, the designer can solve a simple weighted sum problem as follows: $\min_{\substack{\boldsymbol{\pi}\in\Pi\\ \Omega}} \{TH(\boldsymbol{\pi}) - \varepsilon \times TS(\boldsymbol{\pi})\}$, where ε is a small enough number. By applying this objective function, less importance is given to the TS. As a consequence, the resulting solution, let denoted by TH^*_1 and TS^*_2, will be the lower bounds of the objective functions (i.e., $TH_L = TH^*_1$ and $TS_L = TS^*_1$). Similarly, the designer can solve problem $\min_{\substack{\boldsymbol{\pi}\in\Pi\\ \Omega}} \{\varepsilon \times TH(\boldsymbol{\pi}) - TS(\boldsymbol{\pi})\}$ to obtain an upper bound for them. In contrast to previous objective function, in this one less importance is paid to the TH. So, the resulting solution, let denoted by TH^*_2 and TS^*_2, will provide an upper bound for the objective functions (i.e., $TH_U = TH^*_2$ and $TS_U = TD^*_2$).

4. Suggested hybrid solution algorithm

There are two difficulties in solving the proposed problem (i.e., Eq. (12)). The first one is that the objective function of the problem is a non-linear non-smooth term. The second difficulty is in computational complexity of the problem. To overcome these difficulties, a hybrid solution method based on a combination of SA and DP is presented to effectively solve the problem. The basic idea of this solution method has been inspired from Chiang and Lee (2004). As mentioned earlier, they employed a hybrid SA/DP algorithm for solving the joint problem of CF and its layout assignment. In their problem, the objective was to minimize the inter-cell material flow cost. In this section, we extend their approach in order to make it capable to solve the proposed bi-objective problem.

In the suggested solution method, the SA is used to create feasible permutations of machines (i.e., $\boldsymbol{\pi}$) and the DP algorithm is implemented to obtain the optimal objective function of these permutations. To do so, we define $\alpha_1 = \alpha/(TH_U - TH_L)$ and $\alpha_2 = (1 - \alpha)/(TS_U - TS_L)$. Thus, Eq. (12) is rewritten as follows:

$$\begin{aligned}
\min_{\substack{\boldsymbol{\pi}\in\Pi\\ \Omega}} TC(\boldsymbol{\pi}) &= \min_{\substack{\boldsymbol{\pi}\in\Pi\\ \Omega}} \alpha_1\left(TH(\boldsymbol{\pi}) - TH_L\right) + \alpha_2\left(TS_U - TS(\boldsymbol{\pi})\right) \\
&\equiv \min_{\substack{\boldsymbol{\pi}\in\Pi\\ \Omega}} \left\{\alpha_1\left(\overline{TH}(\boldsymbol{\pi}) - TH'(\boldsymbol{\pi}) - TH_L\right) + \alpha_2\left(TS_U - TS(\boldsymbol{\pi})\right)\right\} \\
&\equiv \min_{\boldsymbol{\pi}\in\Pi} \left\{\alpha_1\left(\overline{TH}(\boldsymbol{\pi}) - TH_L\right) + \alpha_2 TS_U + \min_{\Omega}\left\{-\left(\alpha_1 TH'(\boldsymbol{\pi}) + \alpha_2 TS(\boldsymbol{\pi})\right)\right\}\right\}.
\end{aligned} \tag{13}$$

Since $\min(-z) = -\max z$, Eq. (13) becomes:

$$\min_{\boldsymbol{\pi}\in\Pi} TC(\boldsymbol{\pi}) = \min_{\boldsymbol{\pi}\in\Pi} \left\{\alpha_1\left(\overline{TH}(\boldsymbol{\pi}) - TH_L\right) + \alpha_2 TS_U - \max_{\Omega}\left\{\alpha_1 TH'(\boldsymbol{\pi}) + \alpha_2 TS(\boldsymbol{\pi})\right\}\right\}. \tag{14}$$

Finally, Eq. (14) is rewritten in the following form:

$$\min_{\boldsymbol{\pi}\in\Pi} TC(\boldsymbol{\pi}) = \min_{\boldsymbol{\pi}\in\Pi} \left\{\alpha_1\left(\sum_{k=1}^{M-1}\sum_{k'=k+1}^{M} F^E_{\pi(k),\pi(k')} - TH_L\right) + \alpha_2 TS_U - \psi^*(\boldsymbol{\pi})\right\}, \tag{15}$$

where Ω is the set of constraints (5)–(7) and $\Psi^*(\boldsymbol{\pi})$ is the optimum objective function value of the following optimization problem:

$$\psi^*(\boldsymbol{\pi}) = \max_{\Omega}\left\{\sum_{l=1}^{L}\sum_{k=b_{l-1}+1}^{b_l-1}\sum_{k'=k+1}^{b_l}\left[\alpha_1\left(F^E_{\pi(k),\pi(k')} - F^A_{\pi(k),\pi(k')}\right) + \alpha_2 S_{\pi(k),\pi(k')}\right]\right\}. \tag{16}$$

In the next section, we show that $\Psi^*(\boldsymbol{\pi})$ can be obtained by DP.

ACCEPTED MANUSCRIPT

4.1. Dynamic programming

By using DP, problem (16) can be sequentially solved as a partitioning problem in stages from 1 to $C^{\max}$. Let $g_l(b'_l,b_l)$ indicates the objective function value at stage l, when at this stage permutation $\boldsymbol{\pi}$ is partitioned from breaking node b'_l to breaking node b_l. Also, let $g^*_l(b_l)$ shows the optimum objective function value of breaking node b_l in stage l. By using forward recursion, the partitioning problem at stage l becomes:

$$g(b'_l,b_l) = g^*_{l-1}(b'_l) + \sum_{k=b'_l+1}^{b_l-1} \sum_{k'=k+1}^{b_l} \left[\alpha_1 \left(F^E_{\pi(k),\pi(k')} - F^A_{\pi(k),\pi(k')} \right) + \alpha_2 S_{\pi(k),\pi(k')} \right]. \tag{17}$$

Subject to:

$$\max\{l, M-(C^{\max}-l)NM\} \le b_l \le \min\{M, l \times NM\}, \forall l = 1,...,C^{\max}, \tag{18}$$

$$\max\{l-1, M-(C^{\max}-l+1)NM, b_l - NM\} \le b'_l \le \min\{M,(l-1)\times NM, b_l - 1\}, \forall l = 1,...,C^{\max}, \tag{19}$$

where $g^*_0(b'_1 = 0) = 0$ and $g^*_l(b_l) = \max_{\max\{l-1,M-(C^{\max}-l+1)NM,b_l-NM\} \le b'_l \le \min\{M,(l-1)\times NM,b_l-1\}} \{g_l(b'_l,b_l)\}$. Equation (17) is the forward recursive equation. Also, constraints (18) and (19) ensure that the cell size limit and the maximum allowable number of cells are not violated within partitioning procedure.

Note that, the DP partitions $\boldsymbol{\pi}$ into exactly $C^{\max}$ cells. Therefore, the optimum objective function value of DP, $\Psi^*(\boldsymbol{\pi})$, and the optimum number of cells, $L^*(\boldsymbol{\pi})$, for permutation $\boldsymbol{\pi}$ are obtained by Eqs. (20) and (21), respectively.

$$\psi^*(\boldsymbol{\pi}) = \min_{\{l|b_l=M\}} \{g^*_l(b_l)\}, \tag{20}$$

$$L^*(\boldsymbol{\pi}) = \arg\min_{\{l|b_l=M\}} \{g^*_l(b_l)\}. \tag{21}$$

Finally, the objective function value of permutation $\boldsymbol{\pi}$, is computed by:

$$TC(\boldsymbol{\pi}) = \alpha_1 \left(\sum_{k=1}^{M-1} \sum_{k'=k+1}^{M} F^E_{\pi(k),\pi(k')} - TH_L \right) + \alpha_2 TS_U - \psi^*(\boldsymbol{\pi}). \tag{22}$$

The pseudo code of the DP algorithm is given in Appendix A. In the worst case, when $NM = C^{\max} = M$, the time complexity of the DP algorithm is $O((M^3 + 5M)/6)$. This means that the DP is a polynomial algorithm. For instance, if we have 25 machines, the DP requires 2625 computations.

4.2. Simulated annealing

SA is a stochastic search method for solving combinatorial optimization problems which uses the idea of the annealing process of solid. It was independently proposed by Kirkpatrick et al. (1983) and Cerny (1985). In the annealing process, a solid is heated until it melts. Then, the temperature of the solid is slowly decreased by an appropriate annealing schedule until it reaches the lowest energy state or the ground state. As mentioned earlier, both the CF and layout problems are NP-hard. Thus, the proposed problem is also NP-hard (because it integrates the CF and layout problems). In recent years, the SA algorithm has been successfully applied by researchers for solving CF and layout problems (For instance, see (Xambre & Vilarinho, 2003; Chiang & Lee, 2004; Al-Araidah et al., 2006; Arkat et al., 2007; Wu et al., 2009; Paydar et al., 2010; Şahin et al., 2010; Şahin, 2011). The advantages of the SA over other meta-heuristic optimization techniques (such as GA and TS) include its ease of implementation, convergence properties, and ability to escape from local optima (Glover & Kochenberger, 2002). Also, it is an effective and robust algorithm which can find high-quality solutions that do not strongly depend on the choice of the initial solution. Furthermore, it has been proven that the computation

ACCEPTED MANUSCRIPT

time of the SA has a polynomial upper bound (Arats & Korst, 1989). These considerations motivated us to employ the SA algorithm to solve the problem. The details of the proposed methodology are explained below.

In Eq. (15), there is a solution space Π which contains every possible permutation of M machines (i.e., $M!$ permutations) and a real function $TC(\boldsymbol{\pi})$ which is acquired via DP. The purpose is to find a solution $\boldsymbol{\pi} \in \Pi$ such that minimizes $TC(\boldsymbol{\pi})$ over Π. For this purpose, the SA can be employed to perform the search process over Π.

The initial solution (permutation) is randomly generated. This solution corresponds to the layout sequence of machines in the S-shaped layout, i.e., $\boldsymbol{\pi}$. For instance, permutation $\boldsymbol{\pi}$ = (7, 10, 6, 12, 9, 3, 2, 4, 8, 11, 1, 5) corresponds to the layout sequence of machines shown in Fig. 1(b). After obtaining the coordinates of machines by using the algorithm presented in Appendix A, the DP algorithm is employed to partition permutation $\boldsymbol{\pi}$ into machine cells and obtain $\Psi^*(\boldsymbol{\pi})$ (see, Eq. (16)). Then, we can calculate the objective function value of permutation $\boldsymbol{\pi}$, $TC(\boldsymbol{\pi})$, from Eq. (22).

[Please Insert Fig 2 here]

The algorithm starts with an initial temperature T_0, then the temperature is gradually reduced by a cooling function. The *Geometric Decrement* function, $T_t = \theta \times T_{t-1}$, originally suggested by Kirkpatrick et al. (1983) is used as the cooling function. In this function, T_t is the temperature at t-th iteration and θ ($0 < \theta < 1$) is the cooling rate.

At each temperature (iteration), a generation process called *Move* is applied to convert the current permutation $\boldsymbol{\pi}'$, into a neighboring (new) permutation $\boldsymbol{\pi}$. To do so, three move operators, namely *Swap*, *Change* and *Invert* are applied. The *Swap* operator swaps the order of two randomly selected machines; the *Change* operator changes the order of a randomly selected machine; and the *Invert* operator reverses the order of machines between two randomly selected points. An example of these operators is given in Fig. 2.

Once a neighboring solution was created, the change in the objective function value is calculated by $\Delta = TC(\boldsymbol{\pi}) - TC(\boldsymbol{\pi}')$. In minimization problems, if the change in each move leads to an improvement in the objective function value (i.e., $\Delta < 0$), the new solution is accepted. Otherwise, the non-improving solution is accepted with a specified probability function $\exp(-\Delta/T_t)$. By accepting non-improving solutions, the SA algorithm can escape from a local minimum. This process is repeated N times at each iteration, where N is a control parameter and called the epoch length. These cooling and transition processes are repeated until the best solution is not improved after a specified number of transitions $I^{\max}$.

5. Computational results

A computer program was developed by Embarcadero® Delphi XE7 for the proposed Hybrid SA (HSA) which can be implemented in Microsoft Windows XP operating system or higher versions. In order to show the advantage of the S-shaped layout, this approach is compared with two recently developed approaches in the literature. However, before performing comparisons, we should tune the parameters of the algorithm.

It should be noted that all experiments are done on a personal computer having Windows 7 operating system, with Intel(R) Core(TM)2 Quad Q6600 2.4GHz CPU and 2GB Ram.

5.1. Parameters setting

It is well known that parameter values used in the SA algorithm may have significant influence on solution quality. Thus, parameter setting is very important. The parameters of the HSA are as follows: initial temperature (T_0), cooling rate (θ), epoch length (N) and maximum number of transitions ($I^{\max}$). The initial temperature can be chosen in such a way that the acceptance probability of non-improving transitions in the first iteration of the annealing process is 95%. In this relation, 100 pairs of solutions are randomly generated. Then, the value of T_0 is

ACCEPTED MANUSCRIPT

ACCEPTED MANUSCRIPT

calculated by $T_0 = \sum_{n=1}^{100} \left| TC(\boldsymbol{\pi}_n^1) - TC(\boldsymbol{\pi}_n^2) \right| \Big/ (-100 \times \ln(0.95))$, where $\boldsymbol{\pi}^1{}_n$ and $\boldsymbol{\pi}^2{}_n$ are two solutions that are randomly generated at *n*-th trial. The value of N can be set proportional to problem size; in this case the number of machines. So, we set $N = N' \times M$, where N' is a coefficient and M is the number of machines.

Six test problems adopted from Krishnan et al. (2012) were used in the parameter setting of the HSA. Four levels were considered for each parameter. These levels include: (0.8, 0.85, 0.9, 0.95) for θ, (30000, 40000, 50000, 60000) for I^{max}, and (4, 5, 6, 7) for N'. As SA is a stochastic search method, 10 independent runs were performed on each parameter combination (in total $6 \times 4^3 \times 10 = 3840$ runs were carried out). The computation results were analyzed using DOE toolbox in Minitab 16 statistical software. Finally, by considering a trade-off between solution quality and computation time, the following values were selected for the HSA parameters: $N' = 5$ (i.e., $N = 5M$), $\theta = 0.95$ and $I^{max} = 50000$.

5.2. Comparison to Krishnan et al. (2012)'s approach

In this section, we compare the S-shaped layout with the layout approach presented by Krishnan et al. (2012). They developed a method based on the flow between machines for obtaining machine cells. A GA-based machine placement procedure was employed for placing machines in a facility layout matrix (grid) as a QAP. They used six numerical examples from the literature in order to benchmark the developed procedure. These numerical examples are solved by the HSA and the results are compared with the solutions reported in Krishnan et al. (2012). The data sets of these examples are available in the reference paper. According to the solutions reported in (Krishnan et al. 2012), all machines are assumed equal-sized, i.e., $w_k = h_k = 1$, $\forall$ k. Both the vertical and horizontal aisle widths (L^X and L^Y) are set to 0. For all parts, the unit inter- and intra-cell material handling costs per unit distance (i.e., $c^E{}_{i,k,k'}$ and $c^A{}_{i,k,k'}$) are assumed 2 and 1, respectively. The similarity between machines ($S_{k,k'}$) is calculated according to Yule's similarity coefficient (see Table 1). The weighting factor, α, is set to 0.5. The other parameters, including the maximum number of cells, C^{max}, the maximum number of machines allowed in each cell, NM, and the width of the shop floor, W, are set according to the solutions given in Krishnan et al. (2012). The lower and upper bounds of objective functions (i.e., TH_L, TH_U, TS_L and TS_U) are determined in the same way as explained in Section 3.2. For each problem, the HSA is executed 30 times and the best result is considered for comparison. A summary of comparison is provided in Table 2. In this table, the values reported in columns 'Imp^1', 'Imp^2' and 'Imp^3' indicate the improvement percent in the TH, TS, and TC, respectively. These values are calculated by: $\text{Imp}^1 = (1 - TH^{\text{HSA}}/TH^{\text{K}}) \times 100$, $\text{Imp}^2 = (TS^{\text{HSA}}/TS^{\text{K}} - 1) \times 100$, and $\text{Imp}^3 = (1 - TC^{\text{HSA}}/TC^{\text{K}}) \times 100$. Also, the final solutions of the S-shaped layout as well as the solutions of Krishnan et al. (2012) are shown in Appendix C, Figs C.1−C.6.

[Please insert Table 2 here]

As it can be seen in Table 2, in problems 1, 2, 4 and 5, the solution of the proposed approach dominates the solution of Krishnan et al. (2012). For problem 6, the total material handling cost in the proposed approach is 29.16% better than that in Krishnan et al. (2012)'s approach. But, for this problem, the total similarity resulting from the proposed approach is 4.37% worse than that in Krishnan et al. (2012)'s approach. However, we can see that for this problem, the normalized objective function value of the proposed approach (considering $\alpha = 0.5$) is 98.23% better than that of Krishnan et al. (2012)'s approach. So, in problem 6, the solution of the proposed approach is preferred. In problem 3, we could not calculate the TC, because the deviations between the lower and upper bounds are zero. In this problem, the total similarity in the solution of the proposed approach is equal to that of Krishnan et al. (2012)'s approach. However, the material handling cost of proposed approach is 9.79% worse than that of Krishnan et al. (2012)'s approach. It should be noted that in these problems the width of the shop floor was set according to Krishnan et al. (2012)'s solutions. However, in problem 3, if we set $W = 2$ and add an artificial machine to the problem (in order to fill the empty location within the layout grid), the total material handling cost is obtained as 146150 units, which is 5.37% better than that calculated for Krishnan et al. (2012)'s solution. Unfortunately, Krishnan et al. (2012) did not mention anything about their computation times. However, it can be seen that the HSA is able to solve the largest problem (i.e., problem 6) in an average time of 6.05 seconds. Based on this comparison, it was shown that a special case of the proposed S-shaped layout is the

ACCEPTED MANUSCRIPT

ACCEPTED MANUSCRIPT

grid layout (or equivalently the QAP). Also, in most cases it was demonstrated that the proposed bi-objective problem with S-shaped layout can produce better solutions than Krishnan et al. (2012)'s approach.

5.3. Comparison to Mohammadi and Forghani (2014)'s approach

The second section of comparison is carried out between the S-shaped layout considered in this study, and the multi-row layout proposed by Mohammadi and Forghani (2014). They considered similar parameters such as machine dimensions and aisle width in the CMS layout. However, the type of layout is different. In their approach, each row represents a machine cell, and the layout of machines within the cells is assumed linear. For simplicity, they did not take the width of the shop floor into consideration. It needs to be mentioned that they employed a GA for solving their problem. Six numerical examples are used in this section in order to compare these approaches. As the data sets of these examples are not available in the reference paper, we present them in Appendix B, Tables B.1−B.12. These data sets include the dimension of machines (w_k and h_k) the operation sequences of parts (that are used to calculate $f_{i,k,k'}$), and the demand of parts (d_k). In all problems, the unit inter- and intra-cell material handling costs per unit distance ($c^E{}_{i,k,k'}$ and $c^A{}_{i,k,k'}$) were assumed 0.15 and 0.1, respectively. Also, the vertical and horizontal aisle widths (L^Y and L^X) are 3 and 1.5 units, respectively. For each problem, the width of the shop floor (W) is set equal to the maximum length of the cells in Mohammadi and Forghani (2014)'s solution. The other parameters, including, the maximum number of cells ($C^{\max}$) and the maximum number of machines allowed in each cell (NM) are taken from the reference paper. The weighting factor (α) is assumed 0.5. The similarity between machines ($S_{k,k'}$) is calculated according to Yule's similarity coefficient (see Table 1). After obtaining the lower and upper bounds of objective functions (i.e., TH_L, TH_U, TS_L and TS_U) using the method explained in Section 3.2, each problem is solved 30 times by the HSA and the best result is considered for comparison. A summary of comparison is given in Table 3. In this table, the values given in the columns 'Imp^1', 'Imp^2' and 'Imp^3', respectively, indicate the improvement percent in the TH, TS and TC when α = 0.5. These values are calculated by: Imp^1 = $(1 - TH^{\text{HSA}}/TH^{\text{M\&F}}) \times 100$, Imp^2 = $(TS^{\text{HSA}}/TS^{\text{M\&F}} - 1) \times 100$ and Imp^3 = $(1 - TC^{\text{HSA}}/TC^{\text{M\&F}}) \times 100$. Also, the values given in the column 'Imp^4' show the improvement percent in the total material handling cost when α = 1 (i.e., only the total material handling cost is minimized). These values are also calculated by Imp^4 = $(1 - TH_L/TH^{\text{M\&F}}) \times 100$. The final solutions of the S-shaped layout as well as Mohammadi and Forghani (2014)'s solutions are shown in Appendix C, Figs C.7−C.12.

[Please insert Table 3 here]

From Table 3, columns 'Imp1' and 'Imp2', we can see that except for one case (i.e., problem 8), in the remaining problems the HSA solution dominates the solution obtained by Mohammadi and Forghani (2014). According to the results, we can see that applying the S-shaped layout on these problems (except for problem 8) leads to better material handling cost in comparison with the multi-row layout. When α = 1, (i.e., only the material handling cost is minimized), this improvement is even further (see column 'Imp4' in Table 3). For instance, in problem 10 the total material handling cost for α = 0.5 is 5612.50 units, while this value for α = 1 (i.e., TH_L) is 5556.25 units. Also, the total similarity in the solutions of the suggested approach is better than (or at least the same as) that of Mohammadi and Forghani (2014)'s approach. In problem 8, the material handling cost resulting from the suggested approach is just 0.04% worse than that of Mohammadi and Forghani (2014)'s approach, which is negligible. The result of computation times shows that the HSA consumes slightly more CPU time than the GA developed by Mohammadi and Forghani (2014). However, the HSA can solve the problems in a reasonable amount of time. For instance, in the worst case (i.e., problem 11), the average CPU time of the HSA is almost 3.38 seconds.

6. Conclusions and directions for further research

In this research, an integrated bi-objective CF and layout problem considering a new layout framework based on S-shaped layout was presented. In this problem, we attempted to minimize the total material handling cost and maximize the total similarity between machines. A normalized weighted sum method was proposed to unify

ACCEPTED MANUSCRIPT

these objective functions. Due to computational complexity of the problem, a hybrid solution method combining the SA and DP algorithms was developed to solve large-sized problems in a reasonable computational time. In the proposed solution method, which is called HSA, partial solutions (permutations) representing the layout of machines on the shop floor are created by the SA. Then, each permutation is partitioned into machine cells using the DP. After setting the HSA parameters, the suggested approach was compared with two recently developed approaches in the literature. In the first section of comparison, we compared the S-shaped layout with the grid layout presented by Krishnan et al. (2012). Six numerical examples were solved and the solutions were compared with those obtained by Krishnan et al. (2012). In five examples the suggested bi-objective problem with S-shaped layout gave better solutions than Krishnan et al. (2012)'s approach. This comparison also indicated that a special case of the proposed S-shaped layout is the grid layout (or equivalently QAP). The second section of comparison was carried out between the S-shaped layout and the multi-row layout proposed by Mohammadi and Forghani (2014). Six numerical examples were solved by the HSA and the solutions were compared with Mohammadi and Forghani (2014)'s solutions. The results indicated that in five examples, the suggested approach gave better solutions than Mohammadi and Forghani (2014)'s approach.

Although the HSA has good performance in solving the proposed bi-objective problem, it also has some potential limitations that should be mentioned. First, as we applied a weighted some method for converting the bi-objective problem into a single objective problem, it was easy to solve the partitioning problem using the DP (see subsection 4.1). However, if we had more than two objectives, it became difficult to obtain Pareto solutions using the DP. Second, the SA algorithm is not efficient in solving problems with continues variables or many constraints. In addition, the SA algorithm is not suitable for parallel computing. These limitations of the SA and DP algorithms should be taken into consideration in later studies.

The development of the S-shaped layout and the hybrid solution method in this paper also results in some directions for future research. First, it would be interesting to extend the proposed problem to include alternative process routings and machine capacity constraints. So, the development of an efficient solution method for addressing this issue seems necessary. Second, the proposed problem can be investigated in uncertain and dynamic situations. In this regard, the demand of parts plays an important role in the problem. To deal with this issue, one approach is to develop a dynamic problem in which the demand of parts is known in each period. In such circumstances, the objective could be minimization of the total material handling cost plus the rearrangement cost of layout (due to the variability of part demands in different periods). The other approach is to use two-stage or multi-stage stochastic programming methods to take the uncertainty into consideration. Finally, it would be interesting to integrate the proposed problem with scheduling problem. So, other objectives like minimization of the total tardiness or minimization of the makespan could be involved in the problem.

Acknowledgements

This research was supported by the Research Affairs of Kharazmi University under the grant No. 4/2757 and this is gratefully acknowledged.

References

Aktürk, M.S., & Turkcan, A. (2000). Cellular manufacturing system design using a holonistic approach. *International Journal of Production Research, 38*, 2327–2347.

Al-Araidah, O., Krishnamurthy, A., & Malmborg, C.J. (2006). A two-stage simulated annealing procedure for block layout problems. *International Journal of Production Research, 44*, 4417–4429.

Arats, E., & Korst, J. (1989). *Simulated Annealing and Boltzan Machnies*. New York: Wiley.

Arıkan, F., & Güngör, Z. (2009). Modeling of a manufacturing cell design problem with fuzzy multi-objective parametric programming. *Mathematical & Computer Modelling, 50*, 407–420.

Arkat, J., Saidi, M., & Abbasi, B. (2007). Applying simulated annealing to cellular manufacturing system design. *International Journal of Advanced Manufacturing Technology, 32*, 531–536.

ACCEPTED MANUSCRIPT

Benjaafar, S. (2002). Modeling and analysis of congestion in the design of facility layouts. *Management Science, 48*, 679–704.

Cerny, V. (1985). A thermodynamical approach to the travelling salesman problem: an efficient simulation algorithm. *Journal of Optimization Theory & Applications, 45*, 41–51.

Chan, F.T.S., Lau, K.W., Chan, P.L.Y., & Choy, K.L. (2006). Two-stage approach for machine-part grouping and cell layout problems. *Robotics & Computer-Integrated Manufacturing, 22*, 217–238.

Chang, C.C., Wu, T.H., & Wu, C.W. (2013). An efficient approach to determine cell formation, cell layout and intracellular machine sequence in cellular manufacturing systems. *Computers & Industrial Engineering, 66*, 438–450.

Chiang, C.P., & Lee, S.D. (2004). Joint determination of machine cells and linear intercell layout. *Computers & Operations Research, 31*, 1603–1619.

Forghani, k., Mohammadi, M., Ghezavati, V. (2014). Integrated cell formation and layout problem considering multi-row machine arrangement and continuous cell layout with aisle distance. *International Journal of Advanced Manufacturing Technology, 78*, 687–705.

Garey, M.R., & Johnson, D.S. (1979). *Computers & intractability: a guide to the theory of NP-completeness*. San Francisco: Freeman.

Ghosh, T., Doloi, B., Dan, P.K. (2015). An Immune Genetic algorithm for inter-cell layout problem in cellular manufacturing system. *Production Engineering*, doi: 10.1007/s11740-015-0645-4.

Glover, F., & Kochenberger, G.A. (2002). *Handbook of metaheuristics*. New Jersey: Kluwer Academic Publishers.

Heragu, S.S., & Kakuturi, S.R. (1997). Grouping and placement of machine cells. *IIE Transactions, 29*, 561–571.

Javadi, B., Jolai, F., Slomp, J., Rabbani, M., & Tavakkoli-Moghaddam, R. (2014). A hybrid electromagnetism-like algorithm for dynamic inter/intra-cell layout problem. *International Journal of Computer Integrated Manufacturing, 27*, 501–518.

Jolai, F., Taghipour, M., Javadi, B. (2011). A variable neighborhood binary particle swarm algorithm for cell layout problem, *International Journal of Advanced Manufacturing Technology, 55*, 327–339.

Jolai, F., Tavakkoli-Moghaddam, R., Golmohammadi, A., & Javadi, B. (2012). An Electromagnetism-like algorithm for cell formation and layout problem. *Expert Systems with Applications, 39*, 2172–2182.

Kirkpatrick, S., Gelatt, C.D., & Vecchi, M.P. (1983). Optimization by simulated annealing. *Science, 220*, 671–680.

Krishnan, K.K., Mirzaei, S., Venkatasamy, V., & Pillai, V.M. (2012). A comprehensive approach to facility layout design and cell formation. *International Journal of Advanced Manufacturing Technology, 59*, 737–753.

Lee, S.D., & Chiang, C.P. (2001). A cut-tree-based approach for clustering machine cells in the bidirectional linear flow layout. *International Journal of Production Research, 39*, 3491–3512.

Leung, M.T., Quintana, R., & Chen, A.S. (2008). A Paradigm for Group Technology Cellular Layout Planning in JIT Facility. *Proceedings of the 2008 IEEE International Conference on Industrial Engineering and Engineering Management*, 1174–1178.

Mahdavi, I., Shirazi, B., & Paydar, M.M. (2008). A flow matrix-based heuristic algorithm for cell formation and layout design in cellular manufacturing system. *International Journal of Advanced Manufacturing Technology, 39*, 943–953.

Mohammadi, M., & Forghani, K. (2014). A novel approach for considering layout problem in cellular manufacturing systems with alternative processing routings and subcontracting approach. *Applied Mathematical Modelling, 38*, 3624–3640.

Mungwattana, A. (2000). *Design of cellular manufacturing systems for dynamic and uncertain production requirements with presence of routing flexibility*. Virginia: Ph.D. Thesis submitted to the Faculty of the Virginia Polytechnic Institute and State University.

Papaioannou, G., & Wilson, J.M. (2010). The evolution of cell formation problem methodologies based on recent studies (1997–2008): review and directions for future research. *European Journal of Operational Research, 206*, 509–521.

ACCEPTED MANUSCRIPT

ACCEPTED MANUSCRIPT

Paydar, M.M., Mahdavi, I., Sharafuddin, I., & Solimanpur, M. (2010). Applying simulated annealing for designing cellular manufacturing systems using MDmTSP. *Computers & Industrial Engineering, 59*, 929–936.

Şahin, R. (2011). A simulated annealing algorithm for solving the bi-objective facility layout problem. *Expert Systems with Applications, 38,* 4460–4465.

Şahin, R., Ertoğral, K., & Türkbey, 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. (2004). A heuristic to minimize makespan of cell scheduling problem. *International Journal of Production Economics, 88*, 231–241.

Tompkins, J.A., White, J.A., Bozer, Y.A., & Tanchoco, J.M.A. (2003). *Facilities planning*. (3rd. ed). New York: John Wiley & Sons.

Wemmerlöv, U., & Hyer, N.L. (1986). Procedures for the part family/machine group identification problem in cellular manufacture. *Journal of Operations Management, 6*, 125–147.

Wu, T.H., Chung, S.H., & Chang, C.C. (2009). Hybrid simulated annealing algorithm with mutation operator to the cell formation problem with alternative process routings. *Expert Systems with Applications, 36*, 3652–3661.

Wu, X., Chu, C.H., Wang, Y., & Yan, W. (2007). A genetic algorithm for cellular manufacturing design and layout. *European Journal of Operational Research, 181*, 156–167.

Xambre, A.R., & Vilarinho, P.M. (2003). A simulated annealing approach for manufacturing cell formation with multiple identical machines. *European Journal of Operational Research, 151*, 434–446.

Yin, Y., & Yasuda, K. (2005). Similarity coefficient methods applied to the cell formation problem: a comparative investigation. *Computers & Industrial Engineering, 48*, 471–489.

Yin, Y., Yasuda, K., & Hu, L. (2005). Formation of manufacturing cells based on material flows. *International Journal of Advanced Manufacturing Technology, 27*, 159–165.

ACCEPTED MANUSCRIPT

Appendix A. Pseudo code of the procedures used in the paper.

Pseudo code of the procedure for calculating $x_k(\boldsymbol{\pi})$ and $y_k(\boldsymbol{\pi})$.
Variables
— k, k', k'': integer;
— $h^{\text{comulative}}$, $w^{\text{comulative}}$,$h^{\max}$,w^{adjust}: real;
— layout_direction ∈ {forward, backward}
Label next_machine, next_row, stop;
— layout_direction ← forward;
— $h^{\max} \leftarrow 0$;
— $h^{\text{comulative}} \leftarrow -L^Y$;
— $k' \leftarrow 0$;
— next_row:
— $h^{\text{comulative}} \leftarrow h^{\text{comulative}} + h^{\max} + L^Y$;
— $k \leftarrow k' + 1$;
— $k' \leftarrow k$;
— $h^{\max} \leftarrow h_{\pi(k)}$;
— $w^{\text{comulative}} \leftarrow w_{\pi(k)}$;
— next_machine:
— **If** ($k' + 1 \leq M$) **and** ($w^{\text{comulative}} + L^X + w_{\pi(k'+1)} \leq W$) **then**
—— $k' \leftarrow k' + 1$;
—— $w^{\text{comulative}} \leftarrow w^{\text{comulative}} + L^X + w_{\pi(k')}$;
—— **If** $h^{\max} < h_{\pi(k')}$ **then**
——— $h^{\max} \leftarrow h_{\pi(k')}$;
—— **End IF**
—— **Goto** next_machine
— **Else**
—— **If** layout_direction = forward **then**
——— $x_{\pi(k')}(\boldsymbol{\pi}) \leftarrow (W - w^{\text{comulative}})/2 + w_{\pi(k')}$;
——— $y_{\pi(k')}(\boldsymbol{\pi}) \leftarrow h^{\text{comulative}} + h^{\max}/2$;
——— **For** $k'' \leftarrow k + 1$ **to** k' **do**
———— $x_{\pi(k'')}(\boldsymbol{\pi}) \leftarrow x_{\pi(k''-1)}(\boldsymbol{\pi}) + (w_{\pi(k''-1)} + w_{\pi(k'')}) / 2 + L^X$;
———— $y_{\pi(k'')}(\boldsymbol{\pi}) \leftarrow h^{\text{comulative}} + h^{\max} / 2$;
——— **End For**;
——— layout_direction ← backward
—— **Else**
——— $x_{\pi(k')}(\boldsymbol{\pi}) \leftarrow (W - w^{\text{comulative}})/2 + w_{\pi(k')}$;
——— $y_{\pi(k')}(\boldsymbol{\pi}) \leftarrow h^{\text{comulative}} + h^{\max}/2$;
——— **For** $k'' \leftarrow k' - 1$ **downto** k **do**
———— $x_{\pi(k'')}(\boldsymbol{\pi}) \leftarrow x_{\pi(k''+1)}(\boldsymbol{\pi}) + (w_{\pi(k''+1)} + w_{\pi(k'')})/2 + L^X$;
———— $y_{\pi(k'')}(\boldsymbol{\pi}) \leftarrow h^{\text{comulative}} + h^{\max}/2$;
——— **End For**;
——— layout_direction ← forward;
—— **End IF**;
—— **If** $k' = M$ **then**
——— **Goto** stop
—— **Else**
——— **Goto** next_row;
—— **End IF**;
— **End IF**;

ACCEPTED MANUSCRIPT

ACCEPTED MANUSCRIPT

— Stop:
End;
Pseudo code of the DP for Calculating $\Psi^{*}(\boldsymbol{\pi})$ and $L^{*}(\boldsymbol{\pi})$.
— $\Psi^{*}(\boldsymbol{\pi}) \leftarrow -\infty$;
— **For** $l = 1$ **to** $C^{\max}$ **do**
—— **For** $b_l = \max\{l, M - (C^{\max} - 1)NM\}$ **to** $\min\{M, l \times NM\}$ **do**
——— $g^{*}_{l}(k) \leftarrow -\infty$;
——— **For** $b'_l = \max\{l-1, M - (C^{\max} - l + 1)NM, b_l - NM\}$ **to** $\min\{M, (l-1)NM, b_l - 1\}$ **do**
———— **If** $g_l^{*}(b_l) < g_{l-1}^{*}(b_l') + \sum_{k=b_l'+1}^{b_l-1}\sum_{k'=k+1}^{b_l}\left[\alpha_1\left(F^{E}_{\pi(k),\pi(k')} - F^{A}_{\pi(k),\pi(k')}\right) + \alpha_2 S_{\pi(k),\pi(k')}\right]$ **then**
————— $g_l^{*}(b_l) \leftarrow g_{l-1}^{*}(b_l') + \sum_{k=b_l'+1}^{b_l-1}\sum_{k'=k+1}^{b_l}\left[\alpha_1\left(F^{E}_{\pi(k),\pi(k')} - F^{A}_{\pi(k),\pi(k')}\right) + \alpha_2 S_{\pi(k),\pi(k')}\right]$;
———— **End If**;
——— **End For**;
——— **If** $b_l = M$ **and** $\Psi^{*}(\boldsymbol{\pi}) < g^{*}_{l}(b_l)$ **then**
———— $\Psi^{*}(\boldsymbol{\pi}) \leftarrow g^{*}_{l}(b_l)$;
———— $L^{*}(\boldsymbol{\pi}) \leftarrow l$;
——— **End If**;
—— **End For**;
— **End For**;
End;

ACCEPTED MANUSCRIPT

Pseudo code of the HSA for obtaining $TC(\boldsymbol{\pi}^*)$.
— **Set** (T_0, N, $I^{\max}$, θ); — $T \leftarrow T_0$;
— $t \leftarrow 1$;
— Generate a random permutation $\boldsymbol{\pi}'$;
— $TC(\boldsymbol{\pi}') \leftarrow \alpha_1 \left(\sum_{k=1}^{M-1} \sum_{k'=k+1}^{M} F^E_{\pi'(k),\pi'(k')} - TH_L \right) + \alpha_2 TS_U - \psi^*(\boldsymbol{\pi}')$;
— $\boldsymbol{\pi}^* \leftarrow \boldsymbol{\pi}'$;
— $TC(\boldsymbol{\pi}^*) \leftarrow TC(\boldsymbol{\pi}')$;
— **Repeat**
—— $n \leftarrow 0$;
—— **Repeat**
——— $m \leftarrow$ m + 1;
——— **Case** Random {0, …, 6} **of**
———— 0: $\boldsymbol{\pi} \leftarrow$ Swap ($\boldsymbol{\pi}'$);
———— 1: $\boldsymbol{\pi} \leftarrow$ Change ($\boldsymbol{\pi}'$);
———— 2: $\boldsymbol{\pi} \leftarrow$ Invert ($\boldsymbol{\pi}'$);
———— 3: $\boldsymbol{\pi} \leftarrow$ Swap and Change ($\boldsymbol{\pi}'$);
———— 4: $\boldsymbol{\pi} \leftarrow$ Swap and Invert ($\boldsymbol{\pi}'$);
———— 5: $\boldsymbol{\pi} \leftarrow$ Change and Invert ($\boldsymbol{\pi}'$);
———— 6: $\boldsymbol{\pi} \leftarrow$ Swap, Change and Invert ($\boldsymbol{\pi}'$);
——— **End Case**;
——— $TC(\boldsymbol{\pi}) \leftarrow \alpha_1 \left(\sum_{k=1}^{M-1} \sum_{k'=k+1}^{M} F^E_{\pi(k),\pi(k')} - TH_L \right) + \alpha_2 TS_U - \psi^*(\boldsymbol{\pi})$;
——— $\Delta \leftarrow TC(\boldsymbol{\pi}) - TC(\boldsymbol{\pi}')$;
——— **If** ($\Delta < 0$) or (Uniform(0, 1) $\leq \exp(-\Delta/T)$) **then**
———— $\boldsymbol{\pi}' \leftarrow \boldsymbol{\pi}$;
———— $TC(\boldsymbol{\pi}') \leftarrow TC(\boldsymbol{\pi})$;
———— $n \leftarrow n + 1$;
———— **If** $TC(\boldsymbol{\pi}) < TC(\boldsymbol{\pi}^*)$ **then** ————— $t \leftarrow 0$;
————— $\boldsymbol{\pi}^* \leftarrow \boldsymbol{\pi}$;
————— $TC(\boldsymbol{\pi}^*) \leftarrow TC(\boldsymbol{\pi})$;
———— **End If**;
——— **End If**;
—— **Until** $n = N$;
—— $t \leftarrow t + 1$;
—— $T \leftarrow \theta \times T$;
—**Until** $t \geq I^{\max}$; **End**;

Appendix B. Data set of the numerical examples selected form Mohammadi and Forghani (2014).

[Please insert Tables B.1−B.12 here]

Appendix C. Solutions of the numerical examples selected from Krishnan et al. (2012) & Mohammadi and Forghani (2014).

[Please insert Figs. C.1−C.12 here]

ACCEPTED MANUSCRIPT

Table 1. Definitions and ranges of twenty well-known similarity coefficients considered in Yin and Yasuda (2005)

No.	Coefficient name	Definition, $S_{k,k'}$	Range
1	Jaccard	$a/(a+b+c)$	0–1
2	Hamann	$[(a+d)-(b+c)]/[(a+d)+(b+c)]$	–1 to 1
3	Yule	$(ad-bc)/(ad+bc)$	–1 to 1
4	Simple matching	$(a+d)/(a+b+c+d)$	0–1
5	Sorenson	$2a/(2a+b+c)$	0–1
6	Rogers and Tanimoto	$(a+d)/[a+2(b+c)+d]$	0–1
7	Sokal and Sneath	$2(a+d)/[2(a+d)+b+c]$	0–1
8	Rusell and Rao	$a/(a+b+c+d)$	0–1
9	Baroni-Urbani and Buser	$[a+(ad)^{1/2}]/[a+b+c+(ad)^{1/2}]$	0–1
10	Phi	$(ad-bc)/[(a+b)(a+c)(b+d)(c+d)^{1/2}]$	–1 to 1
11	Ochiai	$a/[(a+b)(a+c)^{1/2}]$	0–1
12	PSC	$a^2/[(b+a)(c+a)]$	0–1
13	Dot-product	$a/(b+c+2a)$	0–1
14	Kulczynski	$1/2[a/(a+b)+a/(a+c)]$	0–1
15	Sokal and Sneath 2	$a/[a+2(b+c)]$	0–1
16	Sokal and Sneath 4	$1/4[a/(a+b)+a/(a+c)+d/(b+d)+d/(c+d)]$	0–1
17	Relative matching	$[a+(ad)^{1/2}]/[a+b+c+d+(ad)^{1/2}]$	0–1
18	Chandrasekharan and Rajagopalan	$a/\min[(a+b),(a+c)]$	0–1
19	MaxSC	$\max[a/(a+b),a/(a+c)]$	0–1
20	Baker and Maropoulos	$a/\max[(a+b),(a+c)]$	0–1

a: the number of parts visit both machines; *b*: the number of parts visit machine *k* but not *k′*; *c*, the number of parts visit machine *k′* but not *k*; *d*, the number of parts visit neither machine.

Table 2. Comparison between the S-shaped and Grid layouts

Problem					Lower and upper bounds				S-shaped layout				Krishnan et al. (2012)'s approach (Grid layout)					
#	Size ($M \times P$)	C^{max}	NM	W	TH_L	TH_U	TS_L	TS_U	TH^{HSA}	TS^{HSA}	TC^{HSA}	t^{HSA} (s)*	TH^K	TS^K	TC^K	Imp[1] (%)	Imp[2] (%)	Imp[3] (%)
1	8 × 10	3	3	3	18890	20215	1.24	4.59	18890	1.24	0.50	0.20	19630	1.24	0.78	3.77	0.00	35.84
2	13 × 13	3	5	4	27470	31790	17.63	18.41	28570	17.99	0.40	0.68	36240	17.99	1.29	21.16	0.00	69.02
3	15 × 25	4	4	4	166600	166600	18.06	18.06	166600	18.06	-	0.75	151750	18.06	-	-9.79	0.00	-
3†	15 × 25	4	4	2	146150	146150	18.06	18.06	146150	18.06	-	0.79	151750	18.06	-	5.37	0.00	-
4	16 × 43	5	6	4	199861	278075	2.50	17.33	210933	11.73	0.26	1.36	283938	7.58	0.87	25.71	54.79	70.05
5	25 × 40	8	6	5	19340	22840	13.85	27.01	19640	23.70	0.17	4.86	29655	21.62	1.68	33.77	9.62	89.95
6	30 × 30	8	5	5	112470	113345	38.33	42.62	112920	40.76	0.47	6.05	159395	42.62	26.81	29.16	-4.37	98.23

* Average CPU time in 30 runs of the HSA
† Alternative solution for problem 3 considering $W = 2$

Table 3
Comparison between the S-shaped and multi-rows layouts

Problem					Lower and upper bounds				S-shaped layout				Mohammadi and Forghani (2014)'s layout (multi-rows layout)							
#	Size ($M \times P$)	C^{max}	NM	W	TH_L	TH_U	TS_L	TS_U	TH^{HSA}	TS^{HSA}	TC^{HSA}	t^{HSA} (s)*	$TH^{M\&F}$	$TS^{M\&F}$	$TC^{M\&F}$	t^{GA} (s)†	Imp[1] (%)	Imp[2] (%)	Imp[3] (%)	Imp[4] (%)
7	8 × 20	3	4	17	3453.75	3691.25	4.36	7.39	3453.75	4.36	0.50	0.24	3461.25	4.36	0.52	0.20	0.22	0.00	3.06	0.22
8	15 × 30	3	5	21.5	9114.89	9114.89	29.90	29.90	9114.89	29.90	0.00	0.75	9111.29	29.90	0.00	0.65	-0.04	0.00	0.00	-0.04
9	16 × 30	3	7	22	7230.00	8106.25	1.41	25.50	7316.25	13.89	0.29	1.30	7731.25	2.73	0.76	0.64	5.37	408.89	61.75	6.48
10	20 × 20	5	5	20.5	5556.25	5882.50	22.02	24.54	5612.50	23.45	0.30	1.75	5825.00	4.83	4.32	0.25	3.65	385.76	93.02	4.61
11	24 × 40	7	5	22	6825.94	7317.03	28.59	34.19	6851.94	32.59	0.17	3.38	7092.29	25.37	1.06	1.66	3.39	28.47	84.05	3.76
12	25 × 40	7	4	17.5	10021.25	11543.75	16.76	24.74	10316.25	21.23	0.32	2.30	10455.00	18.75	0.52	1.75	1.33	13.21	38.76	4.15

* Average CPU time in 30 runs of the HSA
† Average CPU time in 30 runs of the GA proposed by Mohammadi and Forghani (2014)

ACCEPTED MANUSCRIPT

Table B.1
Processing information of parts for problem 7

Part #	Operation sequence					Part #	Operation sequence				
1	6	5				11	7	3	1		
2	1	3				12	5	7	6		
3	2	1	7	8	5	13	1	3			
4	2	4	7	8		14	1	2	3		
5	6	5				15	5	6			
6	2	4	7	8	5	16	1	3			
7	8	4	7	2		17	3	5	1		
8	1	3				18	4	2	8	7	
9	1	6	3			19	1	3			
10	5	3	4			20	4	2	6	7	8

$d_i = 100$, $c^A_{i,k,k'} = 0.1$ and $c^E_{i,k,k'} = 0.15$ for i, k, k'.

Table B.2
Machine dimensions for problem 7

Machine #	1	2	3	4	5	6	7	8
width (w_k)	4	3.5	2.5	4	2	2	3	4
length (h_k)	2	3	3	2	4	3.5	3	4

Table B.3
Processing information of parts for problem 8

Part #	Operation sequence						Part #	Operation sequence						Part #	Operation sequence				
1	2	3	7	10	11		11	1	2	3	7	10		21	11	12	13	14	
2	4	5	6	8	9		12	5	6	8	9			22	4	5	6	8	9
3	1	2	3	7	10	11	13	4	11	12	13	14	15	23	11	12	13	14	15
4	3	11	12	13	14	15	14	4	5	6	8	9		24	11	12	13	14	
5	4	5	6	8	9		15	4	11	12	13	14	15	25	11	12	13	14	15
6	1	2	3	7	10		16	3	4	5	6	8	9	26	4	5	6	8	9
7	1	2	3	7	10		17	1	2	3	7	10		27	4	5	6	8	9
8	1	2	3	7	10		18	4	5	6	8	9		28	11	12	13	14	15
9	1	2	3	7	10		19	1	2	3	7	10		29	11	12	13	14	15
10	1	2	3	7	10		20	4	5	6	8	9		30	11	12	13	14	15

$d_i = 100$, $c^A_{i,k,k'} = 0.1$ and $c^E_{i,k,k'} = 0.15$ for i, k, k'.

Table B.4
Machine dimensions for problem 8

Machine #	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
width (w_k)	3	3	4	3.5	2	2	4	4	3	2.5	4	3	2.5	3	3
length (h_k)	2.5	3	4	2	2.5	2	2.5	2	4	3.5	4	4	2	2	2

ACCEPTED MANUSCRIPT

Table B.5
Processing information of parts for problem 9

Part #	Operation sequence	Part #	Operation sequence	Part #	Operation sequence
1	2	11	5 10 11 14	21	5 10 14
2	4 7 8 11 12	12	1 7 8 11 12	22	1 4 7 8 12
3	5 13	13	3 13	23	3 6 9 15
4	1 2 3 4 7 8 12	14	5 10 14 16	24	5 10 16
5	1 9 15	15	5 10 14 16	25	3 6 7 9 15
6	5 10 14 16	16	2	26	5 10 15 16
7	1 4 7 8 11 12 16	17	14	27	3 6 9 14 15 16
8	5 10 14 16	18	1 2 4 7 8 10 11 12	28	3 8 9
9	1 4 8 11	19	9 12	29	3 6 13 15
10	2 13	20	2 12 13	30	1 4 7 11 13 15

$d_i = 100$, $c^A_{i,k,k'} = 0.1$ and $c^E_{i,k,k'} = 0.15$ for i, k, k'.

Table B.6
Machine dimensions for problem 9

Machine #	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
width (w_k)	3.5	2	4	3	3.5	2.5	3.5	3	3.5	4	3	2	3.5	3	3	2.5
length (h_k)	2.5	3	2	3.5	2	3	2	2	3.5	3.5	2	2.5	2.5	2	4	3.5

Table B.7
Processing information of parts for problem 10

Part #	Operation sequence	Part #	Operation sequence
1	12 1 9 18 20	11	11 14 3
2	11 3 2	12	9 18 5 12 1
3	8 20 19	13	6 7 15 17
4	3 11 2 10	14	8 10 1 2
5	4 15 6 7	15	13 14 16 17
6	11 14 16 17 5	16	15 7 6 18
7	5 16 17	17	9 1 12
8	15 13 7 9 4	18	8 19 20 10
9	18 9 11 1 12	19	3 2 11 5
10	19 20 8	20	18 10 1 12

$d_i = 100$, $c^A_{i,k,k'} = 0.1$ and $c^E_{i,k,k'} = 0.15$ for i, k, k'.

Table B.8
Machine dimensions for problem 10

Machine #	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
width (w_k)	4	2	2.5	2	3.5	4	4	3.5	2.5	4	3	2	3.5	2.5	3	4	3.5	3.5	4	3.5
length (h_k)	2	3.5	3	4	3	4	4	2	3	2	3	3.5	3	2	3.5	4	4	3	2.5	2

ACCEPTED MANUSCRIPT

ACCEPTED MANUSCRIPT

Table B.9
Processing information of parts for problem 11

Part #	Demand (d_i)	Operation sequence	Part #	Demand (d_i)	Operation sequence	Part #	Demand (d_i)	Operation sequence
1	155	1 13 21 22	15	100	3 20	29	125	9 17
2	160	3 20 24	16	65	1 13 21 22	30	135	6 8 12 18
3	135	7 14 23 24	17	85	1 13 21 22	31	65	3 20 17
4	150	6 8 12 15 18	18	125	6 8 12 15 18	32	90	7 14 23 24 16
5	210	6 8 12 15 18	19	102	4 16	33	100	1 13 21 22 2
6	230	9 10 17	20	105	2 5 11 19 21	34	90	3 20
7	85	9 10 17	21	75	4 16	35	120	5 11 19 21
8	90	4 16	22	100	2 5 11 19 21	36	130	5 11 19 21
9	95	1 13 21 22	23	140	3 20	37	145	16 15
10	86	2 5 11 19 21	24	62	3 20 12	38	250	4 16
11	55	3 20	25	85	7 14 23	39	60	4 16
12	120	3 20	26	185	6 8 15 18 10	40	90	9 10 17
13	142	2 11 19	27	55	6 8 12 15 18			
14	140	2 5 11 19 21	28	130	4			

$c^A_{i,k,k'} = 0.1$ and $c^E_{i,k,k'} = 0.15$ for i, k, k'.

Table B.10
Machine dimensions for problem 11

Machine #	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
width (w_k)	3.5	2	3	4	3.5	3.5	3	2	3	2	2.5	3.5	2	3	2.5	2.5	2	3	2	2.5	3	2	2.5	2
length (h_k)	2	3.5	2	2	4	3.5	3.5	4	2.5	2.5	3.5	4	2	3	2.5	2	2	2	2.5	4	3.5	4	3.5	3

Table B.11
Processing information of parts for problem 12

Part #	Operation sequence	Part #	Operation sequence	Part #	Operation sequence	Part #	Operation sequence
1	10 18 7 16 4 22	11	21 8 13	21	8 10 9	31	17 5 19
2	25 1 2 17	12	1 24 17 3 25	22	17 9 3 8	32	14 13 15 22
3	20 3 11	13	20 11 3	23	19 5 16	33	11 25 20
4	12 23	14	5 11 20 3	24	5 16	34	23 12 24
5	18 12 4	15	16 19 5 2 10	25	6 21 15	35	21 6 22 15
6	23 16 12	16	4 16 7 18	26	23 4 12 15	36	17 1 2 11
7	18 7 4 16 10	17	7 18 10	27	12 22 21	37	23 12 7
8	5 19 16	18	22 15 14	28	9 8 10	38	22 8 9
9	20 25 3 11	19	8 10 9	29	21 6 5	39	12
10	9 8 25	20	23	30	18 7 16 4	40	21 6 15

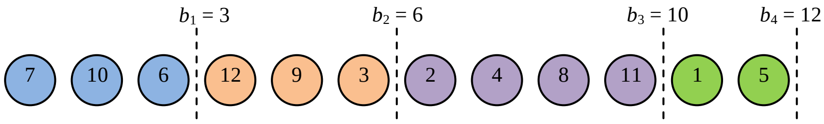
$d_i = 100$, $c^A_{i,k,k'} = 0.1$ and $c^E_{i,k,k'} = 0.15$ for i, k, k'.

Table B.12
Machine dimensions for problem 12

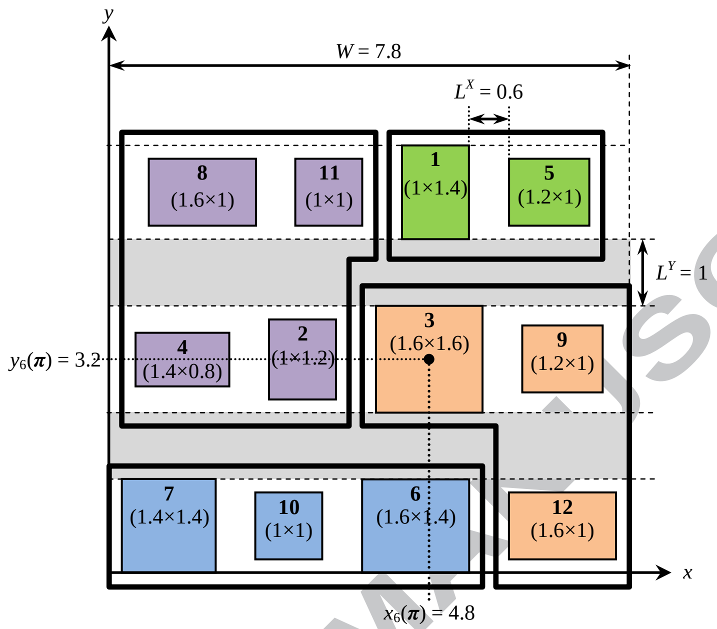
Machine #	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
width (w_k)	4	3.5	2	3.5	2.5	4	3	3.5	2.5	3.5	3.5	4	4	3.5	3	4	2.5	3.5	2.5	4	2	2.5	3.5	4	3.5
length (h_k)	4	2	3	2.5	4	3.5	2.5	3	2	3	2.5	3	3	3.5	2.5	2.5	3.5	4	3	2	3	4	3.5	3	3

ACCEPTED MANUSCRIPT

ACCEPTED MANUSCRIPT



a) A typical permutation of machines and a sample partitioning on it



b) Proposed spiral layout for the given permutation of machines

Fig. 1 An illustrative example for the proposed S-shaped layout

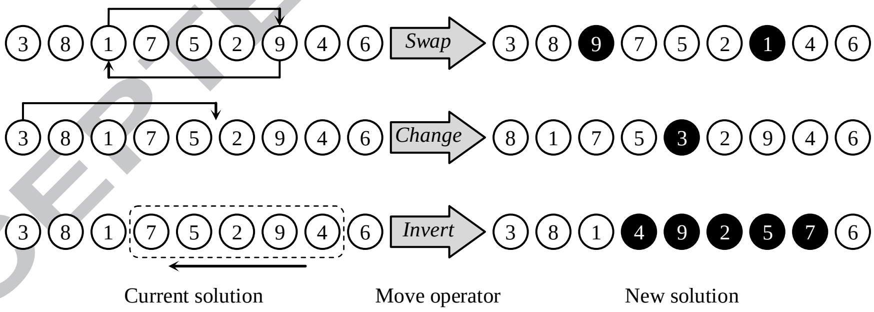


Fig. 2 Example of *Move* operators applied in the SA

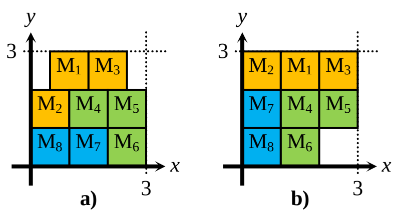


Fig. C.1 Problem 1: **a)** solution of the HSA **b)** solution obtained by Krishnan et al. (2012)

ACCEPTED MANUSCRIPT

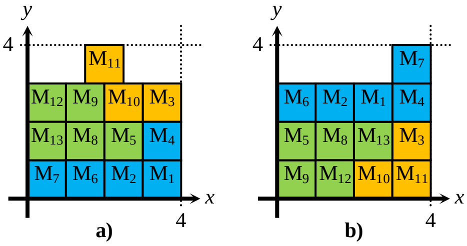


Fig. C.2 Problem 2: **a)** solution of the HSA **b)** solution obtained by Krishnan et al. (2012)

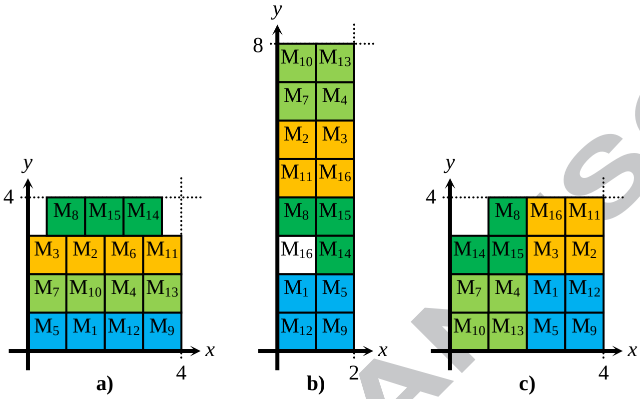


Fig. C.3 Problem 3: **a)** solution of the HSA considering $W = 4$ **b)** solution of the HSA considering $W = 2$ and an artificial machine (i.e., machine 16) **c)** solution obtained by Krishnan et al. (2012)

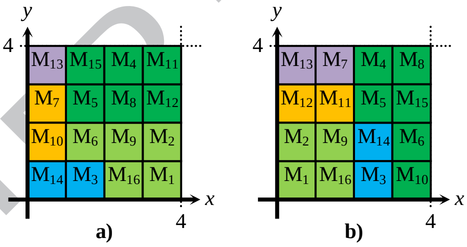


Fig. C.4 Problem 4: **a)** solution of the HSA **b)** solution obtained by Krishnan et al. (2012)

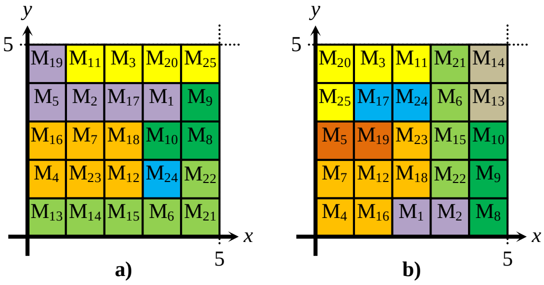


Fig. C.5 Problem 5: **a)** solution of the HSA **b)** solution obtained by Krishnan et al. (2012)

ACCEPTED MANUSCRIPT

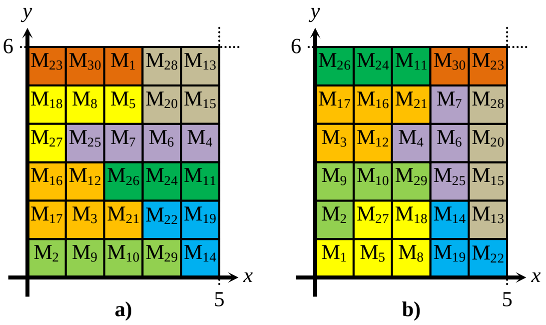


Fig. C.6 Problem 6: **a)** solution of the HSA **b)** solution obtained by Krishnan et al. (2012)

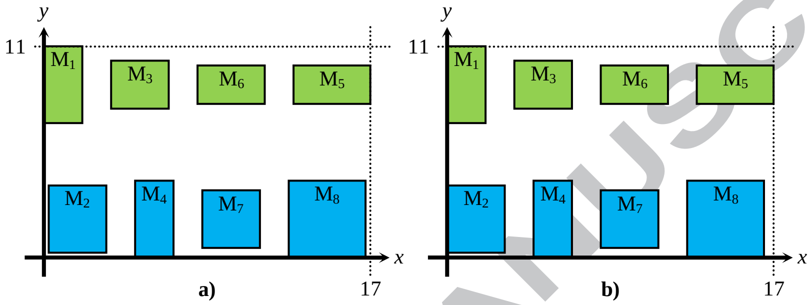


Fig. C.7 Problem 7: **a)** solution of the HSA **b)** solution obtained by Mohammadi and Forghani (2014)

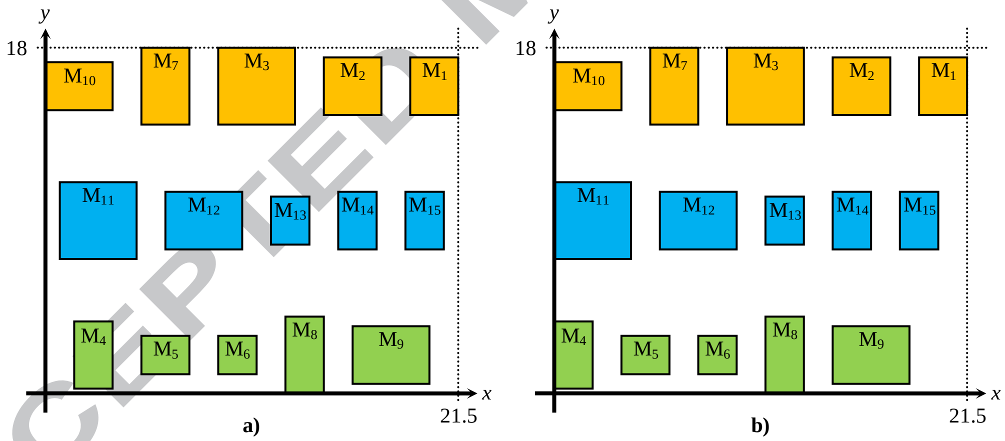


Fig. C.8 Problem 8: **a)** solution of the HSA **b)** solution obtained by Mohammadi and Forghani (2014)

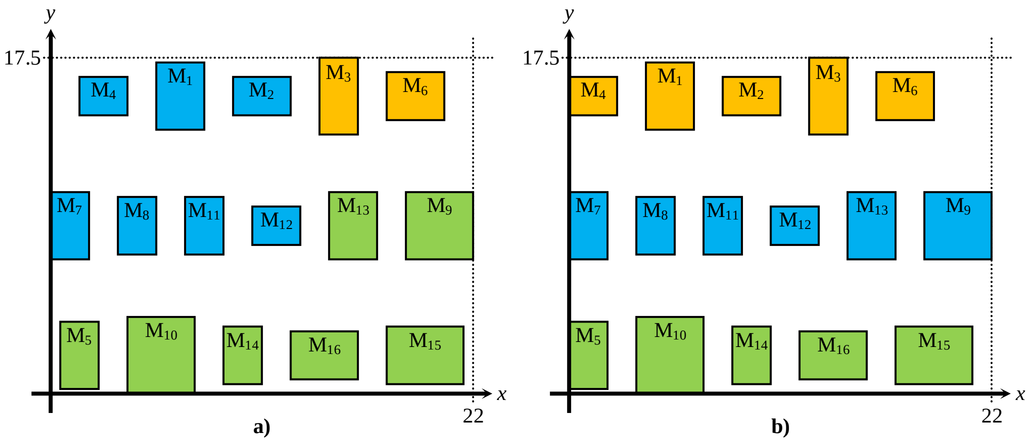


Fig. C.9 Problem 9: **a)** solution of the HSA **b)** solution obtained by Mohammadi and Forghani (2014)

ACCEPTED MANUSCRIPT

ACCEPTED MANUSCRIPT

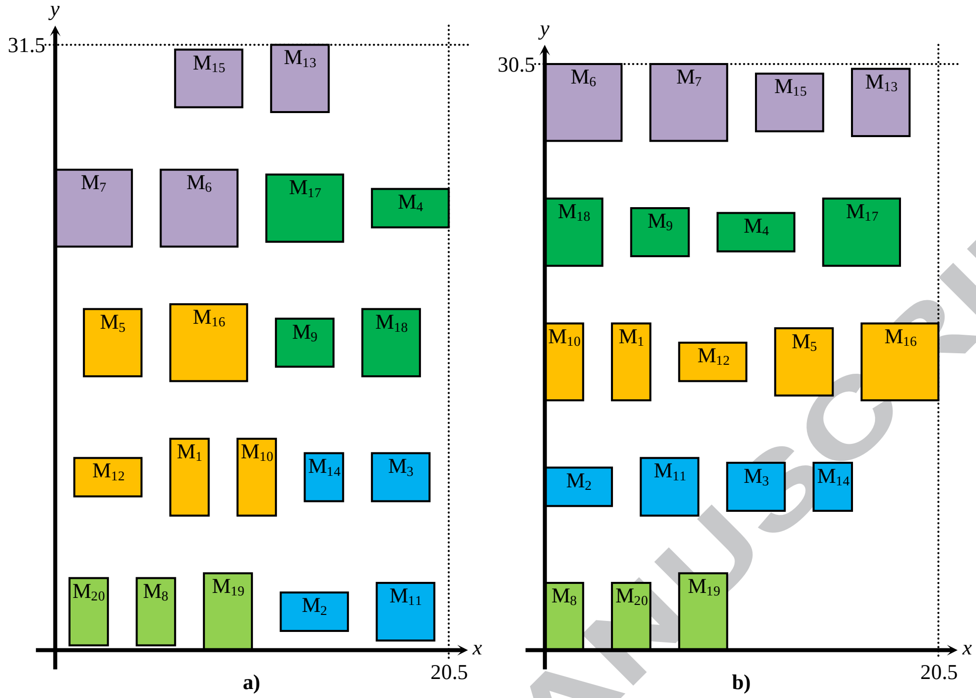


Fig. C.10 Problem 10: **a)** solution of the HSA **b)** solution obtained by Mohammadi and Forghani (2014)

Fig. C.11 Problem 11: **a)** solution of the HSA **b)** solution obtained by Mohammadi and Forghani (2014)

ACCEPTED MANUSCRIPT

ACCEPTED MANUSCRIPT

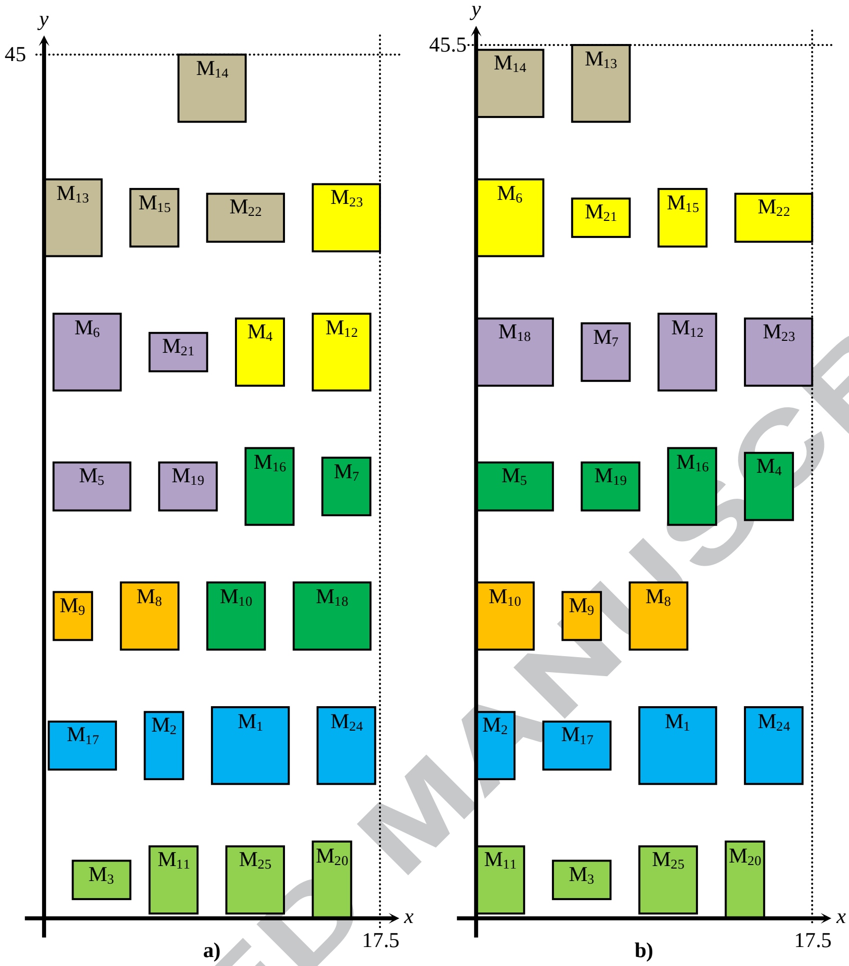


Fig. C.12 Problem 12: **a)** solution of the HSA **b)** solution obtained by Mohammadi and Forghani (2014)

ACCEPTED MANUSCRIPT

ACCEPTED MANUSCRIPT

Highlights

- An integrated bi-objective cell formation and layout problem is addressed.
- A new framework based on a S-shaped layout is proposed for the layout of cellular manufacturing systems.
- The dimensions of machines, the horizontal and vertical aisle widths, and the width of the shop floor are considered in the layout problem.
- A hybrid solution method combining Simulated Annealing and Dynamic programming algorithms is developed.
- The computational results show the advantages of the proposed approach and the good performance of the solution method.

ACCEPTED MANUSCRIPT