Associate Professor Mohammad Mohammadi, PhD
E-mail: mohammadi@khu.ac.ir (corresponding author)
Department of Industrial Engineering, Faculty of Engineering
Kharazmi University, Tehran, Iran
Kamran Forghani, PhD student
E-mail: kamran21f@gmail.com
Department of Industrial Engineering, Amirkabir University of Technology, Tehran, Iran

A HYBRID METHODOLOGY BASED ON DYNAMIC PROGRAMMING AND SIMULATED ANNEALING FOR SOLVING AN INTEGRATED CELL FORMATION AND LAYOUT PROBLEM

Abstract. *The layout design process is an important stage in designing a cellular manufacturing system. The present research investigates the integrated cell formation and layout problem with design parameters such as part demands, sequence data, and machine dimensions. The problem is to assign machines to the cells, find the arrangement of machines within the cells, and obtain the layout of cells, such that the total material handling cost is minimized. Due to the computational complexity of the problem, a hybrid solution procedure based on dynamic programming and simulated annealing is developed to effectively solve it. In the proposed methodology, partial solutions are created by the simulated annealing, and the dynamic programming is applied to complete these partial solutions and evaluate their optimum objective function values. Computational experiments are conducted to evaluate the performance of the proposed algorithm. Our computations indicated remarkable performance both in terms of solution quality and computation time.*

Keywords: *Cellular manufacturing, Cell formation, Layout problem, Dynamic programming, Simulated annealing, Hybrid algorithm.*

JEL Classification: C61, D24, O14

1. Introduction

Cellular Manufacturing system (CMS) decomposes a production system into several manageable and relatively independent subsystems (called manufacturing cells) in order to make the production process more efficient and productive. The major advantages of CMSs involve: decreased set-up times, reduced work-in-process inventories, improved product quality, shorter lead times, reduced tool requirements, improved productivity, better quality and production control, decreased material handling cost, etc., (Paydar et al., 2010). To design a CMS, a set of important decisions should be made carefully. These decisions include:

1) Cell Formation (CF): grouping parts with similar design features or processing requirements into part families and grouping machines into machine cells on the basis of the required operations by the part families,
2) group layout: layout of machines within each cell (intra-cell layout), and layout of cells with respect to one another (inter-cell layout),
3) group scheduling: scheduling parts and part families for production,
4) resource allocation: assignment of tools, manpower, materials, and other resources.

Ideally, these decisions should be made simultaneously in order to attain the best CMS design. However, due to the complex nature of each of these decisions, most researchers have focused on sequential and independent approaches (Wu et al., 2007). Facilities layout is a key area 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). It is estimated that over 20–50% of the manufacturing cost is related to the handling of parts; and an efficient facility layout can reduce it by 10–30% (Tompkins et al., 2003). In practice, machine cells may not be independent and there may be some parts that require machines in two or more cells for processing. These parts are called Exceptional Elements (EEs). The material flow between the cells is an obstacle to achieving the benefits of CMS if its layout is not effectively designed. In recent years, there has been increasing interest in investigating the CF and layout problems using integrative or sequential methods. As the CF and layout problems are NP-hard (Garey and Johnson, 1979), using heuristic and meta-heuristic approaches (such as Genetic Algorithm (GA), Simulated Annealing (SA), Tabu Search (TS), etc.,) is popular among researchers. In this context, Heragu and Kakuturi (1997) attempted to integrate the machine grouping problem with layout problem. The machine cells are first formed by a heuristic algorithm, and then a hybrid SA algorithm is employed to construct near-optimal inter- and intra-cell layouts. Aktürk and Turkcan (2000) proposed a solution methodology to simultaneously solve the CF problem by considering the intra-cell layout. 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 handling cost. It was assumed that cell locations are approximately equally spaced and 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 capacity of cuts. Also, Chiang and Lee (2004) combined a SA algorithm with a Dynamic Programming (DP) for solving the same problem presented by 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 was 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 of parts into a nonlinear mathematical model, and 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. Chan et al. (2006) proposed a two-stage GA-based solution approach for solving the CF problem as well as the cell layout problem. The first stage is to identify machine cells and part families. Also, the second stage is to arrange the layout sequence of machine cells (linear inter-cell layout) in such a way that the total inter-cell material handling cost is minimized. In the suggested approach, the Quadratic Assignment Problem (QAP) was used to represent the inter-cell layout. Wu et al. (2007) developed a GA for solving an integrated CF and group layout problem considering sequence data, workload, machine capacities, part demands, batch sizes, and layout types. Paydar et al. (2010) formulated the integrated CF and intra-cell layout problem as a multiple departures single destination multiple travelling salesman problem and proposed a solution methodology based on SA to solve it. Jolai et al. (2012) presented a modified version of the proposed model in (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 as well as the number of EEs. Chang et al. (2013) formulated a two-stage mathematical programming model to integrate the CF, cell layout, and intracellular machine sequence with the consideration of part demands, sequence data, and alternative process routings. The aim of the first stage is to simultaneously solve the CF and cell layout problems. Whereas the primary function of the second stage is to determine the machine layout in each cell on the basis of the CF determined in the first stage. In this study, the linear single- and double-row layouts were considered as two alternatives for the cell layout. A TS algorithm was employed to solve the proposed problem. Forghani et al. (2015) proposed a heuristic to solve an integrated cell formation and layout problem. They used QAP and continuous facility layout problem to formulate the inter- and intra-cell layout problems, respectively.

Most CF approaches proposed in the literature and some of them reviewed above usually consider one of the inter- or 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, and/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 CMS layout problem. To fill these gaps, this research presents an integrated CF, inter- and intra-cell layout problem with design parameters such as part demands, sequence data, and machines dimensions.

In order to have an accurate layout, the material handling cost is calculated on the basis of the actual location of machines on the plant site. The objective is to form machine cells, find the arrangement of machines within each cell and obtain the layout of cells in such a way that the total material handling cost is minimized. Due to the computational complexity of the problem, a hybrid algorithm combining SA with DP is used to solve it effectively. After setting the parameters of the algorithm, a set of instances are solved and the results are compared with the solutions derived from CPLEX optimization software. Also, by solving several numerical problems from the literature the suggested approach is compared to several conventional approaches. Generally, the main contributions of this research are as follows:

- to address an integrated approach for considering both the inter- and intra-cell layout problems in the CF process by considering part demands, sequence data, and machine dimensions,
- to give a more accurate measure based on the center-to-center distance between machines for calculating the material handling cost,
- to develop an effective hybrid algorithm based on SA and DP for solving the problem,
- to make a comparative study between the proposed integrated approach and other well-known approaches found in the literature.

2. Model description and proposed mathematical model

In CMSs. the intra-cell layout is associated with the layout of machines within each cell and the inter-cell layout is associated with the layout of cells. The flow-line (single line) layout is considered when multi-products with different production volumes and different processing routings need to be manufactured (El-Baz, 2004). It is one of the common layout types that have been used in the design of CMSs. In this research both the inter- and intra-cell layouts are represented by the flow-line layout as shown in Figure 1. Parts are transferred between the machines according to their processing information that is known in advance. The objective function is the minimization of the total material handling cost which is calculated based on the actual location of machines and by considering part demands, sequence data, and machine dimensions.

2.1. Assumptions

The major assumptions of the problem are as follows:

(i) The sequence data is known in advance and the operations of each part must be done according to the given sequence,

(ii) The demand of each part is known and deterministic,

(iii) The distance between two machines (either in the same cell or in distinct cells) is calculated by using rectilinear distance,

(iv) The arrangement of machines within the cells (intra-cell layout), as well as the cell layout (inter-cell layout) are assumed to be flow-line layout as shown in Figure 1,

(v) The maximum number of cells, as well as the maximum number of machines that can be assigned to each cell, are known in advance.

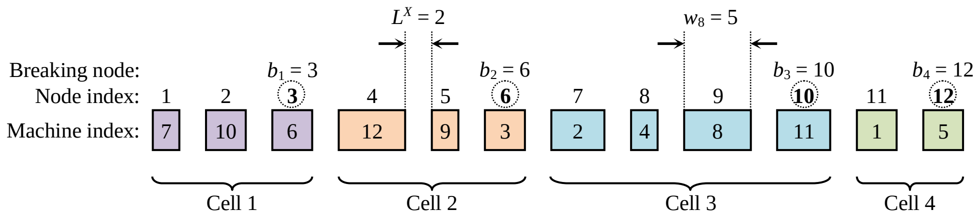

Figure 1. A typical permutation of machines and a sample partitioning on it

2.2. Notation and problem formulation

Sets:

i parts index ($i = 1, \ldots, P$) where P is the number of part types

k, k' machines index ($k, k' = 1, \ldots, M$) where M is the number of part types

l cells index ($l = 1, \ldots, L$) where L is the number of cells to be formed i.e., L is a decision variable

Parameters:

D_i demand of part i (unit/year)

$c^A_{i,k,k'}$ intra-cell material handling cost for transporting part i from machine k to machine k' per unit distance ($/unit)

$c^E_{i,k,k'}$ inter-cell material handling cost for transporting part i from machine k to machine k' per unit distance ($/unit)

w_k width of machine k (meter)

L^X horizontal distance (aisle) between machines (meter)

$f_{i,k,k'}$ 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

S set of possible permutation of machines

$\bar{S}$ permutation of machines to be laid out on the plant site according to the flow -line layout, $\bar{S} \in S$ and $\bar{S} = \{s(1), s(2), \ldots, s(M)\}$, where $s(k)$ represents the machine index placed in kth order

$x_k^{\bar{S}}$ horizontal coordinate of the centroid of machine k in permutation $\bar{S}$

b_l index of breaking node on permutation $\bar{S}$ for forming cell l, where $s(b_l)$ is the last machine on the sequence to be included in cell l

To formulate the problem described above, we introduce two auxiliary variables $F^{A}_{k,k'}$ and $F^{E}_{k,k'}$, respectively representing the total intra- and inter-cell material handling costs between machines k and k' in permutation $\bar{S}$. These auxiliary variables are calculated by Eqs (1) and (2), respectively.

$$F^{A}_{s(k),s(k')} = \sum_{i=1}^{P} D_i c^{A}_{i,k,k} f_{i,,k,k}\left(x^{\bar{S}}_{s(k')} - x^{\bar{S}}_{s(k)}\right), \forall k, k', \quad (1)$$

$$F^{E}_{s(k),s(k')} = \sum_{i=1}^{P} D_i c^{E}_{i,k,k} f_{i,k,k}\left(x^{\bar{S}}_{s(k')} - x^{\bar{S}}_{s(k)}\right), \forall k, k'. \quad (2)$$

Where $x^{\bar{S}}_{s(k)}$ is calculated by Eq. (3).

$$x^{\bar{S}}_{s(k)} = \frac{w_{s(k)}}{2} + \sum_{k'=1}^{k-1}\left(w_{s(k')} + L^{x}\right), \forall k. \quad (3)$$

By these definitions, the integrated CF and layout problem can be represented by (4).

$$\min_{\bar{S}\in S} TH^{*}(\bar{S}). \quad (4)$$

Where $TH^{*}(\bar{S})$ is equivalent to the optimum objective function value of the following optimization problem:

$$TH^{*}(\bar{S}) \equiv \min TH(\bar{S}) = \sum_{l=1}^{L}\left(\sum_{k=b_{l-1}+1}^{b_l -1} \sum_{k'=k+1}^{b_l} F^{A}_{s(k),s(k')} + \sum_{k=b_{l-1}+1}^{b_l} \sum_{k'=1}^{b_{l-1}} F^{E}_{s(k),s(k')}\right). \quad (5)$$

Subject to:

$$b_0 = 0 \text{ and } 1 \le b_1 < \ldots < b_L = M, \quad (6)$$

$$b_l - b_{l-1} \le NM, \forall l = 1, \ldots L, \quad (7)$$

$$L \le C^{\max}. \quad (8)$$

Objective function (5) minimizes the total material handling cost for the given permutation $\bar{S}$. Constraint (6) ensures that each formed cell contains at least one machine and also guarantees that all the machines are included in the cells. Constraint (7) represents that each cell can contain at most NM machines. Finally, constraint (8) prevents the formation of more than $C^{\max}$ cells.

To simplify objective function (5) we can rewrite it as follows:

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

Now, Eq. (9) is rearranged to Eq. (10).

$$TH(\bar{S}) = \sum_{l=1}^{L} \sum_{k=b_{l-1}+1}^{b_l} \left(\sum_{k'=1}^{b_{l-1}} F^{E}_{s(k),s(k')} + \sum_{k'=k+1}^{b_l} F^{E}_{s(k),s(k')} \right) + \sum_{l=1}^{L} \sum_{k=b_{l-1}+1}^{b_l-1} \sum_{k'=k+1}^{b_l} \left(F^{A}_{s(k),s(k')} - F^{E}_{s(k),s(k')} \right). \tag{10}$$

It can be shown that the first term in Eq. (10) is constant. To do so, $TH'(\bar{S})$ is introduced to replace the first term of Eq. (10). Now, $TH'(\bar{S})$ is simplified as follows:

$$TH'(\bar{S}) = \sum_{l=1}^{L} \sum_{k=b_{l-1}+1}^{b_l} \left(\sum_{k'=1}^{b_{l-1}} F^{E}_{s(k),s(k')} + \sum_{k'=k+1}^{b_l} F^{E}_{s(k),s(k')} \right)$$
$$= \sum_{k=1}^{b_1} \sum_{k'=1}^{0} F^{E}_{s(k),s(k')} + \sum_{k=1}^{b_1} \sum_{k'=k+1}^{b_1} F^{E}_{s(k),s(k')} + \ldots + \sum_{k=b_{L-1}+1}^{M} \sum_{k'=1}^{b_{L-1}} F^{E}_{s(k),s(k')} + \sum_{k=b_{l-1}+1}^{M} \sum_{k'=k+1}^{M} F^{E}_{s(k),s(k')}$$
$$= \sum_{k=1}^{M-1} \sum_{k'=k+1}^{M} F^{E}_{s(k),s(k')}. \tag{11}$$

Finally, according to Eqs (10) and (11), the optimum objective function value of the partition problem can be obtained by solving the following optimization problem:

$$TH^{*}(\bar{S}) = TH'(\bar{S}) - \left\{ \begin{array}{l} \max \sum_{l=1}^{L} \sum_{k=b_{l-1}+1}^{b_l-1} \sum_{k'=k+1}^{b_l} \left(F^{E}_{s(k),s(k')} - F^{A}_{s(k),s(k')} \right) \\ \text{Subject to: (6)-(8).} \end{array} \right\}. \tag{12}$$

2.3. Dynamic programming

By using DP, the partition problem (i.e., problem (12)) can be solved sequentially in stages from 1 to $C^{\max}$. To do so, let $f_l(b'_l, b_l)$ and $I_{b'_l,b_l}$, respectively, indicate the objective function value and the improved material handling cost at stage *l*, when at this stage $\bar{S}$ is partitioned from breaking node b'_l to breaking node b_l. Also, let $f_l^{*}(b_l)$ be the optimum objective function value of breaking node b_l in stage *l*. Now, by using forward recursion the partition problem at stage *l* becomes:

$$f(b'_l, b_l) = f^*_{l-1}(b'_l) + I_{b'_l, b_l}. \quad (13)$$

Subject to:

$$\max\{l, M-(C^{\max}-l)NM\} \le b_l \le \min\{M, l\times NM\}, \quad (14)$$

$$\max\{l, M-(C^{\max}-l)NM, b_l - NM\} \le b'_l \le \min\{M, l\times NM, b_l - 1\}. \quad (15)$$

Where, $f^*_0(b'_1) = 0$ and $f^*_l(b_l) = \max_{\max\{l, M-(C^{\max}-l)NM, b_l-NM\} \le b'_l \le \min\{M, l\times NM, b_l-1\}} \{f_l(b'_l, b_l)\}$. Also, $I_{b'_l, b_l}$ i.e., the improved material handling cost from breaking node b'_l to breaking node b_l is calculated by Eq. (16).

$$I_{b'_l, b_l} = \sum_{k=b'_l+1}^{b_l-1} \sum_{k'=k+1}^{b_l} \left(F^E_{s(k),s(k')} - F^A_{s(k),s(k')}\right). \quad (16)$$

Equation (13) is the forward recursive equation. Also, constraints (14) and (15) are auxiliary constraints that avoid infeasible solutions.

Note that, the proposed DP partitions $\bar{S}$ into exactly $C^{\max}$ cells. Therefore, the optimum objective function value, $TH^*(\bar{S})$, and the optimum number of cells, $L^*(\bar{S})$, for permutation $\bar{S}$ are obtained by Eqs (17) and (18), respectively.

$$TH^*(\bar{S}) = \max_{\{l|b_l=M\}} \{f^*_l(b_l)\}, \quad (17)$$

$$L^*(\bar{S}) = \arg\max_{\{l|b_l=M\}} \{f^*_l(b_l)\}. \quad (18)$$

3. Proposed hybrid solution algorithm

SA is a stochastic search method which uses the idea of the annealing process of solid to solve combinatorial optimization problems. In the annealing process, a solid is heated until it melts, and 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 known as NP-hard problem. From the other side, since the problem of this study integrates these problems; we can conclude that this problem is also a NP-problem. It means that the problem is hard to be solved optimally in an acceptable computational time when the problem size increases. In recent years, SA algorithms have been successfully applied by researchers for solving the CF and layout problems, see for example (Chiang and Lee, 2004; Al-Araidah et al., 2007; Wu et al., 2009; Paydar et al., 2010). Based on these considerations, we have been motivated to develop a SA algorithm for solving the problem. The main elements of this methodology are explained in the following subsections.

3.1. Solution encoding and generating initial solution

In the SA implementation, each solution must be represented by a coding scheme. In this research, each solution comprises a permutation of *M* integer values ranged from 1 to *M*, where *M* is the number of machines. The initial permutation is randomly generated. This permutation is associated with the layout sequence of machines on the plant site i.e., $\bar{S}$. For instance, permutation $\bar{S} = (7,10,6,12,9,3,2,4,8,11,1,5)$ corresponds to the layout sequence of the machines given in Figure 1. The optimal partitioning of each permutation plus its objective function value is determined by the DP algorithm which was explained in Section 2.3. It should be noted that applying this coding scheme not only yields optimal solutions for each permutation, but also reduces the string length needed to represent a solution.

3.2. Cooling schedule and moving to a neighboring solution

SA algorithm works with a controlled cooling schedule which is also called the annealing schedule. Starting from the initial temperature, T_0, the temperature is gradually decreased through an appropriate cooling schedule. In this study, the *Geometric Decrement* function, $T_t = \alpha \times T_{t-1}$, is used in the cooling schedule. In this function, T_t is the temperature at *t*-th iteration and α ($0 < \alpha < 1$) is the cooling rate. At each temperature (iteration), a generation mechanism called *Move* is applied to transform the current permutation into a neighboring (new) permutation. Three move operators are proposed, namely *Swap*, *Change* and *Inverse* operators. The *Swap* operator swaps the order of two randomly selected machines, the *Change* operator changes the order of a randomly selected machine, and the *Inverse* operator reverses the order of machines between two randomly selected points. An example of these move operators is given in Figure 2. It should be noted that these operators are applied independently on the current solution to derive a neighboring solution.

3	8	1	7	5	2	9	4	6

Current permutation

3	8	9	7	5	2	1	4	6

New permutation created by *Swap* operator

8	1	7	5	3	2	9	4	6

New permutation created by *Change* operator

3	8	1	4	9	2	5	7	6

New permutation created by *Inverse* operator

Figure 2. Example of *Move* operators used in the SA

Once a neighboring solution was created, the change in the objective function value is calculated by $\Delta = TH^*(\bar{S}^{\text{New}}) - TH^*(\bar{S}^{\text{Current}})$. If the change in each transition represents an improvement in the objective function value (i.e., $\Delta < 0$), the transition to 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 can avoid being trapped on the local minimum. This mechanism at each temperature is repeated until N^{max} accepted transitions are met. Where, the value of N^{max} is assumed to be proportional to the number of machines (i.e., $N^{max} = N \times M$).

3.3. Stopping criteria

The SA algorithm terminates when either a specified number of iterations, I^{max}, is reached or the temperature gets below T_f (frozen temperature).

4. Computational results

In this section, computational experiments are conducted to evaluate the performance of the proposed SA algorithm and show the advantage of the integrated approach being described. First, statistical experiments are carried out in order to set the SA parameters. Then, the proposed hybrid SA is compared with the B&B algorithm. Finally, the proposed integrated approach is compared with the conventional approaches in the literature.

4.1. Parameters setting

The value of parameters used in SA algorithm may have a significant influence on its performance. So, these parameters must be carefully selected. To do this, the frozen temperature, T_f, is fixed at 1 and the initial temperature is calculated by $T_0 = \sum_{n=1}^{100} \left| TH^*(\bar{S}_n^1) - TH^*(\bar{S}_n^2) \right| \Big/ -100 \times \ln(0.95)$, where $\bar{S}_n^1$ and $\bar{S}_n^2$ are two random permutations generated at n-th trial. A pilot experiment is conducted using six randomly generated instances to select the appropriate values for the other parameters. The data set of these instances are randomly generated according to Table 1 and by considering M = 15, 20, 25, 30, 35 and 40.

Table 1. Data set generation based on *M* (number of machines)

Parameter	Value
P	$\lfloor M \times 1.5 \rceil$
No. operations	Random{2, …, 6}, ∀ i
D_i	U ~ (10, 100), ∀ i
w_k	Random{1, 2, 3, 4}, ∀ k
C^{max}	$\lfloor \sqrt{M} \rceil$
NM	$\lfloor M/(NM-1) \rceil$
L^X	1.5
$c^A_{i,k,k'}$	1, ∀ i, k, k'
$c^E_{i,k,k'}$	1.5, ∀ i, k, k'

The symbol $\lfloor x \rceil$ indicates the nearest integer to x

Based on this pilot test, I^{max} (maximum number of iterations) was set to 50000, N^{max} (maximum number of transitions at each temperature) was set to $5 \times M$ and α (cooling rate) was set to 0.95.

4.2. Proposed SA against B&B

In this section, the solution of the SA is compared to that of the B&B algorithm. In this way, the Mixed-Integer Programming (MIP) model of the proposed problem (given in Appendix A) was formulated in the GAMS IDE and the CPLEX was chosen as the solver. 24 instances (with M ranged from 7 to 30) are randomly generated according to Table 1. These instances are solved by the SA and the results are compared to the solutions derived from the CPLEX. As mentioned earlier, the problem is hard to be solved optimally in an acceptable computational time when the problem size increases. In such cases, the solver is interrupted after 7200 seconds and the optimality gap is reported. The comparison results are given in Table 2.

Table 2. Comparison between the SA and CPLEX solutions for the randomly generated instances

Problem #	Size ($M \times P$)	C^{max}	NM	CPLEX (B&B) $TH^{B\&B}$	CPLEX (B&B) CPU time (s)	CPLEX (B&B) Opt. Gap (%)*	SA TH^{SA}	SA μ_{TH}	SA σ_{TH}	SA BSF	SA Mean CPU time (s)	Gap (%)†
1	7×11	3	4	12866.00	12.95	0.00	12866.00	12866.00	0.00	30	0.131	0.00
2	8×12	3	4	19848.75	47.61	0.00	19848.75	19848.75	0.00	30	0.163	0.00
3	9×14	3	5	32853.00	493.42	0.00	32853.00	32853.00	0.00	30	0.252	0.00
4	10×15	3	5	16073.50	1616.20	0.00	16073.50	16073.50	0.00	30	0.266	0.00
5	11×17	3	6	41265.75	>7200	51.76	41189.50	41191.07	5.96	28	0.390	0.18
6	12×18	3	6	38141.00	>7200	48.77	38141.00	38141.00	0.00	30	0.439	0.00
7	13×20	4	4	43718.50	>7200	46.67	43596.00	43596.00	0.00	30	0.381	0.28
8	14×21	4	5	53185.00	>7200	80.10	52968.00	53025.87	97.60	21	0.550	0.41
9	15×23	4	5	66655.25	>7200	87.32	66194.25	66194.25	0.00	30	0.612	0.69
10	16×24	4	5	71234.25	>7200	88.16	67070.00	67070.00	0.00	30	0.639	5.85
11	17×26	4	6	69976.75	>7200	93.72	69227.50	69314.81	333.56	27	0.820	1.07
12	18×27	4	6	81142.25	>7200	97.32	81142.25	81153.65	43.38	28	0.973	0.00
13	19×29	4	6	59984.00	>7200	97.50	49509.25	49509.25	0.00	30	1.294	17.46
14	20×30	4	7	76955.25	>7200	98.20	73913.25	73913.25	0.00	30	1.516	3.95
15	21×32	5	5	170173.75	>7200	99.33	169178.75	169627.13	673.92	13	1.252	0.58
16	22×33	5	6	181581.50	>7200	100.00	179842.00	180466.78	943.65	16	1.718	0.96
17	23×35	5	6	163091.50	>7200	99.90	156457.50	156515.19	190.29	26	2.022	4.07
18	24×36	5	6	173516.75	>7200	100.00	153107.50	154329.83	3120.80	20	2.365	11.76
19	25×38	5	6	212562.75	>7200	100.00	198083.75	198544.93	1395.73	23	2.979	6.81
20	26×39	5	7	199863.00	>7200	100.00	184168.75	184443.55	337.73	10	3.963	7.85
21	27×41	5	7	186269.25	>7200	100.00	173563.75	175420.97	2472.92	15	4.320	6.82
22	28×42	5	7	215700.50	>7200	100.00	193574.00	193645.85	287.18	25	6.318	10.26
23	29×44	5	7	218682.25	>7200	100.00	193134.00	193407.83	500.98	12	6.492	11.68
24	30×45	5	8	296641.00	>7200	100.00	288926.50	289731.63	951.85	14	6.051	2.60

TH^{SA}: Best solution found in 30 runs of the SA implementation; μ_{TH}: Mean of the SA solutions; σ_{TH}: Standard deviation of the SA solutions; Frequency of the best solution in 30 runs of the SA implementation

* Optimality gap found by the CPLEX

† Relative difference between the SA and CPLEX solutions (Gap = $(TH^{B\&B} - TH^{SA})/TH^{B\&B}$)

In Table 2, columns '$TH^{B\&B}$' and 'Opt. Gap', respectively, indicate the objective function value (total material handling cost) and the relative optimality gap obtained by the CPLEX. Also, for the SA, the best objective function in 30 runs of the algorithm is given in column 'TH^{SA}', the mean and standard deviation of the solutions are respectively shown in columns μ_{TH} and σ_{TH}, the number of times that the best solution appeared in the 30 runs of the algorithm is given in column '*BSF*', and the mean of CPU times is shown in column 'Mean CPU time'.

According to the results shown in Table 2, except for problems 1−4 whose optimality gaps are zero, the remaining problems were not solved optimally by the CPLEX in 7200 s. For these problems, we can see that by increasing the problem size, the optimality gap increases. In worst cases (i.e., for problems 16 and 18−24), the optimality gap is equal to 100% which implies that for those problems the CPLEX was not able to even find a lower bound. From the other side, according to the last column of Table 2 (i.e., column 'Gap'), it can be observed that the SA is able to find better (or at least, equally good) solutions in a less computation time compared to the CPLEX solver. The largest gap between the SA and CPLEX solutions is 17.46% which is remarkable. Moreover, according to columns 'σ_{TH}' and '*BSF*', it can be concluded that the SA algorithm is able to consecutively produce good solutions. These demonstrate the superiority of the proposed hybrid SA over the B&B algorithm in terms of the solution quality and computation time.

4.3. Comparison with similar studies

In this section, the suggested integrated approach is compared to several conventional approaches in the literature. In this way, 16 problems adopted from the literature are solved by proposed approach and the results are compared with the solutions derived from the literature. The characteristic of these problems is shown in Table 3.

Table 3. Characteristic of the problems selected from the literature*

Problem #	Size ($M \times P$)	Data set source	Solution source	Design criterion(s)
25	8 × 20	Nair and Narendran (1998)[a]	Mahdavi et al (2013)	A + B + C
26	12 × 19	Irani and Huang (2006)[a]	Ilić (2012)	A + B
27	15 × 25	Saeedi et al (2010)	Saeedi et al (2010)	B + D
28	20 × 20	Harhalakis et al (1990)[a]	Harhalakis et al (1990)	B
29	20 × 20	Harhalakis et al (1990)[a]	Harhalakis et al (1990)	B
30	20 × 20	Harhalakis et al (1990)[a]	Mahdavi et al (2013)	A + B + C
31	20 × 20	Harhalakis et al (1990)[a]	Lee and Chiang (2001)	C
32	20 × 20	Harhalakis et al (1990)[a]	Lee and Chiang (2001)	C
33	24 × 40	Kazerooni et al (1997)	Chan et al (2006)	C
34	25 × 40	Nair and Narendran (1998)[a]	Lee and Chiang (2001)	C
35	30 × 50	Gonçalves and Resende (2004)[a, b]	Gonçalves and Resende (2004)	E
36	30 × 50	Gonçalves and Resende (2004)[a, b]	Gonçalves and Resende (2004)	E
37	30 × 90	Gonçalves and Resende (2004)[a, b]	Gonçalves and Resende (2004)	E
38	37 × 30	Chan et al (2006)[a]	Chan et al (2006)	C
39	40 × 100	Gonçalves and Resende (2004)[a, b]	Lee and Chiang (2001)	C
40	40 × 100	Gonçalves and Resende (2004)[a, b]	Gonçalves and Resende (2004)	E

* For all data sets, the width of each machine type (w_k) is randomly selected from {1, 2, 3, 4}

[a] For this data set the demand of all parts is assumed to be 1 unit

[b] For this data set it is assumed that the process of parts are done according to the machine indexes in increasing order

A: minimization of the intra-cell moves B: minimization of the inter-cell moves; C: minimization of the inter-cell traveled distance; D: minimization of the number of voids; E: maximization of the grouping efficacy

In Table 3, the last column indicate the design criterion(s) applied by different authors in solving these problems, where criterion 'A' is the minimization of the intra-cell moves (an intra-cell move occurs when two consecutive processes of a part are performed within the same cell), criterion 'B' is the minimization of the inter-cell moves (an inter-cell move occurs when a part is moved from one cell to another for processing), criterion 'C' is the minimization of the inter-cell traveled distance (this cost is acquired by the product of the travel distance, travel cost and travel volume between the cells), criterion 'D' is the minimization of the number of voids (a void is a zero value appearing inside the diagonal block of the machine-part matrix), and criterion 'E' is the maximization of the grouping efficacy (grouping efficacy, GE, is defined by $GE = (N_1 - N_1^{Out})/(N_1 + N_0) \times 100$ where N_1 is the total number of 1's in the incidence matrix, N_1^{Out} is the total number of 1's outside the diagonal blocks, and N_0 is the total number of 0's inside the diagonal blocks).

To be able to compare the results, it is necessary to obtain the optimal inter- and intra-cell layouts for the solutions reported in the literature. To do so, the CF results given in the literature (i.e., the assignment of machines to the cells) are treated as the parameter of the model presented in Appendix A (i.e., $z_{k,l}$ is supposed to be a parameter); and this model is then solved optimally by the CPLEX solver in order to obtain their layouts. A summary of the results is given in Table 4.

Table 4. Summary of comparison between proposed approach and conventional approaches in the literature

Problem #	C^{max}	NM	Other approaches		Proposed approach		Imp. (%)
			TH^{O}	CPU time (s)	TH^{SA}	CPU time (s)	
25	3	4	375.5	960	314.25	0.07	16.31
26	9	2	474.75	N/A†	389.75	0.58	17.90
27	3	6	50655.75	(10, 37, 251)*	42907.25	0.57	15.30
28	4	7	885	N/A†	734.25	1.65	17.03
29	5	5	1000.5	N/A†	755.5	1.11	24.49
30	5	5	925.5	8640	755.5	1.11	18.37
31	3	7	738	26.43	734.25	1.64	0.51
32	5	5	770	27.03	755.5	1.11	1.88
33	7	5	68805.5	N/A†	68133	2.48	0.98
34	7	4	1687.25	61.1	1315.75	2.13	22.02
35	11	5	4544.75	52.45	2754	5.90	39.40
36	12	3	4885.75	48.97	1129.75	3.42	76.88
37	9	6	7316.5	81.46	3550	6.77	51.48
38	4	13	1259.75	N/A†	1007.75	18.18	20.00
39	8	6	3706.75	67.6	3527.25	11.14	4.84
40	10	6	4454	152.13	3522.25	11.92	20.92

* The CPU times are correspond to GA, SA and ACO, respectively
† For this case, the CPU time was not available in the source paper

In Table 4, column 'TH^{O}' shows the optimal material handling cost found by the CPLEX solver for the CF results given in the literature, column 'CPU time', associated with other approaches, shows the computation time of various solution methods employed by the others for forming machine cells (note that this column does not corresponds to the CPU time of CPLEX solver), column 'TH^{SA}' indicates the best objective function value found in 30 runs of the SA implementation, column 'CPU time (s)', associated with the proposed approach, shows the average computation time of the SA in 30 runs, and finally column 'Imp. (%)', calculated by $(TH^{O} - TH^{SA})/TH^{O} \times 100$, shows the improvement percent in the total material handling cost.

The results given in Table 4 reveal that the proposed integrated approach gives better solutions with lower material handling cost compared to the other approaches. Based on these problems, the average cost improvement is 21.77%, with the largest cost reduction of 76.88%. Also, according to this table, we can see that the SA is able to solve these problems in considerably better computation time compared to the other solution approaches proposed by the others. For instance, Saeedi et al. (2010) employed three different algorithms, including GA, SA, and ACO for solving problem 27. Their computation indicated that the GA is faster than the other two algorithms with a computation time almost equal to 10 seconds. However, from Table 4, we can see that the SA has solved this problem in just 0.57 seconds.

5. Conclusion

In this research, we presented a hybrid solution procedure base on DP and SA for solving an integrated CF, inter- and intra-cell layout problem. The objective is to minimize the total material handling cost which is calculated based on the actual location of machines on the plant site. To enhance the search process for finding a better solution, a DP based partitioning algorithm was used inside the SA. Partial solutions comprising a permutation of machines are generated by the SA, and the DP is employed to find the optimal partitioning of this permutation (i.e., machine cells). After setting the SA parameters, several test instances were solved and the results were compared with the solutions derived from B&B algorithm. The results demonstrated that the SA is able to obtain better (or at least, equally good) solutions in considerably less computation time compared to the B&B algorithm. The results also indicated that the SA is able to consecutively produce good solutions even for large-sized instances. To compare the proposed approach against the conventional approaches, 16 problems adopted from the literature were solved. The comparisons showed that the suggested integrated approach results in a considerable improvement (in average 21.77%) in terms of the total material handling cost.

ACKNOWLEDGEMENTS

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

Appendix A: The MIP model of the proposed integrated CF and layout problem

The decision variables used in the MIP version of the proposed problem are as follows:

$z_{k,l}$	=1 if machine k is assigned to cell l; 0 otherwise
x_k	horizontal coordinate of the centroid of machine k
x_l^L	horizontal coordinate of the right edge of cell l
$d_{k,k'}^A$	distance between machines k and k' if these machines are assigned to a same cell (intra-cell distance)
$d_{k,k'}^E$	distance between machines k and k' if these machines are assigned to distinct cells (inter-cell distance)
$\lambda_{k,k',l}$	auxiliary variable (used to linearize product term $z_{k,l} z_{k',l}$)
$\alpha_{k,k'}$	auxiliary binary variable (used to liberalize the absolute operator in the layout constraint)

Also, the other parameters are the same as those in Section 2.2.

The MIP model of the integrated CF and layout problem is as follows:

$$\min TH = \sum_{k=1}^{M} \sum_{k'=k+1}^{M} d_{k,k'}^A \left(\sum_{i=1}^{P} D_i c_{i,k,k'}^A f_{i,k,k'} \right) + d_{k,k'}^E \left(\sum_{i=1}^{P} D_i c_{i,k,k'}^E f_{i,k,k'} \right).$$

Subject to:

$$\sum_{l=1}^{C^{\max}} z_{k,l} = 1, \forall k,$$

$$\sum_{k=1}^{M} z_{k,l} \le NM, \forall l,$$

$$\lambda_{k,k',l} \ge z_{k,l} + z_{k',l} - 1, \forall k, k', l,$$

$$\lambda_{k,k',l} \le z_{k,l}, \forall k, k', l,$$

$$\lambda_{k,k',l} \le z_{k',l}, \forall k, k', l,$$

$$x_k - x_{k'} \ge \frac{(w_k - w_{k'})}{2} + L^X - BM \left(1 - \alpha_{k,k'} - \sum_{l=1}^{C^{\max}} \lambda_{k,k',l} \right), \forall k, k', l,$$

$$x_{k'} - x_k \ge \frac{(w_k - w_{k'})}{2} + L^X - BM \left(2 - \alpha_{k,k'} - \sum_{l=1}^{C^{\max}} \lambda_{k,k',l} \right), \forall k, k', l,$$

$$x_l^L = x_{l-1}^L + \sum_{l=1}^{C^{\max}} \sum_{k=1}^{M} (w_k + L^X) z_{k,l}, \forall l,$$

$$x_k - \frac{w_k}{2} - L^X \ge x_{l-1}^L - BM(1 - z_{k,l}), \forall k, l,$$

$$x_k + \frac{w_k}{2} \le x_l^L - BM\left(1 - z_{k,l}\right), \forall k, l,$$
$$d_{k,k'}^A \ge x_k - x_{k'} - BM\left(1 - \sum_{l=1}^{C^{\max}} \lambda_{k,k',l}\right), \forall k' > k,$$
$$d_{k,k'}^A \ge x_{k'} - x_k - BM\left(1 - \sum_{l=1}^{C^{\max}} \lambda_{k,k',l}\right), \forall k' > k,$$
$$d_{k,k'}^E \ge x_k - x_{k'} - BM \sum_{l=1}^{C^{\max}} \lambda_{k,k',l}, \forall k' > k,$$
$$d_{k,k'}^E \ge x_{k'} - x_k - BM \sum_{l=1}^{C^{\max}} \lambda_{k,k',l}, \forall k' > k,$$
$$x_k, x_l^L, d_{k,k'}^A, d_{k,k'}^E, \lambda_{k,k',l,l'} \ge 0, \forall k, k', l' > l, x_0^L = 0,$$
$$z_{k,l}, \alpha_{k,k'} \in \{0,1\}, \forall k' > k, l.$$

where *BM* is a large enough number.

REFERENCES

[1] **Aktürk, M.S., Turkcan, A. (2000),** ***Cellular Manufacturing System Design Using a Holonistic Approach*****.** *International Journal of Production Research*, 38, 2327–2347;

[2] **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;

[3] **Benjaafar, S. (2002),** ***Modeling and Analysis of Congestion in the Design of Facility Layouts*****.** *Management Science*, 48, 679–704;

[4] **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 and Computer-Integrated Manufacturing*, 22, 217–238;

[5] **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 and Industrial Engineering*, 66, 438–450;

[6] **Chiang, C.P., Lee, S.D. (2004),** ***Joint Determination of Machine Cells and Linear Intercell Layout.*** *Computers and Operations Research*, 31, 1603–1619;

[7] **El-Baz, M.A. (2004),** ***A Genetic Algorithm for Facility Layout Problems of Different Manufacturing Environments*****.** *Computers and Industrial Engineering*, 47, 233–246;

[8] **Forghani, K., Mohammadi, M., Ghezavati, V., (2015),** ***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;

[9] **Garey, M.R., Johnson, D.S. (1979),** ***Computers and Intractability: A Guide to the Theory of NP-Completeness***. Freeman, San Francisco;

[10] **Gonçalves, J.F., Resende, M.G.C. (2004),** ***An Evolutionary Algorithm for Manufacturing Cell Formation*****.** *Computers and Industrial Engineering*, 47, 247–273;

[11] **Harhalakis, G., Nagi, R., Proth, J.M. (1990),** ***An Efficient Heuristic in Manufacturing Cell Formation for Group Technology Applications*****.** *International Journal of Production Research*, 28, 185–198;

[12] **Heragu, S.S., Kakuturi, S.R. (1997),** ***Grouping and Placement of Machine Cells*****.** *IIE Transactions*, 29, 561–571;

[13] **Ilić, O.R. (2014),** ***An E-Learning Tool Considering Similarity Measures for Manufacturing Cell Formation*****.** *Journal of Intelligent Manufacturing*, 25, 617–628;

[14] **Irani, S.A., Huang, H. (2006),** ***Cascading Flowlines and Layout Modules Practical Strategies for Machine Duplication in Facility Layouts*****.** *International Journal of Flexible Manufacturing Systems*, 17, 119–149;

[15] **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;

[16] **Kazerooni, M.L., Luong, H.S., Abhary, K. (1997),** ***A Genetic Algorithm Based Cell Design Considering Alternative Routing*****.** *International Journal of Computer Integrated Manufacturing Systems*, 10, 93–107;

[17] **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;

[18] **Mahdavi, I., Teymourian, E., Tahami Baher, N., Kayvanfar, V. (2013),** ***An Integrated Model for Solving Cell Formation and Cell Layout Problem Simultaneously Considering New Situations*****.** *Journal of Manufacturing Systems*, 32, 655–663;

[19] **Nair, G.J., Narendran, T.T. (1998),** ***CASE: A Clustering Algorithm for Cell Formation with Sequence Data*****.** *International Journal of Production Research*, 36, 157–179;

[20] **Paydar, M.M., Mahdavi, I., Sharafuddin, I., Solimanpur, M. (2010),** ***Applying Simulated Annealing for Designing Cellular Manufacturing Systems Using MDmTSP.*** *Computers and Industrial Engineering*, 59, 929–936;

[21] **Saeedi, S., Solimanpur, M., Mahdavi, I., Javadian, N. (2010), *Heuristic Approaches for Cell Formation in Cellular Manufacturing*.** *Journal of Software Engineering and Applications*, 3, 674–682;
[22] **Tompkins, J.A., White, J.A., Bozer, Y.A., Tanchoco, J.M.A. (2003), *Facilities Planning*.** 3rd Edn, *John Wiley and Sons*, New York;
[23] **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;
[24] **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;
[25] **Yin, Y., Yasuda, K., Hu, L. (2005), *Formation of Manufacturing Cells Based on Material Flows*.** *International Journal of Advanced Manufacturing Technology*, 27, 159–165.