SpringerPlus

RESEARCH Open Access

Solving a mathematical model integrating unequal-area facilities layout and part scheduling in a cellular manufacturing system by a genetic algorithm

Ahmad Ebrahimi, Reza Kia* and Alireza Rashidi Komijan

*Correspondence:
rezakia@iaufb.ac.ir
Department of Industrial Engineering, Firoozkooh Branch, Islamic Azad University, Firoozkooh, Iran

Abstract

In this article, a novel integrated mixed-integer nonlinear programming model is presented for designing a cellular manufacturing system (CMS) considering machine layout and part scheduling problems simultaneously as interrelated decisions. The integrated CMS model is formulated to incorporate several design features including part due date, material handling time, operation sequence, processing time, an intra-cell layout of unequal-area facilities, and part scheduling. The objective function is to minimize makespan, tardiness penalties, and material handling costs of inter-cell and intra-cell movements. Two numerical examples are solved by the Lingo software to illustrate the results obtained by the incorporated features. In order to assess the effects and importance of integration of machine layout and part scheduling in designing a CMS, two approaches, sequentially and concurrent are investigated and the improvement resulted from a concurrent approach is revealed. Also, due to the NP-hardness of the integrated model, an efficient genetic algorithm is designed. As a consequence, computational results of this study indicate that the best solutions found by GA are better than the solutions found by B&B in much less time for both sequential and concurrent approaches. Moreover, the comparisons between the objective function values (OFVs) obtained by sequential and concurrent approaches demonstrate that the OFV improvement is averagely around 17 % by GA and 14 % by B&B.

Keywords: Cellular manufacturing system, Machine layout, Part scheduling, Mixed-integer nonlinear programming, Genetic algorithm

Background

Nowadays in modern competitive manufacturing environments, each company will be required to be capable of reacting quickly to sudden unpredictable changes in a market. Hence, flexibility and efficiency in production have been the main targets of many manufacturing systems such as flexible manufacturing systems (FMS) and just-in-time (JIT) production (Adeil et al. 1996; Selim et al. 1997). One of the major approaches to enhance both flexibility and efficiency is cellular manufacturing (CM), which is an important application of group technology (GT) that handle the formation of manufacturing cells in a way that each part family is processed using a machine cell (Wemmerlov and Hyer

Springer Open

1986). Three major considerable steps in a successful design of a cellular manufacturing system (CMS) are: (1) cell formation (CF) (i.e., to group parts with similar processing requirements into part families and machines to machine cells); (2) group layout (GL) (i.e., to assign machines to workstations within each cell, called intra-cell layout, and cells arrangements within shop floor, called inter-cell layout), and (3) group scheduling (GS) (i.e., scheduling of part families) (Jajodia et al. 1992; Wu et al. 2007b).

The cell formation problem is an area that has been widely investigated in the literature (Soleimanpour et al. 2002). Detailed literature reviews can be found in the cell formation problems' review (Papaioannou and Wilson 2010). Cellular layout in the CMS design is the subject of some studies that has not received as much attention as cell formation problem in the past two decades (Wang et al. 2001). Most studies assume the first stage (CF) as a priority and then solve the inter-cell and intra-cell layout problems while some others assume the joint problem of CF and GL. Another decision in the CMSs is GS that has been studied by some researchers but only a few studies have attempted to join GS decision with other decisions (i.e., CF and GL). Recently-published articles considering these decisions are investigated and summarized in Table 1.

Due to the complexity and NP-complete nature of CF, GL, and GS decisions, most researchers have addressed two or three decisions sequentially or independently. However, the benefits gained from CMS implementation are highly affected by how three stages of the CMS design have been performed in collaboration with each other. Hence, all of these decisions should be addressed concurrently with the intention of obtaining the best results (Alfa et al. 1992; Bazargan-lari et al. 2000).

Ranjbar and Najafian Razavi (2012) proposed a new approach to concurrently make the layout and scheduling decisions in a job shop environment and developed a hybrid metaheuristic approach based on the scatter search algorithm. Ripon and Torresen (2013) presented a multi-objective evolutionary method based on a hybrid genetic algorithm by incorporating variable neighborhood search for solving job shop scheduling problem (JSSP) that considers transportation delays and facility layout planning (FLP) as an integrated problem. Halat and Bashirzadeh (2014) developed a concurrent approach for job shop cell scheduling to minimize the makespan in an integer linear programming model by considering exceptional elements, intercellular moves, intercellular transportation times, and sequence-dependent family setup times. They also developed a heuristic approach based on the genetic algorithm.

Wu et al. (2007b) have extended the mathematical models proposed in Wu et al. (2006, 2007a) to develop a new one which integrated three mentioned decisions. Then, they developed a hierarchical genetic algorithm (HGA) to solve the integrated cell design problem. The deficiencies of that model are inaccuracy in determining the layout of cells and probable overlapping of the cells. Tang et al. (2010) developed a scatter search approach to solve a nonlinear mathematical programming model for the problem of parts scheduling in a CMS by considering exceptional parts by minimizing the total weighted tardiness.

Arkat et al. (2012a) have promoted a similar integration by proposing two mathematical models, the first one integrates cellular layout with cell formation to determine optimal cell configuration and the layout of machines and cells in order to minimize the total movement costs. Considering the results obtained by solving this model integrating

Table 1 Studies integrating CF, machines layout, and parts scheduling decisions in CM design

References	Decisions	Integrating approach	Solution method
Chandrasekharan and Rajagopalan (1993)	CF, GL	Sequential	Non-metric multidimensional scaling
Jajodia et al. (1992)	CF, GL	Sequential	Simulated annealing (SA)
Salum (2000)	CF, GL	Sequential	Two phase method based on manufacturing lead time (MLT)
Urban et al. (2000)	CF, GL	Sequential	Mathematical model based on quadratic assignment problem (QAP) and network flow problem (NFP)
Alfa et al. (1992)	CF, GL	Concurrent	Mathematical model-SA
Bazargan-lari et al. (2000)	CF, GL	Concurrent	Three-phase approach
Wang et al. (2001)	CF, GL	Concurrent	Mathematical model-SA
Arvindh and Irani (1994)	CF, GL	Concurrent	An integrated framework
Akturk (1996)	CF, GL	Concurrent	Mathematical model
Chiang and Lee (2004)	CF, GL	Concurrent	SA
Mahdavi et al. (2008)	CF, GL	Concurrent	Heuristic based on flow matrix
Ahi et al. (2009)	CF, GL	Concurrent	Multiple attribute decision making (MADM)
Wu et al. (2006)	CF, GL	Concurrent	Genetic algorithm (GA)
Wu et al. (2007a)	CF, GL	Concurrent	Genetic algorithm (GA)
Wu et al. (2007b)	CF, GL, GS	Concurrent	Mathematical model–Hierarchical GA (HGA)
Mahdavi and Mahadevan (2008)	CF, GL	Concurrent	Heuristic approach
Solimanpur et al. (2004b)	CF, GL	Concurrent	Mathematical model based on QAP-Ant colony optimization (ACO)
Castillo and Westerlund (2005)	CF, GL	Concurrent	An *e*-accurate model
Jolai et al. (2012)	CF, GL	Concurrent	Electromagnetism algorithm
Xie and Sahinidis (2008)	CF, GL	Concurrent	Branch and bound
Javadi et al. (2013)	CF, GL	Concurrent	Mathematical model
Mohammadi and Forghani (2014)	CF, GL	Concurrent	Genetic algorithm (GA)
Sridhar and Rajendran (1993)	CF, GS	Sequential	Hybrid SA
Solimanpur et al. (2004a)	CF, GS	Sequential	Two-stage heuristic algorithm
Atmani et al. (1995)	CF, GS	Concurrent	Mathematical model
Franca et al. (2005)	CF, GS	Concurrent	Evolutionary algorithm
Reddy and Narendran (2003)	CF, GS	Concurrent	Heuristic
Leung et al. (2007)	CF, GS	Concurrent	Heuristic
Lin et al. (2009)	CF, GS	Concurrent	Tabu search (TS)–GA–SA
Hendizadeh et al. (2008)	CF, GS	Concurrent	Tabu Search (TS)
Tavakkoli-Moghaddam et al. (2008)	CF, GS	Concurrent	GA and Memetic algorithm
Tavakkoli-Moghaddam et al. (2010)	CF, GS	Concurrent	Scatter search (SS)
Chen and Cao (2004)	CF, GS	Concurrent	Mathematical model–TS
Arkat et al. (2012a, b)	CF, GL, GS	Sequential	Mathematical model–GA

cellular configuration and layout, the cell scheduling problem becomes a job shop scheduling problem with transportation times. The second model is based on the concurrent design and integrates the GS problem with CF and GL problems. Also, two genetic algorithms were developed to solve the real-sized problems. Arkat et al. (2012b) presented a multi-objective model to make decisions about cell formation, cellular layout and operation sequence simultaneously. The first objective was to minimize total transportation cost of parts and the second objective was to minimize makespan. A multi-objective genetic algorithm was used to solve the model. Zeng et al. (2015) proposed a two-stage GA-based heuristic algorithm to solve a nonlinear mathematical programming model to determine the sequences of the exceptional parts to be transferred via an automated guided vehicle (AGV) in order to minimize the process make-span.

As the main aim of this article, regarding the articles reviewed above, is proposing a new mathematical model with consisting of important manufacturing features such as operation sequence, processing time, transferring time, intra-cell layout and parts scheduling. The presented model is different from the existing models available in the literature because of incorporating some important design aspects simultaneously. In the first aspect, the layout of machines with unequal-areas in the cells is not restricted to linear type. However, dimensions of cells are predetermined by a system designer in a shop floor with a continuous area. In the second aspect, process routings for part types can be flexible. In the third aspect, time and cost of movement are depended on three factors: movement distance, part type, and movement type (inter-cell or intra-cell). In the fourth aspect, despite the fact that three ingredients have been formulated in the objective function including makespan, penalty cost, and material handling costs, however, only one, two, or three of them can be used to form cells based on the desired objective. Finally, in the fifth aspect, the CF, machines layout, and parts scheduling decisions are made simultaneously by an integrated model.

Another aim of this article is developing an efficient genetic algorithm enhanced by a matrix-based chromosome structure consisting of two sections for layout and scheduling, a heuristic procedure generating initial feasible solutions, a procedure calculating the fitness functions of generated solutions, and efficient crossover and mutation operators in order to determine three interrelated stages in designing a CMS simultaneously.

The remainder of this article is organized as follows. In "Mathematical model" section, a mathematical model integrating CMS, machines layout and parts scheduling decisions is formulated. The development of the designed GA is discussed in "Designed genetic algorithm for the integrated layout and scheduling model of a CMS" section. "Experimental results" section illustrates the test problems that are utilized to investigate the features of the proposed model and the performance of the developed algorithm. Finally, a conclusion is given in "Conclusion" section.

Mathematical model

Model assumptions

In this section, a mathematical model is formulated to minimize three ingredients in the objective function including makespan, tardiness penalty, and inter-cell and intra-cell material handling costs under the following assumptions:

1. Each part type has several operations which should be processed in a given sequence. Also, the processing capabilities and processing times of part-operations for each machine type are known and deterministic.
2. All machine types are assumed to be multi-purpose ones which are capable of performing one or more operations without imposing a reinstalling cost. In like manner, each operation of a part type can be performed on different machine types with dif-

ferent processing times. This feature providing flexibility to process plan of parts has been known as alternative process routings. Nevertheless, a part operation should be processed by only one of those machines which are capable of processing that operation.

3. A machine cannot process more than one part at the same time.
4. Parts are moved individually by material handling devices between and within cells. Inter-cell movement happens whenever successive operations of a part type are carried out in different cells. Also, the intra-cell movement happens whenever successive operations of a part type are processed on different machines in the same cell.
5. The rectangular facilities of unequal-areas can be located anywhere in the cells having a predetermined shape with a continual space without any overlaps. In fact, the inter-cell layout and distances between cells are given and intra-cell layout is determined by the model.
6. Each planar cell has a rectangular shape whose length and width is known in advance. Also, the number of cells to be formed is given.
7. The maximum and minimum limit of the cell size in terms of the number of machines is known.
8. The loading and unloading point is at the center of each machine.
9. Machines have a predetermined orientation (i.e., machines may be located either horizontally or vertically). The machine is horizontally located if the longer side of the machine is parallel to the x-axis. On the contrary, the machine is vertically located if the longer side is parallel to the y-axis.
10. The rectilinear distance between the centers of two facilities *i* and *j* with coordinates (x_i, y_i) and (x_j, y_j) is considered to be the distance norm: $d_{ij} = |x_i - x_j| + |y_i - y_j|$.
11. Once an operation of a part starts to be processed on a machine, it cannot be interrupted before being completed.
12. Due date is determined for each part. As a result, tardiness penalty is incurred for each part type per time unit if it is not completed before its due date.
13. Cost and time of handling a part between two locations in the same cell or between different cells depend on three factors: the distance between locations of machines, type of part, and type of movement (inter-cell or intra-cell). Hence, for each part type, three coefficients per distance unit are considered: movement time, inter-cell movement cost and intra-cell movement cost.
14. Machine setup time is negligible.
15. The machines will never breakdown and be available throughout the scheduling period.

The notations used in the model are presented below:

Sets

$i = \{1, 2, \ldots, P\}$	index of parts
$j, j' = \{1, 2, \ldots, M\}$	index of machines
$c, c' = \{1, 2, \ldots, C\}$	index of cells
$k = \left\{1, 2, \ldots, K_p\right\}$	index of operations for part type *p*
$k', k'' = \{1, 2, \ldots, K_m\}$	index of processing positions for machine type *m*

Model parameters

Q	factory costs per time unit
Ply_i	tardiness penalty for part type i per time unit
dd_i	due date for part type i
CO_i	inter-cell material handling cost for part type i per distance unit
CI_i	intra-cell material handling cost for part type i per distance unit
TT_i	material handling time for part type i per distance unit
UBC	upper cell size limit
LBC	lower cell size limit
T_{kij}	processing time of operation k of part type i on machine j
a_{kij}	1 if operation k of part type i can be processed on machine type j; 0 otherwise (i.e., a_{kij} is 1 if $T_{kij} > 0$; 0 otherwise)
L_j	length of the horizontal side of machine type j
H_j	height of the vertical side of machine type j
LX_c	horizontal coordinate of left side of cell c
RX_c	horizontal coordinate of right side of cell c
LY_c	vertical coordinate of lower side of cell c
UY_c	vertical coordinate of upper side of cell c
M	a big positive number

Decision variables

V_{jc}	1 if machine j is assigned to cell c,0 otherwise
$Z_{ki}^{k'j}$	1 if kth operation of part type i is processed at k'th processing position on machine j, 0 otherwise
$CTM_{k'j}$	completion time of k' th processing position of machine j
CTP_i	completion time of part type i
$C_{\max}$	makespan time
α_j	horizontal coordinate of center of machine j
β_j	vertical coordinate of center of machine j
x_j'	horizontal coordinate of left side of machine j
x_j''	horizontal coordinate of right side of machine j
y_j'	vertical coordinate of lower side of machine j
y_j''	vertical coordinate of upper side of machine j
$R_{jj'}^X$	1 if machine j is completely located right side of machine j', 0 otherwise
$R_{jj'}^Y$	1 if machine j is completely located upper side of machine j', 0 otherwise
$d_{jj'}$	distance between machine j and j'

Mathematical model

The mix-integer nonlinear mathematical model is presented as follow:

Minimize

$$Z = Q \times C_{\max} \tag{1.1}$$

$$+\sum_{i=1}^{P} \left(Ply_i \times \max\left\{0, CTP_i - dd_i\right\}\right) \tag{1.2}$$

$$
\begin{aligned}
&+\sum_{k=2}^{K_p}\sum_{i=1}^{P}\sum_{j=1}^{M}\sum_{j'\neq j}\sum_{c=1}^{C}\left(\left(\sum_{k'=1}^{K_m} Z_{ki}^{k'j}\right)\right.\\
&\left.\times\left(\sum_{k'=1}^{K_m} Z_{k-1i}^{k'j'}\right)\times\left[\left(V_{jc}\times V_{j'c}\times CI_i\right)+\left(V_{jc}\times\left(1-V_{j'c}\right)\times CO_i\right)\right]\times d_{jj'}\right)
\end{aligned}
\tag{1.3}
$$

Subject to:

$$\sum_{c=1}^{C} V_{jc}=1 \quad \forall j \tag{2}$$

$$\sum_{j=1}^{M} V_{jc}\leq UBC \quad \forall c \tag{3}$$

$$\sum_{j=1}^{M} V_{jc}\geq LBC \quad \forall c \tag{4}$$

$$\sum_{k'=1}^{K_m}\sum_{j=1}^{M} Z_{ki}^{k'j}=1 \quad \forall\, k,i \tag{5}$$

$$\sum_{k'=1}^{K_m} Z_{ki}^{k'j}\leq a_{kij} \quad \forall k,i,j \tag{6}$$

$$CTM_{1j}=\sum_{i=1}^{P}\left(\sum_{k=2}^{K_p} Z_{ki}^{1j}\times\left[T_{kij}+\sum_{k''=1}^{K_m}\sum_{j'=1}^{M} Z_{k-1i}^{k''j'}\left(CTM_{k''j'}+\left(d_{jj'}\times TT_i\right)\right)\right]+Z_{1i}^{1j}\cdot T_{1ij}\right) \quad \forall j \tag{7}$$

$$
\begin{aligned}
CTM_{k'j}=\sum_{i=1}^{P}&\left(\sum_{k=2}^{K_p} Z_{ki}^{k'j}\times\left[T_{kij}+\mathrm{Max}\left\{CTM_{k'-1j},\quad \sum_{k''=1}^{K_m}\sum_{j'=1}^{M} Z_{k-1i}^{k''j'}\left(CTM_{k''j'}+\left(d_{jj'}\times TT_i\right)\right)\right\}\right]\right.\\
&\left.+Z_{1i}^{k'j}\times\left[T_{1ij}+CTM_{k'-1j}\right]\right) \quad \forall k'>1,j
\end{aligned}
\tag{8}
$$

$$CTP_i=\sum_{k'=1}^{K_m}\sum_{j=1}^{M} Z_{kpi}^{k'j}\cdot CTM_{k'j} \quad \forall i \tag{9}$$

$$C_{\mathrm{max}}=Max\{\forall i: CTP_i\} \tag{10}$$

$$\alpha_j=\frac{1}{2}\left(x_j'+x_j''\right) \quad \forall j \tag{11}$$

$$\beta_j = \frac{1}{2}\left(y'_j + y''_j\right) \quad \forall j \tag{12}$$

$$x''_j - x'_j = L_j \quad \forall j \tag{13}$$

$$y''_j - y'_j = H_j \quad \forall j \tag{14}$$

$$d_{jj'} = \left|\alpha_j - \alpha_{j'}\right| + \left|\beta_j - \beta_{j'}\right| \quad \forall j, j' \quad and \quad j \neq j' \tag{15}$$

$$x''_j \leq RX_c + \left(1 - V_{jc}\right)M \quad \forall j, c \tag{16}$$

$$LX_c \leq x'_j + \left(1 - V_{jc}\right)M \quad \forall j, c \tag{17}$$

$$y''_j \leq UY_c + \left(1 - V_{jc}\right)M \quad \forall j, c \tag{18}$$

$$LY_c \leq y'_j + \left(1 - V_{jc}\right)M \quad \forall j, c \tag{19}$$

$$R^X_{jj'} + R^X_{j'j} + R^Y_{jj'} + R^Y_{j'j} \geq 1 \quad \forall j, j' \quad and \quad j \neq j' \tag{20}$$

$$x''_{j'} \leq x'_j + \left(1 - R^X_{jj'}\right)M \quad \forall j, j' \quad and \quad j \neq j' \tag{21}$$

$$x''_j \leq x'_{j'} + \left(1 - R^X_{j'j}\right)M \quad \forall j, j' \quad and \quad j \neq j' \tag{22}$$

$$y''_{j'} \leq y'_j + \left(1 - R^Y_{jj'}\right)M \quad \forall j, j' \quad and \quad j \neq j' \tag{23}$$

$$y''_j \leq y'_{j'} + \left(1 - R^Y_{j'j}\right)M \quad \forall j, j' \quad and \quad j \neq j' \tag{24}$$

$$Z^{k'j}_{ki}, V_j \ is \ Binary \quad \forall k, i, k', j \tag{25}$$

$$CTM_{k'j}, CTP_i, C_{\max} \geq 0 \quad \forall k', j \tag{26}$$

$$d_{jj'}, \alpha_j, \beta_j, x'_j, x''_j, y'_j, y''_j \geq 0 \quad \forall j, j'j \neq j' \tag{27}$$

$$LX_c, UX_c, LY_c, UY_c \geq 0 \quad \forall c \tag{28}$$

$$R_{jj'}^{X}, R_{jj'}^{Y} \text{ is Binary} \quad \forall j, j' \quad j \neq j' \tag{29}$$

The objective function consists of three terms. Term (1.1) contains makespan time and calculates factory costs during makespan time. Term (1.2) is total tardiness penalties for all parts which have been completed after their due date. Terms (1.3) calculates the inter-cell and intra-cell material handling costs.

Constraints sets (2) ensure that each machine is assigned to only one cell. The cell size limits specified by a designer are enforced through Constraints (3) and (4). Constraint set (5) ensures that each operation of each part is processed by only one machine at one of its processing positions. Constraint set (6) ensures an operation of a part is processed by a machine at one of its processing positions provided that machine is capable of processing the related operation.

Constraints (7) compute completion time of the first processing position of a machine in which an operation of a part is processed. There are two cases: (1) the first operation of part *i*is processed at the first processing position of machine *j*. It can be simply understood that completion time of the first processing position of machine *j*, in this case, is equal to the processing time of the first operation of part *i*; (2) any operation of part *i* except the first one is processed at the first processing position of machine *j*. In this case, the part needs to be moved from the machine j' processing the previous operation (i.e., $k - 1$) to the current machine *j* processing the current operation (i.e., *k*). Considering the distance between machines *j* and j', this movement needs a handling time equal to $\left(d_{jj'} \times TT_i\right)$. Since the previous operation $k - 1$ of part *i* has been finished at the time $CTM_{k''j'}$, the part *i* will be ready for processing the operation *k* on machine *j* at time $\left(d_{jj'} \times TT_i\right) + CTM_{k''j'}$. As a result, in this case, the completion time of the first processing position of machine *j* is when processing the operation *k* of part *i* on machine *j* is finished and it is equal to $T_{kij} + \left(d_{jj'} \times TT_i\right) + CTM_{k''j'}$.

Constraints (8) compute completion time of any processing position except the first one of machine *j* in which an operation of a part is processed. Similarly, there are two cases: (1) the first operation of part *i* is processed at the processing position k' of machine *j*. It can be simply understood that completion time of the processing position k' of machine *j*, in this case, is equal to processing time of the first operation of part *i* on machine *j* plus the completion time of the previous processing position $k' - 1$ of machine *j*; (2) any operation of part *i* except the first one is processed at the processing position *k* of machine *j*. In this case, as it was similarly explained for Constraint (7), the ready time to processes the operation *k* of part *i* on machine *j* is $\left(d_{jj'} \times TT_i\right) + CTM_{k''j'}$. In addition, machine *j* should be idle to process that operation at its processing position k'. It means that the completion of the previous processing position $k' - 1$ of machine *j* should have been reached. To satisfy the limitations of part readiness and machine idleness, the time which is equal to the maximum of $CTM_{k'-1j}$ and $\left(d_{jj'} \times TT_i\right) + CTM_{k''j'}$ is considered as an actual starting time to process that operation of part *i*. As a result, the completion time of the processing position k' of machine *j* in which operation *k* of part *j* is processed is equal to $T_{kij} + \text{Max}\left\{CTM_{k'-1j}, \sum_{k''=1}^{K_m} \sum_{j'=1}^{M} Z_{k-1i}^{k''j'}\left(CTM_{k''j'} + \left(d_{jj'} \times TT_i\right)\right)\right\}$. Constraints (7) and (8) also enforce a machine to not process more than one part at the same time.

Constraint set (9) computes completion time for each part. Constraint set (10) returns makespan time based on the computed completion times of all parts.

Constraints (11) and (12) represent the horizontal and vertical coordinates of the center of each machine, respectively. In the other hand, Constraint sets (13) and (14) return the coordinates of four sides of a machine based on its center coordinates and its length and height. Constraint set (15) computes the rectilinear distance between the centers of two machines j and j'.

Constraint sets (16)–(19) ensure that machines assigned to a cell are placed with regard to coordinates of cell sides (vertically and horizontally). Constraint sets (20)–(24) ensure that machines do not overlap in the horizontal and vertical direction simultaneously. Finally, Constraint sets (25)–(29) are the logical binary and non-negativity necessities on the binary and positive continuous decision variables.

Linearization techniques

The proposed model is nonlinear in both objective function and constraints sets. Hence, some linearization techniques are proposed to convert the nonlinear model into a linearized counterpart as follows:

The non-line function *Max* in the Eq. (1.2) of objective function can be linearized usingthe following transformation $trdns_i = \max\{0, CTP_i - dd_i\}$ under the below set of constraints:

$$trdns_i \geq 0 \quad \forall i \tag{30}$$

$$trdns_i \geq CTP_i - dd_i \quad \forall i \tag{31}$$

Linearization of function *Max* in Eqs. (8) and (10) in constraint sets is exactly similar to that of Eq. (1.2).

The nonlinear absolute components in constraint set (15) can be linearized using the following transformations $|\alpha_j - \alpha_{j'}| = \alpha^{+}_{jj'} + \alpha^{-}_{jj'}$ and $|\beta_j - \beta_{j'}| = \beta^{+}_{jj'} + \beta^{-}_{jj'}$ under the below set of equations:

$$\alpha_j - \alpha_{j'} = \alpha^{+}_{jj'} - \alpha^{-}_{jj'} \quad \forall j, j' \tag{32}$$

$$\beta_j - \beta_{j'} = \beta^{+}_{jj'} - \beta^{-}_{jj'} \quad \forall j, j' \tag{33}$$

$$\alpha^{+}_{jj'}, \alpha^{-}_{jj'} \beta^{+}_{jj'}, \beta^{-}_{jj'} \geq 0 \quad \forall j, j' \tag{34}$$

There are some product terms between binary variables in Eq. (1.3) which make the model non-linear. For instance, let us introduce a new binary variable $ZZ_{kik'jj'}$ which equals to multiplying both binary summations $\sum_{k'} Z^{k'j}_{ki}$ and $\sum_{k'} Z^{k'j}_{k-1i}$ as given in Eq. (1.3) of the objective function. This nonlinear component can be linearized under the following constraint sets:

$$ZZ_{kijj'} \geq \sum_{k'} Z^{k'j}_{ki} + \sum_{k'} Z^{k'j'}_{k-1i} - 1 \quad \forall k, i, k', j, j' (j \neq j') \tag{35}$$

$$ZZ_{kijj'} \leq \sum_{k'} Z^{k'j}_{ki} \forall k, i, k', j, j' (j \neq j') \tag{36}$$

$$ZZ_{kijj'} \leq \sum_{k'} Z_{k-1i}^{k'j'} \quad \forall k, i, k', j, j' \left(j \neq j'\right) \tag{37}$$

Linearization of the other nonlinear terms multiplying binary variables in Eq. (1.3) is similarly done as explained at the above.

There are still some nonlinear terms in the model multiplying binary variables by continuous variables in Eqs. (7)–(9). For example, to linearize the product term $CTM_{k''j'} \cdot Z_{k-1i}^{k''j'}$ in Eq. (7), a new positive continuous variable $ZCTM_{kik''j'}$ which equals to $CTM_{k''j'} \cdot Z_{k-1i}^{k''j'}$ is introduced and the product term can be linearized under the following constraint sets:

$$ZCTM_{kik''j'} \geq CTM_{k''j'} - M \cdot \left(1 - Z_{k-1i}^{k''j'}\right) \quad \forall k, i, k'', j, j' \left(j \neq j'\right) \tag{38}$$

$$ZCTM_{kik''j'} \leq CTM_{k''j'} \quad \forall k, i, k'', j, j' \left(j \neq j'\right) \tag{39}$$

$$ZCTM_{kik''j'} \leq M \cdot Z_{k-1i}^{k''j'} \quad \forall k, i, k'', j, j' \left(j \neq j'\right) \tag{40}$$

Linearization of the other nonlinear terms multiplying binary variables by continuous variables in Eqs. (7)–(9) is similarly done.

Designed genetic algorithm for the integrated layout and scheduling model of a CMS

The GA is an evolutionary search and optimization technique considering the design process as an evolutionary one. It seeks to find the best solution by generating a population of candidate individuals as the current parents. Using a selection mechanism, crossover and mutation operators, solutions (i.e., offsprings) with more fitness values are expected to be generated from the initial population of parents during successive generations. These generations continue until the algorithm finds an acceptable good solution or meets a terminating condition. Genetic algorithms have been implemented in a wide variety of engineering optimization applications (Gen and Cheng 1997; Man et al. 1999), including cellular manufacturing systems (Shiyas and Pillai 2014; Deljoo et al. 2010; Defersha and Chen 2008; Wu et al. 2007a, b; Kia et al. 2014; Vin and Delchambre 2014).

In this section, a genetic algorithm for solving the integrated layout and scheduling model of a CMS is employed. Principle factors for designing the employed GA are described as follows.

Chromosome structure

As represented in the Fig. 1, the structure of each chromosome is a multi-string, where the number of strings is equal to the number of existing machines and the length of each chromosome stringis $K'_m + 1$. There are two separate sections of information including (1) layout gene and, (2) K'_m schedulinggenes.

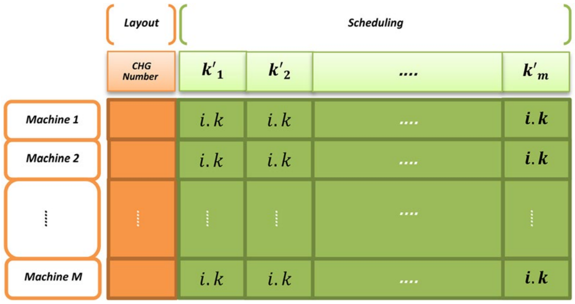


Fig. 1 Chromosome structure

In the first section, each layout gene shows an integer number of set $S_j = \{1, 2, \ldots, TS_j\}$ determining the coordinate location of each machine. The definition of set S_j is discussed at below.

The existing cells are divided according to the constant values of length and width of square grids that contain square houses named as Cell Houses (CH's) and are defined based on a multiplier of the length and width unit. Also, these CH's are divided into smaller grids named as Grid Cells (GC's). For example, if the length and width of cells are measured in the meter and the division multiplier is $\sigma = 2$, it means the cells are divided into 0.5 × 0.5 CH's. As another example, if there is a cell with the dimension of 10 × 10 m and a multiplier $\sigma = 2$, a grilled cell is obtained as shown in Fig. 2.

The total number of CH's in a system with C cells is equal to:

$$TCH_C = \sum_{c=1}^{C} (UX_c - LX_c)(UY_c - LY_c)\sigma^2 \quad (41)$$

For example, in the above sample having a cell with dimension of 10 × 10 m, the total number of CH's is equal to $TCH_1 = 100 \times 2^2 = 400$.

Now, by considering constant length and width values of machines, each machine can be located in finite positions. Each position involves a group of CH's named as Cell Houses Groups (CHG's). The entire positions that a machine j can be located in CG's (CHG's) are defined in set $S_j = \{1, 2, \ldots, TS_j\}$, where the total number of positions is calculated as follows:

$$TS_j = \sum_{c=1}^{C} \left[\left(UX_c - LX_c - L_j\right)\sigma + 1\right]\left[\left(UY_c - LY_c - W_j\right)\sigma + 1\right] \quad (42)$$

CH's and CHG's are numbered from the left-down corner to the right-up corner according to the cell numbers. After numbering a cell, the continuous numbering CH's and CHG's starts from the next cell and is continued until the last cell.

For example, if there are two cells with dimension of 10 × 10 m and 8 × 10 m and a 4 × 2 m machine j, the total number of CH's and candidate locations in set S for this machine are calculated as follows:

$$TCH_2 = (100 \times 4) + (80 \times 4) = 720$$

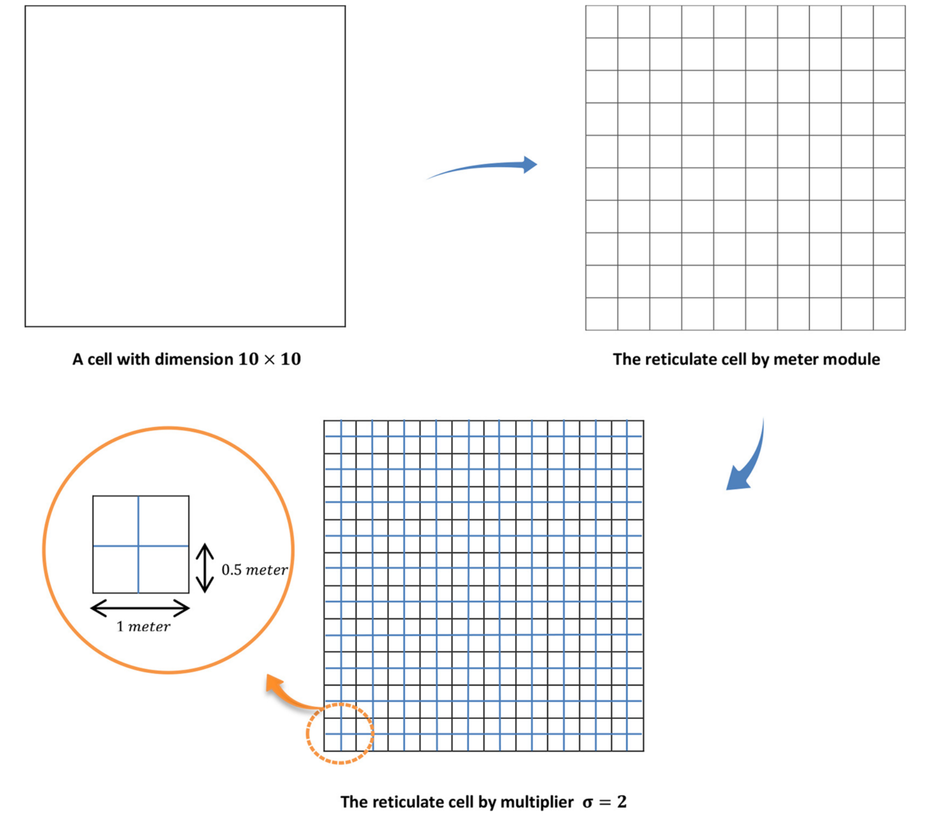

A cell with dimension 10 × 10
The reticulate cell by meter module
0.5 meter
1 meter
The reticulate cell by multiplier σ = 2

Fig. 2 Dividing a cell space into grid cells

$$TS_j = (13 \times 17) + (9 \times 17) = 374$$

$$S_j = \{1, 2, \ldots, 374\}$$

As it was mentioned earlier, each CHG's contains a set of CH's. For instance, in the example shown in Fig. 3, the set S_{109} contains CH's including {165, 166, 167, 168, 169, 170, 171, 172, 185, 186, 187, 188, 189, 190, 191, 192, 205, 206, 207, 208, 209, 210, 211, 212, 225, 226, 227, 228, 229, 230, 231, 232}. Among these CH's, the first house in the left-down corner is considered as a Main House (MH) of each set of CH's (Fig. 4).

The scheduling genes in the second section of each chromosome string are order-based permutation of part operation. In this section, the value in each gene contains information related to part number (i) and part operation (k) that is shown as a decimal number ($i.k$) without having any mathematical value. The first number represents part number and the second one shows part operation.

In the following, the calculation formulas for defined parameters in chromosome structure are discussed:

The total number of CH's for C cells is equal to:

$$TCH_C = \sum_{c=1}^{C} (UX_c - LX_c)(UY_c - LY_c)\sigma^2 \tag{43}$$

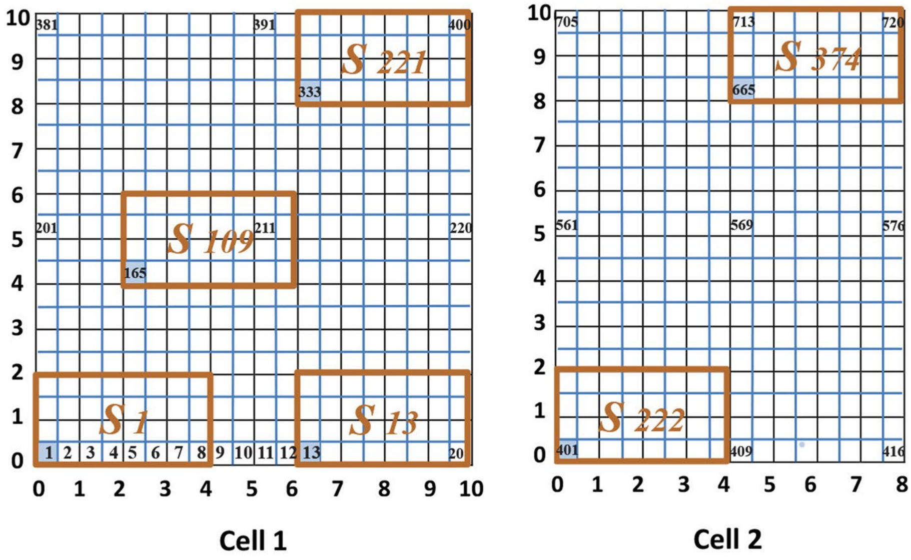


Fig. 3 Numbering CHG's

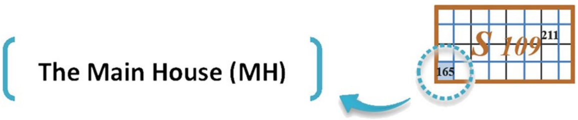


Fig. 4 MH of set S_{109}

The number of locations in a set of CHG's for machine *j* is

$$TS_j = \sum_{c=1}^{C} \left[\left(UX_c - LX_c - L_j \right) \sigma + 1 \right] \left[\left(UY_c - LY_c - W_j \right) \sigma + 1 \right] \tag{44}$$

$$S_j = \left\{ 1, 2, \ldots, TS_j \right\} \tag{45}$$

There are two cases to calculate the cell number where machine *j* is located based on CHG number of machine *j*. In the first case, the cell number where machine *j* is located is one ($c = 1$) if:

$$CHG\ number \leq \left[\left(UX_1 - LX_1 - L_j \right) \sigma + 1 \right] \left[\left(UY_1 - LY_1 - W_j \right) \sigma + 1 \right] \tag{46}$$

And in the second case, the cell number where machine *j* is located is $c\,(c > 1)$ if:

$$CHG\ number > \sum_{c=1}^{C-1} \left[\left(UX_c - LX_c - L_j \right) \sigma + 1 \right] \left[\left(UY_c - LY_c - W_j \right) \sigma + 1 \right] \tag{47}$$

Similarly, there are two cases to calculate the cell number where machine *j* is located based on CH number of selected CHG for machine *j*. In the first case, the cell number where machine *j* is located is one ($c = 1$) if:

$$CH\ number \leq (UX_1 - LX_1)(UY_1 - LY_1)\sigma^2 \tag{48}$$

And in the second case, the cell number where machine *j* is located is *c* (*c* > 1) if:

$$CH\ number > \sum_{C=1}^{C-1} (UX_c - LX_c)(UY_c - LY_c)\sigma^2 \tag{49}$$

The *MH* number calculation based on *CHG* number and cell number *c* is done as follows:

$$\begin{cases} MH\ number = L_j\sigma(CHG\ number - 1) + 1 \quad if\ c = 1 \quad and \quad L_j = UX_c - LX_c \\ MH\ number = CHG\ number + \left[\frac{CHG\ number}{(UX_c - LX_c - L_j)\sigma + 1} - e\right](L_j\sigma - 1) \quad if\ c = 1 \quad and \quad L_j \neq UX_c - LX_c \end{cases} \tag{50}$$

$$\begin{cases} NOSC = CHG\ number - \sum_{c=1}^{C-1} \left[(UX_c - LX_c - L_j)\sigma + 1\right]\left[(UY_c - LY_c - W_j)\sigma + 1\right] \quad if \quad c > 1 \\ MH\ number = L_j\sigma(NOSC - 1) + TCH_{c-1} + 1 \quad ifc > 1 \quad and \quad L_j = UX_c - LX_c \\ MH\ number = NOSC + \left[\frac{NOSC}{(UX_c - LX_c - L_j)\sigma + 1} - e\right](L_j\sigma - 1) + TCH_{c-1} \quad if \quad c > 1 \quad and \quad L_j \neq UX_c - LX_c \end{cases} \tag{51}$$

where *e* is a very small number and sign [ ] represents a floor function.

The calculation of *CH* numbers of each *CHG* based on the *MH* number and cell number would be as follows if it is defined in set *H*:

$$H = \left\{ MH\ number + k + (UX_c - LX_c)m\sigma \middle| (k, m) : \begin{array}{ccc} k = 0 & to & k = L_j\sigma - 1 \\ m = 0 & to & m = W_j\sigma - 1 \end{array} \right. \tag{52}$$

The horizontal and vertical coordinates of machine *j* are calculated based on the number *MH* and related occupied *CHG* as below.

$$\begin{cases} \alpha_j = LX_c + \frac{MH\ number}{\sigma} - \left[\frac{MH\ number}{(UX_c - LX_c)\sigma}\right](UX_c - LX_c) + \frac{L_j}{2} - \frac{1}{\sigma} \\ \beta_j = LY_c + \frac{\left[\frac{MH\ number}{(UX_c - LX_c)\sigma}\right]}{\sigma} + \frac{W_j}{2} \end{cases} \quad if\ c = 1 \tag{53}$$

$$\begin{cases} \alpha_j = LX_c + \frac{MH\ number - TCH_{c-1}}{\sigma} - \left[\frac{MH\ number - TCH_{c-1}}{(UX_c - LX_c)\sigma}\right](UX_c - LX_c) + \frac{L_j}{2} - \frac{1}{\sigma} \\ \beta_j = LY_c + \frac{\left[\frac{MH\ number - TCH_{c-1}}{(UX_c - LX_c)\sigma}\right]}{\sigma} + \frac{W_j}{2} \end{cases} \quad if\ c > 1 \tag{54}$$

Where sign [ ] represents a floor function.

The initial population

The next step after determination of chromosome structure is generating an initial population of chromosomes. In this section, the hierarchical procedure for filling the layout genes and scheduling genes of a chromosome is explained separately.

At first, the filling procedure of scheduling genes with a corresponding pseudo code, shown in Fig. 5, is defined according to the following steps:

```
Start
      For   “Population size”   Do
        Step1;
        Step2;
    End For
End
```

Fig. 5 Pseudo code of filling the scheduling genes

```
Start
Step1;
    For  “Every chromosome in the population”  Do
          While (NOM<M)  Do%for assigning all machines to the cells%
             Set NOM=0;
              Set population(“The genes in the layout chromosome part”)=0;
              For  “ The number of machines” Do
                 Step1;
                 Step2;
                 Step3;
                 Step4;
Step5;
                If   LMC ≤   “The number of machines assigned to cell (step5)”  ≤  UMC
 Step6;
Step7;
                If   “The selected machines don’t overlap the other machines”
               Set  NOM=NOM+1;
            SetPopulation(the layout gene related to machine j) = The gained valuein Step 4
            End For
         End While
    End For
End
```

Fig. 6 Pseudo code of filling the layout genes

Step 1 The operations of all parts are permuted on machines randomly based on decimal numbers *(i.k)* in scheduling genes.

Step 2 A machine is chosen randomly among the machines that do the same operation based on Constraint (5). This means the information of the part and the operations that are related to it should appear only once in each chromosome.

Next, the filling procedure for layout genes with a corresponding pseudo code, shown in Fig. 6, is defined according to the following steps:

Step 1 The value for all CH's is calculated according to the dimension of existing cells and multiplier σ.

Step 2 One machine *j* is selected randomly from the unselected machines.

Step 3 *CHG's* (i.e., series S_j) are calculated through formulas (44) and (45) for the selected machine *j*.

Step 4 A *CHG* from set S_j is selected randomly and is placed in the layout gene.

Step 5 The cell number where machine *j* is located is obtained through formulas (46) and (47) based on the *CHG* number or through formulas (48) and (49) based on *CH* number of *CHG* selected for machine *j*.

Step 6 The MH number is obtained through formulas (50) and (51) based on the cell number and the *CHG* number that the machine *j* has occupied.

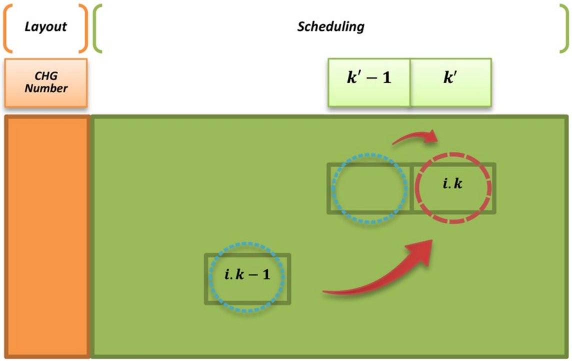


Fig. 7 Evaluating makespan ingredient of objective function for a scheduling gene

Step 7 CH numbers of selected *CHG* for the machine *j* is obtained through formula (52) based on the *MH* number and cell number.

Calculation procedure of fitness function

In this section, the procedure of calculating objective function value is described. Since there are three different ingredients in the objective function, including makespan, tardiness penalty costs and inter-cell/intra-cell movements' costs, they are calculated in three phases to evaluate the objective function for each chromosome.

In the first phase, for evaluating makespan ingredient of the objective function, it is needed some modifications on the chromosome to make the calculation of makespan possible. This is because by considering the constraints (7) and (8), the calculation of the completion time of each processing position k' of machine j ($CTM_{k'j}$) is possible provided that two conditions:

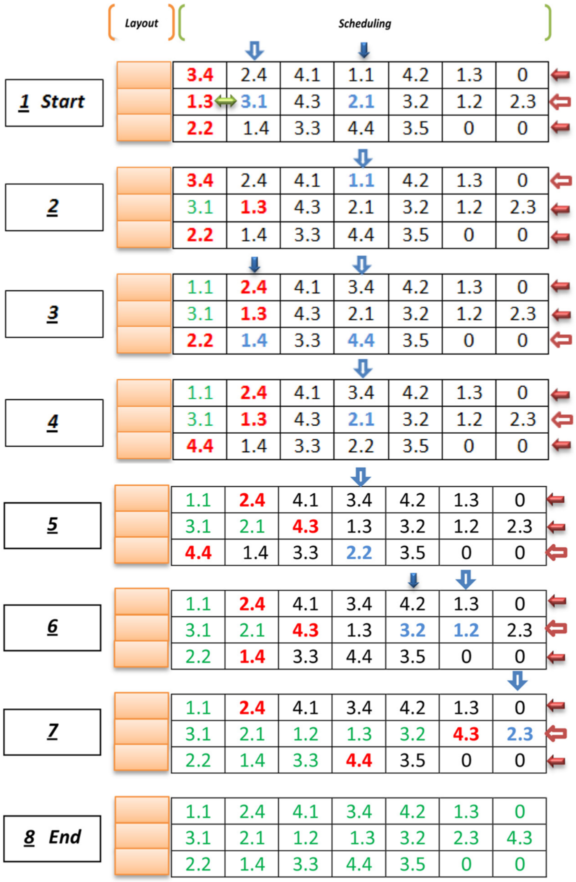
Firstly, the completion time of processing position $k' - 1$ of machine j is calculated in the chromosome. Secondly, the completion time of operation $k - 1$ of part i should be gained in the chromosome if operation k of part i is processed in the processing position k' of machine j (Fig. 7).

Therefore, the following algorithm is presented for modifying permutation in scheduling section of each chromosome until it would be possible to evaluate the makespan in each chromosome.

Step 1 In the scheduling section of each chromosome, the genes for which it would be possible to evaluate their completion time are evaluated by considering two mentioned conditions.

Step 2 For each chromosome string, the first gene that its completion time has not been calculated yet is considered. Then, one of them is chosen randomly.

Step 3 the chromosome strength at was selected in step 2 is considered. Amongt he other genes that their completion time in that chromosome has not gained yet, the gene that has the minimum decimal number is chosen and is replaced with the selected gene in step 2. If there is more than one gene that has the minimum decimal number, one of them is chosen randomly and replaced with the selected gene in step 2.



1 Start	3.4	2.4	4.1	1.1	4.2	1.3	0
	1.3	3.1	4.3	2.1	3.2	1.2	2.3
	2.2	1.4	3.3	4.4	3.5	0	0
2	3.4	2.4	4.1	1.1	4.2	1.3	0
	3.1	1.3	4.3	2.1	3.2	1.2	2.3
	2.2	1.4	3.3	4.4	3.5	0	0
3	1.1	2.4	4.1	3.4	4.2	1.3	0
	3.1	1.3	4.3	2.1	3.2	1.2	2.3
	2.2	1.4	3.3	4.4	3.5	0	0
4	1.1	2.4	4.1	3.4	4.2	1.3	0
	3.1	1.3	4.3	2.1	3.2	1.2	2.3
	4.4	1.4	3.3	2.2	3.5	0	0
5	1.1	2.4	4.1	3.4	4.2	1.3	0
	3.1	2.1	4.3	1.3	3.2	1.2	2.3
	4.4	1.4	3.3	2.2	3.5	0	0
6	1.1	2.4	4.1	3.4	4.2	1.3	0
	3.1	2.1	4.3	1.3	3.2	1.2	2.3
	2.2	1.4	3.3	4.4	3.5	0	0
7	1.1	2.4	4.1	3.4	4.2	1.3	0
	3.1	2.1	1.2	1.3	3.2	4.3	2.3
	2.2	1.4	3.3	4.4	3.5	0	0
8 End	1.1	2.4	4.1	3.4	4.2	1.3	0
	3.1	2.1	1.2	1.3	3.2	2.3	4.3
	2.2	1.4	3.3	4.4	3.5	0	0

Fig. 8 An example for calculation of fitness function

Step 4 The first, second and third steps are repeated until the completion times of all genes ($CTM_{k'j}$) are obtained. Following these steps and considering the constraints (9) and (10), the (CTP_i) values and the makespan value ($C_{\max}$) are returned.

An example for calculation of fitness function is illustrated in Fig. 8, where there is a chromosome with three machines and four parts. In this example, the scheduling genes of the presented chromosome are reordered in eight steps. However, because of random selections in steps 2 and 3, the number of these steps could be more or less.

In the second phase, by considering the amounts of the layout genes of each chromosome (*CHG's*) and formulas (43)–(54), the following values are obtained: the cell number, *MH* number, coordinates of each machine and the distances between machines.

In the third phase, the calculation of fitness function of each chromosome is gained by considering the objective function and the information obtained from the first and second phases.

Selection

The selection function chooses the parents for the production of next generation. In the presented algorithm, a grading function for scaling is used that prevents of more extension of raw scores. In this scaling, the grade related to each individual (chromosome) is a

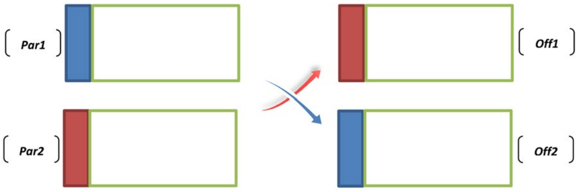


Fig. 9 Crossover operator for the first gene

rank that is assigned to each individual in a population after being sorted. Therefore, the grade related to the best score is 1, for the next is 2 and so forth. The ranking function will assign a rank to each individual as described below:

1. At first, all individuals of a population are ranked based on their fitness function value (i.e., objective function value) increasingly. An individual with rank n is given value $1/\sqrt{n}$.
2. Next, total measure amounts of the whole population are equal with the number of needed parents for the production of the next generation.
3. The value $1/\sqrt{n}$ is placed in the interval (0, 1] and correctness coefficient (α) is obtained from the below formula.

$$\alpha = \frac{Population\ size}{\sum_{n=1}^{Population\ size} \frac{1}{\sqrt{n}}}$$

By multiplying the value $1/\sqrt{n}$ of each individual bycoefficient (α), a scaling number is given toeach individual which is used by roulette wheel rule for selecting parents

Reproduction operators

Offsprings in each new generation are created using recombination operators (i.e., crossover and mutation) as described below:

Crossover

Crossover operator designed in this algorithm makes two offsprings from two selected parents by considering each chromosome string separately. To implement this operator, three crossover points are selected on each string. Clearly, the first crossover point is between the first and the second gene, and two other crossover points are selected after the first crossover point, randomly. This crossover acts in two phases as described below.

In the first phase, the first gene of the first offspring is copied from the first gene of the second parent and the first gene of the second offspring is copied from the first gene of the first parent. A simple example of how crossover operator works is shown in Fig. 9.

In the second phase, the crossover operator for scheduling genes of each chromosome acts distinctly from the layout gene. The length of each chromosome string may be different from other chromosome strings due to the fact that the process routings for part types can be flexible. Therefore, the steps of crossover operator in the scheduling section of each chromosome string are described as below:

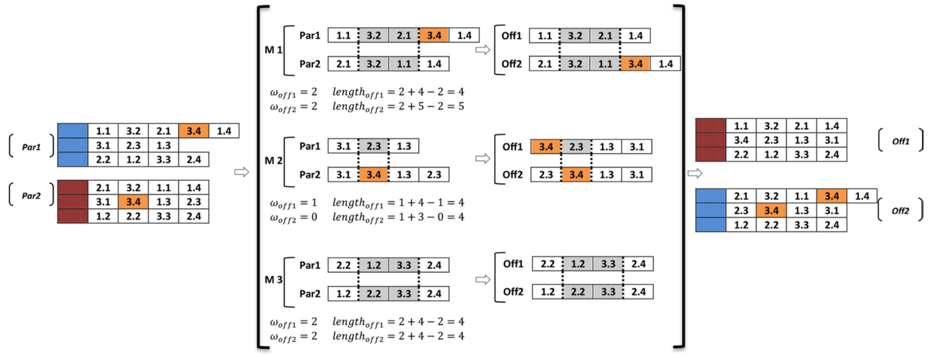


Fig. 10 Crossover operator for scheduling genes

Step 1 Along the length of the two parents, two crossover points are chosen in similar points, randomly. Then, the length of first offspring is calculated by the below formula:

$$\begin{aligned}&\text{length of chromosome string of the first offspring}\\&\quad= \text{“distance between two random points”}\\&\quad+ \text{“length of chromosome of the second parent”} - \omega\end{aligned}$$

where ω is the number of genes that are common along the two chosen points of the first parent and the whole chromosome string of the second parent.

Step 2 The section between two random points of the first parentis copied into the first offspring.

Step 3 Starting from the second crossover point of the second parent, the other unused numbers in an order that they appear are copied in the second parent. If it reaches to the end of the string, it starts from the beginning and copies the remaining genes in order.

Step 4 In this step, the correction of the offsprings is made. The random choice of a machine among the machines that process the same operations based on constraint (5) means each part information and its related operations (*i,k*) should appear only once in each chromosome.

Step 5 The second offspring is produced in the same way by reversing the parents' role.

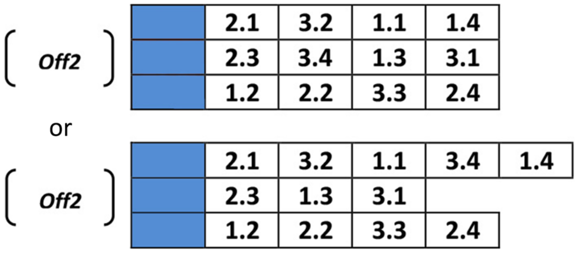
An example of implementing crossover operator for scheduling section is shown in Fig. 10.

As it is clear for *off2* in Fig. 10, the fourth operation of part 3 (i.e., 3.4) is processed by both machine 1 and machine 2. This makes the obtained offspring unfeasible. Hence, the chromosome should be repaired. For that reason, one of them is chosen randomly and the other is deleted from the chromosome (Fig. 11).

Mutation

Mutation operator in the layout and scheduling genes of each offspring operates separately by mutation probability P_m and $P_{m'}$ respectively, as described in the two following phases.

In the first phase, a random resetting mutation operator is done on the layout gene. A random number in the interval (01] is generated and compared with P_m. If the random



	2.1	3.2	1.1	1.4
	2.3	3.4	1.3	3.1
	1.2	2.2	3.3	2.4

	2.1	3.2	1.1	3.4	1.4
	2.3	1.3	3.1		
	1.2	2.2	3.3	2.4	

Fig. 11 Repairing an offspring

number is smaller than P_m, one of thechromosome strings will be selected by chance and an amount of allowable set S_j will be selected in that situation and other amounts of different offspring genes will be changed in an order according to the steps that were explained in "The initial population" section, and new amounts will be replaced.

In the second phase, swap mutation operator is done on the scheduling genes. At first, a random number is produced for each chromosome string. Then, if the random number is smaller than $P_{m'}$, two positions of genes will be randomly selected and their amounts will be replaced by each other.

Termination criterion

Termination criterion in the proposed algorithm is the number of iterations. Therefore, the production of new generations will be continued until the number of iterations is reached. That value depends on the problem size.

Experimental results

Two illustrative numerical examples

In this section, two experiments are performed to validate the proposed model and compare the effects of the sequential and concurrent approach in the CM design.

To verify the proposed model and reveal that the concurrent integration of parts scheduling with CF and machines layout is more effective than sequential approach, two small-sized examples are solved by a Branch and Bound (B&B) method under Lingo 8.0 software on an Intel(R) core(TM) i5 CPU 2.6 GHz personal computer with 4.00 GB RAM.

In the sequential approach, the proposed model is solved to find the optimal values of decision variables for CF and CL. Then, using the obtained solution the optimal solution of the scheduling problem is determined. To implement this approach, the main model is solved by excluding the components (1.1) and (1.2) associating with makespan and tardiness from the objective function in the first step. Then, in the second step, the main model is solved by regarding the values obtained for decision variables of CF and CL as input parameters and excluding the component (1.3) associating with material handling costs from the objective function.

In the concurrent approach, the proposed model is optimally solved including all three components in the objective function to find the optimal solution of integrating parts scheduling with CF and machines layout simultaneously. Finally, the objective function values (OFV) obtained for the optimal solutions of both approaches are compared.

Table 2 Processing times of part operations for the first example

Machines	P1		P2		P3		P4	
	1	2	1	2	1	2	1	2
M1	18			15	3			11
M2	24		5			10	6	
M3		5	19	8			20	

Table 3 Processing times of part operations for the second example

Machine	P1			P2			P3			P4		
	1	2	3	1	2	3	1	2	3	1	2	3
M1	7				5			25		17		
M2			29	14				30				3
M3		25				28	12					8
M4	3					19			8		11	

Table 4 Machines dimensions and coordinates of cells partitions for the first example

Parameter	Machines' information		Cells' information				
	L_j	W_j		LX_c	UX_c	LY_c	UY_c
M1	4	2	C1	3	8	3	13
M2	2	3	C2	11	17	5	11
M3	3	5					

In these two numerical examples, the data is randomly generated which includes process routing for each part type and processing time for each operation. Machines are selected randomly for each process routing and the processing times of operations are random integer numbers between 2 and 30. The machines should be assigned to two cells. The due date for all part types is 40 min. The material handling time for all part types per distance unit is 3 min. The inter-cell and intra-cell material handling costs for all part types per distance unit are 5$ and 2$, respectively. The tardiness penalty for all part types per time unit is 3$. The factory costs per time unit are 25$.

Tables 2 and 3 represent parts' route sheet with the processing time of each operation of each part type on each machine for the first and second example, respectively.

Tables 4 and 5 demonstrate machines dimensions and coordinates of cell partitions for the first and second example, respectively.

Figures 12 and 13 illustrate the location of each machine in each cell for the sequential and concurrent approach of examples 1 and 2, separately. Also, Gantt charts of part operations scheduling have been projected in these figures. A Gantt chart illustrates the sequence of operations of part types processed on machines along with processing time and material handling time. It is depicted for the sequential and concurrent approach of each example, separately.

Now, the solution obtained for the first example is discussed. This example consists of four part types and three machine types. Each part type requires two operations

Table 5 Machines dimensions and coordinates of cells partitions for the second example

Parameter	Machines' information		Cells' information				
	L_j	W_j		LX_c	UX_c	LY_c	UY_c
M1	4	2	C1	2	7	2	10
M2	2	3	C2	10	15	6	14
M3	5	3					
M4	2	2					

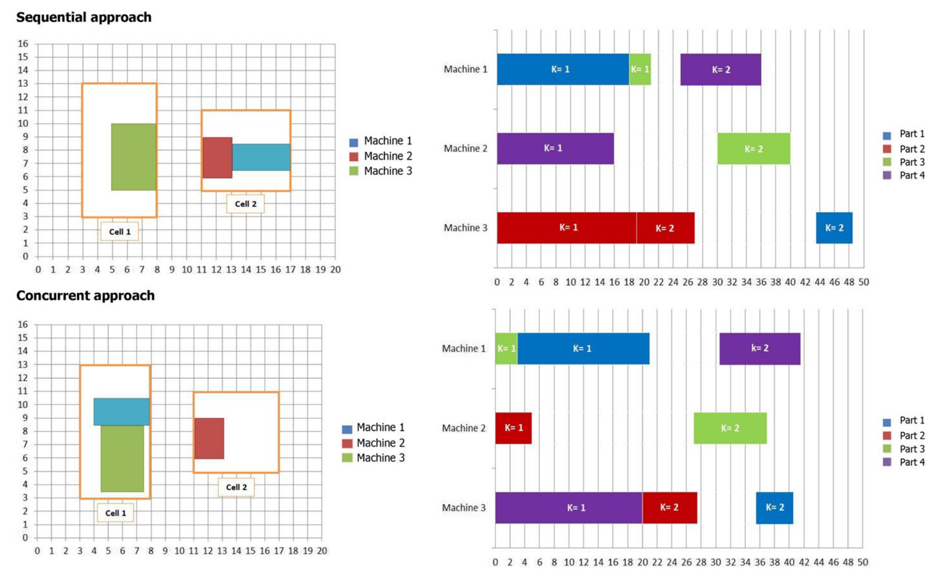


Fig. 12 Intra-cell layout and Gantt chart of part operations scheduling for the first example

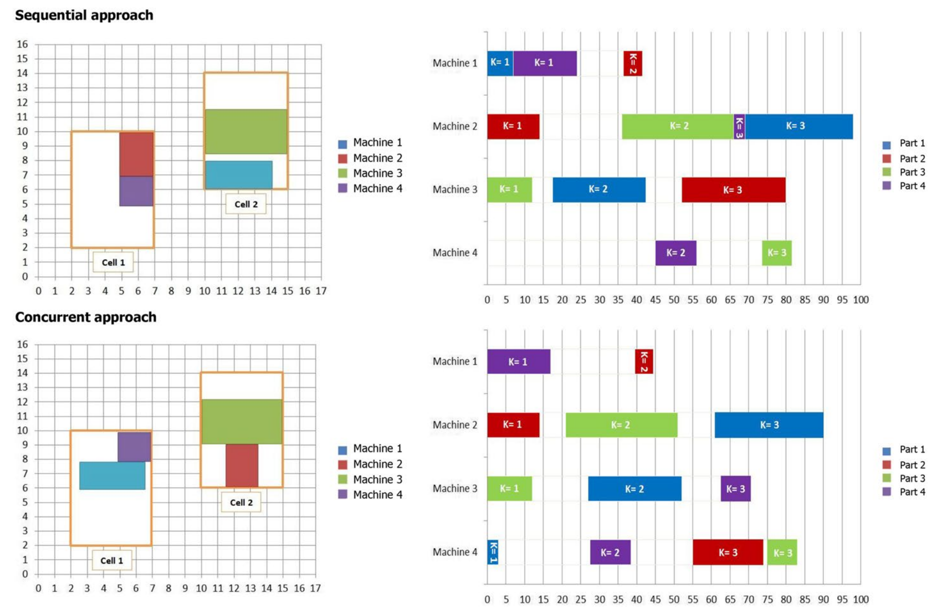


Fig. 13 Intra-cell layout and Gantt chart of part operations scheduling for the second example

Table 6 Objective function value and its cost components for the first example

Cost	Value		
	Sequential approach	Concurrent approach	Improvement by concurrent approach
Factory costs × C_{max}	25 × 48.5 = 1212.5	25 × 41.5 = 1037.5	175 (16.7 %)
Tardiness penalty	25.5	6	19.5 (325 %)
Material handling	54.5	91.5	−37 (67.9 %)
Total (OFV)	1292.5	1135	157.5 (13.9 %)

Table 7 Objective function value and its cost components for the second example

Cost	Value		
	Sequential approach	Concurrent approach	Improvement by concurrent approach
Factory costs × C_{max}	25 × 98 = 2450	25 × 90 = 2250	200 (8.9 %)
Tardiness penalty	505.5	472.5	33 (7 %)
Material handling	176.5	188.5	−12 (6.8 %)
Total (OFV)	3132	2911	221 (7.6 %)

that each one can be processed by only one machine type selected from two alternative machine types with different processing times. For instance, there are two process routings for part type 1, the first operation of part type 1 can be processed either on machine type 1 or machine type 2 while the second one can be done only on machine type 3. Also, there are four, one and two process routings for part types 2, 3 and 4, respectively.

The optimal OFVs obtained for the proposed model in two sequential and concurrent approaches are shown in Table 6 for the first example.

Since in the sequential approach, the objective function is optimized without components related to makespan and tardiness, the model is able to find the optimal value for material handling cost equal to 54.5. However, when the components related to makespan and tardiness are included in the objective function in the concurrent approach, the previous optimal material handling cost increase due to the effect of scheduling parts on machines layout. Switching from sequential approach to concurrent approach reduces the value of makespan from 48.5 to 41.4 and the value of tardiness penalty from 25.5 to 6, which are remarkable improvements for these components. On the other hand, incorporating parts scheduling in the concurrent approach increases the value of material handling cost from 54.5 to 91.5. Totally, OFV is improved about 14 % by switching from sequential approach to concurrent approach. This achievement was expectative since simultaneous decisions making about interrelated decisions machines layout and parts scheduling brings the capability for the model to optimize all components of the objective function as an optimal strategy in designing a CMS.

Next, the second example consists of four part types and three machine types. Each part type requires three operations that each one can be performed by at most two machine types. There are two process routings for each part type.

The optimal OFVs obtained for the proposed model in two sequential and concurrent approaches are shown in Table 7 for the second example.

A similar improvement in the obtained OFV is observed for the second example in Table 7 as it was expected due to the advantage of concurrent approach.

As can be seen in Figs. 12 and 13, when makespan and tardiness components are included in the objective function in the concurrent approach, cell formation, machines layout, and parts scheduling change completely in comparison with sequential approach. It can be concluded that even if factory cost is very high forcing to complete all parts as soon as possible, one should group machines, locate them and schedule operations simultaneously.

To conclude, by comparing the OFVs obtained for examples 1 and 2 in the sequential and concurrent approaches, it is recognized that if cells are configured, machines are located and operations are scheduled sequentially, the optimum strategy with the minimum costs cannot be reached.

Evaluation of the proposed GA

In order to evaluate the performance of the GA in comparison with B&B method for the sequential and concurrent approaches, ten instances have been generated with the random data by inspiration from the literature. The GA has been coded in MATLAB software and run ten replicates on an Intel(R) core(TM) i5 CPU 2.6 GHz computer with 4.00 GB RAM for each test problem, and the best, as well as the average of the obtained solutions in ten runs, have been reported in Table 8. Also, for all test problems, the linearized programming model has been solved by a B&B method under LINGO software. In large-sized problems, the LINGO program has been interrupted after 12 h, and the best-obtained solution has been reported. Table 8 illustrates the information of test problems and the obtained solutions with the computational times by GA and B&B method as an exact method for the sequential and concurrent approaches.

As it can be seen from Table 8, GA has found optimal solutions for problems 1–4 for sequential approach and optimal solutions for problems 1–3 for the concurrent approach in which optimal solutions are obtained by B&B method. No feasible solution is found by B&B for problems 9 and 10 in both approaches due to the complexity of the model. Furthermore, for the rest of problems in both approaches, the best solutions found by GA are better than the solutions found by B&B in much less time. On average, the comparisons between the OFVs obtained by sequential and concurrent approaches indicate that the OFV improvement is around 17 % by GA and 14 % by B&B.

These promising results obtained by the proposed GA prove the efficiency of the designed algorithm enhanced by the matrix-based chromosome structure. Furthermore, the developed GA is elaborately designed to create feasible solutions by using some defined formulas, efficient crossover and mutation operators and established procedures calculating fitness function and generating initial population.

Conclusion

In this article, a novel integrated mathematical model has been formulated for designing a cellular manufacturing system considering three problems simultaneously: cell formation, intra-cell layout, and parts scheduling. The results show that considering

Table 8 Comparison of GA solutions and exact solutions

Problem no.	No. of machines	No. of parts	No. of operations per part	No. of processing positions per machine	No. of cells	Sequential approach					Solution GAP
						B&B		GA			
						OFV	Time (s)	Best OFV	Average OFV	Time (s)	
1	3	4	2	4	2	1292.5*	16	1292.5	1292.5	1.6	–
2	4	4	3	3	2	3474.5*	276	3474.5	3474.5	3.9	–
3	4	4	3	4	2	3132*	20,350	3132	3132	14.6	–
4	4	5	3	4	2	3688.5*	38,405	3688.5	3700.5	15.4	–
5	5	6	3	4	2	5010.5	43,200	4798	4851.7	18.5	−3.3
6	6	8	3	4	2	6490.7	43,200	6206.5	6456.5	23.3	−4.6
7	8	10	4	5	3	13,728.6	43,200	12,802	13,588.4	33.8	−7.2
8	10	12	5	6	3	22,347.4	43,200	20,997	21,190.8	42.4	−6.4
9	15	25	3	5	4	NA	43,200	30,677	30,152.7	61.1	NA
10	20	40		6	6	NA	43,200	86,110	77,090.3	102.5	NA

Problem no.	No. of machines	No. of parts	No. of operations per part	No. of processing positions per machine	No. of cells	Concurrent approach					Solution GAP	OFV Improvement B&B (%)	OFV Improvement GA (%)
						B&B		GA					
						OFV	Time (s)	Best OFV	Average OFV	Time (s)			
1	3	4	2	4	2	1135*	185	1135	1112.5	2.9	–	13.9	13.9
2	4	4	3	3	2	3089*	1203	3089	3089	7.6	–	12.5	12.5
3	4	4	3	4	2	2911*	29,900	2911	2911	18.2	–	7.6	7.6
4	4	5	3	4	2	3557.3	43,200	3520	3691.7	19.2	−1.1	3.7	4.8
5	5	6	3	4	2	4556.7	43,200	4323.5	4482.7	23.3	−5.4	10	11
6	6	8	3	4	2	5764.2	43,200	5486	5543.9	31.4	−5.1	12.6	13.1
7	8	10	4	5	3	10,582.5	43,200	9957.5	10,233.8	56.7	−6.3	29.7	28.6
8	10	12	5	6	3	18,632.4	43,200	17,453	18,255.6	68.47	−6.8	19.9	20.3
9	15	25	3	5	4	NA	43,200	23,543	24,040.3	108.3	NA	NA	30.3
10	20	40		6	6	NA	43,200	67,872	68,253.6	221.1	NA	NA	26.9

* Optimal solution

these three significant decisions in a simultaneous manner contributes to a successful CM implementation in the manufacturing environment. All these problems have been optimized due to three components in the objective function including makespan time, tardiness penalty, and inter-cell and intra-cell material handling cost. By investigating two integration approaches, namely sequential and concurrent, it was revealed that to reach an optimal solution, all stages of CMS (CF, machine layout, and part scheduling) must be designed simultaneously. It In sequential approach, since cells are configured at first in order to decrease inter-cell and intra-cell movements costs, and finally operations are scheduled in order to optimize time factors in the objective function consisting of tardiness and makespan, the global optimal solution is not reached, although it is attainable in a concurrent approach. The advantages of the proposed model were as follows: designing layout of unequal-area machines in cells with continuous space, introducing alternative process routings for parts with different operation sequence, exerting the effects of distances between locations or cells on part scheduling, becoming both cost and time of part movements dependent on (1) traveled distance, (2) part type and (3) movement type (inter-cell and intra-cell), computing the completion time of each part by considering: (1) movement times, (2) processing times and (3) waiting times, and finally, determining three interrelated designing stages (i.e., CF, machines layout, and parts scheduling) in designing a CMS simultaneously. For transforming mixed-integer nonlinear programming formulation into a mixed-integer linear counterpart, some linearization techniques have been proposed. In order to verify the performance of the proposed model, two numerical examples have been solved. It was revealed that concurrent approach surpasses sequential approach in improving the quality of the obtained solutions in the CM system design.

Because of the complexity of the proposed model, Lingo software cannot obtain the optimum solution for medium or large-sized problems. Hence, a genetic algorithm has been developed that its excellent advantages were as follows: determining the exact location of machines in continuous-area cells; computing exact completion time of each operation of each part type, proposing heuristic crossover and mutation operator. The obtained results show that the best solutions found by GA are better than those found by Lingo in much less time especially as the size of problem increases. Incorporating other features, such as uncertainty in part demands, machine time capacity and cost coefficients, integrating with reliability and labor issues and considering dynamic issues will be left to future research.

Authors' contributions

AE carried out the literature review studies, designed the mathematical model and the solution approach, and drafted the manuscript. RK participated in the design of the mathematical model and performed the numerical experiments. ARK conceived of the study and participated in its design and coordination. All authors read and approved the final manuscript.

Acknowledgements

This work was financially supported by Islamic Azad University, Firoozkooh Branch. The authors would like to sincerely thank the anonymous referees for their thorough reviews of the early versions of this paper and their valuable comments.

Competing interests

The authors declare that they have no competing interests.

Received: 7 December 2015 Accepted: 5 July 2016
Published online: 04 August 2016

References

Adeil G, Rajamani D, Strong D (1996) Cell formation considering alternative routings. Int J Prod Res 34(5):1961–1980

Ahi A, Aryanezhad MB, Ashtiani B, Makui A (2009) A novel approach to determine cell formation, intracellular machine layout and cell layout inthe CMS problem based on TOPSIS method. Comput Oper Res 36:1478–1496

Akturk MS (1996) A note onthe within-cell layout problem based on operation sequences. Prod Plan Control 7(1):99–103

Alfa S, Chen M, Heragu SS (1992) Integrating the grouping and layout problems in cellular manufacturing systems. Comput Ind Eng 23(1–4):55–58

Arkat J, Hosseinabadi Farahani M, Hosseini L (2012a) Integrating cell formation with cellular layout and operations scheduling. Int J Adv Manuf Technol 61:637–647

Arkat A, Hosseinabadi Farahani M, Ahmadizar F (2012b) Multi-objective genetic algorithm for cell formation problem considering cellular layout and operations scheduling. Int J Comput Integr Manuf 25(7):625–635

Arvindh B, Irani A (1994) Cell formation: the need for an integrated solution of subproblems. Int J Prod Res 32(5):1197–1218

Atmani A, Lashkari RS, Caron RJ (1995) A mathematical programming approach to joint cell formation and operation allocation in cellular manufacturing. Int J Prod Res 31(1):1–15

Bazargan-Lari M, Kaebernick M, Harraf H (2000) Cell formation and layout design in a cellular manufacturing. Int J Prod Res 38:1689–1709

Castillo I, Westerlund T (2005) An *e*-accurate model for optimal unequal-area block layout design. Comput Oper Res 32:429–447

Chandrasekharan MP, Rajagopalan R (1993) A multidimensional scaling algorithm for group layout in cellular manufacturing. Int J Prod Econ 32:16–65

Chen M, Cao D (2004) Coordinating production planning in cellular manufacturing environment using Tabu search. Comput Ind Eng 46:571–588

Chiang CP, Lee SD (2004) Joint determination of machine cells and linear intercell layout. Comput Oper Res 31:1603–1619

Defersha F, Chen M (2008) A linear programming embedded genetic algorithm for an integrated cell formation and lot sizing considering product quality. Eur J Oper Res 187(1):46–69

Deljoo V, Mirzapour Al-e-hashem SMJ, Deljoo F, Aryanezhad MB (2010) Using genetic algorithm to solve dynamic cell formation problem. Appl Math Model 34:1078–1092

Franca PM, Gupta JND, Mendes AS, Moscato P, Veltink KJ (2005) Evolutionary algorithms for scheduling a flow shop manufacturing cell with sequence dependent family setups. Comput Ind Eng 48:491–506

Gen M, Cheng R (1997) Genetic algorithms and engineering design. Wiley Inter-science Publication, Malden

Halat K, Bashirzadeh R (2014) Concurrent scheduling of manufacturing cells considering sequence-dependent family setup times and intercellular transportation times. Int J Adv Manuf Technol 77(9–12):1907–1915

Hendizadeh SH, Faramarzi H, Mansouri SA (2008) Meta-heuristic for scheduling a flow line manufacturing cell with sequence dependent family setup times. Int J Prod Econ 111:593–605

Jajodia S, Minis I, Harhalakis G, Proth J (1992) CLASS: "computerized layout solutions using simulated annealing". Int J Prod Res 30(1):95–108

Javadi B, Jolai F, Slomp J, Rabbani M, Tavakkoli-Moghaddam R (2013) An integrated approach for the cell formation and layout design in cellular manufacturing systems. Int J Prod Res 51(20):6017–6044

Jolai F, Tavakkoli-Moghaddam R, Golmohammadi A, Javadi B (2012) An electromagnetism like algorithm for cell formation and layout problem. Expert Syst Appl 39(2):2172–2182

Kia R, Khaksar-Haghani F, Javadian N, Tavakkoli-Moghaddam R (2014) Solving a multi-floor layout design model of a dynamic cellular manufacturing system by an efficient genetic algorithm. J Manuf Syst 33(1):218–232

Leung JY-T, Li H, Pinedo M, Zhang J (2007) Minimizing total weighted completion time when scheduling orders in a flexible environment with uniform machines. Inf Process Lett 103:119–129

Lin S-W, Ying K-C, Lee Z-J (2009) Meta-heuristics for scheduling a nonpermutation flow line manufacturing cell with sequence dependent family setup times. Comput Oper Res 36(4):1110–1121

Mahdavi I, Mahadevan B (2008) CLASS: "an algorithm for cellular manufacturing system and layout design using sequence data". Robot Comput Integr Manuf 24:488–497

Mahdavi I, Shirazi B, Paydar MM (2008) A flow matrix-based heuristic algorithm for cell formation and layout design in cellular manufacturing system. Int J Adv Manuf Technol 39:943–953

Man KF, Tang KS, Kwong S (1999) Genetic algorithms: concepts and design. Springer, London

Mohammadi M, Forghani K (2014) A novel approach for considering layout problem in cellular manufacturing systemswith alternative processing routings and subcontracting approach. Appl Math Model 38(14):3624–3640

Papaioannou G, Wilson JM (2010) The evolution of cell formation problem methodologies based on recent studies (1997–2008): review and directions for future research. Eur J Oper Res 206:509–521

Ranjbar M, Najafian Razavi M (2012) A hybrid metaheuristic for concurrent layout and schedulingproblem in a job shop environment. Int J Adv Manuf Technol 62:1249–1260

Reddy V, Narendran TT (2003) Heuristics for scheduling sequence-dependent set-up jobs in flow line cells. Int J Prod Res 41(1):193–206

Ripon KSN, Torresen J (2013) Integrated job shop scheduling and layout planning: a hybridevolutionary method for optimizing multiple objectives. Evol Syst 5(2):121–132

Salum L (2000) The cellular manufacturing layout problem. Int J Prod Res 38(5):1053–1069

Selim H, Askin RG, Vakharia AJ (1997) Cell formation in group technology: review evaluation and directions for future research. Comput Ind Eng 22(2):215–225

Shiyas CR, Pillai VM (2014) A mathematical programming model for manufacturing cell formation to develop multiple configurations. J Manuf Syst 33(1):149–158

Soleimanpour M, Vrat P, Shankar R (2002) A transiently chaotic neural network approach to the design of cellular manufacturing. Int J Prod Res 40(10):2225–2244

Solimanpur M, Vrat P, Shankar R (2004a) A heuristic to minimize makespan of cell scheduling problem. Int J Prod Econ 88:231–241

Solimanpur M, Vrat P, Shankar R (2004b) Ant colony optimization algorithm to the inter-cell layout problem in cellular manufacturing. Eur J Oper Res 157:592–606

Sridhar J, Rajendran C (1993) Scheduling in a cellular manufacturing system: a simulated annealing approach. Int J Prod Res 31(12):2927–2945

Tang J, Wang X, Kaku I, Yung KL (2010) Optimization of parts scheduling in multiple cells consideringintercell move using scatter search approach. J Intell Manuf 21:525–537

Tavakkoli-Moghaddam R, Gholipour-Kanani Y, Cheraghalizadeh R (2008) A genetic and memetic algorithm approach to sequencing and scheduling of cellular manufacturing systems. Int J Manag Sci Eng Manag 3(2):119–130

Tavakkoli-Moghaddam R, Javadian N, Khorrami A, Gholipour-Kanani Y (2010) Design of a scatter search method for a novel multi-criteria group scheduling problem in a cellular manufacturing system. Expert Syst Appl 37:2661–2669

Urban TL, Chiang W, Russell R (2000) The integrated machine allocation and layout problem. Int J Prod Res 38(13):2911–2930

Vin E, Delchambre A (2014) Generalized cell formation: iterative versus simultaneousresolution with grouping genetic algorithm. J Intell Manuf 25:1113–1124

Wang TY, Wu KB, Liu YW (2001) A simulated annealing algorithm for facility layout problems under variable demand in cellular manufacturing systems. Comput Ind 46:181–188

Wemmerlov U, Hyer NL (1986) Procedures for part family/machine group identification problem in cellular manufacturing. J Oper Manag 6(2):125–147

Wu X, Chu CH, Wang Y, Yan W (2006) Concurrent design of cellular manufacturing systems: a genetic algorithm approach. Int J Prod Res 44(6):1217–1241

Wu X, Chu CH, Wang Y, Yue D (2007a) Genetic algorithms for integrating cell formation with machine layout and scheduling. Comput Ind Eng 53:277–289

Wu X, Chu CH, Wang Y, Yan W (2007b) A genetic algorithm for cellular manufacturing design and layout. Eur J Oper Res 181:156–167

Xie W, Sahinidis NV (2008) A branch-and-bound algorithm for the continuous facility layout problem. Comput Chem Eng 32:1016–1028

Zeng C, Tang J, Yan C (2015) Job-shop cell-scheduling problem with inter-cell movesand automated guided vehicles. J Intell Manuf 6:845–859