Decision Science Letters 6 (2017) 165–192

Contents lists available at GrowingScience

Decision Science Letters

homepage: www.GrowingScience.com/dsl

A comprehensive quadratic assignment problem for an integrated layout design of final assembly line and manufacturing feeder cells

Masoud Rabbani[a*], **Saeed Elahi**[a] **and Babak Javadi**[b]

[a]*School of Industrial Engineering, College of Engineering, University of Tehran, Tehran, Iran*
[b]*Department of Industrial Engineering, College of Farabi, University of Tehran, Qom, Iran*

CHRONICLE

Article history:
Received June 25, 2016
Received in revised format:
October 2, 2016
Accepted October 2, 2016
Available online
October 3 2016

Keywords:
Cellular manufacturing system
assembly line design
Quadratic assignment problem
Feeder cells
Genetic algorithm
Memetic algorithm

ABSTRACT

Assembly lines and cellular manufacturing systems (CMSs) design have been widely used in the literature. However the integration of these manufacturing concepts is neglected in an environment where parts need to be assembled after production in different shops. In this paper, a comprehensive quadratic assignment problem is developed for the assignment of machines of each part manufacturing cell, sub-assembly tasks of each sub-assembly cell as well as the assignment of different cells and final assembly tasks within the shop floor in their relevant predetermined locations. A genetic algorithm (GA) as well as a memetic algorithm (MA) consisting of the proposed GA and Tabu search (TS) algorithm are proposed and implemented on different size numerical examples. The obtained results show the efficiency of both algorithms to reach near optimal solutions compared to the optimal solution of small-sized problems.

1. Introduction

Product variety is a challenging issue in today's market. Growing trend in customer's demand change and competitive market requirements have led manufacturers to substitute high-mix/low-volume manufacturing strategies for low-mix/high-volume manufacturing strategies to remain flexible and capture their market share. To deal with such conditions, lots of manufacturing systems have been developed in the literature among which CMS is one of manufacturing systems taking the advantage of job shop's flexibility and the flow shop's efficiency by reducing lead time, setup time, work-in-process (WIP) inventories and material handling costs. Design of CMS involves three stages (Sridhar & Rajendran 1993): 1.cell formation (i.e. grouping parts into part families and machines into machine

* Corresponding author. Tel: +98-2188350642; Fax: +98-2188350642
E-mail address: mrabani@ut.ac.ir (M. Rabbani)

doi: 10.5267/j.dsl.2016.10.001

cells); 2.facility layout (i.e. arranging layout of machines within each cell and cells within shop floor); 3. Scheduling (i.e. schedule of jobs in each cell). These steps have been used either separately or simultaneously by many researchers. In assembly systems, mixed-model assembly line (MMAL) is the one usually used in the presence of above challenges in different layout type consisting of straight line, U-shaped, two-sided layout. Line balancing (i.e. assignment of tasks to workstations) and model sequencing (i.e. production sequence determination of different models) are two problems adopted separately or simultaneously for the design of MMAL. CMS and MMAL concepts have been developed separately in production systems. However, many manufacturing environments are composed of a final assembly line receiving different components of final product mixes from a number of feeder shops (Luss, 1989). In such systems, various raw materials and WIP inventories circulate between manufacturing equipment to produce final products, which necessitate the implementation of an efficient layout. Material handling costs of a plant have been estimated in a range from 20% to 50% of the total manufacturing operating costs. Hence, poor physical layout can reduce some or all of a company's expected benefits (Tompkins & White, 2010).

In this paper, a plant layout design is considered in which parts manufacturing and product assembly are performed, simultaneously. Parts are produced under CMS strategy and served as feeding sub-assembly cells. Then, sub-assembly products are produced in separate U-line assembly cells and served as feeding final mixed model assembly U-line to produce product mixes. In this regard, a comprehensive quadratic mathematical programming model is developed for the assignment of production facilities consisting of part manufacturing cells, sub-assembly cells, machines of each part manufacturing cell, sub-assembly tasks of each sub-assembly cell and final assembly tasks of final assembly tasks in their relevant predetermined locations. For the layout design of the system it is assumed that part manufacturing cells have already formed and production routings of parts have been determined and applied for the layout design. In contrary to the classic MMAL design, the proposed model has nothing to do with line balancing and model sequencing. Instead, the material handling flow of WIP within sub-assembly cells (sub-assembly intra-cell material flow) and final assembly line is taken into consideration as neglected in previous researches. Also, part manufacturing intra-cell, part manufacturing inter-cell, part manufacturing and sub-assembly inter-cell as well as material flow between sub-assembly cells and final assembly line are properly incorporated in the model. Hence, the objective of the proposed model is to minimize the total costs of material handling flows while satisfying facilities assignments and assembly tasks precedence relationships.

The rest of this paper is organized as follows: A brief literature of CMS and feeder shops concepts is presented in Section 2. Formulation of the problem is performed in Section 3. Solution approaches of the problem is clarified in Section 4. In Section 5 parameters of the proposed solution approaches are set. Computational results of different problem sizes are presented in Section 6. Finally, conclusions of the work is discussed in Section 7.

2. Literature review

There is a considerable amount of literature devoted to CMS and MMAL problems reviewing the researches in production areas. Review and classification of CMS problems can be seen in Balakrishnan and Cheng (2007). Review and direction of cell formation (CF) methodologies is presented by Papaioannou and Wilson (2010). Also in assembly line area, Boysen et al. (2008), Battaïa and Dolgui (2013) and Saif et al. (2014) presented comprehensive survey and classifications. A brief literature of layout problems in CMS as well as feeder lines or feeder shops concepts in production environments is addressed as follows:

Wang et al. (2001) formulated a model to tackle the facility layout problem in CMS assuming that the demand rate would change over the product's life cycle. The objective function was considered as the minimization of material handling costs in terms of both intra- and inter-cell layouts simultaneously. A simulated annealing (SA) algorithm was employed to solve the model as to the problem's complexity. Chan et al. (2004) proposed a machine allocation inter-relation (MAIN) algorithm to tackle both static and dynamic layout in an effort to minimize the total part traveling cost and the machine rearrangement cost in a dynamic environment. Tavakkoli-Moghaddam et al. (2007) developed a new mathematical model to minimize the total costs of inter- and intra-cell movements in both machine and cell layout problems in CMSs simultaneously under stochastic demand. They presented different scenarios of machine and cell layouts as the model depends on the attitude of the decision maker toward uncertainty. Ariafar and Ismail (2009) proposed a mathematical model as well as an SA approach for facility layout in CMS that minimizes both inter-cell and intra-cell material handling costs and proved the better quality and less computation time of the proposed SA in comparison with the benchmarked algorithm. Pattanaik and Sharma (2009) presented a methodology for cellular layout design integrated with lean concepts and implemented in an industry dealing with ammunition components for defense applications.

Jolai et al. (2011) proposed a new mathematical model for intra- and inter-cell layout problem in CMS. They developed a binary particle swarm optimization (BPSO) as well a variable neighborhood search (VNS) to solve the model. Kulkarni et al. (2013) developed a GA to deal with the issue of inter-cell layout in CMS design. The novelty of their proposed GA lied in the way the initial solution was constructed. Compared to the simple facility location problem and other existing algorithms, the promising results of the proposed methodology was proven. Javadi et al. (2014) developed a mathematical model devoted to intra- and inter-cell layout problem under dynamic situation. An electromagnetism-like (EM-like) algorithm, a GA and a hybrid algorithm based on the EM-like algorithm and GA were separately adopted to minimize the total costs of rearrangement, intra- and inter-cell material flows. The superiority of the hybrid algorithm than two other algorithms was proven by a number of numerical example. Li et al. (2015) proposed a multi-row layout model to address the remanufacturing dynamic facility problem with variable process capacities, unequal processing and inter-cell material handling. They also developed a modified simulated annealing heuristic to determine optimal layout scheme. Golmohammadi et al. (2016) proposed a hierarchical genetic algorithm (HGA) for cellular manufacturing layout problem under dynamic conditions aiming at minimizing the total costs of movements within and between cells and exceptional parts. The HGA was then compared with a GA by which the applicability of the proposed HGA was proven.

There are also researches performed on solving concurrent decisions of CMSs. In this regard, Bazargan-lari et al. (2000) suggested an integrated approach for the design of CMS in a white-goods manufacturing company where all three steps of CMSs (i.e. parts/machine grouping and intra/inter-cell layout design) were considered simultaneously. Chiang and Lee (2004) addressed the concurrent decisions on CF and inter-cell layout so that machines were assumed to be located along a linear flow line with the objective of minimizing actual inter-cell flow cost instead of the number of inter-cell movements via a GA with optimal partition approach. Wu et al. (2007) proposed a hierarchical genetic algorithm (GA) procedure to simultaneously form manufacturing cells and determine the group layout of a CMS. Mahdavi et al. (2008) developed a flow matrix-based heuristic algorithm and considered CF and layout design simultaneously based on sequence data. They proved the usefulness of the proposed algorithm through numerical results obtained from the available problems in the literature. Bozer and Wang (2012) presented a comprehensive model consisting of important operational and layout design aspects such as unequal dimension of machines and pickup and drop off stations to specify cell formation and layout design simultaneously in CMSs.

Krishnan et al. (2012) proposed an integrated CF and layout design in CMS. They utilized a from-to chart as the primary input to deal with CF problem and a GA-based approach to arrange the machine cells within a layout matrix. Arkat et al. (2012) presented a mathematical model to concurrently study CF, layout and operations sequence as well. A multi-objective genetic algorithm (MOGA) aiming at minimizing total transportation cost of parts as well as minimizing makespan was then developed to solve the model. Mohammadi and Forghani (2014) presented a nonlinear integer programming (NLIP) model for designing CMS in terms of CF and its intra- and inter-cell layouts. Various production factors such as alternative process routings, operations sequence, subcontracting, etc. were incorporated in the model to extend its applicability. They calculated the material handling cost of the system based on the actual positions of the machines within cells regarding machines dimension and aisle distances. Finally, an efficient GA was employed to sole the model and verified through numerical examples adopted from the literature. Kia et al. (2014) presented a mixed-integer programming model integrating a multi-floor layout design of CMS in a dynamic environment and CF problem. Due to the NP-hard class of the problem, a GA was proposed for real size problems. Bagheri and Bashiri (2014) proposed a new mathematical model to simultaneously solve the cell formation, operator assignment and inter-cell layout problems in order to minimize inter-cell movements, machine relocation cost and operator related issues. The model was solved by branch and bound technique and preferred solution was obtained by an LP-metric approach due to the existence of none commensurable statements of objective function.

Despite the applicability of feeder lines or feeder shops concept in production environments, the amount of literature devoted to this area of research is relatively poor. Luss (1989) considered the interrelation between feeder shops and final assembly from sequencing point of view. He derived that manufacturing synchronization between feeder shops and final assembly would be more difficult when the variability of production parameters production intervals increases. Li et al. (2005) considered the reliability statistics application of feeder lines in throughput analysis and continuous improvement of manufacturing systems. They presented a simple approximation approach to estimate the reliability statistics of feeder lines from the associated assembly station's collected blocking and starving information and proved the role of feeder line speed on the approximation improvement. They also showed the efficiency of the presented approach for throughput estimation. Azzi et al. (2012) used the concept of feeder lines for balancing mixed model assembly systems in assemble-to-order (ATO) environments in which some of the components were assembled on the secondary lines as feeder lines and connected to the main line under a 'pull philosophy'.

In spite of the above-mentioned researches, in terms of feeder concept in manufacturing environments, neither of them have considered the crucial aspect of this issue on material handling costs of the system. This shortcoming in the manufacturing literature is tackled in this paper by designing a facilities layout system in which part manufacturing cells serve as the feeder shops for sub-assembly cells and sub-assembly cells fulfill the same role for final assembly line aiming at minimizing the overall material handling costs within the shop floor.

3. Problem formulation

The problem under study is formulated as an NLIP model. The proposed model determines the optimal layout of the machines/tasks within each part manufacturing/sub-assembly cell locations as well as the optimal layout of cells and final assembly line within the shop floor. The objective function of the proposed model containing six terms is defined as the minimization of overall material handling costs of the shop floor. Remarkably, the model is considered as a comprehensive quadratic assignment of facility layout problem.

Generally, the following assumptions are made for the proposed model formulation:

Assumptions

1. Part manufacturing cell formation stage is performed in advance and parts and machine groups have been allocated to cells.
2. Demands for all product models are deterministic and known.

3. Part manufacturing/sub-assembly cells are laid out in predetermined part manufacturing/sub-assembly cell locations which are equal in shape.
4. Parts, in-process sub-assemblies and sub-assemblies are moved within the shop floor in batches which vary depending on their bulk with different movement costs per batch.
5. The process rout of each part type consists of a number of operations with a predetermined sequence on the machines.
6. Allocation of assembly tasks of sub-assemblies and final product models to their relevant locations are performed based on the precedence relations of sub-assemblies and combined precedence relations of product models.
7. Only one kind of sub-assembly is produced in each sub-assembly cell while the final assembly line is a mixed model assembly line producing different models of a product type.
8. The number of machine/task locations of each part manufacturing/sub-assembly cell is equal to the number of machines/tasks existing in the greatest part manufacturing/sub-assembly cell.
9. Parts/Sub-assemblies produced in part manufacturing/sub-assembly cells are used to feed sub-assembly cells/ final-assembly line.
10. Only one kind of task is performable in each task location of sub-assembly cell or final-assembly line.
11. Assembly tasks of sub-assemblies/final products are performed along U-shape layout of sub-assembly cells/final-assembly line.
12. To increase the flexibility of the shop floor the U-shape layouts of sub-assembly cells and final-assembly line are allowed to get optional orientation in clockwise or anticlockwise direction.
13. Material handling costs between each pair of locations increase linearly based on the rectilinear distance between the centroid of each pair of locations except for the distance between part manufacturing and sub-assembly cells as well as sub-assembly cells and final assembly task locations which are based on rectilinear distance between the centroid of part manufacturing cell and the U-line entrance of sub-assembly cell locations, the U-line exit of sub-assembly cell locations and the centroid of final assembly task locations respectively.

Fig. 1 shows a simple example of the corresponding shop floor in which part manufacturing cells one, two and three are arranged in part manufacturing cell locations number one, three and two respectively. Each part manufacturing cell location contains four machine locations, numbered as $ML1,...,ML4$, which is equal to the number of machines incorporated in part manufacturing cell one as the greatest cell according to the fifth assumption. Also, there are sub-assembly cells one, two and three located in their relevant locations number three, two and one respectively. Each sub-assembly cell location contains five task locations, numbered as $TL1,...,TL5$, which is equal to the number of tasks incorporated in sub-assembly cell two as the greatest cell according to the fifth assumption. The vacant location of each cell as a consequence of cell size differences is highlighted in the floor. In addition, there are seven final assembly tasks located in their relevant task locations numbered as $TL1,...,TL7$ to produce different models of a product type. The precedence relations diagram of each sub-assembly tasks as well as the combined precedence relations diagram of final assembly tasks are illustrated in Figure 2 Notably, the line orientation of final assembly line and sub-assembly cell three are arranged in anticlockwise direction, however; the contrary is taken place for sub-assembly cells one and two. Hence, the tasks of final assembly line and sub-assembly cells are arranged based on both, precedence relations and line orientation.

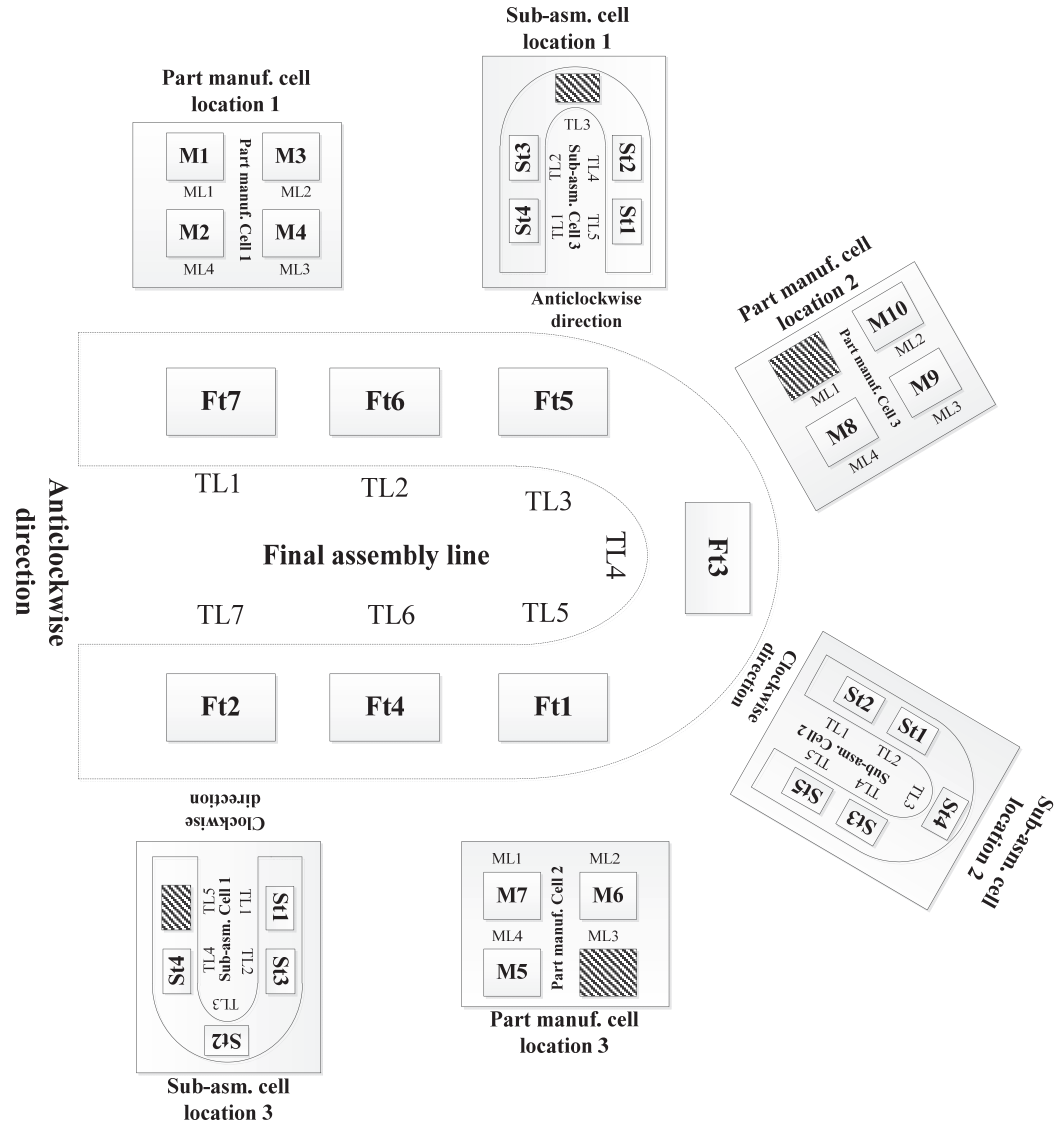


Fig. 1. A simple example of the layout by proposed model

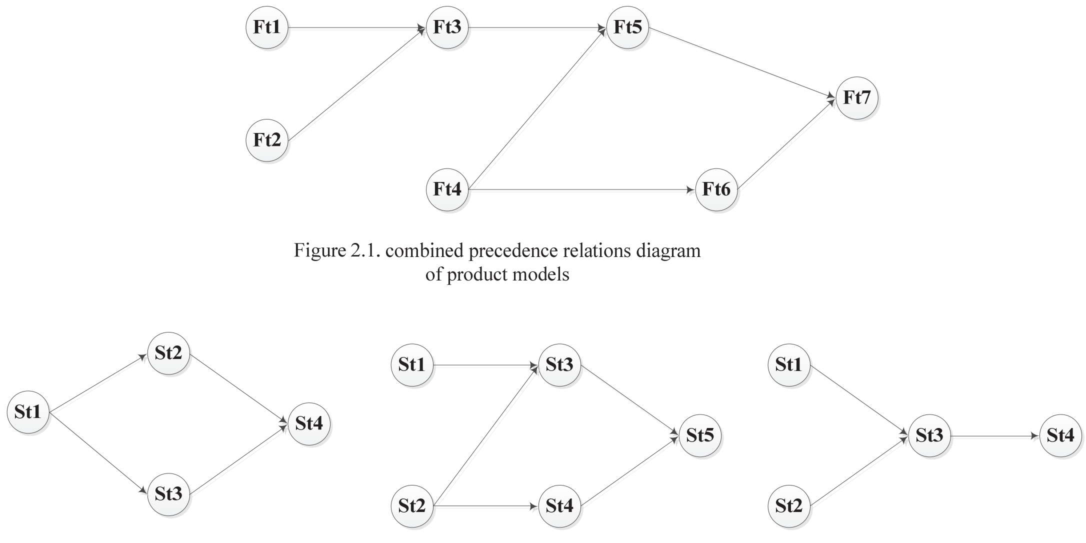


Figure 2.1. combined precedence relations diagram of product models

Figure 2.2.a. precedence relations diagram of sub-assembly 1

Figure 2.2.b. precedence relations diagram of sub-assembly 2

Figure 2.2.c. precedence relations diagram of sub-assembly 3

Fig. 2. Precedence relations of sub-assemblies and product models

Indices

o	Index for product models $(o = 1,...,O)$
p	Index for parts $(p = 1,...,P)$
i, j	Indices for machines $(i, j = 1,...,M)$
k, l	Indices for machine locations of each cell $(k, l = 1,...,ML)$
c, d	Indices for part manufacturing cells $(c, d = 1,...,C)$
e, f	Indices for part manufacturing cell locations $(e, f = 1,...,C)$
s	Index for sub-assemblies or sub-assembly cells $(s = 1,...,Sa)$
q	Index for sub-assembly cell locations $(q = 1,...,Sa)$
m_s, n_s	Indices for sub-assembly tasks $(m_s, n_s = 1,...,T_s)$
t_s, r_s	Indices for sub-assembly task locations $(t_s, r_s = 1,...,TLs)$
m_f, n_f	Indices for final assembly tasks $(m_f, n_f = 1,...,Tf)$
t_f, r_f	Indices for final assembly task locations $(t_f, r_f = 1,...,TLf)$

Parameters

L	A large positive number
O	Number of product models
P	Number of parts
M	Number of machines
M_c	Number of machines dedicated to part manufacturing cell c
ML	Number of candidate machine locations in each part manufacturing cell (Equal to $Max\{M_c\}$)
C	Number of part manufacturing cells
Sa	Number of sub-assemblies or sub-assembly cells
D_o	Demand of model o
D_p	Demand of part p
D_s	Demand of sub-assembly s
$(m_s, n_s) \in \text{Pred}_s$	A precedence relation representing assembly task m_s of sub-assembly s is the predecessor of assembly task n_s
$(m_f, n_f) \in \text{Pred}_f$	A precedence relation representing assembly task m_f of product models is the predecessor of assembly task n_f
U_{ps}^{part}	Usage factor of part p in sub-assembly s
U_{so}^{sa}	Usage factor of sub-assembly s in model o
T_s	Number of tasks to be performed for producing sub-assembly s
TLs	Number of candidate task locations in each sub-assembly cell (Equal to $Max\{T_s\}$)
Tf	Number of tasks to be performed in the final assembly line for producing different models

$F_{m_s n_s}$ Fraction of in-process sub-assembly $s$needed for performing task n_s immediately after task m_s ($(m_s, n_s) \in \mathrm{Pred}_s$)

F_{sm_f} Fraction of sub-assembly s needed for performing task m_f in the final assembly line immediately after production in sub-assembly cell s

$F_{om_f n_f}$ Fraction of in-process model o needed for performing task n_f immediately after task m_f ($(m_f, n_f) \in \mathrm{Pred}_f$)

B_p^{inter} Inter-cell batch size of part p

B_p^{intra} Part manufacturing intra-cell batch size of part p

$B_{m_s}^{intra}$ Intra-cell batch size of in-process sub-assembly s after performing task m_s

B_s Bach size of sub-assembly s handled from sub-assembly cell s to final assembly task locations

$B_{om_f}^{intra}$ Batch size of in-process model o after performing task m_f

d_{kl}^m Distance between machine locations k and l (k not equal l)

d_{ef}^c Distance between part manufacturing cell locations e and f (e not equal f)

dc_{eq}^{cs} Distance between part manufacturing cell location e and the U-line entrance of sub-assembly cell location q if U-line is formed in clockwise direction

dac_{eq}^{cs} Distance between part manufacturing cell location e and the U-line entrance of sub-assembly cell location q if U-line is formed in anticlockwise direction

$d_{t_s r_s}^s$ Distance between sub-assembly task locations t_s and r_s (t_s not equal r_s)

$dc_{qt_f}^{sf}$ Distance between the U-line exit of sub-assembly cell location q and final assembly task location t_f if U-line is formed in clockwise direction

$dac_{qt_f}^{sf}$ Distance between the U-line exit of sub-assembly cell location q and final assembly task location t_f if U-line is formed in anticlockwise direction

$d_{t_f r_f}^f$ Distance between final assembly task locations t_f and r_f (t_f not equal r_f)

C_p^{inter} Inter-cell movement cost of part p per batch, per unit distance

C_p^{intra} Part manufacturing intra-cell movement cost of part p per batch, per unit distance

$C_{m_s}^{intra}$ Intra-cell movement cost of in-process sub-assembly s after performing task m_s per batch, per unit distance

C_s Movement cost of sub-assembly s per batch, per unit distance

$C_{om_f}^{intra}$ Movement cost of in-process model o after performing task m_f per batch, per unit distance

Fp_{cd}^{inter} Material handling cost between manufacturing cells c and d (c is not equal d) per unit distance

Fm_{ijc}^{intra} Material handling cost between machines i and j (i is not equal j) within manufacturing cell c per unit distance

Fps_{cs}^{inter} Material handling cost between manufacturing cell c and sub-assembly cell $s$per unit distance

$Fs^{\text{int}ra}_{m_s n_s}$ Material handling cost between tasks m_s and n_s (m_s is not equal n_s) within sub-assembly cell s per unit distance

$Fsf^{\text{int}er}_{sm_f}$ Material handling cost between sub-assembly cell s and final assembly task t_f per unit distance

$Ff^{\text{int}ra}_{m_f n_f}$ Material handling cost between final assembly tasks m_f and n_f (m_f is not equal n_f) per unit distance

Binary parameter

To_{pc} 1; If the terminal operation of part p is performed on a machine in cell c and 0; Otherwise

Parameters obtained from part manufacturing cell formation phase

a_{ic} 1; If machine i is already assigned to part manufacturing cell c and 0; Otherwise

b_{ijp} 1; If machine j is applied immediately after machine i to process part p and 0; Otherwise

Binary variables

x_{ik} 1; If machine i is located to the machine location k and 0; Otherwise

y_{ce} 1; If part manufacturing cell c is located to the part manufacturing cell location e and 0; Otherwise

α_{sq} 1; If sub-assembly s is located to the sub-assembly cell location q and 0; Otherwise

$\beta_{m_s t_s s}$ 1; If task m_s of sub-assembly s is located to the sub-assembly task location t_s and 0; Otherwise

$\gamma_{m_f t_f}$ 1; If task m_f of final-assembly line is located to the final-assembly task location t_f and 0; Otherwise

Lo_s 1; If U-line orientation of sub-assembly $s$is clockwise and 0; Otherwise

Lo_f 1; If final assembly U-line orientation is clockwise and 0; Otherwise

$$
\begin{aligned}
\min\ z \quad &= \sum_{c=1}^{C}\sum_{i=1}^{M_c}\sum_{j=1}^{M_c}\sum_{k=1}^{ML}\sum_{l=1}^{ML} Fm^{\text{int}ra}_{ijc} d^{m}_{kl} x_{ik} x_{jl} + \sum_{c=1}^{C}\sum_{d=1}^{C}\sum_{e=1}^{C}\sum_{f=1}^{C} Fp^{\text{int}er}_{cd} d^{c}_{ef} y_{ce} y_{df} \\
&+ \sum_{c=1}^{C}\sum_{s=1}^{Sa}\sum_{e=1}^{C}\sum_{q=1}^{Sa} Fps^{\text{int}er}_{cs} y_{ce}\alpha_{sq}\left(dc^{cs}_{eq} Lo_s + dac^{cs}_{eq}(1-Lo_s)\right) \\
&+ \sum_{s=1}^{Sa}\sum_{m_s=1}^{T_s}\sum_{t_s=1}^{TL_s}\sum_{n_s/(m_s,n_s)\in \Pr ed_s}\sum_{r_s=1}^{TL_s} Fs^{\text{int}ra}_{m_s n_s} d^{s}_{t_s r_s}\beta_{m_s t_s s}\beta_{n_s r_s s} \\
&+ \sum_{c=1}^{C}\sum_{d=1}^{C}\sum_{e=1}^{C}\sum_{f=1}^{C} Fsf^{\text{int}er}_{sm_f}\alpha_{sq}\gamma_{m_f t_f}\left(dac^{sf}_{qt_f} Lo_s + dc^{sf}_{qt_f}(1-Lo_s)\right) \\
&+ \sum_{m_f=1}^{T_f}\sum_{t_f=1}^{T_f}\sum_{n_f/(m_f,n_f)\in \Pr ed_f}\sum_{r_f=1}^{T_f} Ff^{\text{int}ra}_{m_f n_f} d^{f}_{t_f r_f}\gamma_{m_f t_f}\gamma_{n_f r_f}
\end{aligned} \tag{1}
$$

subject to

$$x_{ik}x_{jk} + a_{ic}a_{jc} \le 1 \qquad \forall\ i \ne j, k, c \tag{2}$$

$$\sum_{k=1}^{ML} x_{ik} = 1 \qquad \forall\ i \tag{3}$$

$$\sum_{c=1}^{ML} y_{ce} = 1 \qquad \forall\, e \tag{4}$$

$$\sum_{e=1}^{ML} y_{ce} = 1 \qquad \forall\, c \tag{5}$$

$$\sum_{s=1}^{Sa} \alpha_{sq} = 1 \qquad \forall\, q \tag{6}$$

$$\sum_{q=1}^{Sa} \alpha_{sq} = 1 \qquad \forall\, s \tag{7}$$

$$\sum_{m_s=1}^{Ts} \beta_{m_s t_s s} \leq 1 \qquad \forall\, s, t_s \tag{8}$$

$$\sum_{t_s=1}^{Ts} \beta_{m_s t_s s} = 1 \qquad \forall\, s, m_s \tag{9}$$

$$\sum_{t_s=1}^{T_s} t_s \left(\beta_{m_s t_s s} - \beta_{n_s t_s s}\right) \leq L \times (1 - Lo_s) \qquad \forall\, s, (m_s, n_s) \in \mathrm{Pred}_s \tag{10}$$

$$\sum_{t_f=1}^{T_S} t_f \left(\beta_{m_S t_S} - \beta_{n_S t_S}\right) \geq -L \times Lo_S \qquad \forall\, s, (m_s, n_s) \in \mathrm{Pred}_s \tag{11}$$

$$\sum_{m_f=1}^{Tf} \gamma_{m_f t_f} = 1 \qquad \forall\, t_f \tag{12}$$

$$\sum_{t_f=1}^{Tf} \gamma_{m_f t_f} = 1 \qquad \forall\, m_f \tag{13}$$

$$\sum_{t_f=1}^{Tf} t_f \left(\gamma_{m_f t_f} - \gamma_{n_f t_f}\right) \leq L\,(1 - Lo_f) \qquad \forall\, (m_f, n_f) \in \mathrm{Pred}_f \tag{14}$$

$$\sum_{t_f=1}^{Tf} t_f \left(\gamma_{m_f t_f} - \gamma_{n_f t_f}\right) \geq -L \times Lo_f \qquad \forall\, (m_f, n_f) \in \mathrm{Pred}_f \tag{15}$$

$$Fm_{ijc}^{\mathrm{int}ra} = \sum_{p=1}^{P} a_{ic} a_{jc} b_{ijp} C_p^{\mathrm{int}ra} \left\lceil \frac{D_p}{B_p^{\mathrm{int}ra}} \right\rceil \qquad \forall\, i \neq j, c \tag{16}$$

$$Fp_{cd}^{\mathrm{int}er} = \sum_{p=1}^{P} \sum_{i=1}^{M} \sum_{\substack{j=1 \\ j \neq i}}^{M} a_{ic} a_{jd} b_{ijp} C_p^{\mathrm{int}er} \left\lceil \frac{D_p}{B_p^{\mathrm{int}er}} \right\rceil \qquad \forall\, c \neq d \tag{17}$$

$$Fps_{cs}^{\mathrm{int}er} = \sum_{p=1}^{P} To_{pc} C_p^{\mathrm{int}er} \left\lceil \frac{U_{ps}^{part} D_s}{B_p^{\mathrm{int}er}} \right\rceil \qquad \forall\, c, s \tag{18}$$

$$Fs_{m_s n_s}^{\mathrm{int}ra} = C_{m_s}^{\mathrm{int}ra} \left\lceil \frac{F_{m_s n_s} D_s}{B_{m_s}^{\mathrm{int}ra}} \right\rceil \qquad \forall\, s, (m_s, n_s) \in \mathrm{Pred}_s \tag{19}$$

$$Fsf_{sm_f}^{\mathrm{int}er} = C_s \left\lceil \frac{F_{sm_f} \sum_{o=1}^{O} U_{so}^{sa} D_o}{B_s} \right\rceil \qquad \forall\, s, m_f \tag{20}$$

$$Fs_{om_f n_f}^{\mathrm{int}ra} = C_{om_f}^{\mathrm{int}ra} \left\lceil \sum_{o=1}^{O} \frac{F_{om_f n_f} D_o}{B_{om_f n_f}^{\mathrm{int}ra}} \right\rceil \qquad \forall\, o, (m_f, n_f) \in \mathrm{Pred}_f \tag{21}$$

$$D_p = \sum_{s=1}^{Sa} U_{ps}^{part} D_s \qquad \forall p \qquad (22)$$

$$D_s = \sum_{o=1}^{O} U_{so}^{sa} D_o \qquad \forall s \qquad (23)$$

$$x_{ik}, y_{ce}, \alpha_{sq}, \beta_{m_s t_s s}, \gamma_{m_f t_f} \in \{0,1\} \qquad \forall i, k, c, e, s, q, m_s, t_{s,} m_f, t_f \qquad (24)$$

Model description

The objective function of the proposed model presented in Eq. (1) minimizes the overall material handling costs of the system via six terms. The first term computes the part manufacturing intra-cell handling costs. Likewise, the second term concerns with the part manufacturing inter-cell handling costs. Handling costs related to the part transmission from part manufacturing cells to sub-assembly cells (i.e. part manufacturing to sub-assembly inter-cell costs) is computed by the third term. The forth term is the sub-assembly intra-cell handling costs (i.e. the overall costs resulted from in-process material flow between tasks location of each sub-assembly cell). Handling costs regarding sub-assembly flow from sub-assembly cells feeding final assembly line is computed by the fifth term. Finally, the sixth term computes the costs of material flow between task locations within final assembly line. Eq. (2) and Eq. (3) ensure that each machine is assigned to one machine location. It is noteworthy that if machines i and j have already assigned to the same cell c , they cannot be located to the same machine location in that cell. This is also ensured through Eq. (3). Eq. (4) and Eq. (5) ensure that each cell is assigned to one cell location and vice versa. Eq. (6) and Eq. (7) ensure that each sub-assembly cell, producing only one type of sub-assembly, is assigned to one sub-assembly cell location and vice versa. For each sub-assembly, Eq. (8) and Eq. (9) ensure that each sub-assembly task is located to one sub-assembly task location and each task location either contain one task or remain vacant. Eqs. (10-11) are precedence constraint which indicates that for each sub-assembly s, a certain task m_s is assignable to sub-assembly task location t_s , if all of its predecessors have been already located to the previous locations. Eq. (10) and Eq. (11) also specify the U-line orientation of sub-assembly s in clockwise or anticlockwise direction respectively. Eqs. (12-15) have the same applicability as Eqs. (8-11) for final-assembly tasks. To conserve space we withdraw explaining Eqs. (12-15). Eqs. (16-21) calculates the material handling cost of each two facilities per unit distance. The overall demand of each part and sub-assembly is calculated by Eq. (22) and Eq. (23) respectively. Finally, Eq. (24) is the sign constraint of binary variables.

Linearization of the model

As exact solutions of small-sized problems presented in Section 6 for the proposed model are calculated using GAMS-CPLEX mathematical programming software to verify the proposed GA and MA, Linearization of the model seems to be crucial. Linearization reduces complexity of solving model by CPLEX solver to validate the proposed model. Also, optimal solutions are compared with those of the same size solutions obtained by GA and MA to verify these two proposed algorithms. The presented model is a nonlinear integer programming due to several quadratic terms existing in the model. To transform the non-linear terms into linear terms, binary variables A_{ikjl} , B_{cedf} , C_{cesq} , C'_{cesq} , $D_{m_s t_s n_s r_s s}$, $E_{sqm_f t_f}$, $E'_{sqm_f t_f}$ and $F_{m_f t_f n_f r_f}$ are introduced and replaced by quadratic terms of objective function; $x_{ik}x_{jl}$, $y_{ce}y_{df}$, $y_{ce}\alpha_{sq}Lo_s$, $y_{ce}\alpha_{sq}$, $\beta_{m_s t_s s}\beta_{n_s r_s s}$, $\alpha_{sq}\gamma_{m_f t_f}Lo_s$, $\alpha_{sq}\gamma_{m_f t_f}$ and $\gamma_{m_f t_f}\gamma_{n_f r_f}$ respectively. Also, A_{ikjk} is replaced by the quadratic term of Eq. (2) (i.e. $x_{ik}x_{jk}$). Furthermore Eqs. (25-32) are added to the proposed quadratic nonlinear model and a zero-one integer linear programming is obtained.

$$\begin{cases} A_{ikjl} \geq x_{ik} + x_{jl} - 1 \\ A_{ikjl} \leq x_{ik} \\ A_{ikjl} \leq x_{jl} \end{cases} \qquad \forall\ i \neq j,\ k \neq l \tag{25}$$

$$\begin{cases} B_{cedf} \geq y_{ce} + y_{df} - 1 \\ B_{cedf} \leq y_{ce} \\ B_{cedf} \leq y_{df} \end{cases} \qquad \forall\ c \neq d,\ e \neq f \tag{26}$$

$$\begin{cases} C_{cesq} \geq y_{ce} + \alpha_{sq} + Lo_s - 2 \\ C_{cesq} \leq y_{ce} \\ C_{cesq} \leq \alpha_{sq} \\ C_{cesq} \leq Lo_s \end{cases} \qquad \forall\ c, e, s, q \tag{27}$$

$$\begin{cases} C'_{cesq} \geq y_{ce} + \alpha_{sq} - 1 \\ C'_{cesq} \leq y_{ce} \\ C'_{cesq} \leq \alpha_{sq} \end{cases} \qquad \forall\ c, e, s, q \tag{28}$$

$$\begin{cases} D_{m_s t_s n_s r_s s} \geq \beta_{m_s t_s s} + \beta_{n_s r_s s} - 1 \\ D_{m_s t_s n_s r_s s} \leq \beta_{m_s t_s s} \\ D_{m_s t_s n_s r_s s} \leq \beta_{n_s r_s s} \end{cases} \qquad \forall\, m_s \neq n_s,\ t_s \neq r_s, s \tag{29}$$

$$\begin{cases} E_{sqm_f t_f} \geq \alpha_{sq} + \gamma_{m_f t_f} + Lo_s - 2 \\ E_{sqm_f t_f} \leq \alpha_{sq} \\ E_{sqm_f t_f} \leq \gamma_{m_f t_f} \\ E_{sqm_f t_f} \leq Lo_s \end{cases} \qquad \forall\, s, q, m_f, t_f \tag{30}$$

$$\begin{cases} E'_{sqm_f t_f} \geq \alpha_{sq} + \gamma_{m_f t_f} - 1 \\ E'_{sqm_f t_f} \leq \alpha_{sq} \\ E'_{sqm_f t_f} \leq \gamma_{m_f t_f} \end{cases} \qquad \forall\, s, q, m_f, t_f \tag{31}$$

$$\begin{cases} F_{m_f t_f n_f r_f} \geq \gamma_{m_f t_f} + \gamma_{n_f r_f} - 1 \\ F_{m_f t_f n_f r_f} \leq \gamma_{m_f t_f} \\ F_{m_f t_f n_f r_f} \leq \gamma_{n_f r_f} \end{cases} \qquad \forall\, m_f \neq n_f,\ t_f \neq r_f \tag{32}$$

4. Solution approaches

4.1. Genetic algorithm approach

Biologically motivated approaches have gained increasing popularity in solving complex optimization problems in the last decades. GA which was first introduced in 1970s by Holland (1975) is one of the optimization approaches constructed based on evolutionary processes occurring in the natural systems. GA strive to synthesize the good features of different individuals within a population to create individuals who are better suited. For this purpose, GA operators such as crossover, mutation, and selection are utilized to evolve individuals in different iterations in order to reach the desired solution. GA can be used in different types of problems with appropriate genetic representation to create initial solutions, fitness function and operators. GAs have been widely used to solve layout problems and CMS layout in particular either separately or hybridized with other algorithms. For example Wu et al. (2007) and Mohammadi and Forghani (2014) used GA for CF and layout problem in CMS. Javadi et al. (2014) developed GA both, separately and synthesized with EM-like algorithm to solve intra- and inter-cell layout problem. In the proposed layout problem which belongs to NP-hard class of problems, a GA approach is utilized since exact solution methods are highly time consuming for large-sized problems. The proposed GA elements consisting of Initialization, parent selection, crossover and mutation is described in the following discussion.

Initialization

The first step in GA procedure is to initialize a number of feasible solutions as initial populations (i.e. generation 0). For this purpose, sets of chromosomes are generated and each set represents an initial population. There are five chromosomes in each chromosome set where the position of part manufacturing cells, sub-assembly cells, machines, sub-assembly tasks and final assembly tasks are determined by separate chromosomes. Number of genes constructing chromosomes of part manufacturing cells, sub-assembly cells and final assembly tasks are equal to the number of part manufacturing cells (C), number of sub-assembly cells (S) and number of final assembly tasks (Tf) respectively. However, number of genes for the chromosomes of machines and sub-assembly tasks of each cell are equal to $Max\left\{M_c\right\}$ and $Max\left\{T_s\right\}$ respectively, which guarantees the uniformity of part manufacturing cells and sub-assembly cells. Figure 3 shows the chromosome representation equivalent to layout given in Figure 1 for the proposed GA.

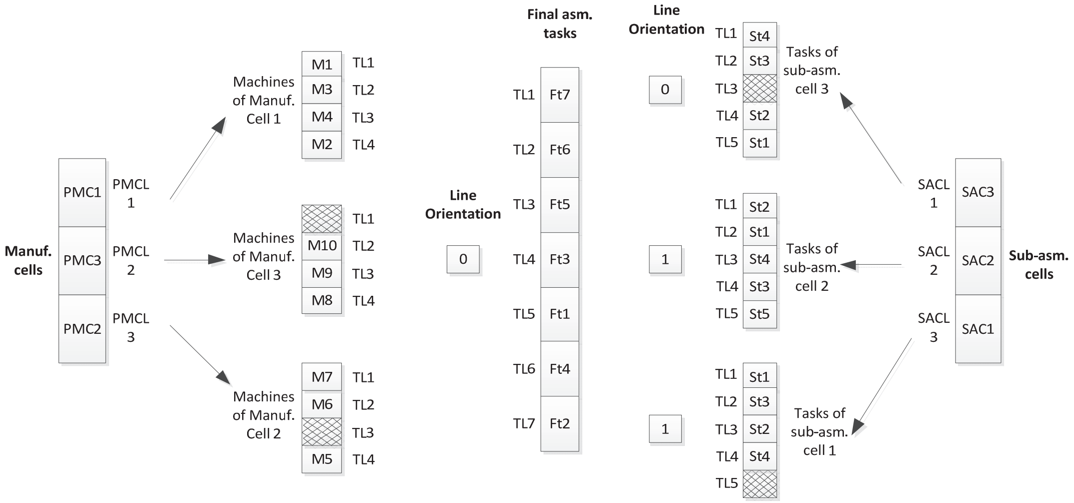


Fig. 3. Solution representation of the layout given in Fig. 1.

Parent selection

The Roulette Wheel selection procedure, proposed by Goldberg (1989), is used to form the next generation. Selection strategy attempts to allow the "fittest" individuals from the current population to be considered more often to reproduce the children for the next generation. For this purpose, a probability value p_i is assigned to each set of chromosome (i.e., each chromosome set of the current population) that is proportional to its fitness value f_i and calculated as follows:

$$p_i = \frac{\dfrac{1-f_i}{N+\sum_{i=1}^{N} f_i}}{N-1} \qquad i=1,\dots,N \tag{33}$$

All chromosome sets are ranked according to their probability value in ascending order. Then, the cumulative probability value c_i is calculated for each chromosome set and a random number r is generated uniformly in the range$[0, 1]$. Chromosome set ch_i is selected as the parent for the next generation if $c_{i-1} < r \le c_i$.

Crossover

Crossover is the main operator of GA procedure by which the information of two parents are combined to produce two children. Different crossover operators such as one-point, two point, uniform and order crossover (OX) have been developed and used depending on chromosome structure and problem type.

For the proposed GA, OX is selected due to the structure of the chromosomes. However the chromosome structure of machines and sub-assembly tasks of each cell is a bit different to that of traditionally used OX due to the existence of vacant locations and a modified version of OX (MOX) is presented.

In primary OX, chromosome structure is formed by a permutation order in which none of two or more genes are the same. For the chromosome structure of machines and sub-assembly tasks of each cell, however, since the number of machines and machine locations or sub-assembly tasks and sub-assembly task locations are not necessarily equal and the existence of repetitive positions is probable accordingly. Probable repetitive positions are vacant locations, if exist, calculated as follows:

$$RP_c = Max\ \{M_c\} - M_c \qquad \forall\ c \tag{34}$$

$$RP_s = Max\ \{T_s\} - T_s \qquad \forall\ s \tag{35}$$

Eq. (34) calculates the number of repetitive positions (i.e. the number of vacant machine location) for part manufacturing cell c. While, Eq. (35) calculates the number of vacant sub-assembly tasks for sub-assembly cell s. Implementation of MOX is clarified by the steps as follows:

Step 1) Select two individuals of current population as parents.

Step 2) Place two brackets within the genes of parents randomly so as to divide each chromosome into three sub-chromosomes.

Step 3) Consider the middle sub-chromosome of parent1/parent2 as the middle sub-chromosome of offspring1/offspring2

Step 4) Consider the left and right sub-chromosomes of parent2/parent1 as the left and right sub-chromosomes of offspring1/offspring2 respectively

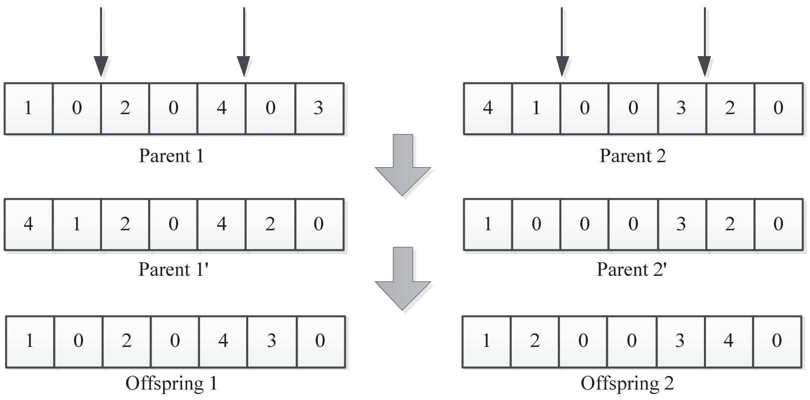
Step 5) Calculate the number of vacant position that the left and right sub-chromosomes of offspring1/offspring2 needs (i.e. the number of vacant position acquired by Eq. 22 or 23 except for those located in the middle sub-chromosome)

Step 6) Form a sequence of genes needed for offspring1/offspring2 consisting of vacant positions calculated in step 5 and non-assigned machines/sub-assembly tasks according to their order in parent2/parent1 from left to right

Step 7) Replace the extra genes of offspring1/offspring2 with those sequence of genes acquired in step 6

The mechanism of MOX crossover on two given locations is illustrated by Figure 4 as follows:

TL5



Parent 1: 1 0 2 0 4 0 3

Parent 2: 4 1 0 0 3 2 0

Parent 1': 4 1 2 0 4 2 0

Parent 2': 1 0 0 0 3 2 0

Offspring 1: 1 0 2 0 4 3 0

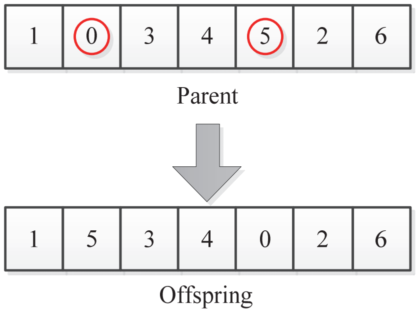
Offspring 2: 1 2 0 0 3 4 0

Fig. 4. MOX crossover on two given locations

The offsprings of sub-assembly task and final assembly task locations obtained from the crossover operator might violate precedence relations constraints. Violation of these chromosomes can be repaired by swapping each pair of tasks violating precedence relations constraint.

Mutation

Another important operator of GA is mutation. This operator increases the diversity of algorithm to reach the near optimum solution and prevent from converging to a local optimum by changing one or more genes of a single parent to produce a new offspring. In the proposed GA two kinds of mutation operator are used; swapping mutation and line orientation (LO) mutation. To preserve the diversity of the algorithm, only one of these two mutations are applied in each iteration of the algorithm based on random choice. In swapping mutation, two genes of the chromosome of a selected parent are swapped to generate a new offspring for the next generation. In the current algorithm, swapping mutation is applied to the chromosome of part manufacturing cell locations, Sub-assembly cell locations and machine locations of each cell within a chromosome set of the selected parent. Figure 5 illustrates the mechanism of swapping mutation on the chromosome of a given location. LO mutation is particularly developed for the current problem by which the line orientation of the final assembly line and each sub-assembly cells are changed from clockwise to anticlockwise direction or conversely.



Parent: 1 0 3 4 5 2 6

Offspring: 1 5 3 4 0 2 6

Fig. 5. Swapping mutation of a given location

4.2. Memetic algorithm approach

GAs often perform well in global search as a diversification tool, but they are typically poor in converging to a local optimum. Hence, memetic algorithms (MAs) are adopted to search over the subspace of local optima by combining GA with local improvement strategies such as local neighborhood search, TS and SA. This is achievable by applying local search to offspring obtained by genetic operators before it is inserted to the next generation. In the proposed MA procedure, a TS-based local search approach is applied to repair a fraction of offsprings produced by crossover operator of proposed GA before they are inserted to the next population. TS-based local search approach of proposed MA is clarified in the following sub-section.

4.2.1. TS-based local search approach

The local search engine is one of the substantial components of MA as mentioned before. In the proposed MA a TS-based local search is used to search in the neighborhood of offspring solutions to obtain solutions with lower cost. TS is one of the effective meta-heuristic methods developed by Glover (1989). This method has been effectively used to find near-optimal solutions for a wide variety of combinatorial problems. Starting from a current solution (CS) in the feasible space, TS strive to find a better solution (BS) in its neighborhood during α times for each iteration. TS algorithm employs a Tabu mechanism to prevent searching over the neighborhood of repetitive solutions. In this regard, special specifications of newly produced solution is saved in a short-term memory, also known as tabu list (TL), and banned for the next θ solutions (Samarghandi & Eshghi 2010). TS also exploits mechanisms such as diversification, intensification and aspiration. Diversification forces the algorithm to search a vast terrain of the solution space before converging to the final solution. Intensification, however, assures to perform a more detailed search in the neighborhood of solutions with desirable specifications. Aspiration override the existence of specific solutions in the TL (Glover & Laguna, 1997).

Initialization

TS algorithm needs a number of feasible solution to start exploration. In this paper TS is applied to crossover population as a local search engine for MA. Accordingly, offspring solutions obtained by crossover are directly stored in a long-term memory called adaptive memory (AM) and used for future explorations.

Selection

The process of selecting an initial solution from AM by a proper probability function to start exploration is the infrastructure of diversification and intensification in TS algorithm. In the proposed TS algorithm the Rolette Wheel selection procedure explained in Section 4.1 is used. The procedure tend to select solutions with desirable specifications more frequently to fulfil intensification. However, to diversify the solutions, less desirable solutions of AM have possibility to be selected proportional to their probability value. As the exploration process continues and BS of each iteration is replaced with the worst solution of AM, the significance of diversification gradually decreases while the importance of intensification increases.

Neighborhood structure

A proper neighborhood structure is a substantial step in TS algorithm converging to final solution. The neighborhood structure of proposed TS algorithm is based upon “move” operator so that two randomly selected part manufacturing/sub-assembly cells of CS are swapped and stored in TL. Besides, two machines of swapped part manufacturing cells are randomly swapped if part manufacturing cells have been selected for “move” operation in the current sub-iteration. However, machines have nothing to do with TL. Notably, sub-assembly and final assembly tasks are not involved in TS due to restriction incurred by precedence relations. In other words, the search mechanism applied to the sub-assembly and final assembly tasks chromosomes are sufficient as precedence relations restrict the search space. The newly produced solution by “move” operator is compared to CS in terms of objective function value. If its objective function value is less than CS, it is replaced with CS if one of the following conditions is fulfilled

a) The exchanged elements (i.e. part manufacturing/sub-assembly cells) does not exist in TL
b) Despite existing in TL, the objective value of the solution is the best one existing in AM.

The second condition mentioned above is the aspiration criteria of the proposed TS algorithm by which tabu mechanism is overridden.

```
%% Initialization
Input: $pop$
Input: $Cr$
Input: $Mr$
$popc \leftarrow Cr * pop$
$popm \leftarrow Mr * pop$
$P \leftarrow$ Random_population();
$Q \leftarrow \emptyset$
%% Main Loop
while Time limit has not reached do
%% Crossover
    for $i \leftarrow 1$ to $(popc)$ do
        $Parents \leftarrow$ Roulette_Wheel($P$);
        $Offsprings \leftarrow$ Crossover($Parents$);
        Insert($Offsprings$, $Q$);
    end for
%% TS local search engine
    Input: TSpr
    Input: $\alpha$
    Input: $\theta$
    Input: L
    $K \leftarrow TSpr * popc$
    AM $\leftarrow$ Q
    for $m \leftarrow 1$ to $(K)$ do
        $CS \leftarrow$ Roulette_Wheel($AM$);
        $BS \leftarrow$ argmin$\{f(AM[m]), \forall m\}$;
        $BC \leftarrow CS$
        for $n \leftarrow 1$ to $(\alpha)$ do
            $TL \leftarrow \emptyset$
            $TSsol \leftarrow$ Move($CS$);
            if $((f(TSsol) < f(BC)$ and $TSsol \notin TL)$ or $TSsol < BS)$
                $BC \leftarrow TSsol$
                Insert($TSsol$, $TL$);
                for $l \leftarrow 1$ to $(L)$ do
                    if ($TL[l]$ has been banned for $\theta$ iterations) do
                        Remove($TL[l]$, $TL$);
                    end if
                end for
            end if
        end for
        Insert($BC$, $AM$);
        WS $\leftarrow$ argmax($f(AM[m]), \forall m$)
        Remove($WS$, $AM$);
    end for
    return AM;
%% Mutation
    for $j \leftarrow 1$ to $(popm)$ do
        $Parent \leftarrow$ Roulette_Wheel($P$);
        $Offspring \leftarrow$ Mutation($Parent$);
        Insert($Offspring$, $Q$);
    end for
    $R \leftarrow P \cup Q$
    Ranking_and_Crowding($R$);
    $P \leftarrow$ Select_Best_Population($R$);
end while
return $P$;
```

Fig. 6. The pseudo code of the proposed MA

The stepwise procedure of the proposed TS is as follows:

Step 1) Set the values of the control parameters, i.e., α (number of attempts to find a better solution in the neighborhood of the CS), 6 (TL size) and k (number of iterations of the algorithm)

Step 2) Set the crossover population of the GA as the AM for the TS approach.

Step 3) Choose a CS by Roulette Wheel operator within AM, search for better solutions in its neighborhood.

Step 4) Exchange the location of two part manufacturing/sub-assembly cells of CS according to "move" operator described above. If this exchange improve the CS, add it to the TL and ban to be exchanged for the next 6 iterations unless it is better than all solutions existing in the AM (aspiration criteria). Repeat this procedure for α times to generate the set of neighborhood solutions of the CS.

Step 5) Compare the best solution of the neighborhood set obtained from Step 4 with the CS. If the best neighborhood solution is better than the CS, exchange it with the worst solution existing in the AM.

Step 6) Proceed Step 3 as long as k iterations are fulfilled.

The pseudo code and flowchart of the proposed MA are given in Fig. 6 and Fig. 7 respectively. Since the pseudo code and flowchart of the proposed GA are obtained from MA after omitting TS local search engine, they are withdrawn to be represented here.

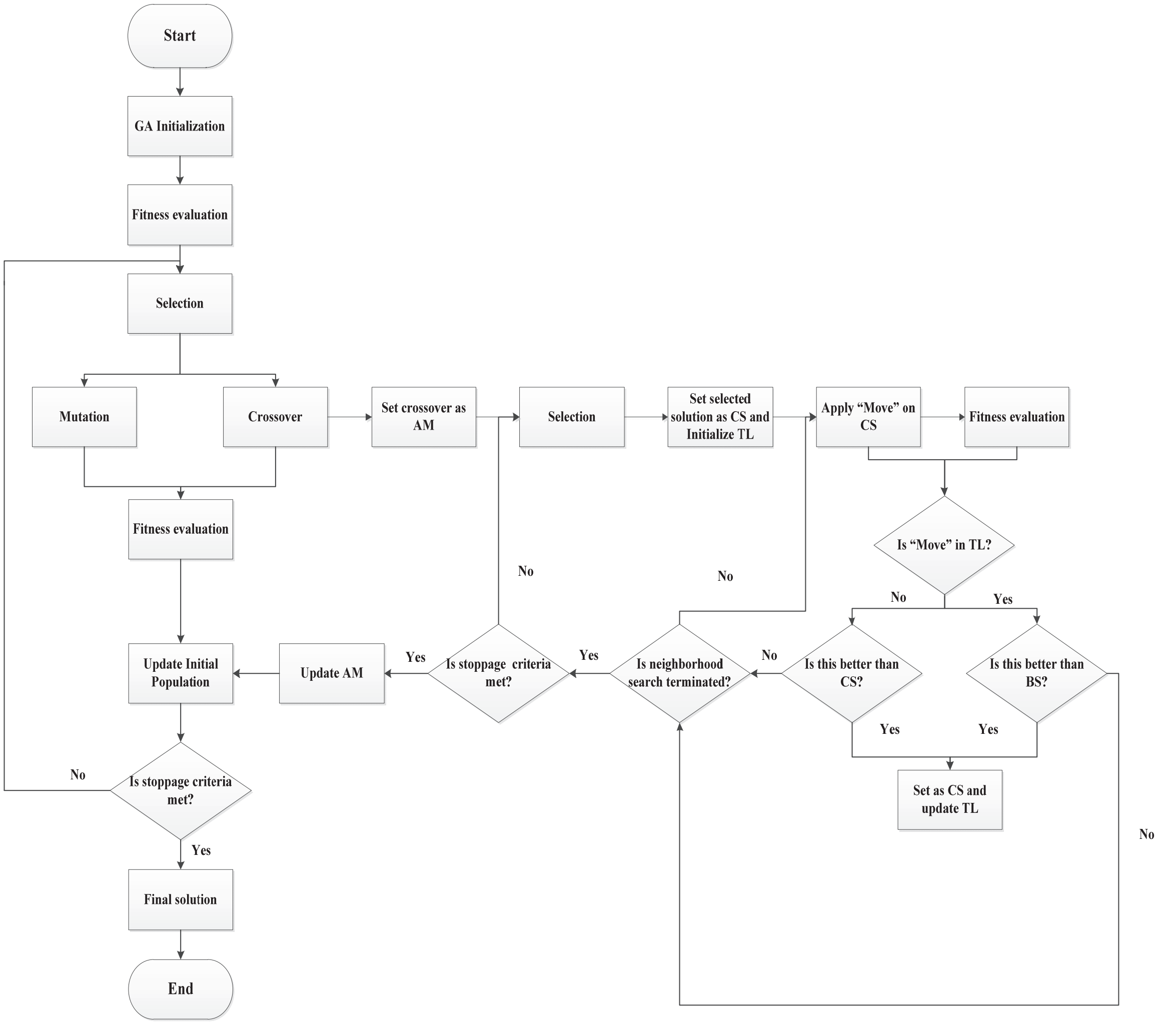


Fig. 7. Flowchart of the proposed MA

4. Parameters setting

The efficiency and effectiveness of meta-heuristic algorithms highly depend on the appropriate adjustment of the parameters. In most of the research studies, parameters are set based on either reference values of the literature or trial and error. Since the optimal parameters of the algorithms vary from one problem to another, specific adjustment of each newly developed algorithm substantially increases the quality of the algorithm solutions. In this regard, experimental design approaches such as full factorial design, response surface methodology (RSM) and Taguchi method have widely used to estimate the appropriate parameters of the algorithms. In this study the parameters optimization of the proposed GA and MA is performed two-fold. First, GA parameters are set according to Taguchi method. Then, the resultant optimal levels of GA parameters proceed to adjust the parameters of the proposed TS algorithm based on Taguchi method.

4.1. Taguchi method

The Taguchi experimental design has been extensively used for optimization problems. This method studies a large number of decision variables with only a small number of well-defined experimental sets (Nourmohammadi & Zandieh, 2011). Taguchi design is based on two major tools: orthogonal array (OA) and signal-to-noise (S/N) ratio. The OA is a matrix of numbers containing experimental schemes based on different levels of factors. The S/N ratio is the measure of variation and guarantees the robustness of this kind of experimental design. The term “signal” represents “mean response variable” as the desirable value and “noise” represents “standard deviation” as undesirable value (Mehdizadeh et al., 2015). In this method parameters are divided into two categories: controllable and uncontrollable (noise) parameters. All the parameters of an algorithm which are denotable in different levels are controllable parameters. However, noise refers to those parameters causing algorithm to deviate from its target value. Taguchi method attempts to minimize the impact of noise and find the best level of controllable parameters simultaneously on the robustness basis (Tsai et al., 2007; Naderi et al., 2009). In such conditions, it is necessary to replicate the experiments in different situations to decrease detrimental effect of noise parameters. In this study, since the proposed GA and MA must be adjusted over various test problems, they are classified into three problem sizes (i.e. small, medium and large) and set separately in order to remove noise effect of problem size. For each size category, five test problems are designed, as shown in Table 5 and implemented by MATLAB software on a computer with Intel® Core 5 CPU, 2.67 GHz speed and 4 GB RAM. Each test problem is run five times due to the stochastic nature of GA and MA in order to obtain more reliable information. Therefore, there are 25 results for each trial of OA to set parameters. The purpose of Taguchi method is to maximize the S/N ratio which is calculated by Eq. (36) for minimization problems for each parameter *i* on its related level *j*.

$$(S/N\ \text{ratio})_{ij} = -10\log_{10}\left(\frac{\sum z_{ij}^{2}}{n}\right) \qquad \forall\ i,j \tag{36}$$

Where z_{ij} is the objective function value using parameter i on level j and n is the number of times level j of parameter i is repeated over the runs of all trials.

4.2. GA parameters setting

The first step of parameter setting is to determine the parameters as control factors and their levels to implement the adopted experimental design. Parameters of the proposed GA includes number of iterations (It) initial population size (pop), crossover population size ($popc$) and mutation population size ($popm$). It is set according to a stoppage criteria which terminates the search over the initial population after searching 15 iterations without improvement, which is obtained based on trial and error. However, $popc$ and $popm$ are dependent to pop and calculated as follows:

$$popc = Cr \times pop \qquad\qquad popm = Mr \times pop \tag{37}$$

GA parameters setting is carried out on different levels of pop, crossover rate (Cr) and mutation rate (Mr) as shown in Table 1.

Table 1

Different levels of GA and MA parameters

Algorithm	Parameters	Level 1	Level 2	Level 3
GA	pop (Small-sized)	60	80	100
	pop (medium-sized)	100	120	140
	pop (large-sized)	140	160	180
	Cr (Small-sized)	0.5	0.6	0.7
	Cr (medium -sized)	0.6	0.7	0.8
	Cr (large -sized)	0.7	0.8	0.9
	Mr (Small-sized)	0.1	0.15	0.2
	Mr (medium -sized)	0.1	0.2	0.3
TS	Mr (large -sized)	0.1	0.2	0.3
	$TSpr$ (All sizes)	0.3	0.5	0.7
	Nf (All sizes)	0.5	1	1.5
	TLr (All sizes)	0.2	0.4	0.6

From the standard table of orthogonal arrays, L_9 (3^3) is chosen as illustrated in Table 2 In this Table, rows denote the level of parameters in each experimental scheme and columns indicate a specific level of a parameter which is changeable in each scheme.

Table 2

L_9 (3^3) adopted from the standard table of orthogonal arrays

	Algorithm	Parameters levels		
Experiment no.	GA	pop	Cr	Mr
	TS	$TSpr$	Nf	TLr
1		1	1	1
2		1	2	2
3		1	3	3
4		2	2	3
5		2	3	1
6		2	1	2
7		3	3	2
8		3	2	1
9		3	3	2

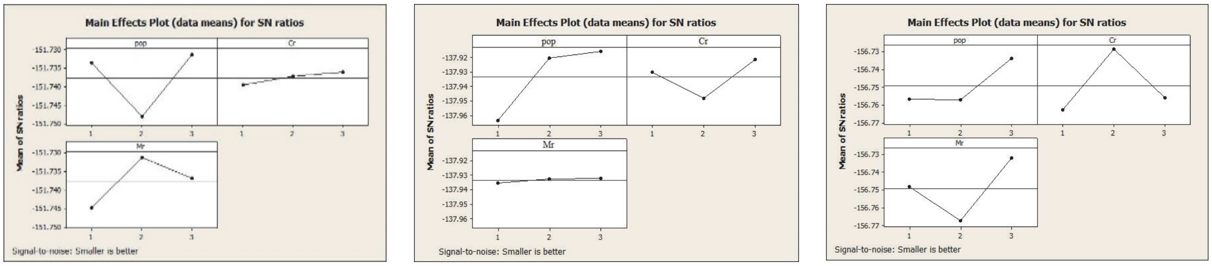
Results of the test problems for GA procedure is achieved as mentioned in Section 4.1 and Taguchi designs of GA parameters for small, medium and large-sized problems are implemented in Minitab14. Therefore, the optimal level of GA parameters for each problem size is shown in Table 3 as depicted in Fig. 8.

4.3. MA parameters setting

The parameters of the GA adjusted in Section 4.2 are directly used for MA. However, the parameters of TS approach as the local search engine for the proposed MA, are adjusted separately in this section. In this regard, the levels of TS parameters including K, α and 6are defined depending on $popc$, number of cells (i.e. the sum of part manufacturing and sub-assembly cells) and α respectively and calculated as follows:

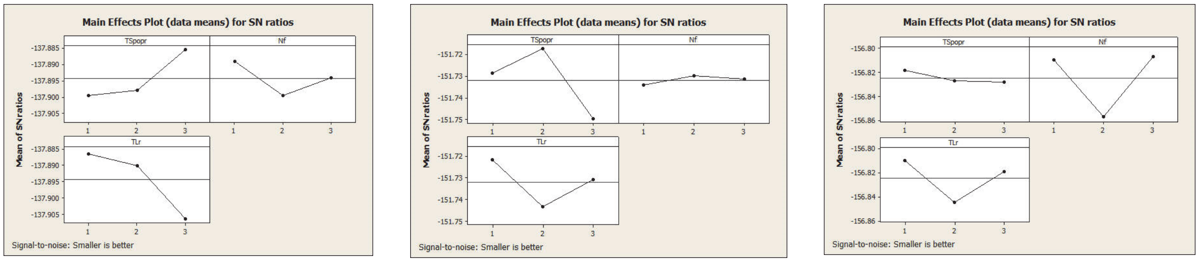
$$K = TSpr \times popc \qquad \alpha = Nf \times (C + Sa) \qquad \theta = TLr \times \alpha \tag{38}$$

TS parameters setting is carried out on different levels of $TSpr$ (TS population rate), Nf (neighborhood iterations factor) and TLr (TL rate) shown in Table 1 for each problem size. L_9 (3^3) is chosen from the standard orthogonal arrays and the implementation of Taguchi designs for TS parameters is just like that of GA. Therefore, the optimal level of TS parameters for each problem size is shown in Table 3 as depicted in Fig. 9.



8 (a) Small-sized problems 8 (b) Medium-sized problems 8 (c) Large-sized problems

Fig. 8. S/N ratio diagrams for different sizes of GA



9 (a) Small-sized problems 9 (b) Small-sized problems 9 (c) Large-sized problems

Fig. 9. S/N ratio diagrams for different sizes of TS

Table 3
Optimal level of GA and TS parameters

Size	Parameters level					
	GA parameters			TS parameters		
	pop	Cr	Mr	$TSpr$	Nf	TLr
Small	100	0.6	0.2	0.7	0.5	0.2
Medium	140	0.7	0.2	0.5	1	0.2
Large	180	0.8	0.3	0.3	0.5	0.2

5. Computational results

In this section the performance of the proposed GA is compared with the proposed MA through numerical examples. Table 4 includes the parameters of the model, which are randomly generated to perform the test problems. The problems are segmented into three size categories (i.e. small, medium and large-sized), as shown in Table 5, and the experiments are implemented on a computer with Intel® Core 5 CPU, 2.67 GHz speed and 4 GB RAM. Each algorithm is run five times for each problem instance and results consisting of minimum, maximum and average values of objective function are given in Table 6 The first set of experiments are carried out on five small-sized problems. Here, small-sized problems are referred to problems which are solvable optimally in GAMS software using CPLEX solver in reasonable CPU time. The optimal solutions of these problems are compared to the solutions obtained from GA and MA procedure, which are coded in MATLAB software, to evaluate the efficiency of the proposed algorithm. According to Table 7 it is obvious that both GA and MA have got eligibility to reach near optimal solutions with the average gap of 0.28% and 0.19% respectively compared with the exact solutions obtained from GAMS software. Hence, the efficient performance of both algorithms to solve small-sized problems permit to generalize them for medium and large-sized problems which are not solvable optimally by GAMS software, CPLEX solver in reasonable time.

Table 4
Randomly generated data

Parameter	**Value range**	**Parameter**	**Value range**
D_o	[75,100]	U_{ps}^{part} , U_{so}^{sa}	[0,3]
$F_{m_s n_s}$, F_{sm_f} , $F_{om_f n_f}$	[0,1]	$B_{om_f}^{\text{int}ra}$	[5,8]
$B_p^{\text{int}er}$	[40,60]	$B_p^{\text{int}ra}$	[15,20]
B_s	[20,30]	$B_{m_s}^{\text{int}ra}$	[10,15]
d_{kl}^m	[3,6]	d_{ef}^c dc_{eq}^{cs} , dac_{eq}^{cs}	[10,20]
$d_{t_s r_s}^s$	[3;10]	$d_{t_f r_f}^f$	[6;18]
$C_p^{\text{int}er}$	[600,800]	$C_p^{\text{int}ra}$	[200,250]
C_s	[500,600]	$C_{m_s}^{\text{int}ra}$	[800,1000]
$C_{om_f}^{\text{int}ra}$	[1000,1200]		

Table 5
Data related to the problem sizes

Problem No.	$O \times P \times M \times C \times S$	$M_1 \times ... \times M_c$	$T_1 \times ... \times T_s$	$Tf_1 \times ... \times Tf_o$
1	3×5×6×2×3	3×3	3×4×4	5×4×4
2	3×5×8×2×3	4×4	5×5×4	5×4×6
3	4×8×8×2×3	4×4	5×5×4	5×4×4×6
4	4×8×6×2×4	3×3	3×4×4×3	5×4×4×5
5	3×6×10×3×3	3×4×3	5×5×4	5×4×6
6	4×8×10×3×4	3×4×3	4×5×5×4	5×4×4×6
7	4×10×12×3×3	3×4×5	5×5×4	5×4×6
8	6×13×12×3×3	3×4×5	5×5×4	5×4×4×6×6×5
9	5×10×10×3×4	3×4×5	4×5×5×4	5×4×4×6×6
10	6×11×10×3×5	3×4×3	4×5×5×4×4	5×4×4×6×6×5
11	6×15×16×4×5	3×4×5×4	6×8×7×8×6	8×6×8×6×10×10
12	7×18×16×4×5	3×4×5×4	6×8×7×8×6	8×6×8×6×10×10×9
13	7×20×16×4×6	3×4×5×4	6×8×7×8×6×7	8×6×8×6×10×10×9
14	8×20×19×5×6	3×4×5×4×3	6×8×7×8×6×7	10×8×12×12×10×8×10×10
15	8×25×19×5×8	3×4×5×4×3	6×8×7×8×8×7×6×7	10×8×12×12×10×8×10×10

Comparing the results of the medium and large-sized problems, the close performance of GA versus MA to obtain near optimum solution is inferred owing to the average gape of 0.33% and 0.07% for medium and large-sized problems respectively. However, the considerable difference between average CPU run time of these two algorithms shown in Table 7 persuade us to prefer GA as the appropriate approach to solve medium and large-sized problems.

To clarify the convergence process of proposed GA and MA, problem number 15 of large-sized instances is chosen and its corresponding convergence diagram is plotted in Figure 10. As mentioned before, the both algorithms stop after 15 iteration without improvement. GA is stopped after 84 iterations while MA is stopped after 104 iterations. Hence, the GA shows faster convergence than MA. However the objective function value obtained from MA is a bit better than GA.

Table 6
Output results of the proposed solution approaches

Problem No.	GAMS solution	GA solution			MA solution		
	Z_{GAMS}	Z_{GA}^{min}	Z_{GA}^{ave}	Z_{GA}^{max}	Z_{MA}^{min}	Z_{MA}^{ave}	Z_{MA}^{max}
1	7093094	7093094	7122682	7171041	7093094	7093094	7093094
2	7305690	7305690	7315356	7322037	7315949	7320866	7327413
3	11981099	11981099	11990745	11999881	11981099	11991335	12000416
4	13551990	13559497	13591189	13603810	13551990	13580543	13619620
5	7800935	7825156	7850731	7890167	7821038	7839766	7849463
Average	**9546562**		**9574141**			**9565121**	
6	–	18043825	18093762	18140753	18067583	18090278	18104105
7	–	12793599	12838525	12869023	12790638	12796317	12804836
8	–	17969281	17969281	18018268	17924278	17927878	17939890
9	–	23816647	23848895	23873380	23726428	23801247	23823254
10	–	38474587	38600193	38911710	38473759	38497534	38529650
Average			**22270131**			**22196537**	
11	–	53123277	53271391	53408832	53101533	53403314	53663237
12	–	65345701	65993938	66430057	65622935	66041529	66447125
13	–	67801375	68622218	69837917	68307354	68635033	68899086
14	–	77808191	78364541	79448017	77795777	78441686	78986946
15	–	121532483	122484890	123384650	120730342	121949804	122514531
Average			**77747396**			**77694273**	

Table 7
Comparison of different solution approaches

Problem No.	Average CPU run time			Average gap %		
	GAMS	GA	MA	GA Vs. GAMS	MA Vs. GAMS	GA Vs. MA
1	00:00:37	00:00:16	00:00:29	0.41	0	0.41
2	00:02:34	00:00:23	00:02:04	0.12	0.19	-0.08
3	00:02:43	00:00:39	00:02:04	0.08	0.09	<-0.01
4	00:00:59	00:00:14	00:00:29	0.28	0.21	0.08
5	00:10:34	00:00:35	00:01:55	0.63	0.49	0.14
Average				**0.28**	**0.19**	**0.09**
6	–	00:01:00	00:06:54	–	–	0.02
7	–	00:00:52	00:04:06	–	–	0.32
8	–	00:00:51	00:08:33	–	–	0.23
9	–	00:01:02	00:05:20	–	–	0.2
10	–	00:01:03	00:06:44	–	–	0.27
Average						**0.33**
11	–	00:06:26	00:16:56	–	–	-0.07
12	–	00:06:54	00:19:51	–	–	-0.07
13	–	00:05:50	00:26:19	–	–	-0.02
14	–	00:09:15	00:57:39	–	–	-0.1
15	–	00:11:01	01:28:52	–	–	0.48
Average						**0.07**

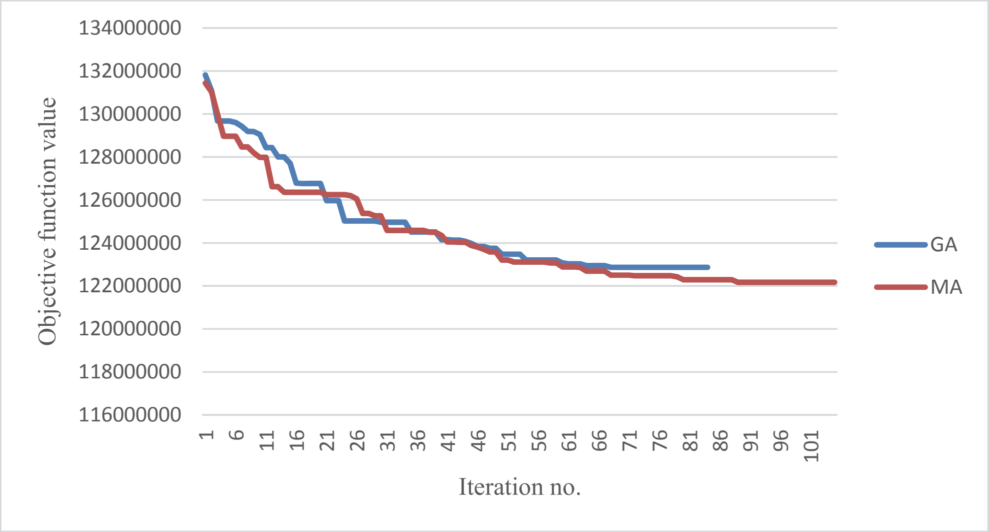


Fig. 10. Comparison of convergence rate between GA and MA

6. Sensitivity Analysis

In the proposed model, assembly U-shape layouts of sub-assembly cells and final assembly line are supposed to be arranged either in clockwise or anticlockwise direction. In this section, the performance of the proposed model is analyzed versus another model in which the line orientation of sub-assembly cells and final assembly line is permitted in only one direction and tasks are located along their relevant predetermined line. In this regard, variables Lo_s and Lo_f as well as Eqs. (11) and (15) of the proposed model are omitted to obtain the model without line orientation assumption. Then the obtained model is solved by proposed GAMS software, CPLEX solver after linearization for small-sized problems to see how sensitive the model is toward the loss of line orientation assumption in terms of cost reduction percent which is calculated in Eq. (39). ZW_{GAMS} is the optimal objective function value for small-sized problems obtained by GAMS software the model without line orientation assumption. Also, $\sigma\%$ is the cost reduction percent resulted from comparing the new model with the main proposed model for each small-sized problem instance calculated by the following equation.

$$\sigma\% = \frac{ZW_{GAMS} - Z_{GAMS}}{Z_{GAMS}} \times 100 \tag{39}$$

According to Table 8, 5.47% cost reduction is fulfilled by adopting the proposed model instead that of without line orientation assumption. This cost reduction was predictable since incorporating line orientation assumption results in more choices for assembly line arrangement and increase the flexibility of the model to attain more appropriate layout accordingly. Figure 11 represents the comparison of two models schematically as well.

Table 8
Comparison of the proposed model with that of without line orientation assumption

Problem no.	Objective function value		Cost reduction
	Proposed model	Model without line orientation	$\sigma\%$
1	7093094	7582977	6.9
2	7305690	7862101	7.61
3	11981099	12909111	7.75
4	13551990	14029937	3.53
5	7800935	7959482	2.03
Average	**9546562**	**10068722**	**5.47**

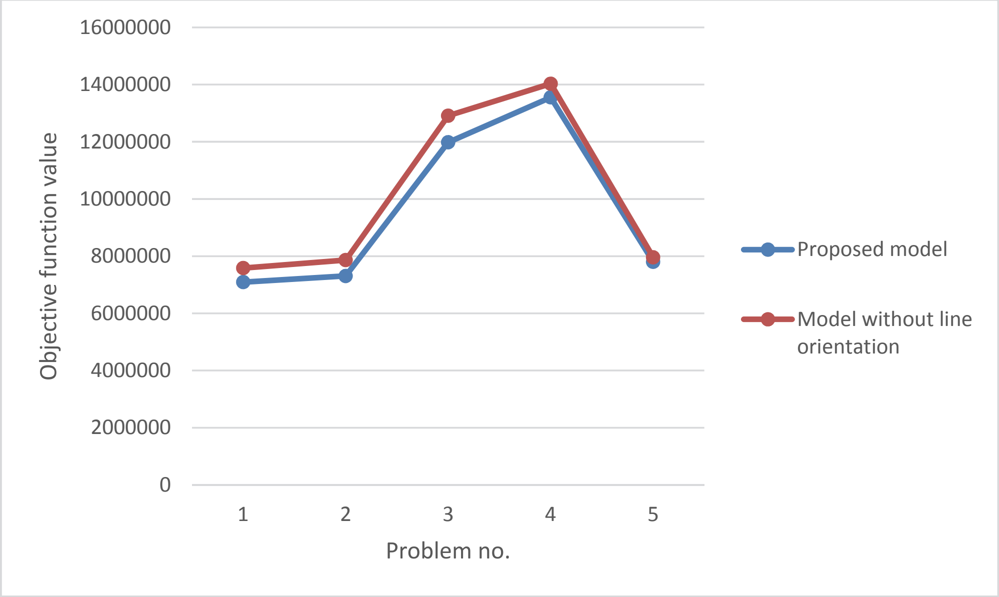


Fig. 11. Comparison between the proposed model and model without line orientation assumption

7. Conclusions

In this paper a comprehensive quadratic assignment problem is presented for the arrangement of facilities including machines, sub-assembly and final assembly tasks, part manufacturing and sub-assembly cells within the shop floor. Part manufacturing cells produce the necessary parts of the sub-assemblies produced in sub-assembly cells to feed final assembly line. The objective function of the proposed model is considered the minimization of material handling costs within the shop floor expressed in six terms including part manufacturing intra- and inter-cell costs, sub-assembly intra-cell cost, part manufacturing and sub-assembly inter-cell costs and material handling costs related to the flow between sub-assembly cells and final assembly line as well as the flow within final assembly line. Due to the complexity of the proposed model for real-sized problems, A GA and a MA constructed by the proposed GA and a TS algorithm as a local search engine is developed to solve the problem and their parameters are set based on Taguchi method. Compared to the optimal solutions obtained from GAMS software CPLEX solver for small-sized problems the efficiency of both algorithms to obtain near optimum solutions was proven owing to the average gap of 0.28% and 0.19% for GA and MA respectively. However, for medium and large-sized problems, the too large CPU run time of MA converging to the near optimum solution make us to forgo the negligible preference of MA in terms of objective function value (i.e. 0.33% for medium-sized problems and 0.07% for large-sized problems) and qualify GA for solving real-sized problems. Besides, a sensitivity analysis is carried out to evaluate the influence of line orientation assumption concerning U-shape layouts of sub-assembly cells and final assembly line. For this purpose, the proposed model was compared to a model lacking the mentioned assumption. Results show the capability of the proposed model versus the other model to reduce 5.47% of material handling costs on average.

Acknowledgement

The authors would like to thank the anonymous referees for constructive comments on earlier version of this paper.

References

Ariafar, S., & Ismail, N. (2009). An improved algorithm for layout design in cellular manufacturing systems. *Journal of Manufacturing Systems*, *28*(4), 132-139.

Arkat, J., Farahani, M. H., & Ahmadizar, F. (2012). Multi-objective genetic algorithm for cell formation problem considering cellular layout and operations scheduling. *International Journal of Computer Integrated Manufacturing*, *25*(7), 625-635.

Azzi, A., Battini, D., Faccio, M., & Persona, A. (2012). Mixed model assembly system with multiple secondary feeder lines: layout design and balancing procedure for ATO environment. *International Journal of Production Research*, *50*(18), 5132-5151.

Bagheri, M., & Bashiri, M. (2014). A new mathematical model towards the integration of cell formation with operator assignment and inter-cell layout problems in a dynamic environment. *Applied Mathematical Modelling*, *38*(4), 1237-1254.

Balakrishnan, J., & Cheng, C. H. (2007). Multi-period planning and uncertainty issues in cellular manufacturing: A review and future directions.*European Journal of Operational Research*, *177*(1), 281-309.

Battaïa, O., & Dolgui, A. (2013). A taxonomy of line balancing problems and their solutionapproaches. *International Journal of Production Economics*,*142*(2), 259-277.

Bazargan-Lari, M., Kaebernick, H., & Harraf, A. (2000). Cell formation and layout designs in a cellular manufacturing environment a case study.*International Journal of Production Research*, *38*(7), 1689-1709..

Boysen, N., Fliedner, M., & Scholl, A. (2008). Assembly line balancing: Which model to use when?. *International Journal of Production Economics*,*111*(2), 509-528.

Bozer, Y. A., & Wang, C. T. (2012). A graph-pair representation and MIP-model-based heuristic for the unequal-area facility layout problem. *European Journal of Operational Research*, *218*(2), 382-391.

Chan, W. M., Chan, C. Y., & Kwong, C. K. (2004). Development of the MAIN algorithm for a cellular manufacturing machine layout. *International journal of production research*, *42*(1), 51-65.

Chiang, C. P., & Lee, S. D. (2004). A genetic-based algorithm with the optimal partition approach for the cell formation in bi-directional linear flow layout. *International Journal of Computer Integrated Manufacturing*, *17*(4), 364-375.

Glover, F., & Laguna, M. (1997). Tabu SearchKluwer Academic Publishers.*Boston, MA*.

Glover, F., & Laguna, M. (2013). *Tabu Search∗* (pp. 3261-3362). Springer New York.

Goldberg, D. E. (1989). Genetic Algorithms in Search, Optimization, and Machine Learning, Addison-Wesley Professional. *Reading, Massachusetts, US*.

Golmohammadi, A., Bani-Asadi, H., Esmaeeli, H., Hadian, H., & Bagheri, F. (2016). Facility layout for cellular manufacturing system under dynamic conditions. *Decision Science Letters*, *5*(3), 407-416.

Holland, J. H. (1975). *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*. U Michigan Press.

Javadi, B., Jolai, F., Slomp, J., Rabbani, M., & Tavakkoli-Moghaddam, R. (2014). A hybrid electromagnetism-like algorithm for dynamic inter/intra-cell layout problem. *International Journal of Computer Integrated Manufacturing*,*27*(6), 501-518.

Jolai, F., Taghipour, M., & Javadi, B. (2011). A variable neighborhood binary particle swarm algorithm for cell layout problem. *The International Journal of Advanced Manufacturing Technology*, *55*(1-4), 327-339.

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. *Journal of Manufacturing Systems*, *33*(1), 218-232.

Krishnan, K. K., Mirzaei, S., Venkatasamy, V., & Pillai, V. M. (2012). A comprehensive approach to facility layout design and cell formation. *The International Journal of Advanced Manufacturing Technology*, *59*(5-8), 737-753.

Kulkarni, P. C., & Shanker, K. (2013, December). Genetic algorithm approach for solving intercellular layout problems in cellular manufacturing systems. In *2013 IEEE International Conference on Industrial Engineering and Engineering Management* (pp. 1587-1591). IEEE.

Li, J., Alden, J. M., & Rabaey, J. R. (2005). Approximating feeder line reliability statistics with partial data collection in assembly systems.*Computers & Industrial Engineering*, *48*(2), 181-203.

Li, L., Li, C., Ma, H., & Tang, Y. (2015). An optimization method for the remanufacturing dynamic facility layout problem with uncertainties. *Discrete Dynamics in Nature and Society*, *2015*.

Luss, H. (1989). Synchronized manufacturing at final assembly and feeder shops. *International journal of production research*, *27*(8), 1413-1426.

Mahdavi, I., Shirazi, B., & Paydar, M. M. (2008). A flow matrix-based heuristic algorithm for cell formation and layout design in cellular manufacturing system. *The International Journal of Advanced Manufacturing Technology, 39*(9-10), 943-953.

Mehdizadeh, E., Tavakkoli-Moghaddam, R., & Yazdani, M. (2015). A vibration damping optimization algorithm for a parallel machines scheduling problem with sequence-independent family setup times. *Applied Mathematical Modelling*, *39*(22), 6845-6859.

Mohammadi, M., & Forghani, K. (2014). A novel approach for considering layout problem in cellular manufacturing systems with alternative processing routings and subcontracting approach. *Applied Mathematical Modelling*,*38*(14), 3624-3640.

Naderi, B., Zandieh, M., Balagh, A. K. G., & Roshanaei, V. (2009). An improved simulated annealing for hybrid flowshops with sequence-dependent setup and transportation times to minimize total completion time and total tardiness. *Expert systems with Applications*, *36*(6), 9625-9633.

Nourmohammadi, A., & Zandieh, M. (2011). Assembly line balancing by a new multi-objective differential evolution algorithm based on TOPSIS.*International Journal of Production Research*, *49*(10), 2833-2855.

Papaioannou, G., & Wilson, J. M. (2010). The evolution of cell formation problem methodologies based on recent studies (1997–2008): Review and directions for future research. *European journal of operational research*,*206*(3), 509-521.

Pattanaik, L. N., & Sharma, B. P. (2009). Implementing lean manufacturing with cellular layout: a case study. *The International Journal of Advanced Manufacturing Technology*, *42*(7-8), 772-779.

Saif, U., Guan, Z., Wang, B., Mirza, J., & Huang, S. (2014). A survey on assembly lines and its types. *Frontiers of Mechanical Engineering*, *9*(2), 95-105.

Samarghandi, H., & Eshghi, K. (2010). An efficient tabu algorithm for the single row facility layout problem. *European Journal of Operational Research*,*205*(1), 98-105.

Sridhar, J., & Rajendran, C. (1993). Scheduling in a cellular manufacturing system: a simulated annealing approach. *The International Journal of Production Reseach*, *31*(12), 2927-2945.

Tavakkoli-Moghaddam, R., Javadian, N., Javadi, B., & Safaei, N. (2007). Design of a facility layout problem in cellular manufacturing systems with stochastic demands. *Applied Mathematics and Computation*, *184*(2), 721-728.

Tompkins, J. A., White, J. A., Bozer, Y. A., & Tanchoco, J. M. A. (2010).*Facilities planning*. John Wiley & Sons.

Tsai, J. T., Ho, W. H., Liu, T. K., & Chou, J. H. (2007). Improved immune algorithm for global numerical optimization and job-shop scheduling problems. *Applied Mathematics and Computation*, *194*(2), 406-424.

Wang, T. Y., Wu, K. B., & Liu, Y. W. (2001). A simulated annealing algorithm for facility layout problems under variable demand in cellular manufacturing systems. *Computers in industry*, *46*(2), 181-188.

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*(1), 156-167.