Computers & Industrial Engineering 91 (2016) 10–29

Contents lists available at ScienceDirect

Computers & Industrial Engineering

journal homepage: www.elsevier.com/locate/caie

A comprehensive mathematical model for dynamic cellular manufacturing system design and Linear Programming embedded hybrid solution techniques

Hüsamettin Bayram [a,*], Ramazan Şahin [b]

[a] Hitit University, Faculty of Engineering, Dept. of Industrial Engineering, Cevre Yolu Bulvari No: 8, 19030 Corum, Turkey
[b] Gazi University, Faculty of Engineering, Dept. of Industrial Engineering, Eti Mahallesi Yükseliş Sk. No: 5, 06570 Maltepe, Ankara, Turkey

ARTICLE INFO

Article history:
Received 10 June 2014
Received in revised form 30 January 2015
Accepted 23 October 2015
Available online 11 November 2015

Keywords:
Dynamic cellular manufacturing system design
Simulated Annealing
Genetic Algorithm
Linear Programming
Group layout
Cell formation

ABSTRACT

Considering the ever changing market conditions, it is essential to design responsive and flexible manufacturing systems. This study addresses the multi-period Dynamic Cellular Manufacturing System (DCMS) design problem and introduces a new mathematical model. The objective function of the mathematical model considers inter-cell and intra-cell material handling, machine purchasing, layout reconfiguration, variable and constant machine costs. Machine duplication, machine capacities, operation sequences, alternative processing routes of the products, varying demands of products and lot splitting are among the most important issues addressed by the mathematical model. It makes decisions on many system related issues, including cell formation, inter- and intra-cell layout, product routing and product flow between machines. Due to the complexity of the problem, we suggest two heuristic solution approaches that combine Simulated Annealing (SA) with Linear Programming and Genetic Algorithm (GA) with Linear Programming. The developed approaches were tested using a data set from the literature. In addition, randomly generated test problems were also used to investigate the performance of the hybrid heuristic approaches. A problem specific lower bound mathematical model was also proposed to observe the solution quality of the developed approaches. The suggested approaches outperformed the previous study in terms of both computational time and the solution quality by reducing the overall system cost.

1. Introduction

Nowadays manufacturing systems are expected to deliver large variety of products in smaller lot sizes with competitive prices. Cellular Manufacturing (CM) is among modern manufacturing philosophies that meets these requirements. In a CM System (CMS), products that are similar in their processing requirements are grouped into part families. The machines that process a family of products are grouped together to attain potential benefits of the CMS. Benefits of the CMS include reducing setup times, reduction in material flow and work-in-process inventory, easier and better system management, improved overall system efficiency and product quality (Baykasoğlu, 2004; Urban, Chiang, & Russell, 2000). However, processing all of the processing requirements of a product family in a single machine cell is an ideal. Under real manufacturing conditions it is either uneconomical or practical to design mutually independent cells. Therefore, exceptional elements is common in CMS manufacturing environments (Wang & Sarker, 2002). An exceptional element is a product that is needed to be produced in more than one cell and it causes inter-cell transfer of materials. In some cases, elimination of exceptional elements is possible, but requires additional machine investment.

When designing a Cellular Manufacturing System (CMS) many decisions must be taken into account. Some of these decisions are as follows: (1) cell formation (CF) through grouping of machines into cells, (2) layout of machines within cells (intra-cell layout) and (3) layout of cells (inter-cell layout) (Wemmerlöv & Hyer, 1986). As stated in Alfa, Chen, and Heragu (1992) these decisions are interrelated and addressing them simultaneously is important for a successful CMS design. However, each of these decisions is proven to be complex (Mak, Wong, & Wang, 2000; Sahni & Gonzalez, 1976), thus addressing of these decisions simultaneously is a difficult task. Therefore, most of the studies either consider

* Corresponding author. Tel.: +90 5056265041.
E-mail addresses: husamettinbayram@hitit.edu.tr (H. Bayram), rsahin@gazi.edu.tr (R. Şahin).

http://dx.doi.org/10.1016/j.cie.2015.10.014
0360-8352/

some of these decisions or they handle all, but in a sequential fashion.

Short product life cycles and rapid changes in product demands require reconfiguration of CMS from time to time. Therefore, CMS design must be carried out taking the changes in the demand into account. In CMS and facility layout literature, in order to handle the changes in demand of products, three main approaches are proposed. In the first approach, resources are rearranged by considering only processing requirements of the imminent future. This approach is called agile strategy and requires availability of agile resources (e.g. machine tools that can be easily relocated). The second approach is called robust strategy. It is based on designing a single layout that would be effective over the planning horizon. Although these approaches are easier and simplifies the multi-period design problem, both of these approaches are able to provide good layout solutions in extreme conditions. For example, agile strategy is useful only if the rearrangement costs are negligible. On the other hand, robust strategy is capable of finding layout solutions if the rearrangement costs are prohibitively high. In these strategies, rearrangement costs are either neglected or not even incurred by not changing the layout. Introduced by Rheault, Drolet, and Abdulnour (1995), Dynamic Cellular Manufacturing System (DCMS) design basically considers changes in product mix and demand. In addition to the single period CMS design decisions, DCMS design involves multi-period cell reconfiguration decisions. The reconfiguration of a manufacturing system involves some costly activities such as machine relocation, installation and uninstallation costs, lost production time and relearning costs (Balakrishnan & Cheng, 2007). In a DCMS design, the length of the time periods should be determined carefully and it must be reasonable to make a trade-off between cumulative increased flow costs of inefficient layout and rearrangement costs. If the time period is selected too short or too long, the problem becomes one of the extreme cases that were discussed above because, relative weight of the cumulative increased flow costs of inefficient layout over rearrangement costs changes significantly. Gupta and Seifoddini (1990) found out that one-third of USA companies rearrange their manufacturing facilities every two years. Moreover, Marsh, Meredith, and McCutcheon (1997) concluded that layout changes could occur within six months from the last rearrangement of a cell.

In this study, we focused on a comprehensive CMS design problem with the consideration of rearrangements in multi-period design horizon. We first present a comprehensive mathematical model that incorporates important DCMS design features including inter-cell layout, intra-cell layout, alternative process routes, duplicated machines, machine capacities, processing times, dynamic product demand, lot splitting, machine installation and uninstallation costs, material handling costs, processing costs, machine purchasing costs, and constant machine costs. We also propose two different Linear Programming (LP) embedded meta-heuristic approaches for solving this problem. The first one is the integration of LP and Simulated Annealing (SAeLP) and the second is the integration of LP and Genetic Algorithm (GAeLP). The efficiencies of the SAeLP and GAeLP are shown by comparing our results with those of a previous study (Kia et al., 2012) and a problem specific lower bound mathematical model. The results have shown that both SAeLP and GAeLP are powerful techniques in terms of both solution quality and computational time. The contribution of this study is manifold: (1) the mathematical model of Kia et al. (2012) is improved, (2) two LP embedded meta-heuristics are suggested and their efficiency is demonstrated, (3) a lower bound mathematical model that provides tight lower bound results for the test samples is provided. A brief review of DCMS design will be given in Section 2. Then, in Section 3 the mathematical model of the problem is introduced. The solution methodology is described in detail in Section 4. In order to illustrate the SAeLP and the GAeLP, the solution steps of a small sample problem are given in Section 5. Finally, comparative computational results and the conclusions are included in the Sections 6 and 7, respectively.

2. Literature

Both CMS and DCMS design literatures are very rich. In this section, only some of the remarkable studies are discussed. Harhalakis, Ioannou, Minis, and Nagi (1994) took product demand changes into account, but they tried to obtain a single design that is effective across the periods in the planning horizon. Rheault et al. (1995) introduced the concept of DCMS design with reconfiguration capability. Their study involves production scheduling, routing and loading of parts. The trade-off between material handling costs (MHC) and reconfiguration costs are presented by using an integer programming model. Wilhelm, Chiou, and Chang (1998) proposed a multi-period cell formation model aimed at minimizing reconfiguration, additional machine purchasing and inter-cell material handling costs. In order to handle the variation in product mix, Askin, Selim, and Vakharia (1997) suggested a four-stage technique. Initially, operations were assigned to machine types, and then operations are assigned to specific machines. In the following stages the manufacturing cells were determined and the design was improved. Chen (1998) developed a mixed integer mathematical programming model for DCMS design with reconfiguration issue. The objective function minimizes inter-cell material handling, reconfiguration and machine costs. Wicks and Reasor (1999) proposed another model with reconfiguration, in which they pursued minimization of the reconfiguration and constant machine costs.

Operation sequence of the products and machine replication were the other aspects considered during DCMS design. Chen and Cao (2004) developed a method to concurrently design CMS and to plan manufacturing activities. Their Tabu Search based method minimizes the sum of inter-cell material handling, inventory holding, cell formation costs. Although they took the machine capacities and machine duplication into account, they assumed that there was a single process plan for each product type. Therefore, processing costs were not included in the model. In their another study (Cao & Chen, 2005), they defined product demand in a probabilistic scenarios and they used a two stage Tabu Search based algorithm to minimize machine costs and inter-cell material costs. Similar to their previous study, they did not add processing costs to the objective function. In their study, Tavakkoli-Moghaddam, Aryanezhad, Safaei, and Azaron (2005a, 2005b) proposed a comprehensive mathematical model assuming alternative process routings, operation sequence, machine capacities and machine duplication. In the objective function, inter-cell material handling, variable and constant machine costs and reconfiguration costs were included. They solved this model using Simulated Annealing, Tabu Search and Genetic Algorithms. In another study, Tavakkoli-Moghaddam et al., 2005a, 2005b solved a similar model using Memetic Algorithms. Defersha and Chen (2008a) focused on cell formation under dynamic manufacturing conditions. In addition to the model properties of Tavakkoli-Moghaddam et al. (2005a, 2005b), they considered workload balancing and machine separation constraints as well. Their objective function comprises the sum of machine maintenance and overhead costs, machine procurement cost, inter-cell material handling cost, machining and setup costs, tool consumption cost, and system reconfiguration cost. Then, they solved this model by using a parallelized Genetic Algorithm. Nsakanda, Diaby, and Price (2006) included the option of outsourcing in their model while. Aryanezhad, Deljoo, and Mirzapour Al-e-hashem (2009) integrated worker assignment decisions into the dynamic cell formation decisions.

Egilmez, Süer, and Huang (2012), considered uncertainty of product demand and process durations in CMS system design and determined cell and product family formations while minimizing the risk. Mahdavi, Aalaei, Paydar, and Solimanpur (2012), solved the cell formation problem by employing three dimensional machine part worker assignment incidence matrix. In their study, they proposed a mathematical model in order to minimize exceptional elements and voids in a CMS. They mainly focused on cell formation problem. Chang, Wu, and Wu (2013) developed a methodology that makes cell formation, inter-cell layout and intra-cell layout decisions simultaneously. They implemented Tabu-Search meta-heuristic for the solving of the comprehensive problem. Operation sequences, alternative process routings are taken into account. Mahdavi, Teymourian, Baher, and Kayvanfar (2013) presented a mathematical model that addresses cell formation and cell layout decisions simultaneously. Their model minimizes costs of inter-cell and intra-cell movements and the number of exceptional elements. Zeidi, Javadian, Tavakkoli-Moghaddam, and Jolai (2013) developed a multi-objective approach based on Genetic Algorithm and neural networks for incremental CMS design. Mohammadi and Forghani (2014) presented an integrated approach for designing CMS. They considered various design elements such as part demands, alternative process routings, operation sequences, process times, capacity and dimensions of machines. Their model allows subcontracting of parts.

As a recent comprehensive study, we focused on the study of Kia et al. (2012). In this study, authors integrated cell layout and cell formation decisions in a dynamic environment. They reckoned with various CMS design attributes including inter- and intra-cell layout, cell formation, machine duplicates, machine capacities, constant and variable machine costs, lot splitting, operation sequence and alternative process routings, layout and cell reconfiguration, machine purchasing.

As discussed in the previous paragraphs, the DCMS design literature is rich and has been flourishing continuously. Along with the increase in the computational capacity of new generation computers, mathematical models include more and more design attributes. Moreover, some new approaches are developed to solve this challenging problem. One of these approaches is the integration of LP into metaheuristics. Synergies created through integration of LP and metaheuristics improve algorithms in terms of running time and solution quality. Although LP integrated approaches is not as prominent as pure metaheuristic approaches in DCMS design literature, the capabilities of this integration is promising, if it is implemented properly. In order to have a detailed review on the integration of LP into metaheuristics, researchers can refer to the literature survey of Raidl and Puchinger (2008).

There are very few implementations of LP integration in CMS design literature. In one of these studies, Defersha and Chen (2008b) proposed an LP embedded Genetic Algorithm to solve the integrated cell formation and lot sizing problem considering product quality. The algorithm searches over the integer variables of the problem. For each visited integer solution, the corresponding values of the continuous variables are determined by solving an LP sub-problem. In their recent study, Rezazadeh, Mahini, and Zarei (2011), focused on virtual cell formation problem considering operation sequence, alternative routings, machine capacities, lot splitting, maximum cell size and workload balancing. In order to solve the problem, they developed an LP embedded particle swarm optimization algorithm with a Simulated Annealing-based local search engine. To the best of our knowledge, these are the only studies that solve a CMS design problem by employing an LP embedded meta-heuristic approach. These studies only focus on only cell formation issue in cellular manufacturing, but our study focuses on DCMS design with numerous design features including layout issues. Considering the problem's comprehensiveness and hardness, the proposed methodology is suitable for solving such a complex problem. However, LP integration to a meta-heuristic have never been implemented to such a comprehensive DCMS design problem in the CMS design literature.

3. Problem formulation and description of the mathematical model

We propose a mathematical model that considers numerous aspects of DCMS design. The model considers many of the possible aspects of DCMS design. Theses aspects include: inter- and intra-cell layout design, alternative processing routes, operation sequence, lot splitting, machine installation and uninstallation, location based inter-cell and intra-cell material handling costs, variable and constant machine costs as well as machine purchasing decisions and costs. Throughout the development of the mathematical model emphasis was put upon the work of Kia et al. (2012).

3.1. Mathematical model

<u>Model assumptions</u>

The assumptions of the non-linear mixed-integer programming model are as follows:

1. Demand for each product type is varying in subsequent periods and the demand is known deterministically prior to the design. Demand must be satisfied in a given period. Demand for each product is expressed in number units.
2. There is an operation sequence for each product.
3. Each operation can be performed in different types of machines and possibly with different processing times.
4. Machines are assumed to be multi-purpose ones. Namely, they are capable of performing different operations of different products.
5. There is a constant cost for a machine per period. This cost is incurred as an overhead cost and does not depend on the utilization of the machine. This cost is expressed in monetary terms (e.g. $, €, etc.).
6. Each machine has a limited capacity and several duplicates of the machines are allowed. Machine capacities are expressed in time units (e.g. h, days).
7. The variable cost of machines is dependent on the assigned workload. Variable cost of machines is expressed in monetary terms in a unit time (e.g. $/h).
8. If a machine is purchased in a period, it must stay on the shop floor in the following periods. Removal of machines from the shop floor is not assumed.
9. There is no physical partitioning between cells and a location can be assigned to different cells in different periods.
10. Replacement cost of machines consists of installation and uninstallation costs. When a machine is moved from one location to another, both installation and uninstallation costs are incurred. Regardless of the purchase period, installation cost is incurred for every new machine. Both installation and uninstallation costs are expressed in terms of money per each relocation.
11. The cost of carrying items between two locations is proportional to the number of carried products. Both inter-cell and intra-cell material handling costs are linearly proportional to the distance between the locations of the machines. Distance between the locations are expressed in units of length (e.g. m, etc.).

12. All machines have the same dimension. Therefore any machine can be assigned to any location. However, only one machine can be assigned to a location.
13. The maximum number of cells and the minimum and the maximum number of machines in cells are assumed to be known in advance. Cell capacities can be determined by system designers considering some design issues. For example, too many machines may cause complicated controlling of the cell. On the other hand, very few machines in a cell causes increased inter-cellular material movements but, it reduces intra-cellular movements.
14. Positions and shapes of the cells are not predetermined.
15. Splitting of lots is allowed. Namely, an operation of a product can be split between two machines of same or different types, in a given period.

Indexing sets:

p: index for product types,
t: index for periods,
r: index for operations,
c, d: indices for cells,
k, l: indices for locations,
i, j: indices for machine types.

Parameters:

T: *Number of planning periods in the planning horizon.*
P: *Number of product types to be produced.*
N: *Maximum number of cells.*
R_p: *Number of operations for product type p.*
K: *Number of locations in the shop floor.*
M: *Number of machine types.*
E_p: *Inter-cellular material handling cost per product type p, per unit distance (e.g. \$/m).*
A_p: *Intra-cellular material handling cost per product type p, per unit distance (e.g. \$/m).*
D_{pt}: *Demand for product type p, in period t.*
λ_{kl}: *Distance between locations k and l (e.g. m).*
δ_i: *Installation cost of machine type i (e.g. \$).*
θ_i: *Uninstallation cost of machine type i (e.g. \$).*
β_i: *Overhead cost of machine type i in each period (e.g. \$).*
μ_i: *Unit time variable cost of machine type i (e.g. \$/h).*
γ_i: *Purchasing cost of machine type i (e.g. \$).*
U: *The maximum number of machines that can be assigned to a manufacturing cell.*
L: *The minimum number of machines that can be assigned to a manufacturing cell.*
π_{pri}: *Processing time of r^{th} operation of product type p, on machine type i.(e.g. h).*
C_i: *Capacity of machine type i in each period (e.g. h).*
α_{pri} : *1, if r^{th} operation of product p can be processed by machine type i, 0, otherwise.*

Decision variables:

x_{ckit}: 1, if a machine of type i is placed on location k and assigned to cell c in period t, and 0 otherwise.
y_{kit}: 1, if location k is either empty or is assigned to a machine type other than i in period $t-1$, and location k is assigned to machine type i in period t. and 0 otherwise.
y'_{kit}: 1, if a machine of type i is located on location k in period $t-1$, but it is either removed or replaced by a machine of different type in period t.
w_{prckit}: Number of products of type p, processed in operation r on machine type i which is assigned to location k cell c in period t.
$W_{prcdklijt}$: Number of products of type p, processed by operation r, on machine type i, located in location k which is assigned to cell c and moved to be processed by operation $r+1$, on machine type j, located in location l which is assigned to cell d, in period t.
Q_{it}: Total number of machines of type i added to the layout at the beginning of the period t.

Mathematical model:

Objective function

$$\begin{aligned}\min z =& \sum_{t=1}^{T}\sum_{p=1}^{P}\sum_{r=1}^{R_p-1}\sum_{c=1}^{N}\sum_{\substack{d=1\\ d\neq c}}^{N}\sum_{k=1}^{K}\sum_{l=1}^{K}\sum_{i=1}^{M}\sum_{j=1}^{M} W_{prcdklijt}\times\lambda_{kl}\times E_p \\ &+\sum_{t=1}^{T}\sum_{p=1}^{P}\sum_{r=1}^{R_p-1}\sum_{c=1}^{N}\sum_{k=1}^{K}\sum_{l=1}^{K}\sum_{i=1}^{M}\sum_{j=1}^{M} W_{prccklijt}\times\lambda_{kl}\times A_p \\ &+\sum_{t=2}^{T}\sum_{k=1}^{K}\sum_{i=1}^{M}(y_{kit}\times\delta_i)+\sum_{t=2}^{T}\sum_{k=1}^{K}\sum_{i=1}^{M}(y'_{kit}\times\theta_i) \\ &+\sum_{t=1}^{T}\sum_{c=1}^{N}\sum_{k=1}^{K}\sum_{i=1}^{M} x_{ckit}\times\beta_i \\ &+\sum_{t=1}^{T}\sum_{p=1}^{P}\sum_{r=1}^{R_p-1}\sum_{c=1}^{N}\sum_{k=1}^{K}\sum_{i=1}^{M} w_{prckit}\times\pi_{pri}\times\mu_i \\ &+\sum_{t=1}^{T}\sum_{i=1}^{M} Q_{it}\times\gamma_i+\sum_{c=1}^{N}\sum_{k=1}^{K}\sum_{i=1}^{M} x_{cki,1}\times\delta_i\end{aligned} \quad (1)$$

Subject to:

$$w_{prckit}\leqslant x_{ckit}\times\alpha_{pri}\times D_{pt} \qquad \forall \quad p,r,c,k,i,t \quad (2)$$

$$\sum_{c=1}^{N}\sum_{k=1}^{K}\sum_{i=1}^{M} w_{prckit}=D_{pt} \qquad \forall \quad p,r,t \quad (3)$$

$$\sum_{c=1}^{N}\sum_{i=1}^{M} x_{ckit}\leqslant 1 \qquad \forall \quad k,t \quad (4)$$

$$\sum_{k=1}^{K}\sum_{i=1}^{M} x_{ckit}\leqslant U \qquad \forall \quad c,t \quad (5)$$

$$\sum_{k=1}^{K}\sum_{i=1}^{M} x_{ckit}\geqslant L \qquad \forall \quad c,t \quad (6)$$

$$\sum_{p=1}^{P}\sum_{r=1}^{R_p}\sum_{c=1}^{N} w_{prckit}\times\pi_{pri}\leqslant C_i \qquad \forall \quad k,i,t \quad (7)$$

$$\sum_{c=1}^{N}\sum_{k=1}^{K}\sum_{i=1}^{M} W_{prcdklijt}=w_{p,r+1,dljt} \qquad \forall \quad p,d,l,j,t,r=1,\ldots,R_p-1 \quad (8)$$

$$\sum_{d=1}^{N}\sum_{l=1}^{K}\sum_{j=1}^{M} W_{prcdklijt}=w_{prckit} \qquad \forall \quad p,c,k,i,t,r=1,\ldots,R_p-1 \quad (9)$$

$$\sum_{c=1}^{N}\sum_{k=1}^{K} x_{cki,t+1}-\sum_{c=1}^{N}\sum_{k=1}^{K} x_{ckit}=Q_{i,t+1} \qquad \forall \quad i,t=1,\ldots,T-1 \quad (10)$$

$$\sum_{c=1}^{N}\sum_{k=1}^{K} x_{cki,1}=Q_{i,1} \qquad \forall \quad i \quad (11)$$

$$\left(1-\sum_{c=1}^{N}x_{ckit}\right)\times\sum_{c=1}^{N}x_{cki,t+1}=y_{ki,t+1}\qquad\forall\quad k,i,t=1,\ldots,T-1\tag{12}$$

$$\sum_{c=1}^{N}x_{ckit}\times\left(1-\sum_{c=1}^{N}x_{cki,t+1}\right)=y'_{ki,t+1}\qquad\forall\quad k,i,t=1,\ldots,T-1\tag{13}$$

$$x_{ckit},y_{kit},y'_{kit}\in\{0,1\}$$

$$W_{prcdklijt},w_{prckit},Q_{it}\geqslant 0 \textit{ and integer}.$$

The objective function (1) to be minimized consists of eight terms The first term represents inter-cellular material handling cost and it is incurred only if consecutive operations of a product is performed in different cells ($d \neq c$). The second term is for the intra-cellular material handling cost and it is incurred when a product's consecutive operations are processed in two different locations of a cell. If two consecutive operations are processed on the same location, neither intra-cellular nor inter-cellular material handling cost is incurred, because there is not any material flow occurring between locations.

The third term of the objective function is the installation cost of the machines. This cost is incurred if a new machine is purchased or an existing machine is relocated. In a similar fashion, the forth term denotes the uninstallation costs and it is incurred only when a machine is uninstalled for relocation, since total uninstallation and removal of a machine from the layout is not allowed in the model.

As discussed in the assumptions, for any machine present in the layout, a constant overhead cost is incurred, in every period. The fifth term of the objective is for this cost. On the other hand, the sixth term is to consider the operating cost in all machines. The seventh term is the purchasing cost of machines. The installation costs of machines at the beginning of the first period are represented by the last term in the objective.

The terms (2)–(13) are the constraints of the mathematical model. The term (2) ensures that an operation of a given product type can only be processed in a given location if a machine, which is capable of this operation, is assigned to the location. In addition, this inequality guarantees that the total number of processed parts cannot exceed the demand for the product. The equality (3) denotes that the total number of product operations processed anywhere in the factory floor must be equal to the total product demand. The term (4) guarantees that at most one machine can be assigned to a location and a location can be belong to only one cell. Terms (5) and (6) impose the lower and upper limits for the cell size. The term (7) guarantees that the total processing time of the operations routed to a machine cannot exceed its capacity, which is defined in terms of time.

Eqs. (8) and (9) are the flow conservation constraints. Briefly, the term (8) ensures that the total number of incoming products (p) from all other locations (k) to a location (l) for its next operation ($r+1$) is equal to the number of products (p) which receive their next operation ($r+1$) in the given location (l). In a similar fashion, the term (9) shows that total the number of moving products (p) from a given (k) to any location (l) for its next operation ($r+1$) is equal to the number of products (p) which receive their current operation (r) in the given location (k).

Eq. (10) is related to the total number of added machines, starting from the beginning of the period 2. On the other hand, the term (11) calculates the number of available machines at the beginning of the planning horizon. The constraints (12) and (13) impose the relationship between x and y decision variables. The former constraint is related to installation and the latter constraint is related to uninstallation.

3.2. Model properties

In this section, some important properties of the mathematical model are discussed.

3.2.1. Sequence of operations

In real manufacturing systems, operation sequences are inevitable. Products undergo processing operations in a predefined order. Prior to the design of a manufacturing system, sequence of operations is readily available from the route sheet of the parts. In some earlier CMS design studies (especially in studies that are focused on cell formation) part machine incidence matrices were used as the input of the problem. Thus, operation sequences were overlooked. Operation sequence of parts is discussed among the major practical issues and the studies that does not take operation sequence into account are criticized (Papaioannou & Wilson, 2010; Onwubolu & Mutingi, 2001). Recently, many researchers have emphasized the importance of operation sequences in CMS design because, it does not only show processing requirements of the parts, but also the flow patterns of the parts. (Sarker & Xu, 2000). It is also an important contributor for the accurate calculation of the material movements in the layout (Cheng, Goh, & Lee, 1996; Harhalakis, Nagi, & Proth, 1990).

3.2.2. Alternative process routes

Consideration of alternative process routes in a design approach enlarges the solution space and increases the problem complexity. Although it is a complicating factor for the problem to be solved, consideration of alternative process routes helps system designer to obtain better CMS designs. If alternative process routes are not considered, a single process route is assumed by neglecting other machines that are capable of processing required operations of the products. However, if alternative process plans are taken into account, throughput rate of the system can be increased and in-process inventory can be reduced. In addition, consideration of alternative process routes helps CMS designer to minimize the total volume of exceptional elements. Logendran, Ramakrjshna, and Sriskandarajah (1994), Adil, Rajamani, and Strong (1996), Papaioannou and Wilson (2010), thus, better cell designs can be obtained.

3.2.3. Inter-cell and intra-cell material handling

In conventional CMS design approaches, machine cells are formed considering the processing requirements of the products and product families. Ideally, formation of clear-cut machine cells and product families is pursued but, this is generally impractical, if not impossible (Urban et al., 2000). Formation of independent machine cells requires substantial amount of machine investment and eventually reduces utilization. Therefore, transfer of parts between machine cells, namely inter-cell material handling is mostly preferred over machine investment. However, inter-cell movements decrease the efficiency of CM by complicating control and increasing material handling requirements and flow time (Aryanezhad et al., 2009). Inter-cell movements reduce the total possible benefit that could be attained through implementation of CMS philosophy. Inter-cell material transfers are usually done in larger lots, thus interferes the unity among the operations of a cell. If these movements increase too much and if the CMS is not rearranged in a counteracting manner, control over the overall CMS system becomes quite harder and controlling of material flow becomes a difficult task than even that of a functional layout. Reduction of inter-cell movements and related costs have been the primary concern of the CMS design literature.

Along with the inter-cell flow of the materials, consideration of intra-cell material flow has a significant effect on the successful CMS design. If layout of machines within a cell is not properly handled, material flow may become complicated. However, if a good sequence of machines within cells is determined, both the intra-cellular movements are minimized and the general flow patterns within cells emerge. In this study, minimization of both inter-cell and intra-cell MHCs through making of inter-cell and intra-cell layout decisions. Both inter-cell and intra-cell MHCs are calculated based on the distance between the locations, as opposed to some studies that does not regard that issue. Thus, relative positions of the machines with respect to each others are properly determined.

3.2.4. Layout reconfiguration

In this study, product demands are assumed to be exactly known prior to the design stage (deterministic demand). Product demands are different in subsequent periods (dynamic demand). As a result of this, CMS can be reconfigured by considering the costs of machine relocation. In our study, we assumed that rearrangement of cells without machine relocation does not impose any cost.

3.2.5. Fractal layout and fractal cells

Suggested by Venkatadri, Rardin, and Montreuil (1997) and discussed by Montreuil (1999), fractal layout converts a functional layout into physically separated cell as in conventional CMSs. However, fractal cells are not created based on product families. It is based on "factory within factory" concept and involves duplication of processes in the layout. As a part of "factory within factory" concept, a fractal cell is capable of manufacturing most of the products and the total number of workstations for most of the processes are equally distributed across several fractal cells. In general, fractal cells have the ability to produce wider variety of parts compared to GT cells. Therefore, in this strategy, formation of identical cells is a common approach as opposed to conventional CMS. However, if a fractal cell is specialized on processing a group of products without sharing the total demand with other cells, it may become a conventional GT cell.

In this study, we inspire from the fractal layout concept and the fractal cells as in previous studies of Kia et al. (2012) and Khaksar-Haghani, Kia, Mahdavi, and Kazemi (2013). Our approach is so flexible that it is possible that all processing on a product can take place at one and only one cell as in a conventional CMS or some portion of the total demand of a product can be processed in one cell and the rest can take place in another cell. Namely, lots can be split among cells. In addition to that, parts are allowed to cross cell boundaries with resulting inter-cell movements. In our model, we allow model itself to determine a layout strategy in order to benefit from the advantages of both fractal cells and conventional CMS. If lots are split among different cells, it means that cells resemble fractal cells. However, if products types are distributed among different cells considering their processing requirements without splitting product lots, manufacturing environment resembles a conventional CMS.

3.2.6. Lot splitting

Lot splitting is dividing of large orders into smaller batches. It provides simultaneous processing of products in more than work center. In a conventional CMS under ideal conditions, a type of a part is produced in a single cell without transferring of parts between cells and splitting of demand among different cells. Therefore, splitting of lots among cells or different machines are not common in CMS design literature. There are very few studies that consider this aspect of the manufacturing (Ahkioon, Bulgak, & Bektas, 2009; Defersha & Chen, 2006; Khaksar-Haghani et al., 2013; Kia et al., 2012; Logendran & Ramakrishna, 1995; Saxena & Jain, 2011; Saxena & Jain, 2012) in CMS design. In this study, we assume that the demand on a product can be split among different cells, as in the case of fractal cells. Consideration of lot-splitting in design stage improves machine utilization, reduces inter-cell movements, operation costs and required machine investments (Defersha & Chen, 2006).

3.2.7. Machine capacities and machine duplication

One of the other realistic assumptions of our model is the machine capacities. In a design approach where process durations of operations and alternative process routings are taken into account, machine capacities should also be considered. Each machine has a limited capacity, expressed in time units (e.g. h) during each time period. In this model, since we assume that machine capacities are limited, purchasing and locating multiple copies of machines in a single cell is allowed.

3.2.8. Number of cells and cell size

In this study, we allow the system designer to specify the number of cells and impose constraints on the maximum and the minimum number of machines that can be assigned to a cell. As stated by Heragu (1994) and Black (1983) presetting the number of cells (or maximum number of cells) might be desired in some cases. In addition, if the number of cells is not wanted to be determined in advance, by assigning a relatively higher value to number of cells and by not imposing a minimum cell size, the number of cells can be determined by solving the mathematical model. This, of course, requires the maximum number of machines in a cell to be defined in advance. Otherwise, the model will converge to a result with only a single cell with too many machines. This will reduce inter-cell material handling costs but will deteriorate the control and the operational efficiency of the cells. As it is easier to coordinate material flows in a smaller cell, not imposing a lower bound to cell size is a reasonable preference (Gupta & Tompkins, 1982; Nsakanda et al., 2006).

3.3. Linearization of the model

The terms (12) and (13) include multiplication of decision variables. This violates linearity of the model. For the linearization of the model, the terms (12) and (13) are replaced by the following terms.

$$0.5 + y_{ki,t+1} + \sum_{c=1}^{N} x_{ckit} - \sum_{c=1}^{N} x_{cki,t+1} \geqslant 0 \qquad \forall \quad k, i, t = 1, \ldots, T-1 \tag{14}$$

$$1.5 \times y_{ki,t+1} + \sum_{c=1}^{N} x_{ckit} - \sum_{c=1}^{N} x_{cki,t+1} - 1 \leqslant 0 \qquad \forall \quad k, i, t = 1, \ldots, T-1 \tag{15}$$

$$0.5 + y'_{ki,t+1} + \sum_{c=1}^{N} x_{cki,t+1} - \sum_{c=1}^{N} x_{cki,t} \geqslant 0 \qquad \forall \quad k, i, t = 1, \ldots, T-1 \tag{16}$$

$$1.5 \times y'_{ki,t+1} + \sum_{c=1}^{N} x_{cki,t+1} - \sum_{c=1}^{N} x_{cki,t} - 1 \leqslant 0 \qquad \forall \quad k, i, t = 1, \ldots, T-1 \tag{17}$$

3.4. Computational complexity

In order to reflect real manufacturing environment to a broader extent, the proposed model integrates many design features of cellular manufacturing design. Along with its dynamic reconfiguration properties, it comprises cell formation, inter- and intra-cell layout features. Furthermore, in the model, cell size is flexible and splitting of lots is allowed. Part routing with alternative routings, duplicated machines and operation sequence are also considered. All these attributes contribute to the complexity of the model.

Cell formation itself is a combinatorial complex problem. When the cell reconfiguration under dynamic conditions is considered, its complexity increases (Chen, 1998). Generally, Quadratic Assignment Problem (QAP) formulation is used to formulate inter-cell layout and intra-cell layout problems in CMS design. The QAP formulation is extensively used in facility layout literature and it has been proven to be an NP-Hard optimization problem (Sahni & Gonzalez, 1976). Determination of process routing among alternative process routings is also a problem of complexity class NP-Hard (Logendran et al., 1994). Therefore, the mathematical model of the problem is of a NP-Hard complexity class, since it integrates all of the problems given above and even with some other design features.

4. Solution approaches

Due to the complexity of the problem, two hybrid solution methods based on Simulated Annealing and Genetic Algorithm is developed. Both of the approaches utilize Linear Programming in a similar fashion. The new solution approaches are discussed in detail in the following sections.

4.1. The LP embedded Simulated Annealing approach

We handle three aspects of dynamic cellular manufacturing system design: the cell formation problem, inter-cell layout problem and intra-cell layout problem. As discussed in Section 3.4, the proposed model is an NP Hard problem and is not expected to be solved to optimality for large problems with the use of off-the-shelf optimization software. Hence, solving this problem by using exact methods is inefficient and time consuming. Therefore, here we propose a Linear Programming and Simulated Annealing based hybrid solution approach.

As described by Rutenbar (1989), components of SA algorithm are configurations, move set, cost function and cooling schedule. Configurations are the representations of possible solutions over which a good answer is searched. Move set is a set of allowable moves which are performed to move from a legal configuration to another. Cost function is used to measure the fitness of visited configurations by using a move set. Cooling schedule determines the starting temperature, when to reduce the temperature, how much the temperature should be lowered and the termination condition of the SA.

SA and many of other heuristic techniques are usually applied to combinatorial optimization problems where all the decision variables are discrete. The proposed model involves both integer variables and binary variables. Although, binary variables can easily be handled by SA, integer variables are hard to be managed in SA.

Compared to integer programming or binary programming, LP is an efficient method in terms of solution time. However, LP is not suitable for solving problems with integer variables. The divisibility assumption of LP ensures that every decision variable can take fractional values (continuous variable). Although the proposed mathematical model does not involve any continuous variables, in some cases, rounding optimal solution's variables to an integer value may yield reasonable solutions. Therefore, relaxing problem's integer variables to continuous variables (without relaxing binary variables), solving this part to optimality by LP and rounding off optimal solution to integers seem to be practical and efficient both in terms of solution quality and solution time (Winston & Goldberg, 2003).

Our problem consists of six types of decision variables ($x_{ckit}, y_{kit}, y'_{kit}, W_{prcdklijt}, w_{prckit}, Q_{it}$), four of which are directly related to the layout of machines and determination of cells. These are $x_{ckit}, y_{kit}, y'_{kit}, Q_{it}$. Once the layout of machines and cells formations is determined for every period, the values of these variables are known. Then, the only problem that remains to be solved is the flow of materials between locations. SA and the LP share the work of optimization of these variables considering their capabilities.

In the hybrid approach (Fig. 1), LP is embedded into Simulated Annealing. Using SAeLP's solution representation ($[DL]$), layout of machines, formation of cells and layout of cells are determined. Then, LP is used to determine the optimal material flow between locations considering the capacities, capabilities and processing cost of machines in these locations and the cost of material handling between locations. LP determines the optimal material flow and part routing by solving the sub-problem, which is given in Eqs. (23)–(30). Afterwards, SAeLP moves to a neighbor solution (a new machine layout and cell formation), then by employing LP, new material flow and part routing are determined for the neighbor solution. These steps are repeated until SAeLP reach the minimum temperature.

Referring to the main mathematical model (1)–(17), if the values of the binary cell-location-machine assignment variables (x_{ckit}) and the other dependent variables ($y_{kit}, y'_{kit}, Q_{it}$) are determined, some of the indices and some of the terms in the main model are no longer needed. For example, if machine-cell-location assignments are known for every period, machine purchasing costs, machine installation and uninstallation costs and machine constant costs can be calculated. In addition, capabilities of machines can be expressed in terms of the capabilities of the locations as the machines assigned to each location are known. Since the locations in the same cell are known, the type of material handling (inter- or intra-cell) between the locations can be determined. The sub-problem is solved by using LP for every accepted solution in SA during the search in discrete solution space. The sub-problem is described in Section 4.2.

4.1.1. Solution representation scheme and neighbor generation

In the meta-heuristic part of this approach, machine-location assignments and cell formations are determined by SA. SA uses a three dimensional matrix, $[DL]_{3\times(U\times N)\times T}$ for the representation of each machine-cell-location assignments in different periods. For the sake of simplicity in graphical representation, let us assume that $[DL]$ is split into T sub-matrices, each of represents a layout solution in a period ($[DL]^1, \ldots, [DL]^T$). An example solution representation and the corresponding layout of machines in subsequent periods are given in Fig. 2. In the example, there are three types of machines, eight locations, three cells and three periods. The maximum cell size is three and the minimum cell size is two for this example representation. Each two dimensional sub-matrix ($[DL]^n$) represents machine-cell-location assignment for a period. Rows of this matrix correspond to cell numbers, location numbers and machine type numbers, from the top to bottom row, respectively. In order to ensure minimum and maximum number of machines in each cell, the matrix is divided into two parts.

Elements of the part on the left side are only allowed to take non-zero values. This rule is checked whenever an initial solution or a neighbor solution is generated. In the top row of the left section, each cell number is repeated only L (minimum cell size) times. Thus, the minimum cell size is guaranteed for every cell, while ensuring that all elements in these columns are non-zero. However, the elements of the right section can be zero, except

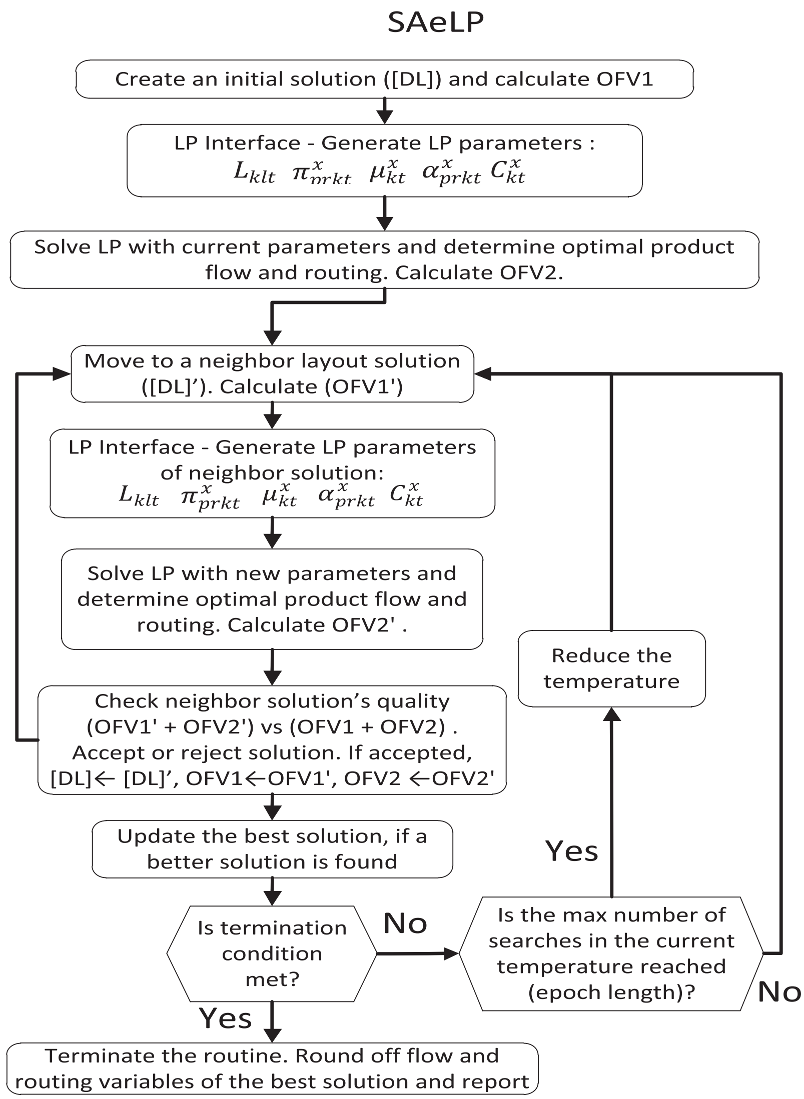


Fig. 1. Summary of the algorithmic steps of SAeLP.

the elements of the top row. Hence, the cell sizes cannot exceed the maximum number of allowed machines (U) in a cell. Each column without zero elements represents a machine-cell-location assignment. For example, the fifth column of the $[DL]^1$ sub-matrix refers to the assignment of a machine type one to location seven, which is assigned to cell number three. As it might be inferred from the figure, all elements of the first section of this matrix correspond to different assignments. For instance, the seventh column of the $[DL]^1$ sub-matrix does not correspond to any solution, due to the zero value in its second row. Similarly, the eighth column of the same matrix does not refer to an assignment, because of the zero value in its third row.

The SAeLP approach moves from a solution to its neighbor by employing a number of different neighborhood mechanisms. Each neighborhood mechanism is employed based on a predefined probability. These mechanisms are briefly described below:

Swap locations: Select a period t, randomly. Randomly select two positions of middle row of $[DL]^t$ matrix. If elements in these positions are non-zero, then swap them directly. If at least one of them is zero, then swap them only if they belong to the second section of the matrix.

Swap machines: Select a period t, randomly. Randomly select two positions of bottom row of $[DL]^t$ matrix. If elements in these positions are non-zero, then swap them directly. If at least one of them is zero, then swap them only if they belong to second section of the matrix.

Swap cells: Select a period t, randomly. Randomly select two columns of $[DL]^t$ matrix. If all elements in these columns are non-zero, then swap columns, except the top row. If some elements of these columns are zero, swap them if both of them belong to the second section of the matrix.

Add machine: Select a random period t. Select a random position in the bottom row of the $[DL]^t$ matrix. Assign a random machine type number to that position. Add that machine number to the representations of the following periods ($[DL]^t, \ldots, [DL]^T$) too.

Remove machine: Select a random period t. Randomly select a non-zero value in the bottom row of the $[DL]^t$ matrix. Assign zero to that position. Remove that machine number from the representations of previous periods ($[DL]^1, \ldots, [DL]^T$) too.

Copy period: Select a random period t. Copy $[DL]^t$ matrix and paste that on representations of either all previous ($[DL]^1, \ldots, [DL]^t$) or following periods ($[DL]^t, \ldots, [DL]^T$).

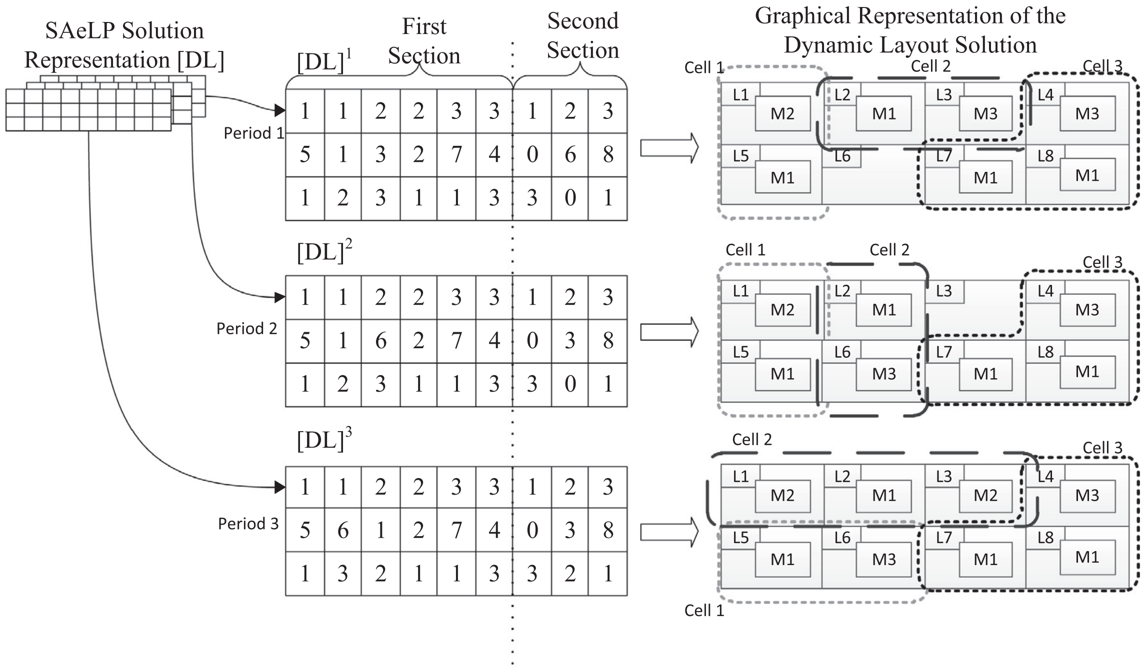


Fig. 2. Solution representation scheme in SAeLP.

4.1.2. Simulated Annealing parameters

SA requires some parameters to be defined in advance. These parameters include initial temperature, epoch length (Markov Chain Length), cooling rate, and termination condition. In this study these parameters are determined based on the problem parameters.

Initial temperature is a contributing factor to the quality of solutions found by SA (Meller & Bozer, 1996). The initial temperature must be high enough to allow acceptance of worse solutions with a high probability at the initial iterations. However, if the determined temperature is very high, then the running time is increased without providing any advantage on the best found solution. In this study, initial temperature is calculated so that the probability of accepting non-improving solutions is 0.95. In order to determine the initial temperature, the average of absolute difference between 500 random feasible solutions and their feasible neighbors are used. Let Z_a and Z'_a be the OFVs of random feasible initial solution and its feasible neighbor in a^{th} iteration, respectively (Eq. (18)). $\overline{OFV}$ is the average difference between initial solutions and their neighbors. The initial temperature is then calculated by using the Eq. (19).

$$\sum_{a=1}^{500} |Z_a - Z'_a|/500 = \overline{OFV} \tag{18}$$

$$T_{max} = \frac{-\overline{OFV}}{\ln 0.95} \tag{19}$$

Before the temperature is reduced in accordance with the cooling schedule, the SA steps are repeated until the search reaches to equilibrium. Epoch length is the number of visited solutions in each iteration. It is observed that as the problem complexity is increased, the epoch length should be increased for better solutions. Therefore, in this study epoch length (Markov Chain Length) is calculated based on the following formulation.

$$EpochLength = K \times T \tag{20}$$

In our implementation of SA, the search stops when the final temperature (T_{min}) is reached. As a lower final temperature leads to a finer search of the space, lower temperatures are desired. However, very low temperatures increase running time without improving solution quality. In this study, final temperature is determined in such a way that the acceptance probability of a neighbor solution is equal to 0.01 when the difference between OFVs of incumbent solution and its non-improving neighbor is equal to the 10% of the average OFV difference ($\overline{OFV}$).

$$T_{min} = \frac{-\overline{OFV} \times 0.1}{\ln 0.01} \tag{21}$$

Maintaining a slower cooling is essential for a better search of problem space. A cooling schedule with a cooling factor (α) of 0.975 is employed for all problems. Let T_n be the temperature at the nth iteration and α be the cooling rate, then:

$$T_n = T_{max} \times \alpha^{n-1} \tag{22}$$

4.2. The LP embedded genetic algorithm approach

In this section, we propose another solution approach (GAeLP), which is based on GA. In GA, solution representations are referred to as "chromosomes" and any given set of chromosomes are referred to as "population". The main idea behind the method is sequentially generating populations by employing some mechanisms derived from "natural selection" and "evolution" theories. Mainly, these mechanisms are selection, cross-over and mutation. Selection mechanism determines the chromosomes that will transmit their genetic material to the following generations through cross-over operator. Cross-over operator is the combination of two chromosomes (parent) from the previous generation (or population) in order to produce new chromosomes (child or offspring). In other words, selection operator determines the parent chromosomes to be crossed over and the cross-over operator produces their offspring. If only these two operators are applied throughout the generation of all populations, then, the chromosomes eventually become exact copies of each other. Therefore, in order to avoid this convergence, mutation operator is used. Mutation operator makes minor changes on the chromosomes and protects population from being stuck in local optima. The method will be discussed in detail after discussing solution representation and GA operators.

4.2.1. Solution representation

Considering the new solution generation mechanisms in GA, such as cross-over, directly using of solution representation of SaeLP ($[DL]$) is inefficient. In our initial studies, we have seen that direct using of SAeLP's solution representation scheme yield many infeasible solutions and require too many repair heuristics to be performed. In GAeLP, for the representation of a solution, three additional representations are used along with the $[DL]$ matrix. These three components are initially used by the GAeLP operators (cross-over and mutation) for the creation of new chromosomes. Then, in order to calculate objective function values (fitness values), these representations are converted into $[DL]$ representation matrix. First one of these is a two dimensional matrix, $[FP]_{3\times(U\times N)}$. In this representation is simply the layout of the first period and it s the exact copy of $[DL]^1$ sub-matrix. As in the case of $[DL]$ matrix, it consists of two sections and none of the elements in the first section is allowed to be zero. Thus, minimum cell size constraint is satisfied automatically. In the second section, zero elements are allowed and if there is a zero element, other elements of that column become ineffective.

Second element of the solution representation is a three dimensional matrix, $[Add]_{2\times((U-L)\times N)\times(T-1))}$. This matrix shows the types of machines to be added to the layout in each period, following the first period. Therefore, the size of the third dimension is $T-1$. For the sake of simplicity, let us assume that $[Add]$ matrix consists of $T-1$ sub-matrices ($[Add]^1_{2\times((U-L)\times N)},\ldots,[Add]^{T-1}_{2\times((U-L)\times N)}$). Each column of the $[Add]^n$ sub-matrix corresponds to a column in second section of $[DL]^n$ sub-matrix. In the first row $[Add]^n$ sub-matrix, the machine type numbers to be added are stored. Second row of the $[Add]^n$ sub-matrix stores binary key values. If the key value is 1 and if the corresponding column in $[DL]^n$ has at least one zero element in second or third row, then the machine is located to that position. Otherwise, adding of the machine to the layout representation is not performed. As it might be inferred, the number of columns in this representation is $(U-L)\times N$ because, no empty position is allowed in the first section of the $[DL]^n$ sub-matrix. Therefore, the number of columns in $[Add]^n$ sub-matrix is equal to the number of columns in the $[DL]^n$ matrix's second section.

The third element of the GAeLP representation scheme is the three dimensional $[Rel]_{3\times R\times(T-1)}$ matrix. The size of its second dimension (R) is arbitrarily determined. In our experiments we have seen that setting the R value to half of the number of available locations (K) is sufficient. This matrix determines the relocations on the solution representation in each period. For the sake of simplicity, let us assume that $[Rel]$ matrix consists of $T-1$ sub-matrices ($[Rel]^1_{3\times R},\ldots,[Rel]^{T-1}_{3\times R}$). The first and the second rows of the $[Rel]^n$ sub-matrix stores the column numbers of the elements to be swapped in the $[DL]^n$ while creating $[DL]^{n+1}$ matrix. The elements of the third column are ternary key values. If the key value is 0, the positions are not swapped. If the value is 1, only the corresponding third row elements of $[DL]^n$ sub-matrix (machine numbers) are swapped. Namely, only machines are relocated. If the key value is 2, both the corresponding second and third row elements of $[DL]^n$ sub-matrix (location and machine numbers) are swapped. In all of the swap operations, in the first section of the $[DL]^n$ sub-matrix, only non-zero elements are allowed. Any swap operation which would place zero element in the first section is cancelled.

Before the GAeLP can proceed, these three elements of the solution representation should be used for the construction of $[DL]$ solution representation. This construction procedure described in Fig. 3. As $[FP]$ matrix is as same as $[DL]$ matrix's sub-matrix $[DL]^1$, it is directly transferred to the representation. By using $[Add]^1$ and $[Rel]^1$, machines are added and relocated. Thus, the layout of second period ($[DL]^2$) is obtained. Implementation order of $[Add]^n$ and $[Rel]^n$ matrices affect the ultimate $[DL]$ matrix. In this study, during the construction of $[DL]^{n+1}$ matrix, firstly $[Add]^n$ and then $[Rel]^n$ is used. As long as it is consistently applied throughout the algorithm, reverse order of implementation can also be preferred. As in the case of $[DL]$ matrix, some elements without any effect on ultimate layout may appear in $[Add]$ and $[Rel]$ solution representations. These elements are intentionally stored in solution representation. After implementation of some GA operators, these elements may become effective and help better searching of the solution space by producing new unvisited solutions.

4.2.2. Crossover, mutation and selection mechanisms

In GAeLP approach, neighbor solutions are generated by using solution representations, $[FP]$, $[Add]$ and $[Rel]$. For each component of the representation scheme, different cross-over and mutation operators are employed. These are summarized below.

Cross-Over of [FP]: For $[FP]$ component of solution representation, one of two types of cross-over operator is applied. In the first type of the cross-over, the third row of the parent 1's $[FP]$ and the second row of parent 2's $[FP]$ are combined. Thus, child 1's $[FP]$ matrix is obtained. Then, the third row of the parent 2' $[FP]$ and the second row of the parent 1's $[FP]$ are combined. Thus, child 2's $[FP]$ matrix is obtained. In the second type of cross-over, two swap column points are selected and the information between these two cross-over points are swapped, thus child solutions are obtained. When conducting second type of cross-over, zero elements in the first section are removed. In addition, as the second row of $[FP]$ holds location numbers, it is ensured that each non-zero location number is repeated only once.

Cross-over of $[Add]$: Two types of cross-over is applied for $[Add]$ matrix. In the first type, child solutions are generated by cross-matching of parents' first and second rows of $[Add]$ matrices. In the second type of cross-over, two swap column points are determined and the information between these two points are swapped among parents. Thus, children are obtained. Since $[Add]$ matrix is a three-dimensional matrix, it should be mentioned that the same swap points are used for each of the $T-1$ periods.

Cross-over of $[Rel]$: As opposed to the other components of solution representation of GAeLP, single type of cross-over is applied for $[Rel]$ component of the solution representation. In this cross-over operator, different pairs of swap column points are determined for each of the $T-1$ periods. The information between these points are swapped among $[Rel]$ matrices of the parents.

Mutation of $[FP]$: Mutation of the solutions are performed right after creation of $[FP]$ representation components of children. Two types of mutation operators are used for $[FP]$. In the first type of the mutation, after randomly selecting a pair of columns in a child solution's $[FP]$, either the second row, or the third row or both the second and the third row elements in these columns are swapped. While swapping the elements, entering of zero values to the first section of $[FP]$ matrix is avoided.

Mutation of $[Add]$: Two types of mutation operators are used. In the first operator, one of the $T-1$ sub-matrices of $[Add]$ is chosen and one of the machine numbers in the first row of the sub-matrix is changed. In the second operator, one of the $T-1$ sub-matrices of $[Add]$ is chosen and a key value is selected

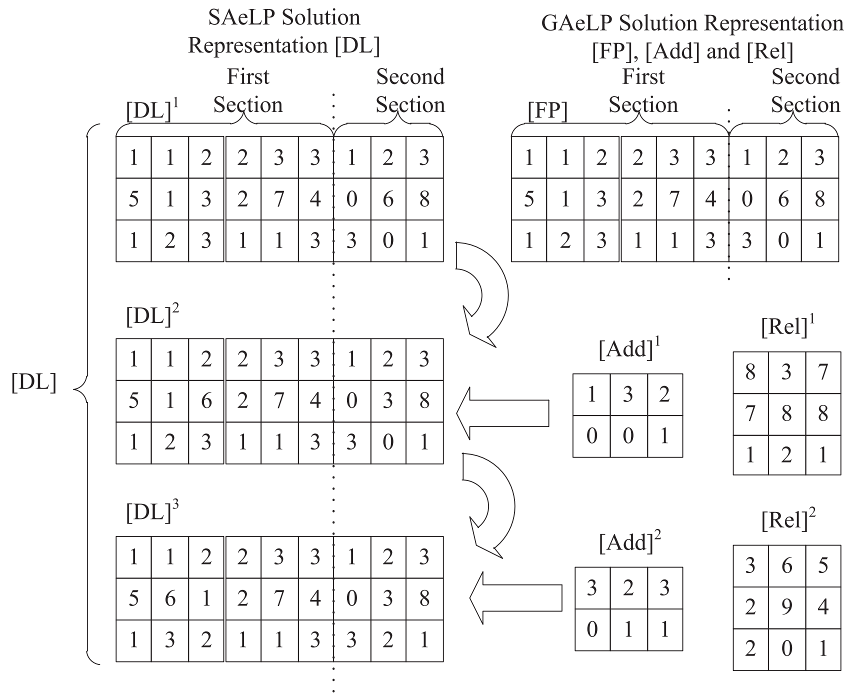


Fig. 3. Illustration of GAeLP's solution representation scheme and conversion procedure.

randomly. Then, this value is switched either from 1 to 0 or from 0 to 1.

Mutation of [*Rel*]: Two types of mutation operators are used. In the first one, values of only the first and the second row elements are changed. These elements are randomly selected. In the second type of the mutation operator, the elements of the key value row are changed with randomly generated key values.

In the above given paragraphs, mutation and cross-over operators of the solution methodology were summarized. Another operator of GA is the selection operator. It determines parent chromosomes of the cross-over. The chromosomes selected by selection operator transmit their genes to the following populations. In this study, we applied tournament selection operator, which is easier to implement. In order to be able to select parents by employing tournament selection, fitness values of the all parent solutions must be already calculated. In tournament selection, a smaller group of candidate solutions are randomly selected among the whole population. The best one among the sub-group is selected as the first parent. This is repeated again and the second parent of the two offspring solutions is determined. Then, cross-over and mutation operators are performed and children chromosomes are determined. These steps are repeated until all children of the new population are produced.

4.2.3. Description of the GAeLP

Our GA based solution methodology (GAeLP) is described in Fig. 4. The LP is integrated to GAeLP as it was done with the SAeLP. GAeLP method begins with the randomly producing of an initial population. Firstly, by using all three matrix components of the GAeLP solution representation ([*FP*], [*Add*] and [*Rel*]), [*DL*] matrices are created. Actually, throughout the algorithm, three components of the solution representation can be directly used but, reading the solution from these three components is not easy and calculation of objective function values from this representations is more complex. Therefore, we convert this representation into [*DL*] matrices.

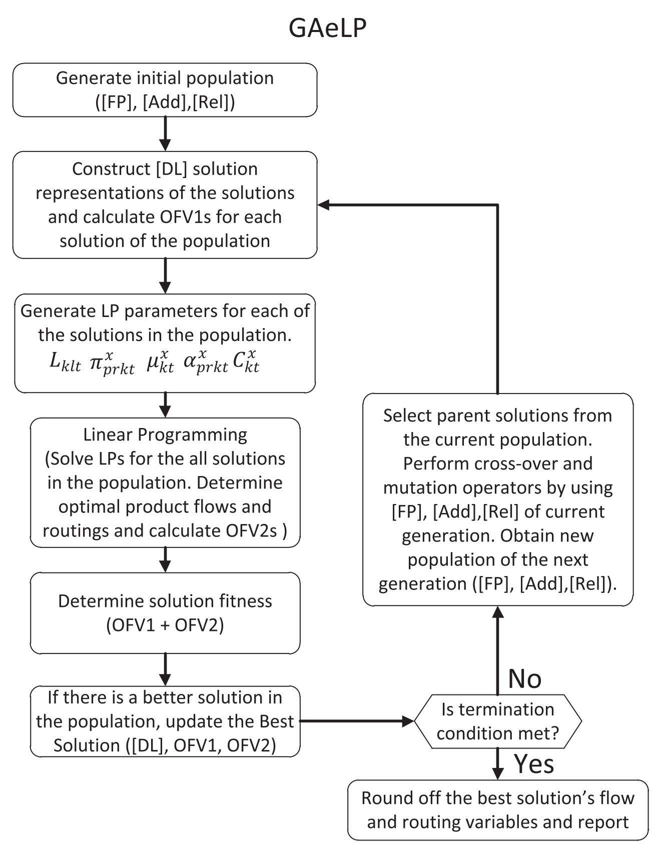


Fig. 4. Summary of the algorithmic steps of GAeLP.

[*FP*], [*Add*], [*Rel*] and [*DL*] representations are stored for all of the chromosomes in the population until they are replaced by their offsprings.

Since [*DL*] matrices store the layout related information, the objective function's certain parts can be calculated. Only by using [*DL*] representation, purchasing, constant, installation and uninstallation costs of the machines can be determined (OFV1). Right after the completion of the conversion, OFV1 values of the solutions in the population are calculated. In order to determine material routing and material flow related costs (OFV2: processing cost, intercell and intra-cell material handling costs) LP is used. LP requires some parameters to be modified. This steps will be discussed in detail, in the following sections. After the modification of these parameters for every solution representation, a series of LP are run. For example, if there are 100 chromosomes in a population, 100 different LP problems are run in order to obtain OFV2s of the layout solutions. Then, the total cost of layouts (OFV1 and OFV2) are determined. Total cost is also used as the fitness value during the selection operator.

Tournament selection operator of GA selects the solutions to be used for the reproduction of children, by comparing their fitness values (OFV1 + OFV2). For the cross-over and mutation of the solutions three-component solution representations ([*FP*], [*Add*], [*Rel*]) of the solutions are used. After the production of new generation, solution representations are converted to [*DL*], and OFV1s of the solutions are calculated, LP parameters are generated for all of the chromosomes in the population. Then, an LP problem for each solution is solved and OFV2s are calculated. These steps are repeated until the GAeLP's termination condition is met.

4.3. The mathematical model of the sub-problem

Determination of machine-cell-location assignments either by SA or GA dramatically simplifies the mathematical model to be solved by LP by reducing the number of variables, indices and constraints and eliminating all integer and binary variables. This sub-problem is solved every time SAeLP or GAeLP visits a layout solution. Therefore, during the search stage of this approaches, LP problem is solved tens thousands times with different parameters.

Indexing sets:

t: index for periods,
p: index for product types,
r: index for operation types,
k, l: index for locations.

Parameters:

T: Number of planning periods.
P: Number of product types.
R_p: Number of operations for product type p.
K: Number of locations.
E_p: Inter-cellular material handling cost per product type p, per unit distance.
A_p: Intra-cellular material handling cost per product type p, per unit distance.
L_{klt} : 1, if locations k and l are in the same cell in period t, 0 otherwise.
D^*_{pt}: Increased *demand for product type p, in period t.*
λ_{kl}: Distance between locations k and l.
π^x_{prkt}: Unit processing time of r^{th} *operation of product type p, in location k in period t.*
μ^x_{kt}: Unit variable cost of machine in location k, *in period t.*
C^x_{kt}: Capacity of machine in location k in period t.
α^x_{prkt} : 1, if r^{th} operation of product type p in location k, in period t, 0 otherwise.

Decision variables:

G_{prklt}: Number of products of type p, processed by operation r, in location k and moved to location l to be processed by operation $r + 1$, in period t.
g_{prkt}: Number of products of type p, processed by operation r, in location k in period t.

Objective function:

$$\min z = \sum_{t=1}^{T}\sum_{p=1}^{P}\sum_{r=1}^{R_p-1}\sum_{k=1}^{K}\sum_{l=1}^{K} G_{prklt} \times \lambda_{kl} \times E_p \times (1 - L_{klt}) \tag{23}$$

$$+ \sum_{t=1}^{T}\sum_{p=1}^{P}\sum_{r=1}^{R_p-1}\sum_{k=1}^{K}\sum_{l=1}^{K} G_{prklt} \times \lambda_{kl} \times A_p \times L_{klt} \tag{24}$$

$$+ \sum_{t=1}^{T}\sum_{p=1}^{P}\sum_{r=1}^{R_p}\sum_{k=1}^{K}\sum_{l=1}^{K} g_{prkt} \times \pi^x_{prkt} \times \mu^x_{kt} \tag{25}$$

Subject to:

$$g_{prkt} \leqslant \alpha^x_{prkt} \times D^*_{pt} \qquad \forall \quad p,r,k,t \tag{26}$$

$$\sum_{k=1}^{K} g_{prkt} = D^*_{pt} \qquad \forall \quad p,r,t \tag{27}$$

$$\sum_{p=1}^{P}\sum_{r=1}^{R_p} g_{prkt} \times \pi^x_{prkt} \leqslant C^x_{kt} \qquad \forall \quad k,t \tag{28}$$

$$\sum_{k=1}^{K} G_{prklt} = g_{p,r+1,lt} \qquad \forall \quad p,l,t,r = 1,\ldots,R_p - 1 \tag{29}$$

$$\sum_{l=1}^{K} G_{prklt} = g_{prkt} \qquad \forall \quad p,k,t,r = 1,\ldots,R_p - 1 \tag{30}$$

$G_{prklt}, g_{prkt} \geqslant 0$ (Continuous).

As discussed above, once the machines are located and are assigned to cells, the third, fourth, fifth, seventh and eighth terms in the objective function (1) of the main mathematical model can be determined. Only the inter- and intra-cell MHC and variable costs of machine processing are needed to be determined by LP because these costs are dependent on the decision variables of product routing, which is determined by LP.

The term (23) imposes the inter-cell MHC to the model if the location k and l belong to different cells. Nevertheless, if these two locations are assigned to the same cell, intra-cell MHC is imposed by the term (24). The last term of the objective function, namely the term (25), calculates the processing costs of the products in all periods.

(26) satisfies the condition that the quantity of certain product operation in a period cannot exceed the demand for that product in that period and can only be processed in locations which are capable of processing that operation. The equality (27) ensures that all products of a given type are processed for their every operation. The inequality (28) guarantees that the capacities of locations, which are equal to the capacities of the machine on those locations, are not exceeded. The inequalities (29) and (30) are flow conservation constraints. It must be kept in mind that the decision variables of the sub-problem are continuous.

After LP sub-problem is solved, all terms of the objective function of the main model are determined. In accordance with these results and the dynamics of SA, SA moves to a new layout solution. Then, the new LP problem with new parameters is solved. These steps are repeated until SA termination condition is met. Finally, in the best solution found in SA, the material flow and part operation variables are rounded to the closest integer considering the machine capabilities, machine capacities, etc.

4.4. LP interface

After generating a layout solution by SAeLP or a population of solutions by GAeLP, an LP sub-problem must be solved for each layout solution, in order to determine the optimal material flow between locations and the process routings. This step is totally same for any generated layout solution, which can either be generated by SAeLP or GAeLP. As it might be noticed, in the sub-problem's mathematical model, the indices that represent machine types (i,j) and cells (c,d) are removed, since the machines in the locations are already determined by solution representations of SAeLP and GAeLP. Some parameters of the LP are dependent on machine-location-cell assignments therefore, they are needed to be updated in every call of the LP according to the dynamic layout plan generated by the solution approach. Updating the parameters is performed by the interface between LP and the meta-heuristics (SA or GA). The updated parameters are $\pi^x_{prkt},\ \mu^x_{kt},\ \alpha^x_{prkt},\ C^x_{kt}$ and L_{klt}.

The parameters $\pi^x_{prkt},\ \mu^x_{kt},\ \alpha^x_{prkt}$ and C^x_{kt} are related to the original mathematical model's parameters, $\pi_{pri},\ \mu_i,\ \alpha_{pri}$ and C_i, respectively. The parameters superscripted with the ^{x} symbol are also dependent to machine-location-cell assignments, which are denoted by x_{ckit} in the main model. Therefore, these parameters are needed to be modified considering the machine types assigned to the locations in every call of LP. Updating of these variables is quite straightforward. For example, if a machine of type i is assigned to location k in period t, then $\pi^x_{prkt} = \pi_{pri}\ \forall p,r$. Other parameters are updated similarly. Namely, capacities (C^x_{kt}), capabilities (α^x_{prkt}), processing times (π^x_{prkt}) and variable costs (μ^x_{kt}) of locations are updated based on the machine type assigned to given location in a given period. Assuming that x^*_{ckit} represents a feasible dynamic layout solution's machine-location-cell assignments, mathematical description of the parameter update is given in (31)–(34). These equations are given in order to illustrate the mathematical and the logical basis of this interfacing procedure. In the executable programs of SAeLP and GAeLP, this procedure is carried out by directly using of $[DL]$ matrices and by implementing some logical expressions. Actually, x^*_{ckit} variables are neither created nor used by the interface.

$$\pi^x_{prkt} = \sum_{c=1}^{N}\sum_{i=1}^{M}(x^*_{ckit} \times \pi_{pri}) \qquad \forall \quad p,r,k,t \tag{31}$$

$$\alpha^x_{prkt} = \sum_{c=1}^{N}\sum_{i=1}^{M}(x^*_{ckit} \times \alpha_{pri}) \qquad \forall \quad p,r,k,t \tag{32}$$

$$\mu^x_{kt} = \sum_{c=1}^{N}\sum_{i=1}^{M}(x^*_{ckit} \times \mu_i) \qquad \forall \quad k,t \tag{33}$$

$$C^x_{kt} = \sum_{c=1}^{N}\sum_{i=1}^{M}(x^*_{ckit} \times C_i) \qquad \forall \quad k,t \tag{34}$$

In addition to those updated parameters, a new parameter L_{klt} is produced. This parameter is also generated by the interface between meta-heuristic and LP according to the layout solution visited by SAeLP or GAeLP during the search. This parameter denotes whether two different locations are assigned to the same cell in a given period. This parameter is important for the correct calculation of the inter-cell and intra-cell MHCs. This parameter is also generated by directly using of $[DL]$ solution representation, without creating x^*_{ckit} variables. The mathematical basis of the generation of this parameter is as follows:

$$L_{klt} = \sum_{c=1}^{N}\left(\sum_{i=1}^{M}x^*_{ckit} \times \sum_{j=1}^{M}x^*_{cljt}\right) \qquad \forall \quad k,l,t \tag{35}$$

4.5. Integer solution

Although the original problem only allows splitting of lots into integer values, the variable values returned by the solving of LP sub-problems are mostly non-integer. After the termination of SAeLP or GAeLP's search stage, the flow and routing variables of the best layout solution should be rounded off to integer values. Rounding off of these solutions to closest integer may cause violation of capacity constraints of machines. In fact, machine capacity constraints are vulnerable to rounding to the closest and larger integer, because any such rounding may exceed machine capacity. In addition, flow conservation constraints are also needed to be taken into account during the rounding off of non-integer values. Therefore, directly rounding off of the continuous variables requires consideration of many constraints. In the proposed approaches, in order to round off these variables, the LP sub-problem is solved for the last time, with the imposition of integrality constraints for G_{prklt} and g_{prkt} variables. Thus, the LP turns into a MIP problem. This step is carried out only once, for the best layout solution, which is found at the end of search stages of GAeLP and SAeLP. Although MIP problems are harder to solve compared to their LP counterparts, this step has a very limited impact on the run time because, it is only executed only once and does not require the proven optimal integer solution to be found. In the case of very complex sub-problems with too many products, operations, locations and periods, a time limit can be imposed to the rounding procedure. So, a relatively good feasible integer solution can be reported. Even with our large test samples, we have rarely encountered such a situation.

5. Numerical example

In this section a numerical example is provided. In this example, rather than discussing all steps of SAeLP and GAeLP, some important steps are focused on.

5.1. Problem parameters

The sample problem used in this section for the illustration purpose is sample number 2. The parameters of the example are given in Tables 1 and 2.

The illustrative example consists of two part types ($P = 2$), three operations for each part type ($R_p = 3$), three machine types ($M = 3$) and 5 machine locations ($K = 5$). Two cells must be formed ($N = 2$). Maximum cell capacity is three machines ($U = 3$) and each cell must have at least one machine ($L = 1$). Planning horizon is two periods ($T = 2$). Unit distance inter-cell and intra-cell material handling costs for a unit of product of any type are 50$/m and 5$/m, respectively ($E_p = 50$, $A_p = 5$). The remaining parameters are given in Tables 1 and 2.

5.2. Dynamic layout solution and LP solution

LP is used repetitively for every layout solution generated by SAeLP and GAeLP. LP parameters are generated and the corresponding LP is solved to determine the optimal material flow and product routings. Let's assume that the layout solution shown in Fig. 5 is obtained in an intermediate iteration of SAeLP or in a generation of GAeLP. Using this layout solution, machine purchasing costs ($80,000), installation costs ($2775), uninstallation costs ($775) and overhead costs ($14,400) can be calculated. The total of these costs (OFV1*) is $97,950 for the solution given in Fig. 5.

After obtaining a layout solution, LP parameters are needed to be generated considering the layout solution and the main model parameters. Then, the LP parameters are generated employing

Table 1
Machine capability, capacity and cost parameters of sample 2.

	Machine information						π_{pri} (h) and α_{pri} [a]					
							p = 1			p = 2		
	γ_i ($)	β_i ($)	δ_i ($)	θ_i ($)	μ_i ($/h)	C_i (h)	r = 1	r = 2	r = 3	r = 1	r = 2	r = 3
i = 1	18000	1800	450	450	9	500	0.54	0.79			0.80	
i = 2	15000	1500	375	375	7	500		0.53		0.45		0.76
i = 3	16000	1600	400	400	6	500	0.77		0.33		0.91	0.80

D_{pt}	p = 1	p = 2
t = 1	400	300
t = 2	500	200

[a] $\alpha_{pri} = 1$ *if* $\pi_{pri} > 0, 0$ *otherwise*.

the terms (31)–(35). Values of x^*_{ckit}, which correspond to the layout solution shown in Fig. 5, are as follows:

$x^*_{1,1,1,1} = 1, \quad x^*_{2,2,2,1} = 1, \quad x^*_{1,3,3,1} = 1, \quad x^*_{2,4,2,1} = 1, \quad x^*_{1,1,1,2} = 1,$
$x^*_{1,2,3,2} = 1, x^*_{2,3,2,2} = 1, \quad x^*_{1,4,2,2} = 1, \quad x^*_{2,5,3,2} = 1$
for all other, $x^*_{ckit} = 0$.

The LP parameters calculated by the LP interface are given below. Multi-dimensional matrices are represented as described by Solo (2010).

$$\mu^x_{kt} = \begin{bmatrix} 9 & 9 \\ 7 & 6 \\ 6 & 7 \\ 7 & 7 \\ 0 & 6 \end{bmatrix}, \quad C^x_{kt} = \begin{bmatrix} 500 & 500 \\ 500 & 500 \\ 500 & 500 \\ 500 & 500 \\ 0 & 500 \end{bmatrix}, \quad \alpha^x_{prkt} = \begin{bmatrix} \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \end{bmatrix} & \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \end{bmatrix} \\ \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix} & \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix} \\ \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix} & \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix} \\ \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix} & \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix} \\ \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} & \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix} \end{bmatrix},$$

$$\pi^x_{prkt} = \begin{bmatrix} \begin{bmatrix} 0.54 & 0.79 & N/A \\ N/A & 0.80 & N/A \end{bmatrix} & \begin{bmatrix} 0.54 & 0.79 & N/A \\ N/A & 0.80 & N/A \end{bmatrix} \\ \begin{bmatrix} N/A & 0.53 & N/A \\ 0.45 & N/A & 0.76 \end{bmatrix} & \begin{bmatrix} 0.77 & N/A & 0.33 \\ N/A & 0.91 & 0.80 \end{bmatrix} \\ \begin{bmatrix} 0.77 & N/A & 0.33 \\ N/A & 0.91 & 0.80 \end{bmatrix} & \begin{bmatrix} N/A & 0.53 & N/A \\ 0.45 & N/A & 0.76 \end{bmatrix} \\ \begin{bmatrix} N/A & 0.53 & N/A \\ 0.45 & N/A & 0.76 \end{bmatrix} & \begin{bmatrix} N/A & 0.53 & N/A \\ 0.45 & N/A & 0.76 \end{bmatrix} \\ \begin{bmatrix} N/A & N/A & N/A \\ N/A & N/A & N/A \end{bmatrix} & \begin{bmatrix} 0.77 & N/A & 0.33 \\ N/A & 0.91 & 0.80 \end{bmatrix} \end{bmatrix},$$

$$L_{klt} = \begin{bmatrix} \begin{bmatrix} 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \\ \begin{bmatrix} 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \end{bmatrix} \end{bmatrix}$$

Table 2
Distance matrix of locations.

λ_{kl} (m)	l = 1	l = 2	l = 3	l = 4	l = 5
k = 1	0	1	1	2	2
k = 2	1	0	2	1	3
k = 3	1	2	0	1	1
k = 4	2	1	1	0	2
k = 5	2	3	1	2	0

After the determination of LP parameters, solution of the LP sub-problem is obtained. The corresponding optimal LP solution (flow and routing) for the given layout is illustrated in Tables 3 and 4. Using the LP solution, processing costs of the parts, inter- and intra-cell material handling costs are calculated. The total of these costs is $52,249.06 (OFV2*) and the total cost (OFV1* + OFV2*) is $150,199.06.

5.3. Integer solution

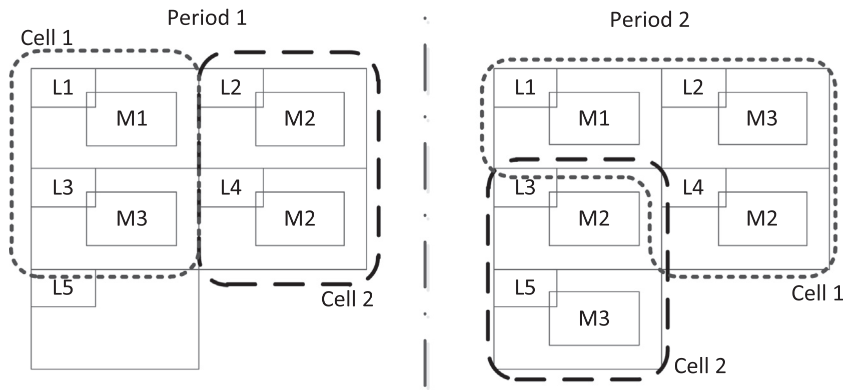
After the completion of SA, the best solution found contains some continuous variables. Therefore, the final solution must be rounded off to integer values. Fig. 6, Tables 5 and 6 illustrate the best found solution's machine layout, part routing and material flow, respectively. The total objective function value (OFV1 + OFV2) for the continuous best found solution is $103434.0.

As it can be seen in Tables 5 and 6, product routing and product flow solutions still include some fractional values. Therefore, the final solution is needed to be rounded off to integer values. In order to improve computational efficiency, rounding is performed only for the final solution. Tables 7 and 8 show the values of the feasible integer solution. The solution found is also the optimal solution of the original problem (103,434). However, it must be reminded that the rounded values are not necessarily the optimal integer solution. Nevertheless, we observed that, even in large problems, rounding procedure is generally able to find optimal integer solutions.

6. Computational results and discussion

Both of the proposed approaches are written in MATLAB programming language. In order to test the performance of the algorithm, written programming code was run ten times for each problem instance. A desktop PC with Intel i5 processor (3.1 GHz), Windows 7 OS and 4 GB of RAM was used.

In order to validate the effectiveness of the proposed approaches, a number of problem samples from the literature (Kia et al., 2012) were solved. The obtained results were compared with those of the previous study and branch and bound algorithm (B&B). The results of Kia et al. (2012) are comparable to those of this study because assumptions of the both models are similar,



(M1 represents an instance of machine type 1. L1 denotes the location 1.)

Fig. 5. A dynamic layout solution.

Table 3
Product routings obtained by LP (continuous).

	Machine info			Product and operation info					
				Product 1			Product 2		
	Cell	Loc.	Mach.	Op1	Op2	Op3	Op1	Op2	Op3
Period 1	C1	L1	M1	340.74	400				
		L3	M3	59.26		400		300	61.71
	C2	L2	M2						
		L4	M2				300		238.29
Period 2	C1	L1	M1	375.94	375.94				
		L2	M3			375.94		200	200
		L4	M2				200		
	C2	L3	M2		124.06				
		L5	M3	124.06		124.06			

Table 4
Transfer of parts between locations obtained by LP (continuous).

	Product 1		Product 2	
	Op1 ⇒ Op2	Op2 ⇒ Op3	Op1 ⇒ Op2	Op2 ⇒ Op3
Period 1	L1 ⇒ L1 (340.74) L3 ⇒ L1 (59.26)	L1 ⇒ L3 (400)	L4 ⇒ L3 (300)	L3 ⇒ L3 (61.71) L3 ⇒ L4 (238.29)
Period 2	L1 ⇒ L1 (375.94) L5 ⇒ L3 (124.06)	L1 ⇒ L2 (375.94) L3 ⇒ L5 (124.06)	L4 ⇒ L2 (200)	L2 ⇒ L2 (200)

except the relocation cost. In Kia et al. (2012), it is assumed that the unit installation and the unit uninstallation costs of the machines were equal, which is a restrictive assumption. Thus, in their model, when a machine installation or uninstallation occurs, only half of the unit relocation cost is incurred. However, in our model this assumption is generalized. Thus, installation and uninstallation costs are not necessarily equal in our model, as they are imposed separately. In addition to Kia et al. (2012)'s test samples, 13 random test problems were also generated and solved. All of the problems were also run by using GAMS/Cplex optimization software (B&B) for 10^5 CPU seconds. The best results obtained by the optimization software are also reported. Problem size data is given in Table 9. Comparative results are given in Table 10.

GAeLP's parameters are determined after some initial experiments. The cross-over rate is 0.9, mutation rate is 0.1, population size is 200 and maximum number of generations is 1000. However, if best solution cannot be improved in 50 subsequent generations, search is terminated and the best found solution is reported after rounding off of the product flow and routing variables.

We observed that in Kia et al. (2012)' study, OFVs were miscalculated. Mostly, the differences between the reported results and the corrected results are very small. For example in the first example the machine installation costs in the first period was not included in the OFV although it was included in the mathematical model. Therefore, OFVs of all solutions provided by Kia et al. (2012) were recalculated and are given in Table 10. All the comparisons were made using the corrected results.

In the computational experiments we observed that both SAeLP and GAeLP outperformed the approach developed by Kia et al. (2012) in terms of solution quality. All of the results found by SAeLP and GAeLP are better than those found by Kia et al. (2012). While the relative improvement in the results is small for smaller problems, as the problem size increases, higher relative improvements in total minimized cost are obtained. In addition to being able to reach better solutions, SAeLP and GAeLP are also superior to Kia et al. (2012)'s technique in terms of computational time. Even for larger problem instances, computational time is quite acceptable. As the used computer systems are similar and the programming platform (MATLAB) are same, the obtained solution times are directly comparable to those of Kia et al. (2012).

As the problems provided by Kia et al. (2012) considers very few number of products, these problems are not adequate to make inferences about the performances of the solution approaches. Therefore, 13 random test samples are generated and performances of the approaches are also tested against these problems. After the tests, we observed that especially for larger problems, solution approaches are capable of finding better solutions than those found by the optimization software. However, some problem

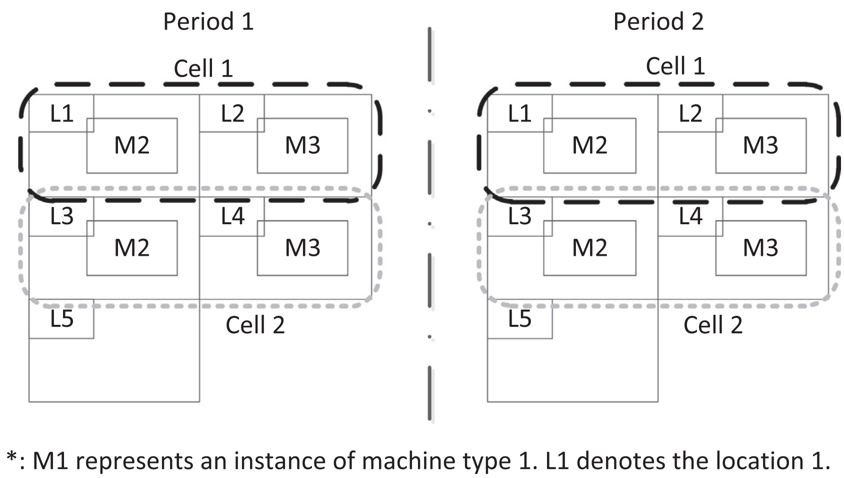


Fig. 6. Dynamic layout of the best obtained solution.

Table 5
Product routings of the best solution (continuous).

	Machine info			Product and operation info					
				Product 1			Product 2		
	Cell	Loc.	Mach.	Op1	Op2	Op3	Op1	Op2	Op3
Period 1	C1	L1	M2				292.4		
		L2	M3					292.4	292.4
	C2	L3	M2		400		7.6		
		L4	M3	400		400		7.6	7.6
Period 2	C1	L1	M2		45.45		200		
		L2	M3	45.45		45.45		200	200
	C2	L3	M2		454.55				
		L4	M3	454.55		454.55			

Table 6
Transfer of parts between locations for the best solution (continuous).

	Product 1		Product 2	
	Op1 ⇒ Op2	Op2 ⇒ Op3	Op1 ⇒ Op2	Op2 ⇒ Op3
Period 1	L4 ⇒ L3 (400)	L3 ⇒ L4 (400)	L1 ⇒ L2 (292.40) L3 ⇒ L4 (7.60)	L2 ⇒ L2 (292.40) L4 ⇒ L4 (7.60)
Period 2	L2 ⇒ L1 (45.45) L4 ⇒ L3 (454.55)	L1 ⇒ L2 (45.45) L3 ⇒ L4 (454.55)	L1 ⇒ L2 (200)	L2 ⇒ L2 (200)

Table 7
Product routings of the best solution (integer).

	Machine info			Product and operation info					
				Product 1			Product 2		
	Cell	Loc.	Mach.	Op1	Op2	Op3	Op1	Op2	Op3
Period 1	C1	L1	M2		400		8		
		L2	M3	400		400		8	8
	C2	L3	M2				292		
		L4	M3					292	292
Period 2	C1	L1	M2		46		200		
		L2	M3	46		46		200	200
	C2	L3	M2		454				
		L4	M3	454		454			

samples are so complex that B&B algorithm could not even find a feasible solution after running of the problem for 10^5 s. This can be anticipated in advance because some larger problems have almost 0.2 billion integer and binary variables and almost half a million constraints. Hence, we developed a mathematical model to obtain lower bounds for such problems.

Table 8
Transfer of parts between locations for the best solution (integer).

	Product 1		Product 2	
	Op1 ⇒ Op2	Op2 ⇒ Op3	Op1 ⇒ Op2	Op2 ⇒ Op3
Period 1	L2 ⇒ L1 (400)	L1 ⇒ L2 (400)	L1 ⇒ L2 (8) L3 ⇒ L4 (292)	L2 ⇒ L2 (8) L4 ⇒ L4 (292)
Period 2	L2 ⇒ L1 (49) L4 ⇒ L3 (451)	L1 ⇒ L2 (49) L3 ⇒ L4 (451)	L1 ⇒ L2 (200)	L2 ⇒ L2 (200)

6.1. A lower bound for the problem

Since the relative gap between the obtained results' objective function value and those of Kia et al. (2012)'s solutions are large and since we couldn't find any feasible integer solution by using optimization software for some larger problem instances, we have developed a lower bound, in order to test our solution's quality. In this section, the mathematical model of the lower bound is given.

Indexing sets:

t: index for time periods,
p: *index for product types,*
r: *index for operations,*
i, j: *indices for machine types,*
c, d: *indices for cells.*

Parameters:

T: *Number of planning periods in the planning horizon.*
P: *Number of product types to be produced.*
R_p: *Number of operations for product type p.*
N: *Maximum number of cells.*
K: *Number of available locations in the shop floor.*
M: *Number of machine types.*
A: *Intra-cellular material handling cost per product type p, per unit distance.*
E: *Inter-cellular material handling cost per product type p, per unit distance.*
D_{pt}: *Demand for product type p, in period t.*
λ^*: Shortest distance in λ_{kl} distance matrix ($\lambda^* = \min_{k \neq l} \lambda_{kl} | k$ *and* $l \in K$).
δ_i: *Installation cost machines of type i.*
β_i: *Overhead cost of machine type i in each period.*
π_{pri}: *Processing time of* r^{th} *operation of product type p, in machine type i.*
μ_i: *Unit time variable cost of machine type i.*
γ_i: *Purchasing cost of machine type i.*
U: *The maximum number of machines that can be assigned to a manufacturing cell.*
L: *The minimum number of machines that can be assigned to a manufacturing cell.*
C_i: *Capacity of machine type i in each period.*
α_{pri}: *1, if* r^{th} *operation of product p can be processed by machine type i, 0, otherwise.*

Decision variables:

X_{cit}: The number of machines of type i that is assigned to cell c, in period t.
$F_{prcdijt}$: Number of products of type p, processed by operation r, on machine type i, which is assigned to cell c and moved to be processed by operation $r+1$, on machine type j, which is assigned to cell d, in period t.
f_{prcit}: Number of products of type p, processed in operation r on machine type i which is assigned to cell c in period t.
Q_{it}: Total number of machines of type i added to the layout at the beginning of the period t.

Objective function:

$$\min z = \sum_{t \in T} \sum_{p \in P} \sum_{r \in R} \sum_{c \in N} \sum_{\substack{d \in N \\ c \neq d}} \sum_{i \in M} \sum_{j \in M} F_{prcdijt} E \lambda^* \tag{36}$$

$$+ \sum_{t \in T} \sum_{p \in P} \sum_{r \in R} \sum_{c \in N} \sum_{i \in M} \sum_{\substack{j \in M \\ j \neq i}} F_{prccijt} A \lambda^* \tag{37}$$

$$+ \sum_{t \in T} \sum_{p \in P} \sum_{r \in R} \sum_{c \in N} \sum_{i \in M} f_{prcit} \pi_{pri} \mu_i \tag{38}$$

$$+ \sum_{t \in T} \sum_{i \in M} Q_{it} \gamma_i \tag{39}$$

$$+ \sum_{t \in T} \sum_{c \in N} \sum_{i \in M} X_{cit} \beta_i \tag{40}$$

$$+ \sum_{t \in T} \sum_{i \in M} Q_{it} \delta_i \tag{41}$$

Table 9
Problem size data, number of variables and constraints and Simulated Annealing parameters.

No	Problem size data						Number of variables			Number of constraints	Simulated Annealing parameters			
	P	R_p	K	T	N	M	Binary	Integer	Total		Initial temp.	Final temp.	Epoch length	Cooling rate
1	2	2	4	2	2	2	48	1156	1204	316	7.3×10^4	165	8	0.975
2	2	3	5	2	2	3	90	7566	7656	936	11.6×10^4	265	10	0.975
3	3	3	6	2	2	4	144	28520	28664	2158	16.3×10^4	371	12	0.975
4	4	3	7	2	3	4	224	114920	115144	4874	2.2×10^5	503	14	0.975
5	4	4	9	3	3	4	468	425100	425568	13317	3.3×10^5	745	27	0.975
6	5	3	8	2	3	5	320	291610	291930	8628	2.1×10^5	491	16	0.975
7	5	3	8	3	3	5	520	437415	437935	12982	2.5×10^5	581	24	0.975
8	5	4	11	3	3	5	715	1235040	1235755	25261	3.0×10^5	696	33	0.975
9	6	3	10	4	3	6	1080	1568184	1569264	31000	3.9×10^5	884	40	0.975
10	8	4	12	3	4	8	1536	10653720	10655256	93012	2.3×10^5	521	36	0.975
11	12	4	15	2	4	8	1200	16634896	16636096	115838	2.0×10^5	458	30	0.975
12	15	3	8	3	4	8	1024	5932824	5933848	81295	3.9×10^5	899	24	0.975
13	20	4	9	2	4	10	900	15609620	15610520	144574	3.5×10^5	809	18	0.975
14	25	3	9	2	4	10	900	13014020	13014920	126564	2.0×10^5	463	18	0.975
15	30	3	9	2	4	10	900	15616820	15617720	151794	4.9×10^5	1114	18	0.975
16	40	3	12	2	5	10	1440	57744020	57745460	336784	3.8×10^5	871	24	0.975
17	20	3	16	2	6	12	2688	106306584	106309272	323528	10.9×10^4	248	32	0.975
18	15	3	16	3	6	12	4224	119594916	119599140	364479	12.2×10^4	279	48	0.975
19	15	4	16	2	6	15	3360	186796830	186800190	433166	1.7×10^5	389	32	0.975
20	20	3	16	2	8	12	3456	188928024	188931480	431056	13.7×10^4	314	32	0.975
21	15	3	8	3	4	8	1024	5932824	5933848	81295	2.7×10^5	620	24	0.975
22	25	3	9	2	4	10	900	13014020	13014920	126564	2.9×10^5	671	18	0.975

Table 10
Comparison of solutions obtained by SAeLP, GAeLP, Kia et al. (2012) and B&B.

No	B&B (10^5 s)	Kia et al. (2012)	Objective function value (OFV)										Time (s)		
			SAeLP					GAeLP							
						Improvement (%)					Improvement (%)				
			Best	Worst	Average	Kia et al. (2012)	B&B	Best	Worst	Average	Kia et al. (2012)	B&B	Kia et al. (2012)	SAeLP (avg.)	GAeLP (avg.)
1	67608^{OPT}	67705.43	67608	67608	67608	0.14	0.00	67608	67608	67608	0.14	0.00	57	2.85	14.11
2	103434^{OPT}	104273.40	103434	103434	103434	0.80	0.00	103434	103434	103434	0.80	0.00	179	7.12	30.11
3	148843.8^{OPT}	153772.66	148843.8	148843.8	148843.8	3.21	0.00	148843.8	148843.8	148843.8	3.21	0.00	1033	16.89	57.11
4	190399.3^{OPT}	202016.58	190409.4	191452.9	191083.9	5.75	−0.01	190409.4	194303.6	191530.1	5.75	−0.01	1036	32.67	136.40
5	284905.2	319125.53	288806.1	301164.9	295142.4	9.50	−1.37	287911.8	290986.3	289738.2	9.78	−1.06	3270	324.89	794.62
6	205117.4	239174.64	205133.1	211351	208117.3	14.23	−0.01	205117.4	211611.2	207907.1	14.24	0.00	2186	62.73	290.52
7	271360.3	336417.29	275258.2	281036.4	277760.5	18.18	−1.44	275258.2	277773.6	276101.9	18.18	−1.44	3706	151.93	585.76
8	341606.2	428877.71	341980.3	353699	349710.4	20.26	−0.11	347523.4	363752.5	351188	18.97	−1.73	5581	846.23	2623.99
9	407068.9	483149.76	378553.8	385837	381460.5	21.65	7.00	380822.2	411036.8	395998.9	21.18	6.45	10013	962.74	2861.08
10	474810.5	–	469141	488243.8	477797.7		1.19	476927.1	508241.6	491397.6	–	−0.45	–	2347.63	8788.67
11	–	–	379892.9	388076.3	383595.6		–	381998.3	403619.9	389178	–	–	–	3586.74	13168.83
12	423441	–	413861.5	419593.5	418026		2.26	412755.6	427228.5	419949.5	–	2.52	–	850.24	2742.92
13	586390.5	–	574457	590143	580642.2		2.04	574457	606517	590676.6	–	2.04	–	1320.72	5952.57
14	487464.4	–	477167	478252.4	477801		2.11	478964	485625	481793.2	–	1.74	–	1013.94	6078.54
15	587246	–	565887.7	583989	575733.8		3.64	566050.2	594196	581770.5	–	3.61	–	1352.90	6224.57
16	–	–	684599.3	687213.2	686116.6		–	673591.3	695834.4	682263.3	–	–	–	6686.82	33645.97
17	–	–	456537.1	463566.6	458352		–	456819.9	491323	470879.5	–	–	–	5625.74	18533.43
18	–	–	486322.2	498785	492467.1		–	486322.2	526871.7	503706.4	–	–	–	10385.00	28432.59
19	–	–	462504.5	469169.2	465474.4		–	470964.5	511801.9	490177.9	–	–	–	6585.13	25950.24
20	–	–	433782	475377.3	443452.4		–	439758.2	465688.9	453556.6	–	–	–	5493.33	19163.39
21	392462.5	–	392462.5	403567.9	395265.2		0.00	392462.5	423753.5	406283.2	–	0.00	–	871.42	3585.33
22	482446.1	–	473515.7	479492.6	477934		1.85	476125.4	481454.2	477683.9	–	1.31	–	1031.50	5408.03

Subject to:

$$\sum_{p\in P}\sum_{r\in R} f_{prcit}\pi_{pri} \leqslant C_i X_{cit} \quad \forall\ c\in N, i\in M, t\in T \tag{42}$$

$$\sum_{c\in N}\sum_{i\in M} f_{prcit} = D_{pt} \quad \forall\ p\in P, r\in R, t\in T \tag{43}$$

$$Q_{i,1} = \sum_{c\in N} X_{ci,1} \quad \forall\ i\in M \tag{44}$$

$$Q_{i,t+1} = \sum_{c\in N} X_{ci,t+1} - \sum_{c\in N} X_{cit} \quad \forall\ i\in M, t\in\{1,\ldots,T-1\} \tag{45}$$

$$\sum_{c\in N}\sum_{i\in M} F_{prcdijt} = f_{p,r+1,djt} \quad \forall\ p\in M, r\in\{1,\ldots,R-1\}, d\in N, j\in M, t\in T \tag{46}$$

$$\sum_{d\in N}\sum_{j\in M} F_{prcdijt} = f_{prcit} \quad \forall\ p\in M, r\in\{1,\ldots,R-1\}, c\in N, i\in M, t\in T \tag{47}$$

$$f_{prcit} \leqslant D_{pt}\alpha_{pri} \quad \forall\ p\in M, r\in R, c\in N, i\in M, t\in T \tag{48}$$

$$\sum_{c\in N}\sum_{i\in M} X_{cit} \leqslant K \quad \forall\ t\in T \tag{49}$$

$$\sum_{i\in M} X_{cit} \leqslant B^u \quad \forall\ t\in T, c\in N \tag{50}$$

$$\sum_{i\in M} X_{cit} \geqslant B^l \quad \forall\ t\in T, c\in N \tag{51}$$

$$X_{cit} \geqslant 0,$$
$$Q_{i,t} \geqslant 0,$$
$$F_{prcdijt} \geqslant 0,$$
$$f_{prcit} \geqslant 0.$$

The mathematical model of the lower bound is actually a simplified form of the original mathematical model. This model does not consider machine relocations and cost of intra-cell material handling among the machines of the same type. In addition, it assumes that costs of any type material handling movement (either inter-cell or intra-cell) is calculated as if it was transferred between the closest locations (λ^*) on the shop floor.

As the binary variables increase the solution time and the complexity of a mathematical model, removing of such variables is useful when developing a lower bound mathematical model. The model determines the integer number of machines to be assigned to each cell (X_{cit}), rather than determining their exact locations in cells and on the shop floor. Similarly, in order to reduce the complexity of the problem, flow and routing variables ($F_{prcdijt}$ and f_{prcit}) are assumed to be continuous. Almost all parameters and the terms of the mathematical model of the lower bound are similar to those of the original mathematical model. Since these terms are explained in the previous sections, in order to avoid repetition, the mathematical model of the lower bound will not be discussed in detail.

The mathematical model of the lower bound was solved to optimality by using GAMS/Cplex software. The comparison of the results are given in Table 11. As it can be inferred from Table 11, solutions found by SAeLP and GAeLP for the original problem are quite close or equal to the optimal solutions of the original problem, since $\mathrm{OFV_{LB}} \leqslant \mathrm{OFV_{OPT}} \leqslant \mathrm{OFV_{SAeLP}}$ and $\mathrm{OFV_{LB}} \leqslant \mathrm{OFV_{OPT}} \leqslant \mathrm{OFV_{GAeLP}}$. Therefore, the effectiveness of the proposed approaches to find solutions that are closer to the optimal solutions is shown.

Table 11
Comparison of solutions found by using SAeLP, GAeLP and the lower bound.

Sample no.	Lower bound for objective function value	Gap (%)	
		SAeLP	GAeLP
1	67578.5	0.04	0.04
2	103434.0	0.00	0.00
3	148575.9	0.18	0.18
4	190199.3	0.11	0.11
5	278481.5	3.57	3.28
6	204483.7	0.32	0.31
7	271104.0	1.51	1.51
8	334865.5	2.08	3.64
9	373642.3	1.30	1.89
10	464322.5	1.03	2.64
11	369832.6	2.65	3.18
12	399951.5	3.36	3.10
13	571460.5	0.52	0.52
14	476378.1	0.17	0.54
15	559300.1	1.16	1.19
16	665792.0	2.75	1.16
17	456312.8	0.05	0.11
18	486318.4	0.00	0.00
19	461797.9	0.15	1.95
20	423163.2	2.45	3.77
21	378447.0	3.57	3.57
22	472924.0	0.12	0.67

7. Conclusion

Dynamic cellular manufacturing system design is an NP hard problem. In this study, we present a new mixed-integer programming model for the problem, with numerous design attributes including inter-cellular layout, intra-cellular layout, cell formation, alternative process routings, lot splitting, duplicated machines, operation sequence, processing time, machine capacity and reconfigurable layout. The model shares common properties with Kia et al. (2012)'s study except the assumption that the installation and uninstallation costs of a machine type are equal. In the mathematical model proposed in this study, installation and uninstallation costs are allowed to be unequal. Moreover, when the linearized versions of these models are compared, it is obvious that the proposed model has less constraints and variables. This is an advantage of the proposed model in terms of solution time.

Due to the complexity of the problem, two Linear Programming based meta-heuristic approaches (SAeLP and GAeLP) were developed. In these approaches, Linear Programming is embedded into Simulated Annealing and Genetic Algorithm with problem specific solution representation and neighborhood mechanisms. The developed approaches were tested by using some test data taken from the literature (Kia et al., 2012) and randomly generated larger-size test problems. In addition, we have also developed a lower bound for the original problem solved in this study. Comparing the results of hybrid meta-heuristic approaches with that of the lower bound, the closeness of the found results to optimality was shown. It is found out that our solution approaches are capable of finding near-optimal solutions in reasonable times.

For the future, it is recommended to incorporate some other design features of CMS design into the model. Some interesting features that can be incorporated are time value of the money, inventory holding and outsourcing. In addition, integration of other meta-heuristics with LP can also be taken into account. In reality, demand forecasts are mostly inaccurate and unreliable. Therefore, introducing uncertainty along with the dynamic conditions in CMS design would also be an important contribution to the CMS design literature.

References

Adil, G. K., Rajamani, D., & Strong, D. (1996). Cell formation considering alternate routeings. *International Journal of Production Research, 34*(5), 1361–1380.

Ahkioon, S., Bulgak, A. A., & Bektas, T. (2009). Integrated cellular manufacturing systems design with production planning and dynamic system reconfiguration. *European Journal of Operational Research, 192*(2), 414–428.

Alfa, A. S., Chen, M., & Heragu, S. S. (1992). Integrating the grouping and layout problems in cellular manufacturing systems. *Computers & Industrial Engineering, 23*(1), 55–58.

Aryanezhad, M. B., Deljoo, V., & Mirzapour Al-e-hashem, S. M. J. (2009). Dynamic cell formation and the worker assignment problem: A new model. *The International Journal of Advanced Manufacturing Technology, 41*(3–4), 329–342.

Askin, R. G., Selim, H. M., & Vakharia, A. J. (1997). A methodology for designing flexible cellular manufacturing systems. *IIE Transactions, 29*(7), 599–610.

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.

Baykasoğlu, A. (2004). A meta-heuristic algorithm to solve quadratic assignment formulations of cell formation problems without presetting number of cells. *Journal of Intelligent Manufacturing, 15*(6), 753–759.

Black, J. T. (1983). Cellular manufacturing systems reduce setup time, make small lot production economical. *Industrial Engineering, 15*(11), 36–48.

Cao, D., & Chen, M. (2005). A robust cell formation approach for varying product demands. *International Journal of Production Research, 43*(8), 1587–1605.

Chang, C. C., Wu, T. H., & Wu, C. W. (2013). An efficient approach to determine cell formation, cell layout and intracellular machine sequence in cellular manufacturing systems. *Computers & Industrial Engineering, 66*(2), 438–450.

Chen, M. (1998). A mathematical programming model for system reconfiguration in a dynamic cellular manufacturing environment. *Annals of Operations Research, 77*, 109–128.

Chen, M., & Cao, D. (2004). Coordinating production planning in cellular manufacturing environment using Tabu search. *Computers & Industrial Engineering, 46*(3), 571–588.

Cheng, C. H., Goh, C. H., & Lee, A. (1996). Solving the generalized machine assignment problem in group technology. *Journal of the Operational Research Society*, 794–802.

Defersha, F. M., & Chen, M. (2006). A comprehensive mathematical model for the design of cellular manufacturing systems. *International Journal of Production Economics, 103*(2), 767–783.

Defersha, F. M., & Chen, M. (2008a). A parallel genetic algorithm for dynamic cell formation in cellular manufacturing systems. *International Journal of Production Research, 46*(22), 6389–6413.

Defersha, F. M., & Chen, M. (2008b). A linear programming embedded genetic algorithm for an integrated cell formation and lot sizing considering product quality. *European Journal of Operational Research, 187*(1), 46–69.

Egilmez, G., Süer, G. A., & Huang, J. (2012). Stochastic cellular manufacturing system design subject to maximum acceptable risk level. *Computers & Industrial Engineering, 63*(4), 842–854.

Gupta, R. M., & Tompkins, J. A. (1982). An examination of the dynamic behaviour of part-families in group technology. *The International Journal of Production Research, 20*(1), 73–86.

Gupta, T., & Seifoddini, H. I. (1990). Production data based similarity coefficient for machine-component grouping decisions in the design of a cellular manufacturing system. *The International Journal of Production Research, 28*(7), 1247–1269.

Harhalakis, G., Nagi, R., & Proth, J. M. (1990). An efficient heuristic in manufacturing cell formation for group technology applications. *The International Journal of Production Research, 28*(1), 185–198.

Harhalakis, G., Ioannou, G., Minis, I., & Nagi, R. (1994). Manufacturing cell formation under random product demand. *International Journal of Production Research, 32*(1), 47–64.

Heragu, S. S. (1994). Group technology and cellular manufacturing. *IEEE Transactions on Systems, Man and Cybernetics, 24*(2), 203–215.

Khaksar-Haghani, F., Kia, R., Mahdavi, I., & Kazemi, M. (2013). A genetic algorithm for solving a multi-floor layout design model of a cellular manufacturing system with alternative process routings and flexible configuration. *The International Journal of Advanced Manufacturing Technology, 66*(5–8), 845–865.

Kia, R., Baboli, A., Javadian, N., Tavakkoli-Moghaddam, R., Kazemi, M., & Khorrami, J. (2012). Solving a group layout design model of a dynamic cellular manufacturing system with alternative process routings, lot splitting and flexible reconfiguration by simulated annealing. *Computers & Operations Research, 39*(11), 2642–2658.

Logendran, R., Ramakrjshna, P., & Sriskandarajah, C. (1994). Tabu search-based heuristics for cellular manufacturing systems in the presence of alternative process plans. *International Journal of Production Research, 32*(2), 273–297.

Logendran, R., & Ramakrishna, P. (1995). Manufacturing cell formation in the presence of lot splitting and multiple units of the same machine. *The International Journal of Production Research, 33*(3), 675–693.

Mahdavi, I., Aalaei, A., Paydar, M. M., & Solimanpur, M. (2012). A new mathematical model for integrating all incidence matrices in multi-dimensional cellular manufacturing system. *Journal of Manufacturing Systems, 31*(2), 214–223.

Mahdavi, I., Teymourian, E., Baher, N. T., & Kayvanfar, V. (2013). An integrated model for solving cell formation and cell layout problem simultaneously considering new situations. *Journal of Manufacturing Systems, 32*(4), 655–663.

Mak, K. L., Wong, Y. S., & Wang, X. X. (2000). An adaptive genetic algorithm for manufacturing cell formation. *The International Journal of Advanced Manufacturing Technology, 16*(7), 491–497.

Marsh, R. F., Meredith, J. R., & McCutcheon, D. M. (1997). The life cycle of manufacturing cells. *International Journal of Operations & Production Management, 17*(12), 1167–1182.

Meller, R. D., & Bozer, Y. A. (1996). A new simulated annealing algorithm for the facility layout problem. *International Journal of Production Research, 34*(6), 1675–1692.

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.

Montreuil, B. (1999). Fractal layout organization for job shop environments. *International Journal of Production Research, 37*(3), 501–521.

Nsakanda, A. L., Diaby, M., & Price, W. L. (2006). Hybrid genetic approach for solving large-scale capacitated cell formation problems with multiple routings. *European Journal of Operational Research, 171*(3), 1051–1070.

Onwubolu, G. C., & Mutingi, M. (2001). A genetic algorithm approach to cellular manufacturing systems. *Computers & Industrial Engineering, 39*(1), 125–144.

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.

Raidl, G. R., & Puchinger, J. (2008). Combining (integer) linear programming techniques and metaheuristics for combinatorial optimization. In C. Blum, M. J. Blesa Aguilera, A. Roli, & M. Sampels (Eds.), *Hybrid metaheuristics, an emerging approach to optimization. Series: Studies in computational intelligence* (Vol. 114, pp. 31–62).

Rezazadeh, H., Mahini, R., & Zarei, M. (2011). Solving a dynamic virtual cell formation problem by linear programming embedded particle swarm optimization algorithm. *Applied Soft Computing, 11*(3), 3160–3169.

Rheault, M., Drolet, J. R., & Abdulnour, G. (1995). Physically reconfigurable virtual cells: A dynamic model for a highly dynamic environment. *Computers & Industrial Engineering, 29*(1), 221–225.

Rutenbar, R. A. (1989). Simulated annealing algorithms: An overview. *Circuits and Devices Magazine IEEE, 5*(1), 19–26.

Sahni, S., & Gonzalez, T. (1976). P-complete approximation problems. *Journal of the ACM, 23*(3), 555–565.

Sarker, B. R., & Xu, Y. (2000). Designing multi-product lines: Job routing in cellular manufacturing systems. *IIE Transactions, 32*(3), 219–235.

Saxena, L. K., & Jain, P. K. (2011). Dynamic cellular manufacturing systems design—A comprehensive model. *The International Journal of Advanced Manufacturing Technology, 53*(1–4), 11–34.

Saxena, L. K., & Jain, P. K. (2012). An integrated model of dynamic cellular manufacturing and supply chain system design. *The International Journal of Advanced Manufacturing Technology, 62*(1–4), 385–404.

Solo, A. M. (2010). Multidimensional matrix mathematics: Notation, representation, and simplification, Part 1 of 6. In *Proceedings of the world congress on engineering* (Vol. 3).

Tavakkoli-Moghaddam, R., Aryanezhad, M. B., Safaei, N., & Azaron, A. (2005a). Solving a dynamic cell formation problem using metaheuristics. *Applied Mathematics and Computation, 170*(2), 761–780.

Tavakkoli-Moghaddam, R., Safaei, N., & Babakhani, M. (2005b). Solving a dynamic cell formation problem with machine cost and alternative process plan by memetic algorithms. In O. B. Lupanov, O. M. Kasim-Zade, A. V. Chaskin, & K. Steinhofel (Eds.). *Lecture notes in computer science* (vol. 3777, pp. 213–227). Berlin: Springer.

Urban, T. L., Chiang, W. C., & Russell, R. A. (2000). The integrated machine allocation and layout problem. *International Journal of Production Research, 38*(13), 2911–2930.

Venkatadri, U., Rardin, R. L., & Montreuil, B. (1997). A design methodology for fractal layout organization. *IIE Transactions, 29*(10), 911–924.

Wang, S., & Sarker, B. R. (2002). Locating cells with bottleneck machines in cellular manufacturing systems. *International Journal of Production Research, 40*(2), 403–424.

Wemmerlöv, U., & Hyer, N. L. (1986). Procedures for the part family/machine group identification problem in cellular manufacturing. *Journal of Operations Management, 6*(2), 125–147.

Wicks, E. M., & Reasor, R. J. (1999). Designing cellular manufacturing systems with dynamic part populations. *IIE Transactions, 31*(1), 11–20.

Wilhelm, W. E., Chiou, C. C., & Chang, D. B. (1998). Integrating design and planning considerations in cellular manufacturing. *Annals of Operations Research, 77*, 97–107.

Winston, W., & Goldberg, J. B. (2003). *Operations research: Applications and algorithms* (4th ed.). USA: Thomson Brooks/Cole.

Zeidi, J. R., Javadian, N., Tavakkoli-Moghaddam, R., & Jolai, F. (2013). A hybrid multi-objective approach based on the genetic algorithm and neural network to design an incremental cellular manufacturing system. *Computers & Industrial Engineering, 66*(4), 1004–1014.