Decision Science Letters 5 (2016) 219–244

Contents lists available at GrowingScience

Decision Science Letters

homepage: www.GrowingScience.com/dsl

Evaluating impact of market changes on increasing cell-load variation in dynamic cellular manufacturing systems using a hybrid Tabu search and simulated annealing algorithms

Aidin Delgoshaei*, Masih Parvin and Mohd Khairol Anuar Ariffin

Department of Mechanical and Manufacturing Engineering, University Putra Malaysia, 43400 UPM, Serdang, Kuala Lumpur, Malaysia

C H R O N I C L E

Article history:
Received June 25, 2015
Received in revised format:
October 12, 2015
Accepted December 13, 2015
Available online
December 14 2015

Keywords:
Facilities planning
Cell Scheduling
Part Programming

A B S T R A C T

In this paper, a new method is proposed for scheduling dynamic cellular manufacturing systems (D-CMS) in the presence of uncertain product demands. The aim of this method is to control the process of trading off between in-house manufacturing and outsourcing while product demands are uncertain and can be varied from period to period. To solve the proposed problem, a hybrid Tabu Search and Simulated Annealing are developed to overcome hardness of the proposed model and then results are compared with a Branch and Bound and Simulated Annealing algorithms. A Taguchi method (L_27 orthogonal optimization) is used to estimate parameters of the proposed method in order to solve experiments derived from literature. An in-depth analysis is conducted on the results in consideration of various factors. For evaluating the system imbalance in dynamic market demands, a new measuring index is developed. Our findings indicate that the uncertain condition of market demands affects the routing of product parts and may induce machine-load variations that yield to cell-load diversity. The results showed that the proposed hybrid method can provide solutions with better quality.

1. Introduction

Cellular manufacturing systems (CMS) is considered as effective way of using group technology by defining manufacturing system as a hybrid system of cells linking the advantages of both the jobbing (flexibility) and mass (efficient flow and high production rate) production approaches (Papaioannou & Wilson, 2010). In CMS studies, two main work-in-process (WIP) movements can be recognized. Intra-cellular WIP transferring involves transferring materials among machines that are located in a cell. By contrast, in inter-cellular movements, materials are planned to shift between cells to perform some operations. Although intracellular WIP transferring may seem more desirable because of the lesser transfer cost involved than that for intercellular movements, intercellular WIP movements are

* Corresponding author. Tel/Fax: + (60) 1123091810
E-mail address: delgoshaei.aidin@gmail.com (A. Delgoshaei)

doi: 10.5267/j.dsl.2015.12.002

unavoidable. In the following subsections, some of the most important reasons for the increase in material transferring costs and the solutions provided by scientists will be investigated in detail.

Determining the best combination of machines that can be used in the consecutive operations of a part (during or after cell generation) is the aim of addressing part routing problems. From another perspective in part routing problems, each part can be completed in more than one way because of the existence of parallel machines. Choosing different permutations of various machines inevitably causes different inter and intracellular movements and entails material transferring costs accordingly.

To minimize material transferring costs, scientists have chosen one or a combination of two strategies. The first is assigning machines to generate part families. The other is assigning parts to machines that are grouped according to function similarity or requirements. Gupta et al. (1995) focused on minimizing intercellular movements and maximizing machine utilization simultaneously during the process of grouping machines in cells. Heragu (1989) developed a three-stage knowledge-based approach for cell forming. In the first stage, a clustering algorithm was used to make part families and machine groups. Then, with a distance-based model, the suitable locations for grouped machines were estimated. The third stage enabled the authors to finalize the layout of the machine cells in terms of WIP transferring paths, shape, and size of the machine cells. Gravel et al. (1998) employed a double-loop GA for simultaneous machine grouping and the part assigning problem in which an outer loop assigns machines to cells, and an inner loop determines the optimal part routing. Yu & Sarker (2006) proposed a quadratic assignment problem model to minimize the total intercellular flows by considering bottleneck parts in which the output of other group formation methods can be used as an input of their method. Goncalves Filho and José Tiberti (2006) proposed a GA to minimize number of inter-cellular material transferring and cell load variations. Afterward, Haleh et al. (2009) presented a hybrid Memetic algorithm and revised TOPSIS method to minimize cell load variation and inter-cellular WIP transferring. Nsakanda et al. (2006) presented a multi-process plan problem to determine the best part routing of each process plan. They also used outsourcing as a useful strategy to meet the rest of the part demands.

In many cases, the irrelevant location of machines has been observed to increase material transferring costs. To solve this problem, Wang and Sarker (2002) proposed a QAP method for a machine cell location problem. Later, Sarker and Yu (2007) focused on the spatial coordinates in a machine-cell location problem in which bottleneck machines and parts may exist. They also used QAP method to minimize intercellular WIP movements in which vertical and horizontal directions for material transferring were allowed.

Another drawback that emerges during the part routing process is the underutilization of machines in cells. Seifoddini and Djassemi (1993) argued that underutilization in CMS models can cause long queues in some machines, whereas other parallel machines remain idle. To solve this shortcoming, the authors used the dynamic part assignment method (DPA). Later, Seifoddini and Djassemi (1996) used DPA for the re-routing of parts among machines to improve machine utilization. The results show that the proposed DPA improves system performance by reducing the congestion in machines with long waiting lines. Gonçalves and Resende (2004) applied a hybrid local search and GA to minimize intercellular movements and maximize the utilization of machines within a cell.

One reason for the emerging underutilization of some machines or cell load variation is ignoring the capacity of machines during the scheduling process. Grznar et al. (1994) proposed a method to minimize material movement costs in a multi-part routing process problem while material movements and machine capacity are restricted. Zhou and Askin (1998) also argued that machine capacity and budget constraints should not be ignored while part routings are determined. Moussa and Kamel (1998) considered the machine capacity in the process of part-machine assignment to minimize intercellular movements. Logendran and Talkington (1997) developed a simulating model to compare cell and flow

shop layouts based on preventive maintenance and emergency reparation and batch size. Then, Ahkioon et al. (2009) presented a CMS where maintenance activities can affect system capacity. To overcome such drawback, they also considered using outsource services and machine relocating. But their model was failed to solve large scale problems.

In some cases, cell size constraints were employed to control the addition or removal of machines in cells and to determine the best values of transferring materials within or between cells accordingly. Onwubolu and Mutingi (2001) also considered the lower and upper bounds for cell sizes to control material transferring and provide more flexible schemes. Boulif and Atif (2006) considered a situation in which cohabitation and non-cohabitation machines exist. In actual practice, such a condition may indeed happen because of safety reasons. They proposed a new approach to minimize WIP transferring between pairwise machines while the maximum number of cells is considered aside from the maximum sizes of these cells. Ariafar and Ismail (2009) addressed the problem of material transferring while the shape of the machines was assumed equal. A center-based distance calculation between machines was employed.

Aside from the strategies and solutions discussed, during the last two decades, the idea of reconfiguring cells by shifting of machines or re-arranging of part families to smooth material transferring within and between cells (or as a result of changing part demands during periods) has become noteworthy. Almost in all cases, the focus is on the modification of cell layouts during the manufacturing process to determine the best part routing. Harhalakis et al. (1990) developed a two-step heuristic to minimize intercellular movements. In the first step, they used aggregate planning concepts to form cells with minimum intercellular movements. In the second step, they applied another procedure to determine and shift essential machines that must be relocated to achieve improved results. Safaei et al. (2008) investigated the issue of machine time capacity and maximum cell size in a reconfigurable dynamic CMS in which machine relocating was allowed. A few years later, Rafiee et al. (2011) developed a similar mathematical programing method for a reconfigurable manufacturing system in which more aspects of a real system, such as preventive and maintenance activities, finished and unfinished parts inventory, and defective parts replacement costs, were considered. Elmi et al. (2011) developed a situation that was previously proposed by Won and Currie (2007) in which some parts need to visit a machine more than once in a non-consecutive manner (re-entrant parts). In the proposed model, they presented a new method to schedule both bottleneck and re-entrant parts in cells. Paydar et al. (2010) considered operation sequences in a multiple travelling salesman formulation for the cell forming and machine locating problem, with multiple departures and a single destination considered.

In most real cases, part demands are different from one planning horizon to another. Such a criterion is known as dynamic part demand. Market changes, changes in product designs, and the manufacture of new products are some of the reasons for the change in part demands through different time periods. These conditions may cause emerging imbalances in part routings and bottleneck machines. They will be explained as a separate section because of their importance.

Wang et al. (2001) argued that dynamic demands can increase the complexity of such models. Therefore, they applied simulated annealing algorithm to solve the problems involved. Tavakkoli-Moghaddam, Safaei, et al. (2005) employed triangular fuzzy numbers to estimate uncertain demands of each part type. For this purpose, a fuzzy nonlinear mixed integer programming method was developed with the aim of minimizing constant machine, intercellular WIP transferring, and reconfiguration costs. Balakrishnan and Cheng (2005) proposed a two-stage procedure to minimize material handling and machine relocation costs in the midst of part uncertainties. In the same year, Tavakkoli-Moghaddam et al. (2005) minimized material transferring costs in the dynamic condition of part demands by using alternative process plans and machine relocation and replications. Defersha and Chen (2006) used parallel machines and outsource services to overcome dynamic part demand defects in cell forming process. Jeon and Leep (2006) presented a model for scheduling dynamic cells where

machine failures can cause waiting times and reduce system capacity accordingly. Tavakkoli-Moghaddam et al. (2007) considered dynamic part demands and parts mixed for a reconfigurable part routing problem; minimizing operating (constant and variable), machine relocating, and intercellular WIP transferring costs was considered as the objective of the proposed model. Tavakkoli-Moghaddam, Javadian, et al. (2007) considered the normal distribution function to estimate the part demands in a stochastic model; minimizing material transferring movements was the main objective of the method. Safaei and Tavakkoli-Moghaddam (2009a) also argued that machine capacity and part demands should not be considered fixed and showed how such uncertainties can influence the cell configuration through time horizon. During the scheduling of a dynamic manufacturing system, the system capacity may be inadequate to meet customer demand at a specific period. Hence, Safaei and Tavakkoli-Moghaddam (2009b) addressed a dynamic scheduling problem to find the tradeoff values between in-house production and outsourcing while cells are supposed to be reconfigurable. This time, they considered intercellular movements in addition to intracellular ones.

The other solution to address part uncertainties is forming new cells as a result of market changes. This strategy was discussed by Zhang (2011). Aggregate planning while minimizing operation, inventory, and material movement costs was used. A few years later, Egilmez & Süer (2014) evaluated the impact of risk level in an integrated cell forming and scheduling problem using Monte Carlo Simulation. Ariafar et al. (2014) focused on the impact of dynamic product demand on facility layout problem. The main objective of the proposed model was minimizing material transferring by arranging the machine cells within the shop-floor, and the machines within each of the machine cells. Afterward, Renna and Ambrico (2015) also proposed three models for designing, reconfiguring and scheduling cells in dynamic condition of product demands. In their models, they considered minimizing system costs including intercellular movements, machining and reconfiguring costs as well as maximizing net-profit. This section presented a review of the issue on material transferring during the part routing process in CMS. The most significant drawbacks that emerge through cell forming or scheduling are illustrated, and the proposed solutions in various situations are investigated. The literature review shows that when the part routing issue emerges, considering multi process plans, machine relocating, cell decomposition, and cell reconfiguring are the most common solutions offered by scientists. However, some gaps are also found in this area. An in depth survey in literature of Material Transferring and System Capacity in CMS studies shows that the issue trading off between in-house manufacturing and outsourcing while part demands and machine availability are considered uncertain is less developed.

2. Research Methodology

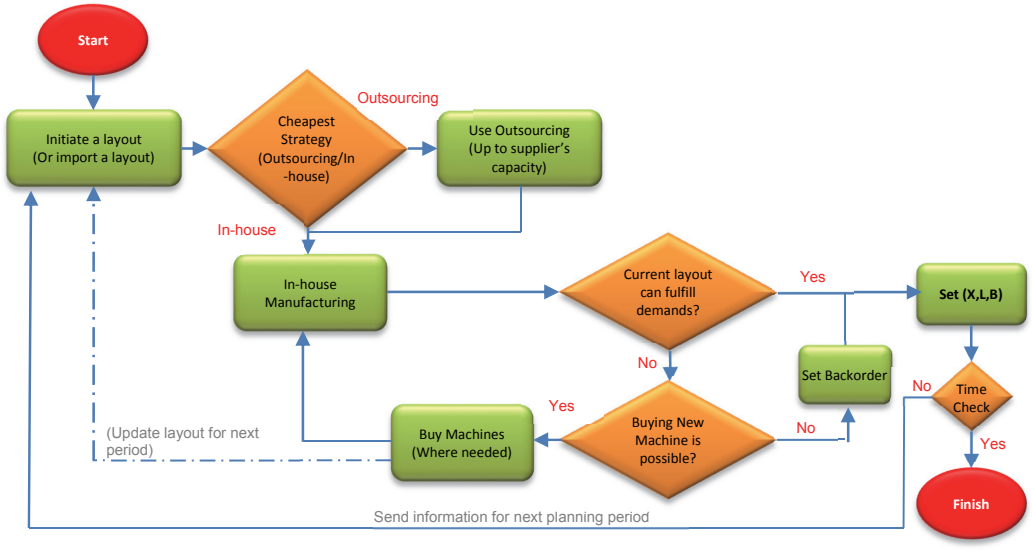
This model is prepared to determine best trading off values between in-house manufacturing and outsourcing over planning horizon while limited backorders are allowed and product demands are considered uncertain. The proposed model can be applied for both job shop problems and cellular manufacturing systems. During formulating the model, all system costs including group setup, operating, machine purchasing, outsourcing and backorder costs are taken into consideration. Product demands are also considered dynamic and may be vary from time to time.

2.1 Features and novelties of the proposed model

The features of the proposed model can be listed as:
1) Determining the tradeoff values between in-house manufacturing and outsourcing.
2) Considering part sequences using operation process charts of activities (OPC)
3) Considering machine capacity
4) Considering both intra-cellular and inter-cellular material transferring during manufacturing process
5) Considering dynamic part demands and using batch sizes for transferring parts
6) Using subcontractor's services (being able to use both restricted and unrestricted subcontractors services (outsourcing))
7) Considering geometrical distance between machines

8) Considering cell sizes
9) Considering machine purchasing ability
10) The model can solve in 2 modes of allowed or restrict backorders.

Novelties of the proposed model can also summarize as:
1) A new method for calculating inter- & intra-cellular material transferring while bottleneck machines, parallel machines, voids and exceptional elements are existed.
2) A new way to find best trading off values between in-house manufacturing, outsourcing and backorders.
3) Periodically (daily) scheduling plans for each machine as an outcome (note that all previous models could only determine the total volume of part assigning to each machine but the best part routes one by one).
Fig. 1 show the flowchart of the proposed method.



X: in-house manufacturing; *L* is outsource services and *B* shows Backorders

Fig. 1. Flowchart for finding the best trade off values between in-house manufacturing, outsourcing and backorder while machine purchasing is allowed

2.1 Mathematical model

In this section, a non-linear mixed integer programming NL-MIP (method) is developed to survey how in-process materials change their production routes in the presence of dynamic part demands and how such entropies can influence the system performance.
As a brief the advantage of the proposed model is highlighted as: considering uncertainty in product demands, inter cellular and intra cellular movements with the constant batch size, existence of parallel machines, alternative process routes for part types considering operation sequence. A number of assumptions are used to formulate the mathematical model.
1) Product types have number of operations must be performed based on manufacturing priorities.
2) The lower bound and upper bound for each cell is known in advance.
3) Machine purchasing and machine relocating are allowed.
4) Using subcontractor services is allowed.
5) The demands for part types in each scheduling period are not fixed and will be expressed by normal distribution function.
6) The machine capacities must be considered while scheduling.

7) Backorders are allowed but restricted. The beginning inventory is considered zero and last period backorder is not allowed.

Inputs:

$i: number\ of\ parts$
$j: type\ of\ machines$
$k: number\ of\ cells$
$l: number\ of\ sub-contractors$
$t: planning\ periods\ (time\ slots)$

Parameters:

$d_{i,t}: demand\ of\ part\ i\ in\ period\ t$

$$d_{i,t} \sim N(\mu_{i,t}, \sigma_{i,t}) \quad (1)$$

$S_j: Setup\ cost\ of\ machine\ j$
$OP_{i,j}: cost\ of\ processing\ part\ i\ using\ machine\ j$
$OS_l: cost\ of\ performing\ one\ part\ by\ subcontractor\ l$
$BC_i: backorer\ cost\ of\ part\ i\ for\ a\ period$
$\alpha_i: intracellular\ movement\ cost\ of\ part\ i$
$\beta_i: interrcellular\ movement\ cost\ of\ part\ i$
$K_j^+: purchasing\ cost\ of\ mchine\ j$
$K_j^-: selling\ cost\ of\ mchine\ j$
$length: is\ a\ lenth\ of\ the\ cell$
$width: is\ a\ width\ of\ the\ cells$
$Cl: is\ cell\ size\ and\ refers\ to\ maximum\ number\ of\ machines\ can\ be\ located\ in\ a\ cell$

$$CL = length \times width \quad (2)$$

Input Matrixes:

Product demand ($D_{i,t}\ [\because]_{i,t}$)
Batch size ($BS_i\ [\because]_i$)
Machine Component Incidence Matrix ($MCIM_{i,j}\ [\because]_{i,j}$)
Machine capacity ($MA_j\ [\because]_j$)
Sub-contractor capability ($SC_l\ [\because]_l$)
Allowed backorder ($AB_{i,t}\ [\because]_{i,t}$)
Initial number of machines ($NOM_j\ [\because]_j$)
Operation Cost ($OP_i\ [\because]_i$)
Setup Cost ($S_j\ [\because]_j$)
Intercellular Cost ($\alpha_i\ [\because]_i$)
Intracellular Cost ($\beta_i\ [\because]_i$)
Machine purchasing Cost ($K_j\ [\because]_j$)
Backorder Cost ($BC_i\ [\because]_i$)
Outsourcing Cost ($OS_l\ [\because]_l$)

Variables:

$X_{i,f,j,k,t}: number\ of\ part\ i\ which\ is\ performing\ by\ fth\ of\ machine\ type\ j\ in\ cell\ k\ in\ period\ t\ (int.)$

$Z_{i,f,j,k,t}$ *if part i performs using fth of machine type j in cell k in period t* $(bin.)$
$Y_{l,t}$: *number of parts which manufactured by sub − contractor l in period t* $(int.)$
$B_{i,t}$: *number of part i which is decided to postpone for next period* $(int.)$
$N^{+}{}_{j,k,t}$: *number of machine type j added to cell k during period t* $(int.)$
$N^{-}{}_{j,k,t}$: *number of machine type j removed from cell k during period t* $(int.)$

Mathematical Model:

$$Min: \sum_{t}^{T}\sum_{k}^{K}\sum_{j}^{J}\sum_{f}^{F}\sum_{i}^{I} S_j.Z_{ifjkt}.(X_{ifjkt}/BS_i) + \sum_{t}^{T}\sum_{k}^{K}\sum_{j}^{J}\sum_{f}^{F}\sum_{i}^{I} MCIM_{i,j}.OP_{i,j}.X_{ifjkt}.Z_{ifjkt} \quad (3)$$

$$+ \sum_{t}^{T}\sum_{k}^{K}\sum_{j}^{J}[(K_j{}^{+}.N^{+}_{j,k,t}) - (K_j{}^{-}.N^{-}_{j,k,t})] + \sum_{t}^{T}\sum_{l}^{L} OS_l.Y_{l,t} + \sum_{t}^{T}\sum_{k}^{K}\sum_{i}^{I}\sum_{j}^{J}\sum_{f}^{F}\left[\left({}^{X_{i,j,f,k,t}}/_{BS_i}\right) - d_{i,t}\right].BC_i \quad (4)$$

$$+ \sum_{t=1}^{TH}\sum_{k=1}^{K}\sum_{i=1}^{I}\sum_{m=1}^{m-1}\sum_{j}^{J}\sum_{f=1}^{F}\left(\alpha_i.\boldsymbol{Z_{m,i,f,j,k,t}}.\left(\left(\sum_{j=1}^{J}\boldsymbol{Z_{m+1,i,f,j,k,t}}.\boldsymbol{X_{m+1,i,f,j,k,t}}\right) - \boldsymbol{X_{m+1,i,f,j,k,t}}\right)\right) \quad (5)$$

$$+ \sum_{t=1}^{TH}\sum_{k=1}^{K}\sum_{i=1}^{I}\sum_{m=1}^{m-1}\sum_{j=1}^{J}\sum_{f}^{F}\left(\beta_i.\boldsymbol{Z_{m,i,f,j,k,t}}.\left(\left(\sum_{k=1}^{K}\sum_{j=1}^{M}\boldsymbol{Z_{m+1,i,f,j,k,t}}.\boldsymbol{X_{m+1,i,f,j,k,t}}\right) - \sum_{j=1}^{M}\boldsymbol{X_{m+1,i,f,j,k,t}}\right)\right) \quad (6)$$

subject to

$$\sum_{k}^{K}\sum_{j}^{J}\sum_{f}^{F}\sum_{i}^{I}\left[\left({}^{X_{i,f,j,k,t}}/_{M}\right)\right] + \sum_{l}^{L} Y_l + \sum_{l}^{L} B_{i,t} \geq d_{i,t} \;\; \forall\, t, i; \quad (7)$$

$$X_{i,f,j,k,t}.(1 - MCIM_{i,j}) \leq 0 \;\; \forall\, t, k, j, i; \quad (8)$$

$$X_{i,f,j,k,t} \leq MA_j \;\; \forall\, t, k, j, i; \quad (9)$$

$$\sum_{l}^{L} Y_l \leq SC_l \;\; \forall\, l, t; \quad (10)$$

$$\sum_{i}^{I} B_{i,t} \leq AB_{i,t} \;\; \forall\, t, i; \quad (11)$$

$$\sum_{j}^{J} N^{+}_{j,k,t} - \sum_{j}^{J} N^{+}_{j,k,t} + N_{k,j,t-1} = N_{k,j,t} \;\forall\, t > 1, k; \quad (12)$$

$$N_{k,j,1} = NOM_j \;\; \forall\, k, j; \quad (13)$$

$$LP \leq \textstyle\sum_j^J N_{j,k,t} \leq CL \;\; \forall\, k, t; \quad (14)$$

$$\sum_{k}^{K}\sum_{i}^{I}\sum_{f}^{F} X_{i,f,j,k,t} \leq MA_j.\sum_{k}^{K} N_{k,j,t} \;\; \forall\, t, j; \quad (15)$$

$$X_{i,f,j,k,t};\; Y_{l,t},;B_{i,t};\; N^{+}_{j,k,t};\; N^{-}_{j,k,t}: integer \quad (16)$$

$$Z_{i,f,j,k,t} : binary \quad (17)$$

The first term of objective function represents the setup cost for each machine that may be different from period to period. The next term shows the operating cost including machinery cost of each part. The third sentence is to show the purchasing or removing costs of machines. The fourth sentence shows the outsourcing cost and the fifth sentence represents the backorder costs. The sixth and seventh sentences show intra- and inter-cellular material transferring costs respectively.

The first set of constraints guarantees that demands for each part will be satisfied by in-house manufacturing, using outsource services or postponing to next periods. Second constraint is developed for ensuring that operations of parts will be performs based on MCIM information. The third constraint ensures that each machine will be allocated based on its capacity. Similarly, the fourth constraint is to guarantee that using sub-contractor services will not be more than their announced capacity. The fifth constraint controls the values of backorders considering the backorder limits. The sixth set of constraints ensures that the amount of purchased or removed machines will stay in a logic manner during manufacturing horizon. The seventh set of constraint represents the initial sets of each machine type in the first period. The eighth sets of constraints controls the size of cells during manufacturing process and the ninth set of operations shows that the amount of producing each part should not be more than the available number of that machine type. The next 2 set of constraints are used to control domain of the variables.

2.2 Analyzing the proposed model

The proposed model is a non-linear mixed integer programing model that has both binary variable and integer variables. Table 1 and table 2 show the number of variable, and number of constraints of the proposed model. Based on these information number of available solutions (both feasible and infeasible) can be estimated. For example in table 1 and table 2 these values are calculated for a sample with 5 product, 3 operations, 4 machine types, 3 cells and 4 periods.

Table 1
Variables and domains of the proposed model

Variable	Domain	Calculation for example	Value
X(i,f,j,k,t)	i×f×j×k×t	5×3×4×3×4	720
Z(i,f,j,k,t)	i×f×j×k×t	5×3×4×3×4	720
Y(l,t)	l×t	5×4	20
B(i,t)	i×t	5×4	20
N+(j,k,t)	j×k×t	4×3×4	48
N-(j,k,t)	j×k×t	4×3×4	48
Total Number of Variables Including Integer and Binary			1576

Table 2
Constraints and domains of the proposed model

Constraint	Domain	Calculation for example	Value
Constraint 1	i,t	5×4	20
Constraint 2	t,k,j,i	4×3×4×5	240
Constraint 3	t,k,j,i	4×3×4×5	240
Constraint 4	l,t	5×4	20
Constraint 5	t,i	4×5	20
Constraint 6	t-1,k	3×3	9
Constraint 7	k,j	3×4	12
Constraint 8	k,t	3×4	12
Constraint 9	t,j	4×4	16
Constraint 10	i,f,j,k,t	5×3×4×3×4	720
Constraint 11	l,t	5×4	20
Constraint 12	i,t	5×4	20
Constraint 13	j,k,t	4×3×4	48
Constraint 14	j,k,t	4×3×4	48
Constraint 15	i,f,j,k,t	5×3×4×3×4	720
Total Number of Constraints			2165

Then, if all constraints are assumed linear the total number of basic solutions including feasible and infeasible will be calculated as:

$$C_m^{n+m} = \frac{(n+m)!}{n!\,m!} = \frac{(1576+2165)!}{1576!\,2165!} \tag{18}$$

The result will be so big that cannot be calculated even with Microsoft Office Excel® 2010. It seems that the proposed model (like many related problems in CMS) is too complex to be able to solve with regular solvers.

3. A hybrid Tabu Search with Simulated Annealing and Genetic Algorithm

Glover (1986) developed TS to overcome defects of searching neighborhood spaces. The logic of the TS is based on using short term memory to prohibit revisiting those solutions that had been rejected before or those whom are banned by the algorithm for some reason (long term memory). Tabu search enhances the performance of local search techniques by defining such memory structures in a way that if a potential solution has been visited in a certain period before or if it violates a rule, it will be marked as a "tabu" (forbidden) movement and will not be considered for a certain period in next iterations as shown by Fig. 2.

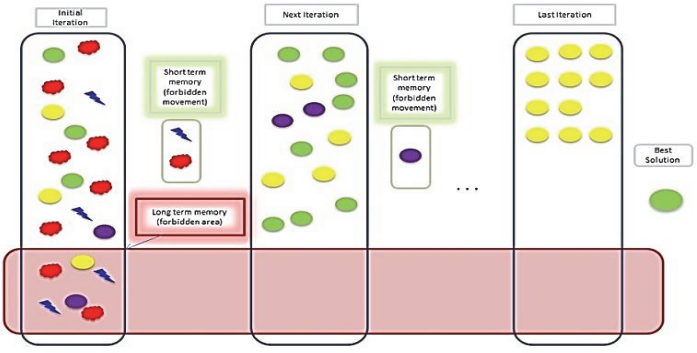


Fig. 2. Scheme of Tabu Search performance

The algorithm starts by generating (or importing) a layout but before performing any further process the algorithm will checks whether the mentioned layout is inside the long term memory. If so the algorithm will recall the selecting operator. Then for each product demand the TS checks whether outsourcing is more economic or in-housing. While outsourcing is cheaper, the algorithm assigns the product demands to subcontractors considering their capacity. In continue for the rest of product demands (if any) the algorithm will choose the best part routes and assign the part on machines considering their available capacity, product batch size and remained product demands. If a part route has a long distance it will be banned in short term memory until the end of the developing process of the current solution. The mechanism of purchasing new machines is same as other developed algorithms. If not, the algorithm will set the rest of the product demands (if any) as backorder. while a solution is ready the algorithm will checks it with the best observed solution gained so far and if the total system cost of the developed solution is more than all observed solutions in the iteration, TS will consider its layout as a member of long term memory which means that the mentioned algorithm is not eligible for more investigation. In addition, the algorithm will update the long term memory at the end of the developing process. Table 3 shows pseudo code of the proposed TS for solving the developed mathematical model.

There are some parameters must be set before using the proposed hybrid TS and SA which are Tabu list size, inspiration rate, neighborhood radius, number of solutions in each iteration, number of iterations and local escaping rate.

Table 3
Pseudo code for TS-SA

Insert Dataset
- Input data sets ($D_{i,t}$; BS_i; $MCIM_{i,j}$; MA_j; SC_l; $AB_{i,t}$; NOM_j; OP_i; S_j; α_i; β_i; OC_j; SR_n; $TB_{j,t}$; CC_n; H_j; F_j; CS_j; W_i ; K_j; DR_j; $PL_{j,t}$; PST_j ; EMT_j; FR_j; PMC_j; ESC_j; BC_i; OS_l; Ir_t)
- Initialize m, k, itr, ζ, Tbs & LEP (m: number of solutions; k= neighbourhood radius; itr: number of iterations; ζ : inspiration rate; Tbs: tabu size; LEP=local escape probability)

While n<itr

Step 1)
- if n=1
- Generate a layout
- else
- Call Selecting operator (tournament list)

Step 2)
- if $X \in Long.memo$
- Recall the Selecting Operator

Step 3)
- for i ∈ I
- if $OS_l < OP_i$
- O_i=min($D_{i,t}$;Out source capacity l)
- calculate Remained $D_{i,t}$

Step 4)
- find $MCIM_{i,j}$
- Generate candidate list considering neighbor radius (n^m) ($Part.route$ operator)
  - for each member (k)
    - if Part. route $_k \in Shortterm.memo$
    - Recall part route operator
  - else
    - $Part.route_k$= $\sum|intercellular\ travel| + |intracellular\ travel|$ (for consecutive machines)
    - if $Manhattan_k$>min($Manhattan_{1:1:k}$)
    - $Shorterm.memo_k = Part.route_k$
- Find the Best Part route in the X
- x_i= min(MA_j, Remained $D_{i,t}$, BS_i)
- calculate Remained $D_{i,t}$

Step 5)
- for i ∈ I
- if Remained $D_{i,t}$>0
- if Machine Purchasing < Outsourcing or $OS_l = 0$
- $NOM_j = NOM_j + 1\ for\ needed\ machine$
- Go step 3

Step 6)
- else
- Set b_i = Remained $D_{i,t}$

Step 7)
- Calculate OFV_i
- if $OFV_i \leq \min_{itr}(OFV)$
- $Tournament\ list_i$=X_i
- OFV^{best} =OFV_i
- $X^{best} = X_k{}^{itr}$

Step 8)
- else
- R=rand(1)
- if R≤ LEP
- $Tournament\ list_i$=X_i
- else
- $X_k{}^{itr} \in Longterm.memory^{itr}$

Step 9)
- if $nnz(Longterm.memory_s^{itr}) > Tbs$
- Update $Longterm.memory^{itr}$
  - $Longterm.memory_1^{itr} = 0$
  - for $p \in (s > 1)$
    - $Longterm.memory_{p-1}^{itr} == Longterm.memory_p^{itr}$
    - $Longterm.memory_s^{itr} = X_k{}^{itr}$

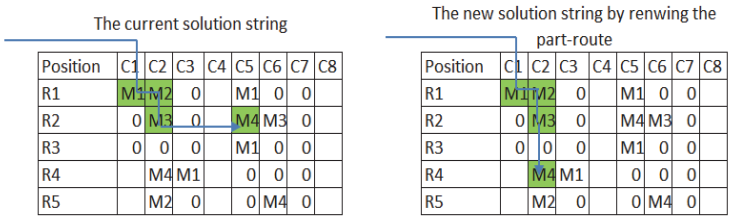
Step 10)
- For $s \in Tbs$
  - if $inspiration.rate_s > \zeta \rightarrow Longterm.memory_s{}^{itr} = 0$
  - Rearrange $Longterm.memory^{itr}$

Step 11)
- Check Stopping Criteria

End

3.1 Number of solutions (Neighborhood radius)

In classic form of Branch and Bound algorithm, all possible neighbors in each branch must be developed. Although such strategy helps searching each branch more deeply but at the same time, it is more time consuming (Fig. 3).



Position	C1	C2	C3	C4	C5	C6	C7	C8
R1	M1	M2	0		M1	0	0	
R2	0	M3	0		M4	M3	0	
R3	0	0	0		M1	0	0	
R4		M4	M1		0	0	0	
R5		M2	0		0	M4	0	

Position	C1	C2	C3	C4	C5	C6	C7	C8
R1	M1	M2	0		M1	0	0	
R2	0	M3	0		M4	M3	0	
R3	0	0	0		M1	0	0	
R4		M4	M1		0	0	0	
R5		M2	0		0	M4	0	

Fig. 3. Generating a new solution string by replacing two related elements in part routing process (left to right)

The strategies of finding neighbors are different and depended on the nature of each problem. Replacing elements, improving an element or zero-one shifting are among popular methods. In this case, considering the solution representing scheme as shown by Fig. 6, such elements are defined by replacing two related elements (parallel machines) during part routing process.

In the proposed method, for each product in a layout (solution), the neighborhood size for considering part routes will be n^m where n shows the number of neighborhood radius which can be determined by decision maker and m is number of applicable machines according to MCIM. For example if neighborhood radius is set 3 by a decision maker and for completing a part (let's say i) required 4 machines, then algorithm will generates 81 neighbors (part routes) in candidate list.

$$n^m \rightarrow 3^4 = 81 \tag{19}$$

Suppose $X(p_1, p_2, \dots, p_m, \dots, p_n; x_1, x_2, \dots, x_n; o_1, o_2, \dots, o_l, b_1, b_2, \dots, b_n)$ is an solution string where p_i is the position of elements in part routing, x_i repersents the amount of parts that allocated to each member of the part routing, o_i indicates the amount of outsourcing and b_i shows the amount of backorders. Then, $X'\begin{pmatrix} p_1, p_2, \dots, p'_m, \dots, p_n; x_1, x_2, \dots, x'_m, \dots, x_n; \\ o_1, o_2, \dots, o_l; b_1, b_2, \dots, b_n \end{pmatrix}$ is a new neighbor which can be generated by shifting the m^{th} element in the initial solution (p_m) to another machine (p'_m).

After generating a candidate list of feasible part routes for a product, the algorithm will use a Manhattan based formula for calculating the intercellular and intracellular distances between consecutive machines. Then the best part route that provided lowest total distance will be chosen for being assigned.

Choosing a candidate from neighbourhood is as described above but in TS after choosing an element and before any changes in part routing, it must be checked that whether the selected elements belong to short term memory. If so the algorithm will check for another element.

$$X(p_1, p_2, \dots, \boldsymbol{p_n}; s_1, s_2, \dots \boldsymbol{s_n}; x_1, x_2, \dots, \boldsymbol{x_n}; o_1, o_2, \dots, \boldsymbol{o_l}) \tag{20}$$

$$X'(p_1, p_2, \dots, \boldsymbol{p_n}'; s_1, s_2, \dots \boldsymbol{s_n}'; x_1, x_2, \dots, \boldsymbol{x_{n+\epsilon}}'; o_1, o_2, \dots, \boldsymbol{o_{l-\epsilon}}') \tag{21}$$

3.2 Tabu list size

Choosing appropriate tabu list size is critical in searching the solution space. Choosing large size tabu list may cause providing bad solutions or early stage convergence. Besides, small tabu lists may affect the since of direction or increase computation time (because of re-visiting low quality solutions).

During the searching process, if a layout can not provide a good basis for improving objective function, it will conside as a long term memory member which means that in next iterations, TS doesnot consider it as a bais for finding best part routings.

3.3 Long term memory

As mentioned before, the framework starts by generating initial cell layouts randomly which are then promoted using part routing improving, machine purchasing and training. In such models, many layouts may be not good enough due to weak arrangement of machines in clusters. Hence, focusing on such layouts may waste the time and increase the computations. Hopefully, TS provides the ability of avoiding redundant calculations using long term memory.
Suppose after preparing a set of clusters (layout), if due to weak arrangement of machines and part routings the achieved objectives function of the layout is worse than all other members in previous iterations, the cell layout is not eligible candidate for further considering in improvement process. It is also means that more likely the mentioned area of solution space does not carry the optimum solution and hence it is not deserve to search more. In such condition, there is no need to focus on the mentioned area. Hence, the initial layout which was generated during first step of the frame work is forbidden for coming iterations. Note that in continue it will be explained that using inspiration rate avoids missing such layouts.

a) suppose $X(p_1, p_2, \dots, p_n; s_1, s_2, \dots s_n; x_1, x_2, \dots, x_n; o_1, o_2, \dots, o_l)$ is an initial cell layout ε is an elected element in $(p_1, p_2, \dots, p_n)$ that is a candidate for creating a new neighbour $(suppose\ \varepsilon = p_2)$

$$X(p_1, \boldsymbol{p_2}', \dots, p_n; s_1, \boldsymbol{s_2}', \dots s_n; x_1, \boldsymbol{x_2}', \dots, x_n; o_1, \boldsymbol{o_2}', \dots, o_l) \rightarrow F_{\varepsilon_1} \quad (22)$$

$$\text{b) if } F_{\varepsilon_1}{}^{itr=n} > \max\left\{F_{\varepsilon_1}{}^{itr=n-1}, F_{\varepsilon_2}{}^{itr=n-1}, \dots, F_{\varepsilon_n}{}^{itr=n-1}\right\};\ \forall\ \varepsilon\epsilon(\varepsilon_1, \varepsilon_2, \dots, \varepsilon_n), t \quad (23)$$

then $X(p_1, p_2, \dots, p_n; s_1, s_2, \dots s_n; x_1, x_2, \dots, x_n; o_1, o_2, \dots, o_l)$ places in $Long\_term\_memory_i$

3.4 Short term memory

Short term memory is used to ban a part of a solution string temporarily as it found not suitable for improvement. But this short rule is not general and that's why it is considered as a short term memory. In the mathematical model, the short term memory can be defined as a part routes that causes a worse material transferring movements. Therefore they will be eliminated from a candidate list of part routes for a specific solution string. The proposed part routs is banned temporarily for current lay out but there is no reason for banning it in next solution that have different layouts.

a) Suppose a part route is selected among candidate list in a layout. If the selected part route provides greater geometrical distances (using Manhattan distance formula) comparing to other part routes, it will be banned for this layout.

$$\text{b) If } P_m(x_1, y_1) \text{ and } Q_m(x'_1, y'_1) \text{ then: } Z_{pq} = |(x_1 - x_1')| + |(y_1 - y_1')| \quad (24)$$

$$\text{c) If } Z_{PQ_m} > min\ (Z_{PQ_1}, Z_{PQ_2}, \dots, Z_{PQ_n}) \quad (25)$$

$$\text{Then: } Short.list_i = PQ_m \quad (26)$$

3.5 Updating tabu list

At the end of solution developing process in iteration, the Tabu list will be updated. In this manner, the algorithm checks if the long term memory is full (based on tabu list size) and there is still a solution must be added to long term memory. In the mentioned considtion, the TS will remove the oldest member and pas all other members to the prior position. Then, the new member will be placed in last position in long term memory.

a) Suppose $X'(p_1', p_2', \dots, p_n'; s_1', s_2', \dots s_n'; x_1', x_2', \dots, x_n'; o_1', o_2', \dots, o_l')$ is the first member of long-term memory

$(Long\_term\_memory_1)$. (27)

b) If $Long\_term\_memory_i > 0 \; \forall \; n \; \epsilon \; Tabulist.size \;\; \&$ (28)

$X''(p_1'', p_2'', \dots, p_n''; s_1'', s_2'', \dots s_n''; x_1'', x_2'', \dots, x_n''; o_1'', o_2'', \dots, o_l'')$ is a new candidate for long-term memory, then:

$Long\_term\_memory_1 = 0$ (29)

$\& \; Long\_term\_memory_{i-1} == \quad Long\_term\_memory_i \; \forall \; i > 1$ (30)

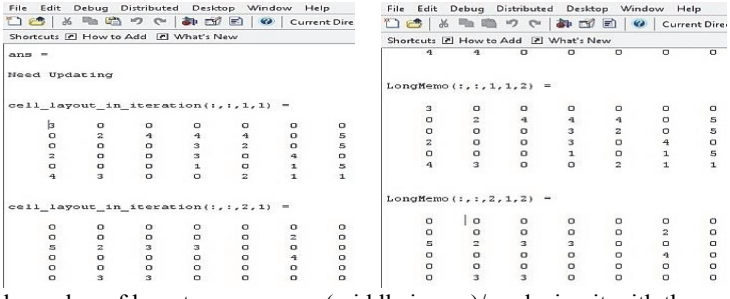


Fig. 4. Removing old member of long term memory (middle image)/ replacing it with the next member (right image)

Fig. 4 shows how TS-SA algorithm is trained to update its long term memory to avoid early stage convergence. As seen, while the number of members in long term memory is equal to tabu list size the algorithm will remove the oldest member and update other members position. This will let the algorithm to avoid converging to a specific area while other solution space areas are completely thew away.

3.6 Intensification

Through the process of searching local neighbors, the algorithm may encounter with a lot of elite neighbours in a layout. In such condition, it is better to have a mechansim for increase searching for such layouts. Note that this doesnot mean that the algorithm will only focus to a dominant layout since it may leads to early stage convergence. But instead, the algorithm will increase the chance of choosing such layouts more. Using such strategy helps to provide a good structure for cocentrating more comprehensively on elite members in local areas. Moreover, using a good short-term memory is an effective way for searching elites in a local area as it prevents searching worse elements as what shown by Fig 5.

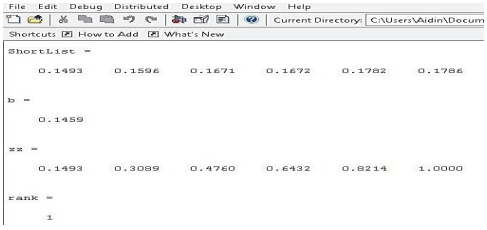


Fig. 5. Increasing the chance of visiting more elite layouts through searching process (intensification)

3.7 Diversification

Diversification is another common issues while using TS algorithm. As mentioned before, TS algorithm works based on functions that prohibit searching areas in solution space which have been

searched before with no imrpovement. But using such strategy may cause early stage convergence since many areas may be phorbidden in searching process. To prevent such problem 2 strategies are used:
1) Using updating long term memory at the end of the processing of each solution by considering the tabu list size.
2) Using an inspiration rate helps to fade old members of tabu list even if the long term memory is not full. Hence, if an area is banned in an itteration, it will not last until the end of the searching process but instead after a period (even if the tabu list is not updated) the older members will be faded one by one. Using such strategy the areas that have been forbidden in early stages of the solving process will have a chanse to consider in a later itteratations again.
a) Suppose $X'(p_1', p_2', \dots, p_n'; s_1', s_2', \dots s_n'; x_1', x_2', \dots, x_n'; o_1', o_2', \dots, o_l')$ is an old member of long term memory ($Long\_term\_memory_i$).

b) If $F_t > F_{t-1} \rightarrow inspiration_{rate} = inspiration_{rate} + \varepsilon$ (31)

c) If $inspiration_{rate} > \zeta \rightarrow Long_{term_{memory_i}} = 0$ (ζ: maximum allowed inspiration rate) (32)

Then: $Long\_term\_memory_1 = 0$ & $Long_{term_{memory_{i-1}}} == Long\_term\_memory_i \; \forall \; i > 1$ (33)

3.8 Fitness function operator

In many optimization cases, objective function are used to evaluate the power of solutions as what did by Tavakkoli-Moghaddam et al. (2005) while in other cases authors defined different fitness functions for their model (Banerjee & Das, 2012). For the heuristics and metaheuristics that designed to solve the mathematical model that developed, the objective function of each proposed model will be used as fitness function operator.

$$\begin{aligned} &\sum \text{Group Setup Cost} && + \sum \text{Operating Cost} + \\ &\sum \text{Outsourcing Cost} && + \sum \text{Backorder Cost} + \\ &\sum \text{Machine Purchasing Cost} && + \sum \text{Intercellular Transferring Cost} + \\ &\sum \text{Intracellular Transferring Costs} \end{aligned} \quad (34)$$

Then each solution will be evaluated if it can improve the minimum total system costs observed so far. If so, the TS-SA will consider it as a member of tournament list.

a) Suppose $X_k^{itr}(x_1, x_2, \dots, x_i; o_1, o_2, \dots, o_l; b_1, b_2, \dots, b_i)$ is the k^{th} solution in itr^{th} iteration.

b) If $F(X_k^{itr}) \leq \min\left(F(X_1^{itr}), F(X_2^{itr}), \dots, F(X_{k-1}^{itr}); F^{best}(X_k^{itr-1})\right) \; ; \; \forall \, k \in i$ (35)

c) Then, $\text{Tournament.list}_{m+1} = X_k^{itr}$ (suppose the tournament list already has m members) (36)

3.9 Local optimum escaping operator

Falling into local optimum traps is a big concern in optimizing problems. One dominant feature of the proposed metaheuristics is using the ability of Simulated Annealing algorithm to escape from local optimum traps. The proposed TS-SA uses such local optimum escaping operator which avoids falling into local optima as shown by Fig. 6. After calculating the fitness function for the developed solution in iterations (sayX_k^{itr}), the TS-SA algorithm checks them with the best observed value achieved so far ($F^{best;itr-1}$). If the fitness function value (F_k^{itr}) is less than the $F^{best;itr-1}$ TS-SA replaces the X_k^{itr} with $X^{best;itr-1}$. But at the same time if the value of F_k^{itr} is more than $F^{best;itr}$ it will be withdrawn immediately. As shown by Fig. 6, in proposed method, even after achieving the worse fitness function, the algorithm provides a base to keep them with a small probability.

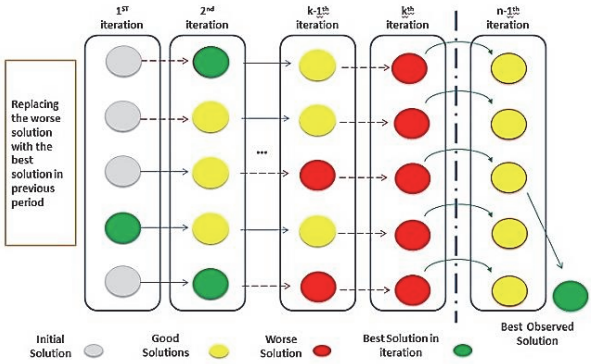


Fig. 6. Solving process scheme of TS-SA with the ability of escaping from local optimum traps

Such strategy lets the algorithm to keep searching to find better solutions as shown by Fig. 7. Such local escaping operator is added to the algorithm by using a function which described:

$$F^{best;itr} = \begin{cases} F_k^{itr}; & if\ F_k^{itr} \le \min\left\{F_i^{itr}, F^{best;itr-1}\right\} \forall\, i \in itr \\ F_k^{itr}; & if\ R \le LEP \\ F^{best;itr-1}; & otherwise \end{cases} \tag{37}$$

where R is a normal random number between (0,1) and LEP is a local escaping parameter which is defined by decision maker. Note that the exact amount of LEP cannot be determined and may be different from case to case but it can be approximately estimated using design of experiments.

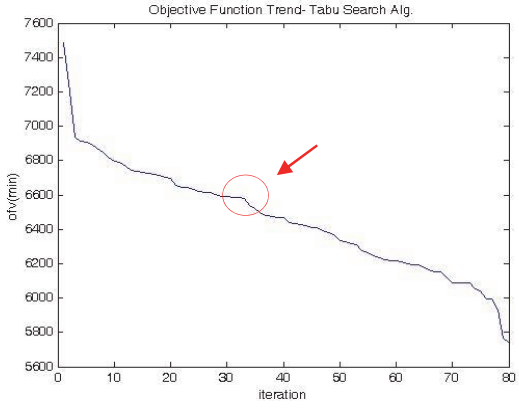


Fig. 7. Sample of escaping from local optimum traps by using local optimum escaping operator

3.10 Number of iterations

Number of solutions is a pre-determined value which shows number of solutions must be completed in iteration. Number of iterations must be completed through the searching process. It is obvious that large number of iterations increases the accuracy of the solving algorithm but at the same time it increases the time of computations. Therefore it must be chosen in a logic manner. For the experiments in this research 3 levels of number of iterations and number of solutions in iteration are considered for small size, medium size and large scale problems. However the more accurate input values for parameters will be estimated in DOE section.

3.11 Stopping criteria

The stopping criteria in the proposed TS-SA algorithm are set as:

1) Reaching to maximum number of pre-defined iterations.

2) If there is no choice in tournament list in iteration which means none of the solutions in the iteration can improve the fitness function so there will be no choice for improving the algorithm.

a) Suppose $X_k^{itr}(x_1, x_2, \dots, x_i; o_1, o_2, \dots, o_l; b_1, b_2, \dots, b_i)$ is the k^{th} solution in itr^{th} iteration.

b) If $F(X_k^{itr}) > \min\left(F(X_1^{itr}), F(X_2^{itr}), \dots, F(X_{k-1}^{itr}); F^*(X_k^{itr-1})\right)$; $\forall\, k \in i$ (38)

Then $\text{Tournament.list}^{itr} = \emptyset$ for next iteration (39)

4. Design of Experiments

As mentioned previously, the experimental design helps estimate the significance of each setting parameter and also determine the interactions between these parameters. Therefore, the Taguchi method designs are used using Minitab® 17. For the solving algorithm the design depends on the number of input parameters of the algorithm. For example, as shown by Table 5, since the proposed method has 4 input parameters, an L_{27} (3^4) orthogonal optimization is taken into account. For DOE experiments, the levels of each parameter must be estimated. Eiben et al. (1999) in their invaluable research over setting input parameters in evolutionary algorithm mentioned that for setting input parameters two strategy can be considered. First is using mathematical formula for each parameter and then estimate the parameter considering other conditions. Second is using a range of data for a parameter and then choosing appropriate value considering other condition of the problem. For example for population size in genetic algorithm (which is similar to number of solution in other metaheuristics in this research) a range between [30 80] is opted by Schaffer et al. (1989) and a range between [50 100] is offered by De Jong (1975). However, estimating the appropriate value for this parameter and other input parameter depends on other condition of the problem like, the scale of the problem, complexity of the problem, the chance of falling in the local optima. In this research due to complexity of the mathematical model and the solving algorithms it is very hard to estimate a comprehensive mathematical formula for each of the input parameters. Moreover, the results of using such formulas in complex conditions may not be trustful. Thus, as used by many other scientists (Roeva et al., 2013), for each parameter, the model runs for a range of values while other parameters are fixed at maximum point and then the most reasonable value will be chosen by comparing the observed objective function. The results for the factor "number of solutions" in a small size CCDCMS are shown by Table 4. Table 4 shows that while using TS-SA for small size CCDCMS experiments, the appropriate input value for "number of solutions in the iterations" must be approximately 50. Any value more than this can result worthless calculations and smaller values also could not provide proper minimizations (Fig. 8).

Table 4

Results of running TS-SA to find approximate levels of factors of CCDCMS (small size)

Number of solution	5	10	20	50	100	150	200
First-Last Obj	10452-7552	10652-7652	11077-7598	11052-7552	11152-7652	10452-7852	11052-7952
ΔOBJ	2900	3000	3479	3500	3500	2600	3100

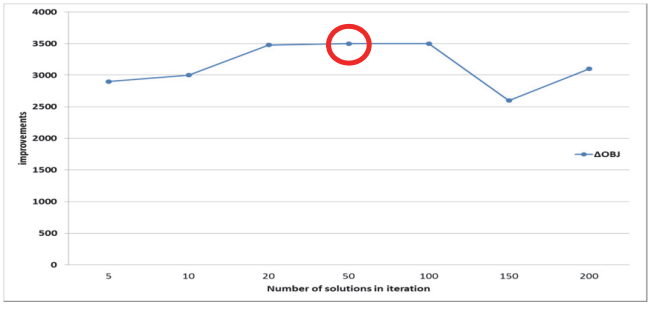


Fig. 8. The trend line of maximing improvements by considering different number of solutions in iterations

The same strategy is employed for other parameters of solving algorithms. The best estimated values for each parameter of metaheuristics are shown by the Table 5:

Table 5
Initial estimation for levels of factors

Algorithm	Factor	Small scale			Medium scale			Large scale		
		L1	L2	L3	L1	L2	L3	L1	L2	L3
TS-SA	Number of Iterations	30	50	100	50	100	150	50	100	200
	Number of Solutions in each iteration	20	50	100	30	50	80	50	80	100
	Tabu list size	2	3	5	2	3	5	2	3	5
	Local Escaping rate	0	0.1	0.3	0	0.1	0.3	0	0.1	0.3

The levels of factors, which are shown by Table 5, are then used for DOE in order to find the best estimating value for each parameter, significant parameters and interactions between them. Table 6 provides results of implementing the CCDCMS using the hybrid TS-SA and SA. R1, R2 and R3 show the results of implementing the hybrid method for small, medium and large scale CCDCMS, respectively.

Table 6
Results of implementing the L27(3^4) experiments for the proposed hybrid TS and SA

Experiment Number	Factor				SMALL SCALE			MEDIUM SCALE			LARGE SCALE		
	Number of Iterations	Number of Solutions in each iteration	Tabu list size	Local Escaping rate	CCDCMS			CCDCMS			CCDCMS		
	(A)	(B)	(C)	(D)	FIRST	LAST	R1	FIRST	LAST	R2	FIRST	LAST	R3
1	1	1	1	1	9770	9244	526	10688	7887	2801	14228	13005	1223
2	1	1	1	1	9784	9175	609	11253	7472	3781	13414	12057	1357
3	1	1	1	1	10177	8948	1229	9880	6876	3004	13445	12462	983
4	1	2	2	2	13446	10177	3269	8977	6548	2429	17307	13160	4147
5	1	2	2	2	13772	9302	4470	9506	6750	2756	16615	12589	4026
6	1	2	2	2	13078	9281	3797	9375	6847	2528	16610	13244	3366
7	1	3	3	3	13047	9618	3429	9582	7246	2336	17697	13201	4496
8	1	3	3	3	13046	10106	2940	10756	7917	2839	16714	13438	3276
9	1	3	3	3	14312	9347	4965	9080	6573	2507	17116	12954	4162
10	2	1	2	3	13913	9314	4599	9376	6878	2498	17389	11945	5444
11	2	1	2	3	13941	10042	3899	9778	6884	2894	17515	13507	4008
12	2	1	2	3	13743	9750	3993	9181	6680	2501	18187	12793	5394
13	2	2	3	1	9910	8847	1063	9896	6338	3558	13263	11598	1665
14	2	2	3	1	9011	8744	267	10489	7963	2526	12766	11853	913
15	2	2	3	1	9673	8612	1061	9576	6881	2695	13368	12307	1061
16	2	3	1	2	14470	9042	5428	9485	6980	2505	18706	14057	4649
17	2	3	1	2	13277	8878	4399	9477	7176	2301	17630	12130	5500
18	2	3	1	2	14271	9604	4667	9641	6672	2969	18120	13394	4726
19	3	1	3	2	13992	9144	4848	10180	6645	3535	17454	12495	4959
20	3	1	3	2	14149	9080	5069	10476	6538	3938	17610	12864	4746
21	3	1	3	2	14110	9137	4973	10555	7567	2988	18584	11937	6647
22	3	2	1	3	13844	9775	4069	10072	6775	3297	17433	12718	4715
23	3	2	1	3	13986	9739	4247	9683	6678	3005	19453	13919	5534
24	3	2	1	3	14406	9576	4830	10174	6548	3626	17374	11755	5619
25	3	3	2	1	9082	8715	367	9677	6675	3002	13999	12514	1485
26	3	3	2	1	9285	8450	835	11143	7294	3849	12735	11944	791
27	3	3	2	1	8980	8443	537	9975	6375	3600	12651	11745	906

Upon implementing the experiments designed for the Taguchi method, the obtained results show that the algorithm is sensitive to the number of iterations, number of solutions and the interaction between number of iterations and number of solution (Fig. 9).

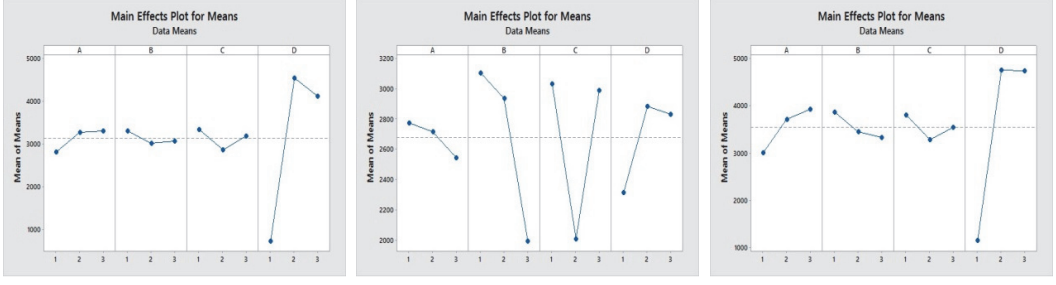


Fig. 9. The main effects plot of input parameters for small, medium and large scale problems while using the proposed hybrid TS-SA (left to right)

The high slope of the curves related to the local escaping rate shows that while using TS-SA for small, medium and large scale problems, L2 will provide better results. With the same logic, Tabu list size should not have more than 2 members for small, medium and large scale problems. The number of iterations and number of solutions in iteration have moderate impact on minimizing the objective function. Fig. 9 also shows that the local escaping rate has the highest impact on maximizing the algorithms power (minimizing total system costs). The interaction between factors for large scale problems is monitored using the regression equation in encoded units that estimated based on actual values:

$$R= 3244 + 591.5\ \text{Iterations} - 362.3\ \text{Number of Solutions} + -168.2\ \text{Tabu list size} + 1996\ \text{Local Escaping rate} + 173.8\ \text{Iterations} \times \text{Number of Solutions} + 221.7\ \text{Iterations} \times \text{Local Escaping rate} + 270.0\ \text{Number of Solutions} \times \text{Local Escaping rate} - 321.7\ \text{Iterations} \times \text{Local Escaping rate} + 181.5\ \text{Iterations} \times \text{Number of Solutions} \times \text{Local Escaping rate} \quad (40)$$

As seen, in this equation, the local escaping rate and the iterations have the highest impact on R respectively, the number of solutions and Tabu list size cause negative interaction on total system function and should not be increase simultaneously. Therefore, for the larger scale experiments the number of iterations must be increased. Hence, the appropriate ranges for number of iterations and number of solutions in iterations can be set as what shown by Table 7. After implementing DOE experiments for solving the developed model using proposed hybrid method, the best observed values will be record as good estimation for input parameters.

Table 7
Estimated values for input parameters of TS-SA algorithm

Algorithm	Factor	SMALL SCALE	MEDIUM SCALE	LARGE SCALE
TS	Number of Iterations	100	50	200
	Number of Solutions in each iteration	20	30	50
	Tabu list size	2	2	2
	Local Escaping rate	0.1	0.1	0.1

5. Results and Discussion

5.1 Solving algorithms using datasets from the literature

For solving such problem we use the strategy that offered by Li et al. (2012) as they run their metaheuristics 10 times for each experiments and then reported the best solution. Table 8 shows the results of solving different experiments using proposed solving algorithms. In each case, solving algorithms are solved 10 times and the best result in reported then. The results in table 8 show for small and medium size problems of CCDCMS the solving algorithms reported same results but for the rest of the cases, in most of the cases TS-SA provided better solutions than Branch and Bound and Simulated Annealing algorithms.

Table 8
Solving mathematical problems with TS-SA and Branch and Bound

	Algorithm	Run 1	Run 2	Run 3	Run 4	Run 5	Run 6	Run 7	Run 8	Run 9	Run 10	Mean	Std.	Max	Min
CC-DCMS (SMALL SIZE)	BnB Heuristic	4454	4454	4454	4454	4454	4454	4454	4454	4454	4454	4454	0	4454	4454
	SA	4454	4454	4454	4454	4454	4454	4454	4454	4454	4454	4454	0	4454	4454
	Hybrid TS-SA	4454	4454	4454	4454	4454	4454	4454	4454	4454	4454	4454	0	4454	4454
CC-DCMS (MEDIUM SIZE)	BnB Heuristic	9185	9455	9195	9185	9191	9185	9185	9195	9185	9191	9215.2	84.361392	9455	9185
	SA	9185	9195	9195	9185	9461	9193	9187	9185	9191	9191	9216.8	85.89632	9461	9185
	Hybrid TS-SA	9185	9185	9189	9185	9185	9455	9185	9201	9185	9185	9214	84.82793	9455	9185
CC-DCMS-(LARGE SIZE)	BnB Heuristic	16735	16935	16835	16435	15635	15835	15535	16735	17035	17235	16495	611.37368	17235	15535
	SA	18235	17135	15935	16135	17235	17635	17935	17235	16535	17535	17155	750.99933	18235	15935
	Hybrid TS-SA	16435	16235	17235	16235	16235	16835	17135	16335	15035	16735	16445	620.84172	17235	15035

In continue, to verify the proposed hybrid metaheuristics, 17 experiments are solved with data gathered from literature. The results are then compared with results of branch and bound and simulated annealing algorithm as shown by Table 9. Fig. 10, Fig. 11 and Fig. 12 show the minimizing process for some experiments in Table 9.

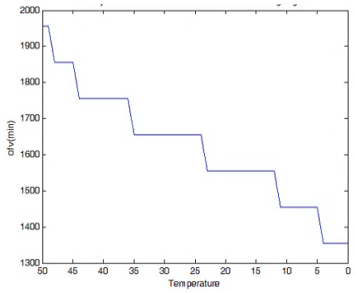


Fig. 10. The objective function graph of TS-SA method for experiment number 5 in Table 9

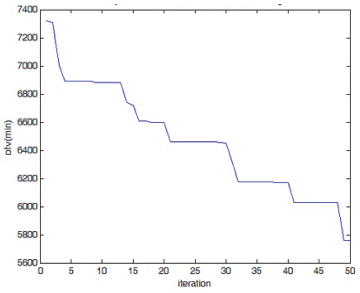


Fig. 11. The objective function graph of TS-SA method for experiment number 8 in Table 9

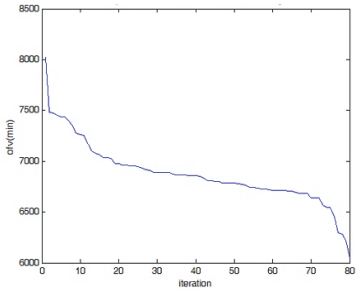


Fig. 12. The objective function graph of TS-SA method for experiment number 17 in Table 9

Table 9
Results of solving numerical examples from the literature for CCDCMS model

Scale	No.	Problem Source	K	I	m	j	L	T	C.S	NOP	BnB	TS-SA	Min	R	Average
small	1	Askin & Huang (2001)	2	2	2	2	2	4	8	[2 4]	4454	4454	4454	0	4454
	2	Askin & Huang (2001)	2	2	4	2	2	4	20	[3 6]	6304	6304	6304	0	6304
	3	Suer & Cedeno (1996)	1	4	4	4	4	4	15	[3 2 3 2]	19550	19550	19550	0	19550
	4	Askin & Huang (2001)	8	2	2	2	2	4	4	[14 6]	1245	1245	1245	0	1245
	5	Askin & Huang (2001)	8	2	2	2	2	4	4	[8 8]	1355	1355	1355	0	1355
medium	6	Mahdavi et al. (2010)	2	4	4	4	4	2	20	[4 5 3 6]	21155	21680	21155	-525	21417.5
	7	Mahdavi et al. (2012)	2	4	4	4	4	4	16	[4 3 6 5]	7030	6978	6978	52	7004
	8	Aryanezhad et al. (2009)	3	3	3	3	3	3	20	[5 4 5]	6030	5760	5760	270	5895
	9	Aryanezhad et al. (2009)	5	4	5	5	4	4	8	[4 3 2 3 3]	5650	5560	5560	90	5605
	10	Aryanezhad et al. (2009)	4	5	5	5	5	4	12	[2 4 3 4 4]	5571	5671	5571	-100	5621
	11	Li et al. (2012)	2	5	5	5	5	4	15	[3 5 4 2 3]	10657	11039	10657	-382	10848
	12	Mahdavi et al. (2012)	2	5	5	5	5	4	15	[7 8 5 9 4]	5706	5600	5600	106	5653
Large	13	Mahdavi et al. (2010)	2	6	6	6	6	3	20	[5 4 3 6 5 4]	10319	10319	10319	0	10319
	14	Norman et al. (2002)	2	6	6	6	6	5	14	[3 5 2 4 3 4]	9455	9455	9455	0	9455
	15	Askin & Huang (2001)	2	8	6	6	8	4	30	[3 2 2 3 2 3]	15697	15609	15609	88	15653
	16	Askin & Huang (2001)	2	8	8	8	8	4	20	[2 4 3 5 4 3 2 4]	17858	17125	17125	733	17491.5
	17	Aryanezhad et al. (2009)	5	5	5	5	5	5	30	[4 3 4 3 3]	6221	6041	6041	180	6131

5.2 Measuring the machine load variation in dynamic cellular manufacturing systems

In this section using concepts of controlling charts in TQM, a suitable method is offered for measuring the values of machine load variation in after scheduling dynamic cellular systems while costs are not fixed. For this purpose, the results of solving example 6 in the Table 9 will be illustrated.

The result of best observed cell layout is displayed in Table 10. The figure indicates that the algorithm generated two block diagonals through agglomeration. These blocks are maintained until the end of the searching process. Upon forming cells and locating the machines inside, the next priority is identifying the best parts routes for allocating the operation sequence of products. Parts routes represent a set of required machines (based on MCIM priorities) that are chosen in consecutive order to complete a product. For this purpose, a new matrix is developed that determines the ideal number of parts routes for production (Table 11).

Table 10
Results of generated block diagonals

Best_Observed_cellposition										
First Period										
Cell 1					Cell 2					
1	2	0	0	0	1	1	3	2	0	
1	1	0	0	0	3	2	1	3	0	
2	0	0	0	0	0	1	3	3	1	
2	0	0	0	0	0	1	2	2	3	
Second Period										
Cell 1					Cell 2					
1	2	0	0	0	1	1	3	2	0	
1	1	0	0	0	3	2	1	3	0	
2	0	0	0	0	0	1	3	3	1	
2	0	0	0	0	0	1	2	2	3	

Table 11
Part-routes observed for the experiment

Part route			Period	Product id	Load Volume
1	0	0	1	4	15
39	36	35	1	1	15
26	31	0	1	3	8
27	32	31	1	1	15
5	29	0	1	3	8
31	0	0	1	2	10
33	35	0	1	3	8
21	33	29	1	1	15
31	0	0	1	2	2
34	0	0	1	2	10
21	0	0	1	4	15
40	0	0	1	2	10
25	0	0	1	4	15
34	0	0	1	2	10
40	0	0	1	2	10
28	26	34	1	1	15
1	0	0	1	4	15
35	0	0	1	2	10
2	3	22	1	1	15
22	0	0	1	2	10
29	0	0	1	2	10
22	0	0	1	2	10
2	4	40	1	1	15
30	26	29	1	1	2
21	33	35	1	1	2
1	0	0	1	4	5
28	0	0	1	4	15
28	0	0	1	4	5
39	0	0	1	4	15
25	0	0	1	4	15
39	0	0	1	4	5

Table 12
The best observed machine loads

Best_observed_machineloads										
Cell 1					Cell 2					
35	8	0	0	0	35	30	35	25	0	
35	0	0	0	0	35	25	17	35	0	
15	0	0	0	0	0	17	35	35	35	
15	0	0	0	0	0	35	15	15	35	
Cell 1					Cell 2					
30	0	0	0	0	30	35	35	25	0	
30	17	0	0	0	35	30	10	35	0	
30	0	0	0	0	0	25	35	35	15	
10	0	0	0	0	0	35	0	30	35	

The "*best-observed-machineloads*" matrix in Table 12 indicates the number of machine loads in each cell. Machines with less or more load than other parallel machines can thus be recognized.

5.3 Proposing a new method for calculating cell load variations and recognizing complex dummy sub cells

In small layouts, cell-load variations are easily recognized. However, these cells are difficult to identify as they grow in number or size. Hence, a mathematical metric must be used. To investigate the effect of uncertain product market on cell-load variation, a mathematical metric is developed that operates on the basis of tracking material transfer between and within each cell in the planning horizon. The metric can indicate the possibility of existing cell-load variation in a region with neighborhood radius R.

$$CDSB_{p,q} = \left(\left\{M_{(p,q),j,k,t} \middle| \left(\frac{Machine-load_{j,k,t}}{\sum_k^K Machine-load_{j,k,t}} > \bar{X}_{j,k,t} + kS_{j,k,t}{}^2\right); \forall j \in \begin{pmatrix} R-1 \le p \le R+1, \\ R-1 \le q \le R+1 \end{pmatrix}; k,t\right\}\right) \quad (41)$$

where $M_{p,q,j,k,t}$ is machine type j that is placed in row p and column q of cell k during period t. $\bar{X}_{j,k,t}$ is the average machine load on machine type j in cell k during period t. $S_{j,k,t}$ is the standard deviation of machine type j. R is the neighborhood radius that shows the cubic area around each center point (p, q).

After determining the parts routes (see Table 11) and machine loads (see Table 12) and calculating the formula presented in Equation 41, each value must be analyzed to identify whether the machine is idle or over-loaded. Moreover, whether or not the achieved value should be considered noise due to product demands is investigated as well. To determine machine load variation in experiment 1, the results derived from Table 10 are analyzed further. Table 13 indicates the number of machine loads for each machine type for the first period in the final solution. For instance, machine type 1 in first cell has a load of 70. This cell contains two type 1 machines, therefore, the loads on machine type 1 can be averaged as 35 (70/2).

Table 13
Results of machine loads in first period of final solution

Loads/ Machines ($t=1$)	C1			C2		
	M1	M2	M3	M1	M2	M3
Number of loads	70	38	0	169	80	210
Maximum load	35	15	0	35	25	35
loading average	35	12.67	0	28.16	20	35

The control limit lines for each machine type in the second period of the final solution are calculated according to the findings regarding machine loads in each cell (Table 13). These results are then considered in drawing a control chart for each machine type and in determining idle or over-allocated machines in a cell (Fig. 13).

Table 14
Control limits for each machine type in best observed solution

Control Limits	t=1			t=2		
	M=1	M=2	M=3	M=1	M=2	M=3
U.C.L	45.89	31.55	35.00	45.98	34.11	35.00
L.C.L	10.44	8.45	35.00	4.02	22.56	35.00

Fig. 13 indicates that all of the charts are within the control limits in the final layout. While drawing the charts k is assumed 1. As discussed previously, parameter K is dependent on the product demands, number of machines, and batch sizes. This parameter can be set by a decision maker.

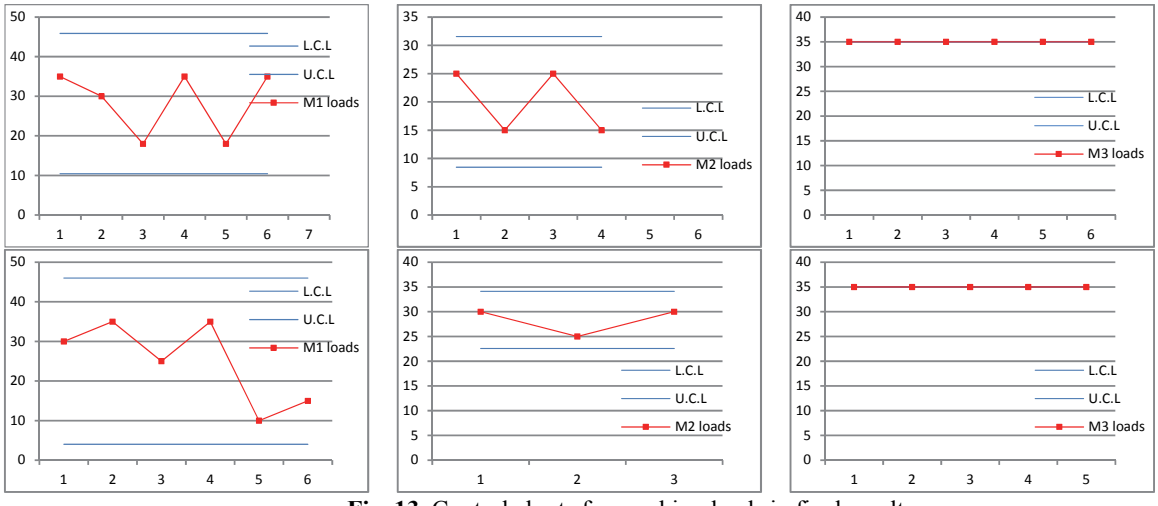


Fig. 13. Control charts for machine-loads in final result

Fig. 14 depicts an example of dummy sub-cell recognition in the first period of the final solution for experiment 8. The equation 41 is used in a macro program of Matlab 2011. This macro works based on the set center point, the considered cubic block around the center point and all of the recognized sets of machines that are over-allocated in this area given the relations defined in MCIM. If the algorithm

detects an overloaded machine in the cell, then all of the processes mentioned earlier are repeated to increase *R* values until all members of the complex sub-cells are recognized. The outcome is the set of over-allocated machines located nearby.

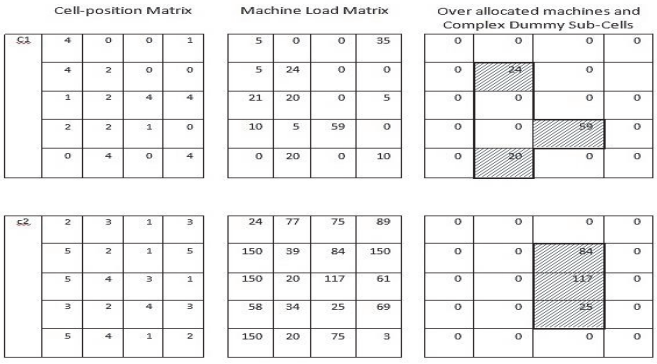


Fig. 14. Results of CDSB Equation for first period in final solution of experiment number 8

The use of this algorithm can help a decision-maker consider complex dummy sub-cells in cubic blocks of various sizes. For example, the dummy sub-cell block of over-allocated machines can be identified as highlighted in Fig. 14 in the second cell of experiment 8 when neighborhood radius (*R*) 1 is considered. However, if *R* assumed 1 for the first cell of experiment 8, then each over-allocated machine cannot identify other nearby over-allocated machines. Instead, if *R* is assumed to be 2, then a dummy complex can be recognized as illustrated in Fig. 14.

5.4 Impact of dynamic market changes on cell load variation

The results of the solving experiments show that demands for dynamic parts can disrupt material transfer fluency (parts routes). Additional short-distance intra-cellular routes are generated by increasing the number of products or product demands. As a result, the number of bottleneck machines increases, whereas other parallel machines remain idle. As per a comparison of the numbers of inter- and intra-cellular movements, intra-cellular WIP movements increased in the time horizon in 53.8% of the studied cases under dynamic parts demand. Although such movements may be useful and may reduce material transfer costs, these movements may induce cell load variation. This variation is concisely called "*complex dummy sub-cells*". This phenomenon emerges as a consequence of increasing the setting of short-cut distances in material transfer on parts routes that enhance cell load variation. Fig. 15 presents the "*complex dummy sub-cells*" as a CMS output.

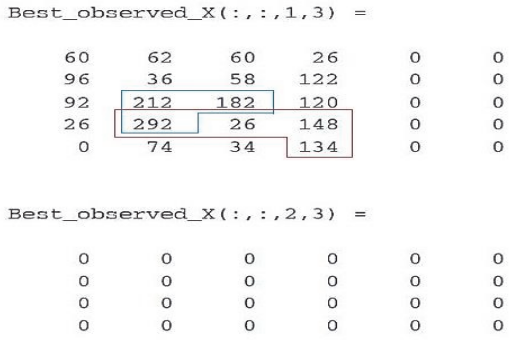


Fig. 15. Emerging complex dummy sub-cells after scheduling in a CMS

The sub-cells do not follow a specific rule and may emerge in any area in a cell depending on the arrangements of the machines and on the parts routes scheduled on the machines. This phenomenon illustrates the tendency of systems to allocate WIPs to a set of machines that are located contiguously. Parallel machines in the same cell are located farther away and remain idle. To examine the level of cell sensitivity to "*complex dummy sub-cells,"* a metric is developed on the basis of the neighborhood

radius of the WIPs. The remaining dummy sub-cells in the solved cases are presented in table 15. As shown although in all cases, the proposed method found the minimum total system cost (which contains feasible part routes) but in many cases dummy sub-cells emerged.

Table 15
Observed dummy sub cells in the solved experiments

No.	t=1	t=2	t=3	t=4	t=5
1	-	-	-	-	-
2	(1,2)	(2,5)	(1,7)	(2,6)	-
3	-	-	-	-	-
4	(10,12)	(7,12,16)	-	-	-
5	(8,9,,19)	(3,8,14,18,19)	(4,8,13,19)	(4,8,9,12,19)	-
6	(12,13,17,18)	(7,10,12,13)	-	-	-
7	(21,27,28)	-	-	-	-
8	(23,32,34,35)	(4,12); (22,28,31)	(4,13,18)	(7,12); (23,29)	-
9	(1,2,4,11); (34,36)	(1,8,19);(34,36)	(5,6);(26,28)		-
10	(7,11);(28,29)	(7,9,10)	(9,11);(26,29)	-	(9,10,12,14);(26,29)
11	(5,11,12)	-	(2,3)	(3,15,24)	
12	-	-	-	-	-
13	-	-	-	-	-
14	-	-	-	-	-
15	-	-	(6,7)	(5,8,11)	-
16	-	-	-	-	-
17	(18,23,26)	(22,25)	-	-	-

6. Conclusion

This paper presents an applicable method for scheduling dynamic cellular manufacturing systems in the presence of product demand uncertainty. The main objective of the research is evaluating the impact of uncertain product demands on material transferring in CMS. For this purpose, a mathematical model is developed for determining best part routes considering different product demands in planning periods. Then, the impact of changing in product demands on material transferring is evaluated. It is shown that in the mentioned condition, closer set of required machines for producing a product are employed more than other parallel machines. This phenomenon can cause increasing the machine loads in such cells and may lead to machine load variation in set of closer machines while other machines are less allocated or left idle. Such distortion emerges temporarily in a period and by the end of the period it may disappear but at the same time they may cause long queues in front of some machines while other parallel machines are remained idle. As a subsidiary conclusion, the results also show that the locations of machines are an important factor in reducing the machine load and cell load variations accordingly. It is shown that (whenever possible) using subcontractor services is a good strategy for preventing demand lost.

Acknowledgments

The Authors would like to thank Professor Mohammad Reza Tavakoli-Moghaddam for his positive comments.

References

Ahkioon, S., Bulgak, A., & Bektas, T. (2009). Cellular manufacturing systems design with routing flexibility, machine procurement, production planning and dynamic system reconfiguration. *International Journal of Production Research, 47*(6), 1573-1600. doi: 10.1080/00207540701581809

Ariafar, S., Firoozi, Z., & Ismail, N. (2014). A Triangular Stochastic Facility Layout Problem in a Cellular Manufacturing System. Paper presented at the International Conference on Mathematical Sciences and Statistics 2013.

Ariafar, S., & Ismail, N. (2009). An improved algorithm for layout design in cellular manufacturing systems. *Journal of Manufacturing Systems, 28*(4), 132-139. doi: 10.1016/j.jmsy.2010.06.003

Balakrishnan, J., & Cheng, C. H. (2005). Dynamic cellular manufacturing under multiperiod planning horizons. *Journal of manufacturing technology management, 16*(5), 516-530.
Banerjee, I., & Das, P. (2012). Group technology based adaptive cell formation using predator–prey genetic algorithm. *Applied Soft Computing, 12*(1), 559-572.
Boulif, M., & Atif, K. (2006). A new branch-&-bound-enhanced genetic algorithm for the manufacturing cell formation problem. *Computers & operations research, 33*(8), 2219-2245.
De Jong, K. A. (1975). Analysis of the behavior of a class of genetic adaptive systems.
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.
Egilmez, G., & Süer, G. (2014). The impact of risk on the integrated cellular design and control. *International Journal of Production Research, 52*(5), 1455-1478.
Eiben, A. E., Hinterding, R., & Michalewicz, Z. (1999). Parameter control in evolutionary algorithms. *Evolutionary Computation, IEEE Transactions on, 3*(2), 124-141.
Elmi, A., Solimanpur, M., Topaloglu, S., & Elmi, A. (2011). A simulated annealing algorithm for the job shop cell scheduling problem with intercellular moves and reentrant parts. *Computers & industrial engineering, 61*(1), 171-178.
Glover, F. (1986). Future paths for integer programming and links to artificial intelligence. *Computers & Operations Research, 13*(5), 533-549.
Goncalves Filho, V., & José Tiberti, A. (2006). A group genetic algorithm for the machine cell formation problem. *International Journal of Production Economics, 102*(1), 1-21.
Gonçalves, J. F., & Resende, M. G. (2004). An evolutionary algorithm for manufacturing cell formation. *Computers & industrial engineering, 47*(2), 247-273.
Gravel, M., Luntala Nsakanda, A., & Price, W. (1998). Efficient solutions to the cell-formation problem with multiple routings via a double-loop genetic algorithm. *European Journal of Operational Research, 109*(2), 286-298.
Grznar, J., Mehrez, A., & Felix Offodile, O. (1994). Formulation of the machine cell grouping problem with capacity and material movement constraints. *Journal of Manufacturing Systems, 13*(4), 241-250.
Gupta, Y. P., Gupta, M. C., Kumar, A., & Sundram, C. (1995). Minimizing total intercell and intracell moves in cellular manufacturing: a genetic algorithm approach. *International Journal of Computer Integrated Manufacturing, 8*(2), 92-101.
Haleh, H., Iranmanesh, H., & Kor, H. (2009). A new hybrid evolutionary algorithm for solving multi objective cell formation problem. Paper presented at the Computers & Industrial Engineering, 2009. CIE 2009. International Conference on.
Harhalakis, G., Nagi, R., & Proth, J. (1990). An efficient heuristic in manufacturing cell formation for group technology applications. *International Journal of Production Research, 28*(1), 185-198.
Heragu, S. S. (1989). Knowledge based approach to machine cell layout. *Computers & industrial engineering, 17*(1), 37-42.
Jeon, G., & Leep, H. R. (2006). Forming part families by using genetic algorithm and designing machine cells under demand changes. *Computers & operations research, 33*(1), 263-283.
Li, Q., Gong, J., Fung, R. Y., & Tang, J. (2012). Multi-objective optimal cross-training configuration models for an assembly cell using non-dominated sorting genetic algorithm-II. *International Journal of Computer Integrated Manufacturing, 25*(11), 981-995.
Logendran, R., & Talkington, D. (1997). Analysis of cellular and functional manufacturing systems in the presence of machine breakdown. *International Journal of Production Economics, 53*(3), 239-256.
Moussa, S. E., & Kamel, M. (1998). A part-machine assignment algorithm for cellular manufacturing with machine capacity constraints. *Computers & industrial engineering, 35*(3), 483-486.
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. doi: 10.1016/j.ejor.2005.01.017

Onwubolu, G., & 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.

Paydar, M. M., Mahdavi, I., Sharafuddin, I., & Solimanpur, M. (2010). Applying simulated annealing for designing cellular manufacturing systems using MDmTSP. *Computers & industrial engineering, 59*(4), 929-936.

Rafiee, K., Rabbani, M., Rafiei, H., & Rahimi-Vahed, A. (2011). A new approach towards integrated cell formation and inventory lot sizing in an unreliable cellular manufacturing system. *Applied Mathematical Modelling, 35*(4), 1810-1819.

Renna, P., & Ambrico, M. (2015). Design and reconfiguration models for dynamic cellular manufacturing to handle market changes. *International Journal of Computer Integrated Manufacturing, 28*(2), 170-186.

Roeva, O., Fidanova, S., & Paprzycki, M. (2013). Influence of the population size on the genetic algorithm performance in case of cultivation process modelling. Paper presented at the Computer Science and Information Systems (FedCSIS), 2013 Federated Conference on.

Safaei, N., Saidi-Mehrabad, M., & Jabal-Ameli, M. (2008). A hybrid simulated annealing for solving an extended model of dynamic cellular manufacturing system. *European Journal of Operational Research, 185*(2), 563-592. doi: 10.1016/j.ejor.2006.12.058

Safaei, N., & Tavakkoli-Moghaddam, R. (2009a). An extended fuzzy parametric programming-based approach for designing cellular manufacturing systems under uncertainty and dynamic conditions. *International Journal of Computer Integrated Manufacturing, 22*(6), 538-548.

Safaei, N., & Tavakkoli-Moghaddam, R. (2009b). Integrated multi-period cell formation and subcontracting production planning in dynamic cellular manufacturing systems. *International Journal of Production Economics, 120*(2), 301-314.

Sarker, B. R., & Yu, J. (2007). A quadra-directional decomposition heuristic for a two-dimensional, non-equidistant machine-cell location problem. *Computers & operations research, 34*(1), 107-151.

Schaffer, J. D., Caruana, R. A., Eshelman, L. J., & Das, R. (1989). A study of control parameters affecting online performance of genetic algorithms for function optimization. Paper presented at the Proceedings of the third international conference on Genetic algorithms.

Seifoddini, H., & Djassemi, M. (1993). A dynamic part assignment procedure in machine cell formation. *Computers & industrial engineering, 25*(1), 473-476.

Seifoddini, H., & Djassemi, M. (1996). Improving the performance of cellular manufacturing by a dynamic part assignment approach. *Computers & industrial engineering, 30*(4), 719-726.

Tavakkoli-Moghaddam, R., Aryanezhad, M.-B., Safaei, N., & Azaron, A. (2005). Solving a dynamic cell formation problem using metaheuristics. *Applied Mathematics and Computation, 170*(2), 761-780. doi: 10.1016/j.amc.2004.12.021

Tavakkoli-Moghaddam, R., Aryanezhad, M.-B., Safaei, N., Vasei, M., & Azaron, A. (2007). A new approach for the cellular manufacturing problem in fuzzy dynamic conditions by a genetic algorithm. *Journal of Intelligent and Fuzzy Systems, 18*(4), 363-376.

Tavakkoli-Moghaddam, R., Javadian, N., Javadi, B., & Safaei, N. (2007). Design of a facility layout problem in cellular manufacturing systems with stochastic demands. *Applied Mathematics and Computation, 184*(2), 721-728. doi: 10.1016/j.amc.2006.05.172

Tavakkoli-Moghaddam, R., Safaei, N., & Babakhani, M. (2005). Solving a dynamic cell formation problem with machine cost and alternative process plan by memetic algorithms *Stochastic Algorithms: Foundations and Applications* (pp. 213-227): Springer.

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

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

Won, Y., & Currie, K. (2007). Fuzzy ART/RRR-RSS: a two-phase neural network algorithm for part-machine grouping in cellular manufacturing. *International Journal of Production Research, 45*(9), 2073-2104.

Yu, J., & Sarker, B. R. (2006). A directional decomposition heuristic for one-dimensional, non-equidistant machine-cell location problems. *Computers & operations research, 33*(1), 64-92.

Zhang, Z. (2011). Modeling complexity of cellular manufacturing systems. *Applied Mathematical Modelling, 35*(9), 4189-4195.

Zhou, M., & Askin, R. G. (1998). Formation of general GT cells: an operation-based approach. *Computers & industrial engineering, 34*(1), 147-157.