Linear programming embedded particle swarm optimization for solving an extended model of dynamic virtual cellular manufacturing systems

H. Rezazadeh*, M. Ghazanfari, S. J. Sadjadi, Mir.B. Aryanezhad, A. Makui

Department of Industrial Engineering, Iran University
of Science and Technology, Tehran, Iran
Tel.: 0098-914-4027521; Fax: 0098-171-2240855;
*Hassan.Rezazadeh@gmail.com
h-rezazadeh@tabrizu.ac.ir

ABSTRACT
The concept of virtual cellular manufacturing system (VCMS) is finding acceptance among researchers as an extension to group technology. In fact, in order to realize benefits of cellular manufacturing system in the functional layout, the VCMS creates provisional groups of resources (machines, parts and workers) in the production planning and control system. This paper develops a mathematical model to design the VCMS under a dynamic environment with a more integrated approach where production planning, system reconfiguration and workforce requirements decisions are incorporated. The advantages of the proposed model are as follows: considering the operations sequence, alternative process plans for part types, machine time-capacity, worker time-capacity, cross-training, lot splitting, maximal cell size, balanced workload for cells and workers. An efficient linear programming embedded particle swarm optimization algorithm is used to solve the proposed model. The algorithm searches over the 0-1 integer variables and for each 0-1 integer solution visited; corresponding values of integer variables are determined by solving a linear programming sub-problem using the simplex algorithm. Numerical examples show that the proposed method is efficient and effective in searching for near optimal solutions.

RESUMEN
El concepto de sistema de manufactura celular virtual (SMCV) está siendo aceptado entre los investigadores como una extensión de la tecnología de grupos. De hecho, para hacer realidad los beneficios del sistema de manufactura celular en el diseño funcional, el SMCV crea grupos provisionales de recursos (máquinas, partes y trabajadores) en la planificación de la producción y el sistema de control. En el presente trabajo se describe el desarrollo de un modelo matemático para diseñar el SMCV en el marco de un entorno dinámico con un enfoque más integrado en donde se incorporan la planificación de la producción, la reconfiguración del sistema y las decisiones relacionadas con los requisitos de la fuerza de trabajo. Las ventajas del modelo propuesto son las siguientes: considera la secuencia de operaciones, planes de proceso alternativos según los tipos de partes, tiempo de trabajo de la máquina, tiempo de trabajo del trabajador, capacitación mixta, división del trabajo, tamaño máximo de la célula y carga de trabajo balanceada para las células y trabajadores. Para resolver el modelo propuesto se usa un algoritmo eficiente de optimización por enjambre de partículas embebidas de programación lineal. El algoritmo busca en las variables enteras 0-1 y cada variable entera 0-1 visitada; los valores correspondientes de las variables enteras se determinan resolviendo una parte de un problema de programación lineal por medio del algoritmo simple. Mediante ejemplos numéricos se demuestra que el método propuesto es eficiente y efectivo en la búsqueda de soluciones casi óptimas.

Keywords: Dynamic virtual cellular manufacturing system; production planning; particle swarm optimization; linear programming

1. Introduction

The Virtual cellular manufacturing system (VCMS) belongs to the family of modern production methods, which many industrial sectors have used beneficially in recent years. A VCMS is aimed at increasing the efficiency of production and system flexibility by utilizing the production control system. Identifying logical groups of resources within the production control system, offers the possibility of achieving advantages of cellular manufacturing in situations where traditional cellular manufacturing systems may not be feasible. Resulting advantages may include the improved flow performance, higher efficiency, simplified production control and better quality. A schema of VCMS including machine and worker sharing between virtual cells for two consecutive periods is shown in Fig.1. In this figure, it is assumed that nine machines and six workers exist in layout. Because of the processing requirements, the logical grouping of machines and workers (virtual cells) is changed from period 1 to period 2.

Research on VCMS has gained momentum during the last decade. Recent studies on VCMS have focused on improvements in queue-related performance measure of the job shop. The emphasis has been on the performance evaluation of VCMS compared to traditional functional and cellular layout. There has been little research until now on the design of VCMS forming the subject of this paper. The objective of this paper is to propose a new mathematical model for integrated virtual cellular manufacturing system designing with production planning, dynamic reconfiguration and workforce requirements.

The remainder of this paper is organized as follows: In Section 2, we review relevant literature on the VCMS. Section 3 presents the mathematical formulation for the VCMS. In Section 4, we introduce a brief review of particle swarm optimization. Implementation of linear programming embedded particle swarm optimization is described in Section 5. Computational results are reported in Section 6 and the conclusion is given in Section 7.

2. Literature review

The concept of VCMS was introduced at the National Bureau of Standards (NBS) to address the specific control problems found in the design phase of the automated manufacturing batches of machined parts (Simpson et al. 1982). Montreuil et al (1992) introduced the idea that the logical system can be separated from the physical system, i.e., it is not necessary to have a functional organization if a process layout is in place and that a product organization is not exclusively

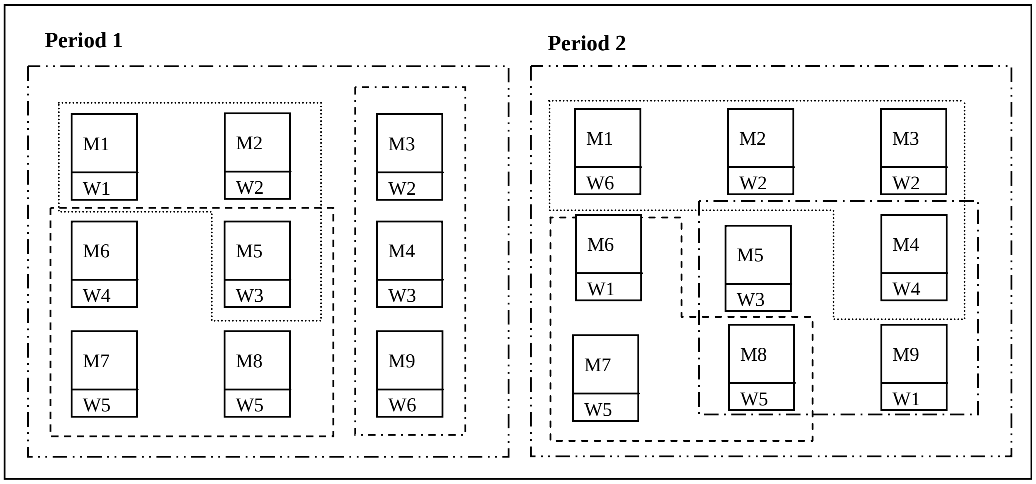

Figure 1. A schema of changing virtual cells in relation to changing production needs

associated with a product layout. Other paper in this stream presents a link with the eventual ability to move resources to accommodate to changing manufacturing requirements, i.e. the possibility to use dynamic cells (Rheault et al. (1995), Drolet et al. (1996)). In this sense, a VCMS is to be associated with a specific parameter range of the dynamic facility layout problem (see, for example, Balakrishnan and Chang (1998)) i.e. , when the product mix and volumes change so much in relation to the relocation equipment costs so that changing the facility layout is never worth the effort. Vakharia et al. (1999) compared the performance of virtual cells and multi-stage flow shops through analytical approximations. Some advantages of this study were the number of processing stages, the number of machines per processing stage, the batch size and ratio of setup to run time per batch for the implementation of the virtual cells. Ratchev (2001) proposed a four phase procedure for the virtual cell formation. In the first phase, processing alternatives are generated; in the second phase, the capability of boundaries of the virtual cell is defined; in the third phase, machine tools are selected, and finally, in the fourth phase, the performance of the system is evaluated. Sarker and Li (2001) suggested an approach for virtual cell formation with special emphasis on job routing and scheduling rather than on cell sharing. The basic feature of their approach lies in the identification of a sequence of machines to minimize a job throughout time in a multistage production system where there are multiple identified machines per stage and a job can only be assigned to one machine per stage. Thomalla (2000) addressed the same problem, but with the objective of minimizing tardiness. In this work, the problem is solved by using a Lagrangian relaxation approach. Irani et al. (1993) proposed a two-stage procedure which was a combination of the graph theoretic approach and the mathematical programming approach for forming virtual manufacturing cells. Subash et al. (2000) proposed a framework for virtual cell formation. In essence, the authors tested several clustering algorithms for the formation of virtual cells. Saad et al. (2002) also presented an integrated framework in a three-step approach for production planning and cell formation. They studied the possibility of using virtual cells as a reconfiguration strategy. Besides the above issues, there is a number of papers published on virtual cell formation. Most of them are controlled or simulation-oriented. Furthermore, those papers that specially address the formation of virtual cells are dedicated to special part families. On the other hand, the aspects of shared cell formulation have not received much attention. Mak and Wang (2002) proposed a new genetic-based scheduling algorithm to minimize the total material and component traveling distance incurred when manufacturing the product with the review to set up virtual manufacturing cells and to formulate feasible production schedules for all manufacturing operations. The proposed algorithm differs from the conventional genetic algorithms in that the populations of the candidate solutions consist of individuals from various age-groups, and each individual is incorporated with an age attribute to enable its birth and survival rates to be governed by predefined ageing patterns. In 2005, Mak and et al. improved their methodology by adding another objective of minimizing the sum of tardiness of all products. Baykasoglu (2003) proposed a simulated annealing algorithm for developing a distributed layout for a virtual manufacturing cell. Nomden et al. (2006) classified the virtual cell formation procedures into three main classes: design, operation and empirical. A comprehensive taxonomy and review of prior research in the area of VCMS can be found in their study. Nomden et al. (2008) studied parallel machine shops that implemented the concept of VCMS for production control. They strived to have a more comprehensive study on the relevance of routing configuration in VCMS.

Some authors proposed that workforce requirements should be taken into account at the cell design stage. Min and Shin (1993) and Suresh and Slomp (2001) proposed cell design procedures in which the complex cell formation problem is solved in two or more phases. The last phase in both procedures concerns workforce requirements. A basic assumption in the problem formulation of Min and Shin (1993) is that workers are linked with the various parts by means of so-called 'skill matching factors'. A skill matching factor indicates to what extent a worker is able to produce a part. These factors are used for the optimization of the worker assignment problem. Cross-training issues were not considered in this work. Suresh and Slomp (2001), in the last phase of their procedure, address various workforce requirements such as the partitioning of functionally specialized worker pools and the required additional training of workers. The need for cross-training is predetermined in their approach by setting minimum and maximum levels for the multi-functionality of workers and the redundancy of machines. They do not determine the need for cross-training analytically. Suer (1996) presented a two-phase hierarchical methodology for operator assignment and cell loading in worker-intensive manufacturing cells. Here, the major concern is determination of the number of workers in each cell and the assignment of workers to specific operations in such a way that worker productivity is maximal. A functional arrangement of tasks was assumed in each cell without considering training and multi-functionality problems. Askin and Huang (2001) focused on the relocation of workers into cells and the training needed for effective cellular manufacturing. They proposed a mixed integer, goal-programming model for guiding the worker assignment and training process. The model integrates psychological, organizational, and technical factors. They presented greedy heuristics as means to solve the problem. Askin and Huang (2001) assumed that the required skills are cell dependent and that workers may need some additional training, again without considering cross-training issues. Norman et al. (2002) presented a mixed integer programming formulation for the assignment of workers to operations in a manufacturing cell. Their formulation permits the ability to change the skill levels of workers by providing them with additional training and training decisions taken in order to balance the productivity and output quality of a manufacturing cell and the training costs. Slomp et al. (2004, 2005) presented a framework for the design of VMCS, specifically accounting for the limited availability of workers and worker skills. They propose a goal programming formulation that first groups jobs and machines and then assigns workers to the groups to form VMCS. The objective is to use the capacity as efficiently as possible, but also to have VMCS in places that are as independent as possible.

3. Problem formulation

In this section, we develop a new mixed-integer programming model to design the VCMS under a dynamic environment with a more integrated approach where production planning, system reconfiguration and workforce requirements decisions are incorporated. Figure 2 presents a graphical description of the model.

The model is formulated under the following assumption.

3.1 Assumptions

1. Each part type has a number of operations that must be processed respectively as numbered.
2. The processing time for all operations of a part type on different machine types are known and deterministic.

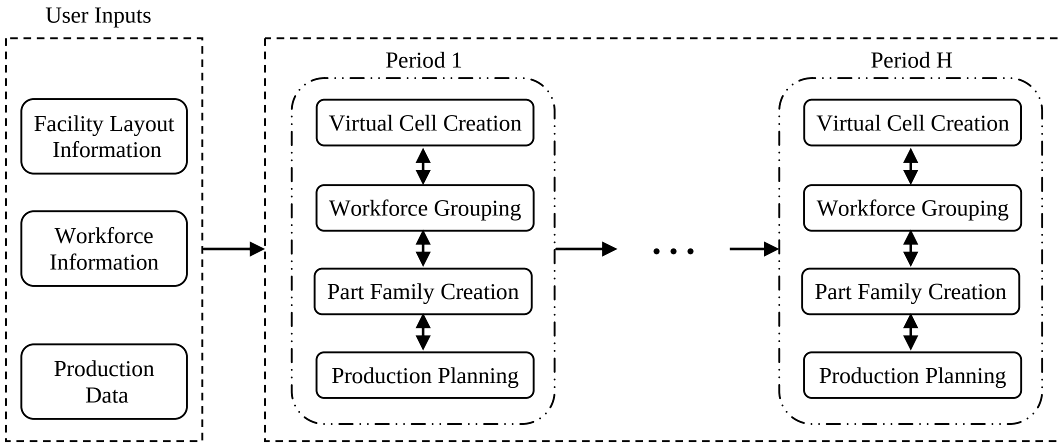

Figure 2. Structure of the model

3. The demand for each part type in each period is known and deterministic.

4. The capabilities and time-capacity of each machine type are known and constant over the planning horizon.

5. The skills and time–capacity of each worker are known and constant over the planning horizon.

6. All workers are assumed to be multi-functional. Thus, each worker can be able to operate at least two machines.

7. The ability of each worker for training on individual machines is known and constant over the planning horizon.

8. The manufacturing cost of each machine type is known. The manufacturing cost implies the operating cost that is independent of the workload allocated to machine.

9. The distance between two machine locations in layout is first known as a prior.

10. All machine types are assumed to be multipurpose. Thus, each part type can have several alternative process routing with different processing times.

11. Backorders are not allowed. All demands must be satisfied in the given period.

12. Finished parts inventory is allowed in the production system.

3.2 Notation

Indexes

- h index for time periods ($h=1,\dots,H$)
- c index for virtual cells ($c=1,\dots,C$)
- w index for workers ($w=1,\dots,W$)
- m index for machine types ($m=1,\dots,M$)
- p index for part types ($p=1,\dots,P$)
- j index for operations belong to part type p ($j=1,\dots,K_p$)
- c_{jp} index for virtual cell used to process operation j of part type p
- m_{jp} index for machine type used to process operation j of part type p

3.3 Input parameters

H	number of time periods
C	number of virtual cells
W	number of workers
M	number of machine types
P	number of part types
K_p	number of operations for part type *p*
D_{ph}	demand for part *p* in period *h*
α_m	operating cost per unit time per machine type *m*
t_{jpm}	processing time required to perform operation *j* of part type *p* on machine type *m*
γ_p	unit cost to move part type *p* between machines
d_{mn}	distance between machine locations *m,n*
s_p	subcontracting cost per part type
i_p	inventory holding cost per part type *p* per time period
β_p	internal production cost per part type *p*
tr_m	cost of training a worker for machine type *m*
T_m	time-capacity of machine type *m* in each period
T_w	time-capacity of worker *w* in each period
q_1	factor of workload balancing between virtual cells
q_2	factor of workload balancing between workers
UB	Maximal virtual cell size
a_{wmh}	= 1 if worker *w* has ability to operating on machine type *m* in each period
b_{wm}	= 1 if worker *w* has capability to training on machine type *m*

3.4 Decision variables

X_{jpmwch} =1 if operation *j* of part type *p* is done on machine type *m* by worker of *w* in virtual cell *c* in period *h*

Q_{pmh} = the quantity of parts of type *p* processed on machine type *m* in period *h*

SC_{ph} = the quantity of parts of type *p* subcontracted in period *h*

IH_{ph} = the quantity of inventory of part type *p* kept in period *h* and carried over to period *h+1*

3.5 Mathematical model

By using the above notation, the nonlinear mathematical formulation for the VCMS is presented as follows:

$$\min \quad Z = \sum_{h=1}^{H}\sum_{c=1}^{C}\sum_{w=1}^{W}\sum_{m=1}^{M}\sum_{p=1}^{P}\sum_{j=1}^{K_p} \alpha_m . t_{jpm} . Q_{pmh} . X_{jpmwch} + \sum_{h=1}^{H}\sum_{c=1}^{C}\sum_{w=1}^{W}\sum_{m=1}^{M}\sum_{p=1}^{P}\sum_{j=1}^{K_p} \beta_p . Q_{pmh} . X_{jpmwch}$$
$$+ \sum_{h=1}^{H}\sum_{w=1}^{W}\sum_{p=1}^{P}\sum_{j=1}^{K_p-1} \gamma_p . d_{(m_{jp})(m_{j+1,p})} . Q_{pmh} . X_{jpm_{jp}wc_{jp}h} . X_{j+1,pm_{j+1,p}wc_{j+1,p}h} + \sum_{h=1}^{H}\sum_{p=1}^{P} i_p . IH_{ph} + \sum_{h=1}^{H}\sum_{p=1}^{P} s_p . \quad (1)$$
$$+ \sum_{h=1}^{H}\sum_{c=1}^{C}\sum_{w=1}^{W}\sum_{m=1}^{M}\sum_{p=1}^{P}\sum_{j=1}^{K_p} b_{wm} . (1-a_{wmh}) . tr_m . X_{jpmwch}$$

Subject to:

$$\sum_{c=1}^{C}\sum_{w=1}^{W}\sum_{p=1}^{P}\sum_{j=1}^{K_p} t_{jpm}.Q_{pmh}.X_{jpmwch} \le T_m \quad \forall m,h \tag{2}$$

$$\sum_{c=1}^{C}\sum_{m=1}^{M}\sum_{p=1}^{P}\sum_{j=1}^{K_p} t_{jpm}.Q_{pmh}.X_{jpmwch} \le T_w \quad \forall w,h \tag{3}$$

$$\sum_{c=1}^{C}\sum_{w=1}^{W}\sum_{m=1}^{M} Q_{pmh}.X_{jpmwch} + IH_{p,h-1} + SC_{ph} - IH_{ph} = D_{ph} \quad \forall j,p,h \tag{4}$$

$$\sum_{w=1}^{W}\sum_{m=1}^{M}\sum_{p=1}^{P}\sum_{j=1}^{K_p} t_{jpm}.Q_{pmh}.X_{jpmwch} \ge \frac{q_1}{C}\sum_{c=1}^{C}\sum_{w=1}^{W}\sum_{m=1}^{M}\sum_{p=1}^{P}\sum_{j=1}^{K_p} t_{jpm}.Q_{pmh}.X_{jpmwch} \quad \forall c,h \tag{5}$$

$$\sum_{c=1}^{C}\sum_{m=1}^{M}\sum_{p=1}^{P}\sum_{j=1}^{K_p} t_{jpm}.Q_{pmh}.X_{jpmwch} \ge \frac{q_2}{W}\sum_{c=1}^{C}\sum_{w=1}^{W}\sum_{m=1}^{M}\sum_{p=1}^{P}\sum_{j=1}^{K_p} t_{jpm}.Q_{pmh}.X_{jpmwch} \quad \forall w,h \tag{6}$$

$$\sum_{m=1}^{M} X_{jpmwch} \le UB \quad \forall j,p,w,c,h \tag{7}$$

$$a_{wmh} = b_{wm}.X_{jpmwc,h-1}.(1-a_{wm,h-1}) + a_{wm,h-1} \quad \forall j,p,m,w,c,h \ge 2 \tag{8}$$

$$X_{jpmwch} = a_{wmh} + b_{wm} - a_{wmh}\,.\,b_{wm} \quad \forall j,p,m,w,c,h \tag{9}$$

$$X_{jpmwch} \in \{0,1\} \quad \forall j,p,m,w,c,h \tag{10}$$

$$Q_{pmh} \ge 0 \text{ and integer.} \quad \forall p,m,h \tag{11}$$

$$SC_{ph} \ge 0 \text{ and integer.} \quad \forall p,h \tag{12}$$

$$IH_{ph} \ge 0 \text{ and integer.} \quad \forall p,h \tag{13}$$

The objective function given in Eq. (1) is to minimize the total sum of the manufacturing cost, material handling cost, subcontracting cost, inventory holding cost, internal production cost and needed cost of cross-training for workers over the planning horizon. The first term represents the manufacturing cost of all machines in all virtual cells over the planning horizon. It is the sum of the product of the time-workload allocated to each machine type and their associated cost. This term causes a balance between the workload assigned to machines at each virtual cell. The third term computes the total distance traveled by the materials and component parts for manufacturing all the incoming demand in each period. So, the part demand can be satisfied in each period through internal production, subcontracting or inventory carried over from the previous period, the second, fourth and fifth terms computes this costs. Finally, the sixth term calculates the total needed costs of cross-training for workers.

Eqs. (2),(3) show how machine and worker capacity constraints are respected. Eq. (4) is the relationship between internal production, inventory holding and subcontracting levels in each period over the planning horizon. Eqs. (5),(6) enforce workload balance among virtual cells and workers, respectively, where the factors of $q_1, q_2 \in [0,1)$, in each inequality, are used to determine the extent of the workload balance. Eq. (7) ensures the maximal virtual cell size is not violated. Eq. (8) updates the skill matrix of workers in the beginning of each period. Eq. (9) ensures that the worker assigned to the machine has the needed skill in each period. Finally, constraint sets of (10)-(13) represent the logical binary and non-negativity integer requirements on the decision variables.

3.6 Linearization of the proposed model

Eq. (1) is a non-linear integer equation. The transformation of the non-linear terms of the objective function into linear terms can be performed by using the procedures given below.

Linearization of the first and second terms: the first and second terms can be linearized by introducing a non-negative variable Y_{jpmwch}. The transformation equation is as follows:

After, these three terms are linearized; the objective function of the integer programming model includes linear terms only. All constraints in the model are also linear. The number of variables and number of constraints in the linearized model are presented in Table 1, based on the variable indices.

$$Y_{jpmwch} = Q_{pmh} \cdot X_{jpmwch} \qquad (14)$$

Where below constraint must be added to the original model.

$$Y_{jpmwch} \leq M \cdot X_{jpmwch} \qquad \forall\, j,p,m,w,c,h \qquad (15)$$

Where, M is a large positive value.

Linearization of the third term: the third term of the objective function can be linearized by introducing a non- negative variable $W_{jpmnwckh}$, and a binary variable $Z_{jpmnwckh}$. The transformation equation is as follows:

$W_{jpmnwckh} = Z_{jpmnwckh} \cdot Q_{pmh}$, where $Z_{jpmnwckh} = X_{jpmwch} \cdot X_{j+1,pnwkh}$, under the following sets of constraints:

$$Z_{jpmnwckh} \geq X_{jpmwch} + X_{j+1,pnwkh} - 1 \qquad \forall\, j,p,m,n,w,c,k,h \qquad (16)$$

$$W_{jpmnwckh} \leq M \cdot X_{jpmwch} \qquad \forall\, j,p,m,n,w,c,k,h \qquad (17)$$

Variable name	Variable type	Variable count	Constraint	Total count
X_{jpmwch}	Binary	KP×M×W×C×H	(2)	M×H
$Z_{jpmnwckh}$	Binary	KP×M^2×W×C^2×H	(3)	W×H
Q_{pmh}	Integer	P×M×H	(4)	KP×W×H
Y_{jpmwch}	Integer	KP×M×W×C×H	(5)	C×H
$W_{jpmnwckh}$	Integer	KP×M^2×W×C^2×H	(6)	W×H
IH_{ph}	Integer	P×H	(7)	KP×W×C×H
SC_{ph}	Integer	P×H	(8)	KP×M×W×C×(H-1)
			(9)	KP×M×W×C×H
			(15)	KP×M×W×C×H
			(16)	KP×M^2×W×C^2×H
			(17)	KP×M^2×W×C^2×H

KP: Total number of operations in all of the parts.

Table 1 Number of variables and constraints in the linearized model

4. Particle swarm optimization (PSO)

4.1 Brief review of particle swarm optimization

The particle swarm optimization (PSO) algorithm was first proposed by Kennedy and Eberhart (Kennedy and Eberhart, 1995) and had exhibited many successful applications, ranging from evolving weights and structure for artificial neural networks (Eberhart and Shi, 1998), manufacture end milling (Tandon, 2000), reactive power and voltage control (Yoshida et al., 1999), to state estimation for electric power distribution systems (Shigenori, 2003). The convergence and parameterization aspects of the PSO have also been discussed thoroughly (Clerc and Kennedy, 2002).

The PSO is inspired by observations of birds flocking and fish schooling. Birds/fish flock synchronously, change direction suddenly, and scatter and regroup together. Each individual, called a particle, benefits from the historical experience of its own and that of the other members of the swarm during the search for food. The PSO models the social dynamics of birds/fish and serves as an optimizer for nonlinear functions.

4.2 Discrete particle swarm optimization

In the discussion above, the PSO is restricted in real number space. However, many optimization problems are set in a space featuring discrete or qualitative distinctions between variables. To meet the need, Kennedy and Eberhart (Kennedy and Eberhart, 1997) developed a discrete version of PSO. The discrete PSO essentially differs from the original (or continuous) PSO in two characteristics: First, the particle is composed of the binary variable; second, the velocity must be transformed into the change of probability, which is the chance of the binary variable taking value one.

Let $X_i^t = (x_{i1}^t, x_{i2}^t, \ldots, x_{iD}^t), x_{id}^t \in \{0, 1\}$ be particle i with D bits at iteration t, where X_i^t being treated as a potential solution has a rate of change called velocity. Denote the velocity as $V_i^t = (v_{i1}^t, v_{i2}^t, \ldots, v_{iD}^t), v_{id}^t \in R$.

Let $P_i^t = (p_{i1}^t, p_{i2}^t, \ldots, p_{iD}^t)$ be the best solution that particle i has obtained until iteration t, and $P_g^t = (p_{g1}^t, p_{g2}^t, \ldots, p_{gD}^t)$ be the best solution obtained from P_i^t in the population (gbest) or local neighborhood (lbest) at iteration t.

As in continuous PSO, each particle adjusts its velocity according to the cognition part and the social part. Mathematically, we have:

$$v_{id}^t = v_{id}^{t-1} + c_1 r_1 (p_{id}^t - x_{id}^t) + c_2 r_2 (p_{gd}^t - x_{id}^t), \quad (18)$$

Where c_1 is the cognition learning factor, c_2 is the social learning factor, and r_1 and r_2 are random numbers uniformly distributed in [0, 1]. Eq. (6) specifies that the velocity of a particle at iteration t is determined by the previous velocity of the particle, the cognition part, and the social part. Values $c_1.r_1$, $c_2.r_2$ determine the weights of the two parts, where their sum is usually limited to 4 (Kennedy and Eberhart, 2001).

By Eq. (6), each particle moves according to its new velocity. Recall that particles are represented by binary variables. For the velocity value of each bit in a particle, Kennedy and Eberhart (Kennedy and Eberhart, 1997) claim that a higher value is more likely to choose 1, while a lower value favors the 0 choice. Furthermore, they constrain the velocity value to the interval [0, 1] by using the following sigmoid function:

$$s(v_{id}^t) = \frac{1}{1 + \exp(-v_{id}^t)}, \quad (19)$$

Where $s(v_{id}^t)$ denotes the probability of bit x_{id}^t taking 1. To avoid $s(v_{id}^t)$ approaching 0 or 1, a constant V_{max} is used to limit the range of v_{id}^t . In practice, V_{max} is often set at 4, i.e., . $v_{id}^t \in [-V_{max}, +V_{max}]$.

Kennedy et al. [19] gave the pseudo-code of discrete PSO as follows (for maximization problem):

Loop
 For i = 1 to Np
  If $G(X_i^t) > G(P_i^t)$ then // G( ) evaluates objective function
   For d = 1 to D bits
    $p_{id}^t = x_{id}^t$ //p_{id}^t is best so far
   Next d
  End if
  g = i //arbitrary
  For j=indices of neighbors (or population)
   If $G(P_j^t) > G(P_g^t)$ then g = j //g is index of best performer in neighborhood (or population)
  Next j
  For d = 1 to D
   $v_{id}^t = v_{id}^{t-1} + c_1 r_1 (p_{id}^t - x_{id}^t) + c_2 r_2 (p_{gd}^t . x_{id}^t)$
   $v_{id}^t \in [-V_{max}, +V_{max}]$
   $s(v_{id}^t) = \frac{1}{1 + \exp(-v_{id}^t)}$,
   If random number < $s(v_{id}^t)$ then $x_{id}^{t+1} = 1$; else $x_{id}^{t+1} = 0$
  Next d
 Next i
Until criterion

5. Linear programming embedded particle swarm optimization

In this section, we develop a linear programming embedded particle swarm optimization algorithm (LPEPSO) in order to solve the model presented in Section 3 efficiently for large data set. For a given solution point, the value of 0-1 binary decision variables (X_{jpmwch}) is obtained by decoding the solution representation. To compute the corresponding values of the integer variables and the value of the objective function, a LP sub-problem is solved using the simplex algorithm in lingo 8.0 software. The main idea of embedding a simplex algorithm in a meta-heuristic is similar to that presented in Teghem et al. (1995). The advantage of embedding an LP sub-problem in the particle swarm optimization algorithm can be explained as follows: For a given solution of 0-1 binary decision variables, there may be infinite combinations of the values for the integer variables.; however, by solving the LP sub-problem, values that optimally correspond to the integer solution can be obtained easily. It is also important to note that the solution of the LP sub-problem satisfies several constraints having integer variables which otherwise may be difficult to satisfy by using particle swarm optimization search alone. The steps of LPEPSO are represented in the flow chart given in Fig. 3.

5.1 Encoding

The most important issue in applying PSO successfully is to develop an effective 'problem mapping' mechanism. The solution encoding of the proposed model involves the 0-1 binary decision variables X_{jpmwch} enabling a randomly generated solution. Fig. 4 illustrates a particle structure assuming P part type (P_1-P_P) are to be processed on M machines (M_1-M_M) by W workers (W_1-W_W) in C cells (C_1-C_C) during H planning period. A segment corresponding to a given time period has three sub-segments: the first sub-segment, labeled ‘‘Machines’’, represents the operation assignment of the parts to various machines, the second sub-segment, labeled ‘‘Workers’’, represents the operation assignment of the parts to various workers and the three sub-segment, labeled ‘‘Cells’’, represents the machine configurations. In this figure, P_P in the sub segments of Machines, Workers and cells in period 1 are shown in detail:

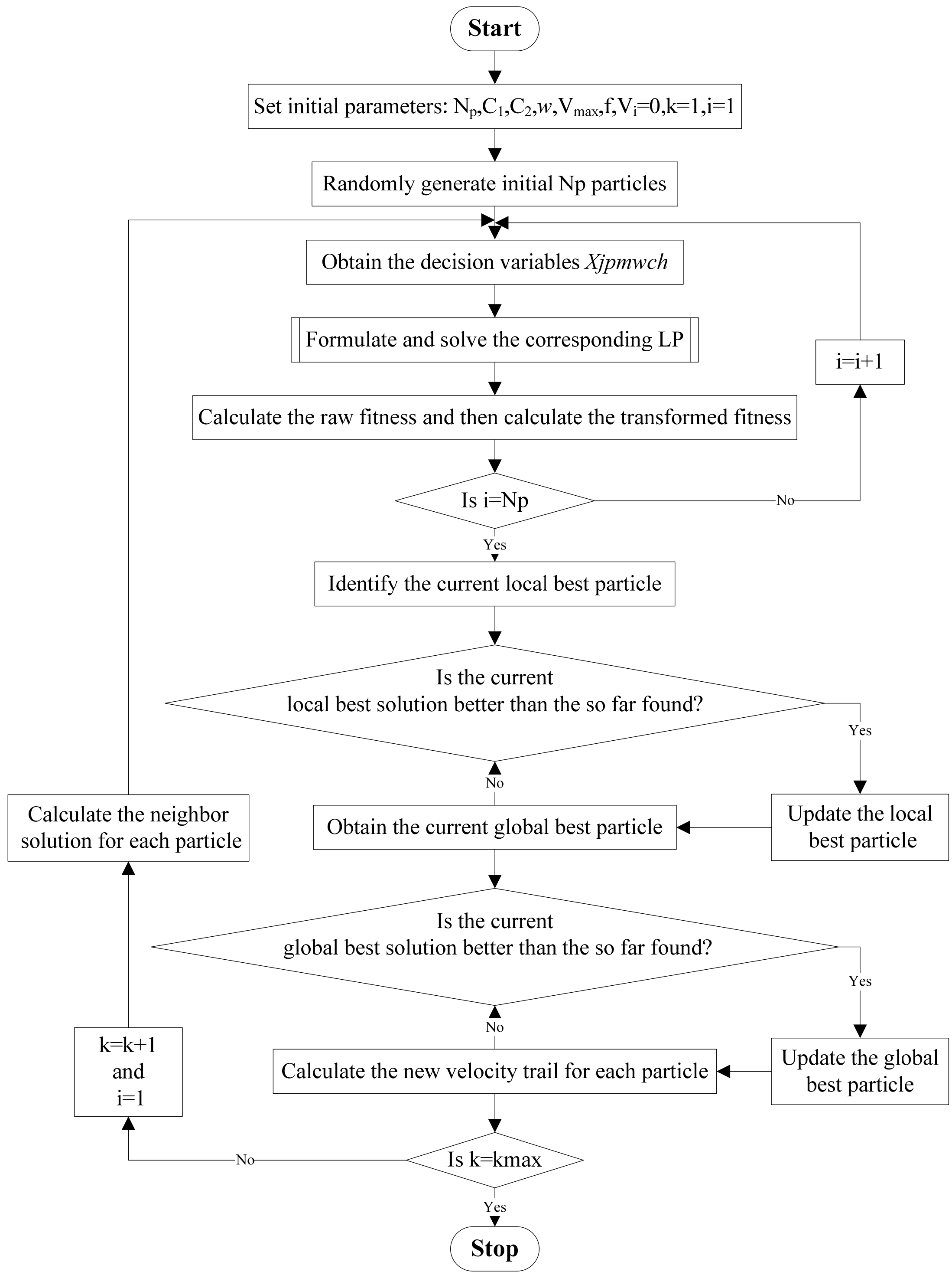

Figure 3. The steps of LPEPSO

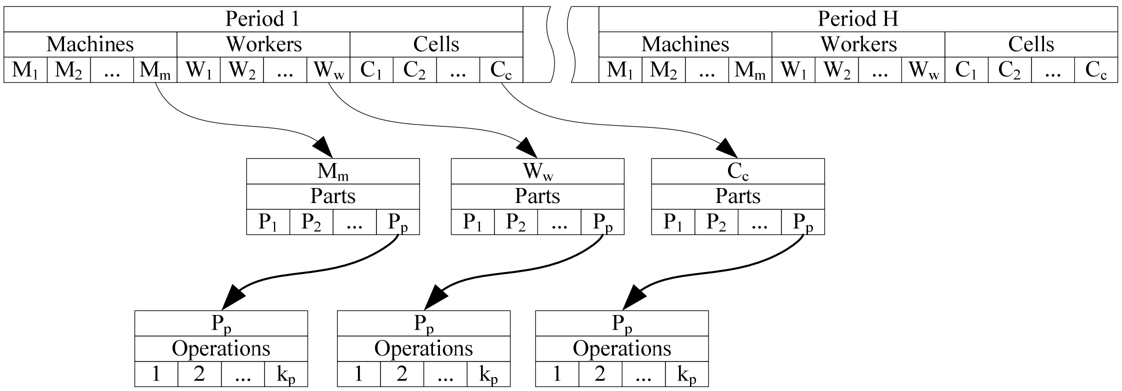

Figure 4. Solution representation

5.2 Definition of discrete particle

We define particle i at iteration t as $X_i^t = (X_{i1}^t, X_{i2}^t, ..., X_{iH}^t)$, as $X_{ih}^t = (x_{ih111}^t, ..., x_{ihK_pPM}^t$, $y_{ih111}^t, ..., y_{ihK_pPW}^t$, $z_{ih111}^t, ..., z_{ihK_pPC}^t)$, $x_{ihjpm}^t, y_{ihjpw}^t, z_{ihjpc}^t \in \{0,1\}$, $h=1,...,H$, where x_{ihjpm}^t equal 1 if operation j of part type p of particle i is assigned to machine type m in period h and 0 otherwise, y_{ihjpw}^t equal 1 if operation j of part type p of particle i is assigned to worker w in period h and 0 otherwise and Z_{ihjpc}^t equal 1 if operation j of part type p of particle i is assigned to cell c in period h and 0 otherwise. The value of decision variable X_{jpmwch} equals 1 if x_{ihjpm}^t, y_{ihjpw}^t and Z_{ihjpc}^t equal 1 and 0 otherwise. For example, suppose the sequence of X_i^t is {(111,1,1,1), (111,2,2,2), (112,2,2,2), (121,2,2,2), (122,1,1,1), (211,1,2,1), (212,1,1,1), (221,2,2,2), (222,2,2,2)}, ((hpj,m,w,c) denotes that operation j of part type p in period h is assigned to mth machine, wth worker and cth cell). By this definition, we have $X_{111111}=1$, $X_{112221}=1$, $X_{212221}=1$, $X_{122221}=1$, $X_{221111}=1$, $X_{111212}=1$, $X_{211112}=1$, $X_{121112}=1$, $X_{222222}=1$. (see Fig. 5).

5.3 A linear programming sub-problem

The values of all the 0-1 binary decision variables obtained by decoding a particle as explained in the previous section. The corresponding values of integer variables Q_{pmh}, SC_{ph}, IH_{ph} determined by solving a linear programming sub-problem given below. This LP sub-problem is to minimize the total sum of the manufacturing cost, material handling cost, subcontracting cost, inventory holding cost and internal production cost, subject to the constraints in Eqs. (2)–(6). In this LP sub-problem, these constraints are renumbered as Eqs. (21)–(25).

	Period 1																							
	Machines								Workers								Cells							
	M_1				M_2				W_1				W_2				C_1				C_2			
Parts	P_1		P2		P_1		P2		P_1		P2		P_1		P2		P_1		P2		P_1		P2	
Operations	o_1	o_2	o_1	o_2	o_1	o_2	o_1	o_2	o_1	o_2	o_1	o_2	o_1	o_2	o_1	o_2	o_1	o_2	o_1	o_2	o_1	o_2	o_1	o_2
	1	0	0	1	1	1	1	0	1	0	0	1	1	1	1	0	1	0	0	1	1	1	1	0

	Period 2																							
	Machines								Workers								Cells							
	M_1				M_2				W_1				W_2				C_1				C_2			
Parts	P_1		P2		P_1		P2		P_1		P2		P_1		P2		P_1		P2		P_1		P2	
Operations	o_1	o_2	o_1	o_2	o_1	o_2	o_1	o_2	o_1	o_2	o_1	o_2	o_1	o_2	o_1	o_2	o_1	o_2	o_1	o_2	o_1	o_2	o_1	o_2
	1	1	1	0	0	0	0	1	0	1	1	0	1	0	0	1	1	1	1	0	0	0	0	1

Figure 5. Definition of particle X_i^t

$$\min Z_{LP} = \sum_{h=1}^{H}\sum_{c=1}^{C}\sum_{w=1}^{W}\sum_{m=1}^{M}\sum_{p=1}^{P}\sum_{j=1}^{K_p} \alpha_m.t_{jpm}.Q_{pmh}.X_{jpmwch} + \sum_{h=1}^{H}\sum_{c=1}^{C}\sum_{w=1}^{W}\sum_{m=1}^{M}\sum_{p=1}^{P}\sum_{j=1}^{K_p} \beta_p.Q_{pmh}.X_{jpmwch}$$

$$+ \sum_{h=1}^{H}\sum_{w=1}^{W}\sum_{p=1}^{P}\sum_{j=1}^{K_p-1} \gamma_p.d_{(m_{jp})(m_{j+1,p})}.Q_{pmh}.X_{jpm_{jp}wc_{jp}h}.X_{j+1,pm_{j+1,p}wc_{j+1,p}h} + \sum_{h=1}^{H}\sum_{p=1}^{P} i_p.\,IH_{ph} + \sum_{h=1}^{H}\sum_{p=1}^{P} s_p.\,SC_{ph} \quad (20)$$

Subject to

$$\sum_{c=1}^{C}\sum_{w=1}^{W}\sum_{p=1}^{P}\sum_{j=1}^{K_p} t_{jpm}.Q_{pmh}.X_{jpmwch} \le T_m \qquad \forall m,h \quad (21)$$

$$\sum_{c=1}^{C}\sum_{m=1}^{M}\sum_{p=1}^{P}\sum_{j=1}^{K_p} t_{jpm}.Q_{pmh}.X_{jpmwch} \le T_w \qquad \forall w,h \quad (22)$$

$$\sum_{c=1}^{C}\sum_{w=1}^{W}\sum_{m=1}^{M} Q_{pmh}.X_{jpmwch} + IH_{p,h-1} + SC_{ph} - IH_{ph} = D_{ph} \qquad \forall j,p,h \quad (23)$$

$$\sum_{w=1}^{W}\sum_{m=1}^{M}\sum_{p=1}^{P}\sum_{j=1}^{K_p} t_{jpm}.Q_{pmh}.X_{jpmwch} \ge \frac{q_1}{C}\sum_{c=1}^{C}\sum_{w=1}^{W}\sum_{m=1}^{M}\sum_{p=1}^{P}\sum_{j=1}^{K_p} t_{jpm}.Q_{pmh}.X_{jpmwch} \qquad \forall c,h \quad (24)$$

$$\sum_{c=1}^{C}\sum_{m=1}^{M}\sum_{p=1}^{P}\sum_{j=1}^{K_p} t_{jpm}.Q_{pmh}.X_{jpmwch} \ge \frac{q_2}{W}\sum_{c=1}^{C}\sum_{w=1}^{W}\sum_{m=1}^{M}\sum_{p=1}^{P}\sum_{j=1}^{K_p} t_{jpm}.Q_{pmh}.X_{jpmwch} \qquad \forall w,h \quad (25)$$

5.4. The fitness function

The purpose of the fitness function is to measure the fitness of the candidate solutions in the population with respect to the objective and constraint functions of the model. For a given solution, its fitness obtained by Eq. (26) as the sum of the objective function of the model (Eq. (1)) and the penalty terms of constraint violations. The value of the model objective function is the sum of the objective function of the LP sub-problem, needed costs of cross-training for workers over the planning horizon. The penalty terms are to enforce the cell size and worker skill constraints. Factors f_{cs} and f_{ws} are used for scaling these penalty terms. In this study, these factors are determined by trial and error where satisfactory values were obtained with little effort. Finally, for a minimization problem, the raw fitness score F needs to be transformed so that the minimum raw fitness will correspond to the maximum transformed fitness. This is achieved by using Eq. (27) where $\tilde{F}$ is the transformed fitness function.

$$\begin{aligned} F = {} & \text{Model Objective Function} \\ & + f_{cs}.\sum_{h=1}^{H}\sum_{c=1}^{C}\sum_{w=1}^{W}\sum_{p=1}^{P}\sum_{j=1}^{k_p} \max\left\{0, \sum_{m=1}^{M} X_{jpmwch} - UB\right\} \\ & + f_{ws}.\sum_{h=1}^{H}\sum_{c=1}^{C}\sum_{m=1}^{M}\sum_{w=1}^{W}\sum_{p=1}^{P}\sum_{j=1}^{k_p} \left\{X_{jpmwch} - a_{wmh} - b_{wm} + a_{wmh}\,.\,b_{wm}\right\} \end{aligned} \quad (26)$$

$$\tilde{F} = \begin{cases} 1 & ; \quad \text{if } F_{max} = F_{min}, \\ \dfrac{F_{max} - F}{F_{max} - F_{min}} & ; \quad \dfrac{F_{max} - F}{F_{max} - F_{min}} > 0.1, \\ 0.1 & ; \quad \text{otherwise.} \end{cases} \quad (27)$$

5.5 Velocity trail

After a period selected, to move a particle to a new sequence, we define the velocity of part of particle *i* at iteration *t* as $V^t_{ih}=(vx^t_{ih111},\ldots,vx^t_{ihK_PPM}, vy^t_{ih111},\ldots,vy^t_{ihK_PPW}, vz^t_{ih111},\ldots,vz^t_{ihK_PPC})$, $vx^t_{ihjpm}, vy^t_{ihjpw}, vz^t_{ihjpc} \in R$, where vx^t_{ihjpm} is the velocity value for operation *j* of part type *p* of particle *i* assigned to machine type *m* in period *h* at iteration *t*, vy^t_{ihjpw} is the velocity value for operation *j* of part type *p* of particle *i* assigned to worker *w* in period *h* at iteration *t* and vz^t_{ihjpc} is the velocity value for operation *j* of part type *p* of particle *i* is assigned to cell *c* in period *h* at iteration *t*. Velocity V^t_{ih}, called velocity trail, is inspired by the frequency-based memory (Onwubolu, 2002). The frequency-based memory records the number of times that an operation of parts visits a particular machine, worker or cells, and it is often used in combinatorial optimization, e.g., the long-term memory of tabu-search, to provide useful information that facilitates choosing preferred moves. Here, we make use of the similar concept to design the velocity trail. A higher value of vx^t_{ihjpm} in the trail indicates that operation *j* of part type *p* is more likely to be processed on machine type *m*, while a lower value favors assigning operation *j* of part *p* out of the *m*th machine; value of vy^t_{ihjpw} in the trail indicates that operation *j* of part type *p* is more likely to be assigned to worker *w*, while a lower value favors assigning operation *j* of part *p* out of the *w*th worker and value of vz^t_{ihjpc} in the trail indicates that operation *j* of part type *p* is more likely to be assigned to cell *c,* while a lower value favors assigning operation *j* of part *p* out of the *c*th cell. The particle's new velocity trail is updated by the following equations:

$$vx^t_{ihjpm} = w.\, vx^{t-1}_{ihjpm} + c_1r_1(p^t_{ihjpm} - x^t_{ihjpm}) + c_2r_2(p^t_{ghjpm} - x^t_{ihjpm})$$
$$vy^t_{ihjpw} = w.\, vy^{t-1}_{ihjpw} + c_1r_1(p^t_{ihjpw} - x^t_{ihjpw}) + c_2r_2(p^t_{ghjpw} - x^t_{ihjpw}) \quad (28)$$
$$vz^t_{ihjpc} = w.\, vz^{t-1}_{ihjpc} + c_1r_1(p^t_{ihjpc} - x^t_{ihjpc}) + c_2r_2(p^t_{ghjpc} - x^t_{ihjpc})$$

Here, $P^t_i=(P^t_{i1},P^t_{i2},\ldots,P^t_{iH})$, as $P^t_{ih}=(px^t_{ih111},\ldots,px^t_{ihK_PPM}, py^t_{ih111},\ldots,py^t_{ihK_PPW}, pz^t_{ih111},\ldots,pz^t_{ihK_PPC})$, $px^t_{ihjpm}, py^t_{ihjpw}, pz^t_{ihjpc} \in \{0,1\}$, $h=1,\ldots,H$, denotes the best solution that particle *i* has obtained until iteration *t*, $P^t_{gh}=(px^t_{gh111},\ldots,px^t_{ghK_PPM}, py^t_{gh111},\ldots,py^t_{ghK_PPW}, pz^t_{gh111},\ldots,pz^t_{ghK_PPC})$, $px^t_{ghjpm}, py^t_{ghjpw}, pz^t_{ghjpc} \in \{0,1\}$, $h=1,\ldots,H$, denotes the best solution obtained from particles in the population at iteration *t* and *w* is the inertia weight proposed by Shi and Eberhart (1998). A constant V_{max} use to limit the range of $vx^t_{ihjpm}, vy^t_{ihjpw}$ and vz^t_{ihjpc}, i.e., $vx^t_{ihjpm}, vy^t_{ihjpw}, vz^t_{ihjpc} \in [-V_{max}, +V_{max}]$.

We now explain the meaning of velocity trail. For simplicity, suppose there exist only the social part in Eqs. (28) and $c_2=r_2=1$. The sequence of X^t_i is assumed to be {(111,1,1,1), 111,2,2,2), 112,2,2,2), 121,2,2,2), 122,1,1,1), 211,1,2,1), 212,1,1,1), 221,2,2,2), 222,2,2,2)}, the first period is selected randomly and the sequence of P_{g1}^t be {(111,1,1,1), 112,2,1,2), 121,1,1,1), 122,1,2,2)}. It is clear that $Vx^t_{i1jpm}=P^t_{g1jpm}-X^t_{i1jpm}=1,0,-1$, $Vy^t_{i1jpw}=P^t_{g1jp}X^t_{i1jpw}=1,0,-1$ and $Vz^t_{i1jpc}=P^t_{g1jpc}-X^t_{i1jpc}=1,0,-1$ (see Fig. 6). Values 1 intensify the assignments of operation *j* of part *p* in the *m*th machine, *w*th worker and *c*th cell, respectively, whereas, values -1 diversify such assignments. In the calculation, we can simply add $P^t_{g1jpm}=1$, $P^t_{g1jpw}=1$ and $P^t_{g1jpc}=1$ to the corresponding Vx^t_{i1jpm}, Vy^t_{i1jpw} and Vz^t_{i1jpc}, subtract $X^t_{i1jpm}=1$, $X^t_{i1jpw}=1$ and $X^t_{i1jpc}=1$ from Vx^t_{i1jpm}, Vy^t_{i1jpw} and Vz^t_{i1jpc}, respectively, and leave others unchanged. If each one of the Vx^t_{i1jpm}, Vy^t_{i1jpw} and Vz^t_{i1jpc} is smaller than $-V_{max}$, then set it with $-V_{max}$; if each one of the Vx^t_{i1jpm}, Vy^t_{i1jpw} and Vz^t_{i1jpc} is greater than $+V_{max}$, then set it with $+V_{max}$.

	Period 1																							
	Machines								Workers								Cells							
	M_1				M_2				W_1				W_2				C_1				C_2			
Parts	P_1		P2		P_1		P2		P_1		P2		P_1		P2		P_1		P2		P_1		P2	
Operations	o_1	o_2	o_1	o_2	o_1	o_2	o_1	o_2	o_1	o_2	o_1	o_2	o_1	o_2	o_1	o_2	o_1	o_2	o_1	o_2	o_1	o_2	o_1	o_2
	0	0	1	0	-1	0	-1	0	0	1	1	-1	-1	-1	-1	1	0	0	1	-1	-1	0	-1	1

Figure 6. The resulting values of $Vx^t_{i1jpm}=P^t_{g1jpm}-X^t_{i1jpm}$, $Vy^t_{i1jpw}=P^t_{g1jpw}-X^t_{i1jpw}$, $Vz^t_{i1jpc}=P^t_{g1jpc}-X^t_{i1jpc}$

The above example and Eq. (28) demonstrate that the velocity trail is gradually accumulated by the individual's own experience and individual's companions' experience. This social behavior of sharing useful information among individuals in searching for the optimal solution is the merit of PSO over more classical meta-heuristics.

As in discrete PSO, the velocity trail values need to be converted from real numbers to the changes of probabilities by the following sigmoid functions:

$$s(vx^t_{ihjpm}) = \begin{cases} \dfrac{1}{1+exp(-vx^t_{ihjpm})} & if \quad t_{jpm} \neq 0 \\ 0 & if \quad t_{jpm} = 0 \end{cases} \tag{29}$$

$$s(vy^t_{ihjpw}) = \frac{1}{1+exp(-vy^t_{ihjpw})}$$

$$s(vz^t_{ihjpc}) = \frac{1}{1+exp(-vz^t_{ihjpc})}$$

Where, $s(vx^t_{ihjpm})$ represents the probability of X^t_{ihjpm} taking value 1, $s(vy^t_{ihjpw})$ represents the probability of X^t_{ihjpw} taking value 1 and $s(vz^t_{ihjpc})$ represents the probability of X^t_{ihjpc} taking value 1. For example $s(vx^t_{i1111})=0.2$ in Fig. 7 represents that there is a *20%* chance that operation 1 of part type 1 of particle *i* will be assigned to the first machine at first period.

*5.6 Construction of a particle sequence (*neighborhood solution*)*

In the proposed algorithm, each particle constructs its new sequence based on its changes of probabilities from the velocity trail. In the conventional approach, particle *i* starts with a null sequence in the selected period and assigns an operation of part according to the following probabilities:

$$qx^t_{ih}(jp, m) = \frac{s(vx^t_{ihjpm})}{\sum_{j\in U} s(vx^t_{ihjpm})},$$

$$qy^t_{ih}(jp, w) = \frac{s(vy^t_{ihjpw})}{\sum_{j\in U} s(vy^t_{ihjpw})}, \tag{30}$$

$$qz^t_{ih}(jp, c) = \frac{s(vz^t_{ihjpc})}{\sum_{j\in U} s(vz^t_{ihjpc})}$$

Where U is the set of all operations of parts. The operations of parts are appended successively to the partial sequence until a complete sequence is constructed.

To reduce the computational effort, we replace U by a smaller set of operations of parts in our algorithm. The basic idea of this approach is to take the information of the best sequence into consideration and reduce the computational effort. We employ parameters f that are determined by experiments. Based on the experiments in our VCMS, the use of the smaller set not only reduces the computation time, but also improves the solution quality. The new probabilities are as follows:

$$qx^{t}_{ih}(jp, m) = \frac{s(vx^{t}_{ihjpm})}{\sum_{j \in F} s(vx^{t}_{ihjpm})},$$

$$qy^{t}_{ih}(jp, w) = \frac{s(vy^{t}_{ihjpw})}{\sum_{j \in F} s(vy^{t}_{ihjpw})}, \qquad (31)$$

$$qz^{t}_{ih}(jp, c) = \frac{s(vz^{t}_{ihjpc})}{\sum_{j \in F} s(vz^{t}_{ihjpc})}$$

Where F is the set among the f randomly selected of operations of parts as present in the best sequence (B) obtained so far. For example, suppose first period is selected randomly, f=10, B={(111,1,1,1),(112,2,2,2),(121,2,2,2),(122,1,1,1)} and $s(vx^{t}_{i1jpm})$, $s(vy^{t}_{i1jpw})$ and $s(vz^{t}_{i1jpc})$ are as given in Fig. 7. We start with the null sequence at first period and select ten operations of parts randomly. By Eq. (31), we calculate the probabilities of selected operations of parts and generate a random number for each one of them, drawn from a uniform distribution in (0, 1). Then, among selected operations of parts, each one has probability greater than its random number, its bit gets value 1 (see Fig. 7).

5.7. Variant of the gbest model

For the neighborhood structure of particles in the social part, we introduce the gbest model but modify the approach of searching for P^{t}_{gk} in our algorithm. In the original PSO approach, P^{t}_{gk} is obtained from $P^{t}_{ik}\ (i = 1, 2, \ldots, Np)$. Based on our computational experiments in the virtual cellular manufacturing system problem, we find that the approach which obtains P^{t}_{gk} from the current particles $X^{t}_{ik}\ (i = 1, 2, \ldots, Np)$ performs better. Although our approach spends more computation time on converging, it increases the probability of leaving a local optimum.

6. Numerical examples

6.1. Model analysis

In this section, we present a numerical example showing some of the basic features of the proposed model and illustrating the need of an integrated approach in manufacturing system analysis. The considered example consists of seven part types, six machine types and two periods in which each part type is assumed to have two operation that must be processed respectively; each operation being able to be performed on two alternative machines. Thus, each part type has 2×2=4 process plans and there are 4^7 combinations to select a process plan for each part type in each period. For the numerical example, we assume that the upper bound for the virtual cell sizes is 6, workload balancing factors

	Period 1																							
	Machines								Workers								Cells							
	M_1				M_2				W_1				W_2				C_1				C_2			
Parts	P_1		P2		P_1		P2		P_1		P2		P_1		P2		P_1		P2		P_1		P2	
Operations	o_1	o_2	o_1	o_2	o_1	o_2	o_1	o_2	o_1	o_2	o_1	o_2	o_1	o_2	o_1	o_2	o_1	o_2	o_1	o_2	o_1	o_2	o_1	o_2
Best particle (B)	1	0	0	1	0	1	1	0	1	1	0	1	0	0	1	0	1	0	0	1	0	1	1	0
$s(v^{t}_{i1})$	0.20	0	0.53	0	0.99	0	0.45	0	0.13	0.28	0.48	.093	0.54	0.68	0.99	0.34	0.50	0.01	0.55	0.75	0.15	0.83	0.94	0.35
Selected operations		**	**					**			**	**		**		**	**			**			**	
$q(v^{t}_{i1})$		0	0.10					0			0.09	0.18		0.13		0.07	0.10			0.15			0.18	
Random numbers		0.01	0.05					0.58			0.48	0.08		0.10		0.24	0.12			0.13			0.68	
Candidated bits			**									**		**						**				
New sequence	1	0	1	1	0	1	1	0	1	1	0	1	0	1	1	0	1	0	0	1	0	1	1	0

Figure 7. The construction of a particle

q_1, q_2 are 0.9 and unit material traveling cost is one dollar (i.e. γ=1 $). After linearization, the proposed model consists of 37520 variables and 8316 constraints under the considered example. Tables 2-4 show related data to the considered example. Table 2 shows distances between the machines. Table 3 provides information about the workforce such as initial skills, ability and inability of workers for training on individual machines. Also table 3 presents cost of cross-training on each machine and time-capacity of each worker in each period. Table 4 presents the production data (time-capacity and operating cost of each machine type, quantity of demand for each part type in each period, processing time, inventory holding cost, subcontracting cost and internal production cost). With the data given in Tables 2–4, the proposed model was solved using the general branch and cut algorithm in LINGO where the solution generated by the proposed particle swarm optimization algorithm was used as a starting incumbent solution. Decisions regarding virtual cell configuration, internal production, subcontracting and inventory level are given in Tables 5 and 6. These tables show some of the characteristics and advantages of the proposed model. The demand for part types 1 is entirely satisfied by subcontracting and the demand for some of the part types, such as part types 3, 6 during period 1 and part types 5, 6 during period 2, is satisfied partially by internal production and partially by subcontracting. Part 3 is entirely processed in virtual cell 1. This is indicated in Table 5

In Table 7 is the convergence history of LINGO and LPEPSO in solving this problem. As can be seen from Table 7, the lower bound (F^{bound}) and the best objective function value (F^{best}) for the problem were 87925 and 92114, respectively, found by LINGO after 46 hours of computation. At this point of the computation, the optimality gap was (92114-87925)/92114×100=4.55%. From this table, it can also be seen that starting from the first 26 seconds of computation time, LPEPSO found solutions better than those generated using LINGO in 48 hours. The optimality gap of the final solution found using LPEPSO with reference to the LINGO lower bound was 0.58%. An improved lower bound for the problem was also determined by solving it to optimality after relaxing the constraints in Eq. (7) and setting the number of virtual cells to 1. The improved lower bound was 87986 and the optimality gap of the final solution found using LPEPSO with reference to this improved lower bound was 0.52 %. This suggested that the optimality gap of the LPEPSO solution with reference to an optimal solution of the original problem is less than 0.52%.

From\To	M1	M2	M3	M4	M5	M6
M1	0	8	3	17	18	13
M2	8	0	9	9	10	5
M3	3	9	0	14	20	10
M4	17	9	14	0	6	5
M5	18	10	20	6	0	11
M6	13	5	10	5	11	0

Table 2 Traveling distances (meters) among the machine locations

Worker	Machine						T_w (hours)
	M1	M2	M3	M4	M5	M6	
W1	S	UA	UA	UA	S	A	600
W2	A	A	S	A	UA	S	600
W3	UA	S	UA	A	UA	S	600
W4	A	UA	A	UA	S	UA	600
W5	UA	A	A	S	A	UA	600
tr_m ($)	1000	1200	1000	1400	1800	2000	

S- means that the worker had the skill required for operating an individual machine
A- means that the worker is able to train on an individual machine

Table 3 Workforce information

Part	Op	Machine						Periodic demand		Costs related to			
		M1	M2	M3	M4	M5	M6	h=1	h=2	s_p	i_p	β_p	γ_p
P1	1	0.55		0.44				400	250	20	0.45	10	10
	2		0.35			0.48							
P2	1		0.45			0.48		0	600	35	0.45	9	10
	2				0.79		0.62						
P3	1	0.20						500	650	50	0.40	9	10
	2		0.44			0.51							
P4	1			0.68				350	400	30	0.45	10	10
	2				0.23		0.33						
P5	1					0.62	0.64	250	350	30	0.25	9	10
	2	0.81	0.73										
P6	1					0.34		750	500	45	0.25	7	10
	2						0.25						
P7	1	0.58			0.44			300	300	25	0.50	9	10
	2		0.12				0.21						
T_m (hours)		600	600	600	600	600	600						
α_m		6	9	7	5	4	7						

Table 4 Production data for the numerical example

Part	Oper.	Machine						Internal production	Inventory holding	Subcontracting
		M1	M5	M6	M2	M3	M4			
P1								0	0	400
P3	1	200/w1						200	0	300
	2		200/w4							
P5	1		144/w4	106/w3				250	0	0
	2	250/w1								
P6	1		300/w4 175/w1					475	2	277
	2			475/w3						
P4	1					350/w2		350	0	0
	2						350/w5			
P7	1						300/w5	300	0	0
	2				300/w5					

Table 5 The production planning of parts for period 1

Part	Oper.	Machine M1	M5	M6	M2	M3	M4	Internal production	Inventory holding	Subcontracting
P1								0	0	250
P2	1		600/w4					600	0	0
	2			125/w3			475/w5			
P3	1	650/w1						650	0	0
	2		274/w4		376/w3					
P5	1			150/w3				150	0	200
	2	150/w1								
P6	1		198/w1					198	0	300
	2			198/w1						
P4	1					400/w2		400	0	0
	2						400/w5			
P7	1						300/w2	300	0	0
	2				300/w3					

Table 6 The production planning of parts for period 2

Time	F^{bound}	F^{BEST}	LPEPSO
00:00:02	87654	112649	92819
00:00:05	87654	112872	92473
00:00:11	87654	112431	92345
00:00:26	87635	111187	91437
00:01:58	87647	105904	90128
00:05:08	87647	101812	89483
00:15:28	87647	99187	88442
00:31:48	87652	96419	88442
01:03:24	87669	95871	*
05:07:37	87674	95782	*
10:03:45	87719	94918	*
14:21:56	87816	94639	*
19:57:42	87855	94204	*
25:16:28	87857	93119	*
31:34:07	87869	92649	*
39:18:06	87882	92372	*
44:36:05	87894	92114	*
48:41:06	87925	92114	*

*A termination criterion was met.

Table 7 Comparison of LPEPSO with LINGO for the example problem

6.2 Computation performance

In addition to the example problem discussed above, several other example problems were developed to evaluate the computational efficiency of the developed particle swarm optimization algorithm. These problems generated randomly based on consideration of similar data in the literature. Also, problems are solved under conditions discussed in Section 3.7. For simplicity, we assume that the capacity of the machines are independent of their type, but depends on the length of the planning horizon. For this purpose, we assume that the planning horizon is a three months period or one season. Also, each period includes 75 workdays and each workday includes 8 hours. Therefore, each period is equal to 75×8 = 600 hours. Consequently, by taking into account the controllable and uncontrollable reasons for interrupting production activities, we consider a 500-hour effective capacity for all machine types. Each problem is allowed 7200 seconds (2 hours). However, because of the computational complexity, the proposed model cannot be optimally solved within 7200 seconds or even more time for medium and large-sized instances. Thus, to solve the small and medium-sized problems, we consider a possible interval for the optimum value of objective function (F^*) that are constructed by the F^{bound} and F^{best} values that are introduced by Lingo software where $F^{bound} \leq F^* \leq F^{best}$. According to the Lingo software's documents, The F^{best} indicates the best feasible objective function value (OFV) found so far.

F^{bound} indicates the bound on the objective function values. This bound is a limit on how far the solver will be able to improve the objective. At some point, these two values may become very close. Given that the best objective value can never exceed the bound, the fact that these two values are close indicates that Lingo's current best solution is either the optimal solution or very close to it. At such a point, the user may choose to interrupt the solver and go with the current best solution in the interest of saving on additional computation time. As mentioned earlier, we interrupt the solver within 7200 seconds.

6.2.1. LPEPSO results

In this section, the performance of LPEPSO developed in Section 5 will be verified. In the preliminary experiments, the following ranges of parameter values from the PSO literature were tested N_p=[5,60], c_k=[1,4], w=(0.8,1.2), V_{max}=[3,20], f=[0.2×k_p×P×(M+W+C), 0.8×k_p×P×(M+W+C)]. Based on experimental results, the best PSO parameter settings are shown in Table 8. By attention to whether the mean and best OFV found by LPEPSO lie in interval [F^{bound},F^{best}] or not, six measures for judgment on the effectiveness of LPEPSO are defined as

1. G^{mean} = Gap between F^{best} and Z^{mean}. We assume that if $Z^{mean} < F^{best}$ then the associated gap will be a negative number.
2. G^{best} = Gap between F^{best} and Z^{best}. We assume that if $Z^{best} < F^{best}$ then the associated gap will be a negative number.
3. Avg (|G^{best}-G^{mean}|) = average of the absolute difference between G^{mean} and G^{best}.
4. N^{mean} = the number of problems of their corresponding Z^{mean} lies in interval [F^{bound},F^{best}].
5. N^{best} = the number of problems of their corresponding Z^{best} lies in interval [F^{bound},F^{best}].
6. CPU time.

Parameter	N_p	w	C_1	C_2	V_{max}	F
Value	30	1	1.5	1.5	10	0.4× k_p×P×(M+W+C)

Table 8 PSO parameter settings

In measures 1–4, F^{best} is a base to evaluate the performance. Thus, a negative value implicates a better performance. Table 9 indicates the comparison between the results obtained from B&B and the LPEPSO algorithm corresponding to the 20 problems. The two last columns of Table 9 show the values of G^{mean} and G^{best} resulted from LPEPSO, respectively. In general, the smaller values of G^{mean}, G^{best} and Avg (|G^{best}-G^{mean}|) are more favorable and implicate solutions with a higher quality. Obviously, measure Avg (|G^{best}-G^{mean}|) is directly dependent to the standard deviation of OFV, in 20 times run. As shown in the last row of Table 9, the average values of G^{mean}, G^{best} and Avg (|G^{best}-G^{mean}|) resulted from LPEPSO are obtained as 0.77%, -1.19% and 4.15%, respectively. In other words, Z^{mean} is worse average 0.77% than F^{best} while Z^{best} is better average 1.19% than F^{best} and also there is average 4.15% difference between Z^{mean} and Z^{best}.

Moreover, we have N^{mean} = 6 and N^{best} = 11, thus, in 30% of problems, Z^{mean} lies in interval [F^{bound},F^{best}] and in 55% of problems, Z^{best} lies in interval [F^{bound},F^{best}]. The average of CPU time is obtained as 686.75 seconds that is a promising result in comparison to the CPU time reported for B&B. The average degree of infeasibility is obtained as 0.45 that is negligible. Fig. 8a and b shows the behavior of Z^{mean} and Z^{best} values obtained by LPEPSO versus F^{best} according to the information provided in Table 9. In these figures, the problems are arranged by terms of F^{best} values. The obtained results show that the LPEPSO is a relatively suitable approach to solve the large-size problems.

No.	Problem info.		B&B			LPEPSO			Gap		
	P×M×W×C×H	UB	F^{best}	F^{bound}	$T_{B\&B}$	Z^{best}	Z^{mean}	T_{LPEPSO}	G^{best} (%)	G^{mean} (%)	$\lvert G^{best}-G^{mean}\rvert$ (%)
1	4×2×2×2×2	3	25,625	25,625	1	25,625	25,625	4	0.00	0.00	0.00
2	5×6×3×2×2	3	61,686	61,686	158	61,686	61,686	7	0.00	0.00	0.00
3	6×6×3×2×2	4	64,591	64,591	4286	64,591	65,257	12	0.00	1.03	1.03
4	8×6×4×2×2	4	98,455	98,455	4562	100,435	98512	45	2.01	0.06	1.95
5	12×10×5×2×2	5	101,145	86,379	>7200	99,176	102,402	164	-1.95	1.24	3.19
6	16×10×6×2×3	5	109,458	106,967	>7200	110,549	112,487	75	1.00	2.77	1.77
7	20×14×8×2×2	5	92,648	92,349	>7200	92,954	94,264	128	0.33	1.74	1.41
8	22×15×8×3×2	6	143,719	141,365	>7200	156,546	152,658	144	8.93	6.22	2.71
9	24×14×7×3×2	6	129,515	127,658	>7200	130,256	125,154	354	0.57	-3.37	3.94
10	28×16×10×3×2	7	134,215	131,544	>7200	134,256	139,125	236	0.03	3.66	3.63
11	28×16×8×4×2	6	139,568	138,697	>7200	139,201	145,654	486	-0.26	4.36	4.62
12	28×18×9×3×2	8	186,657	175,249	>7200	175,625	190,365	684	-5.91	1.99	7.90
13	30×17×10×4×2	10	283,545	243,568	>7200	253,214	292,348	985	-10.70	3.10	13.80
14	30×17×10×3×2	7	297,147	257,549	>7200	301,524	299,345	1085	1.47	0.74	0.73
15	30×20×8×3×2	8	282,258	241,246	>7200	257,245	282,346	1262	-8.86	0.03	8.89
16	30×20×12×3×2	10	301,568	265,456	>7200	303,367	303,687	1150	0.60	0.70	0.11
17	40×22×14×5×2	11	368,467	295,426	>7200	398,596	352,124	1428	8.18	-4.44	12.61
18	40×22×10×3×2	8	395,435	303,102	>7200	381,249	416,597	1453	-3.59	5.35	8.94
19	50×27×15×2×2	12	464,598	400,125	>7200	429,267	442,168	1982	-7.60	-4.83	2.78
20	60×32×20×5×2	15	606,257	501,625	>7200	557,895	575,825	2051	-7.98	-5.02	2.96
Average								686.75	-1.19	0.77	4.15

Table 9 Comparison between B&B and LPEPSO runs

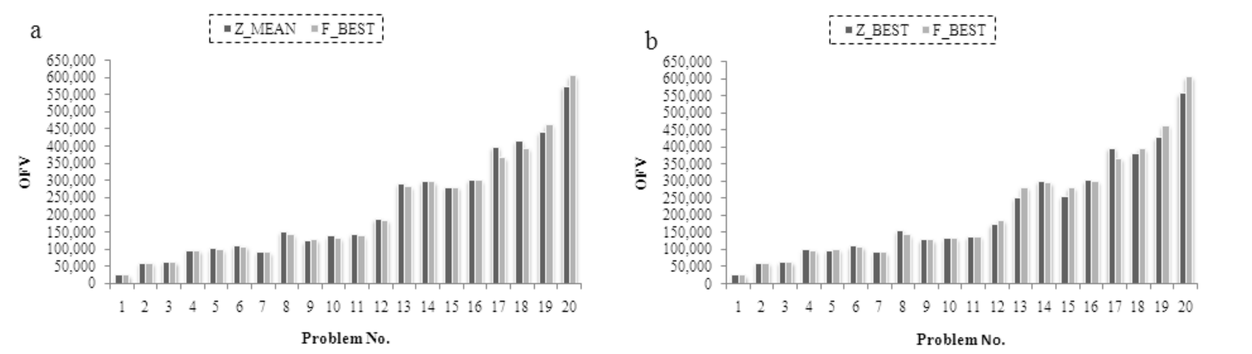

Figure 8. Comparison between the B&B and LPEPSO results (Table 9): (a) Z^{mean} found by LPEPSO vs. F^{best} and (b) Z^{best} found by LPEPSO vs. F^{best}.

7. Conclusion

In this paper a comprehensive mathematical model of a dynamic virtual cellular manufacturing system (DVCMS) is introduced. The advantages of the proposed model are as follows: simultaneous considering dynamic system configuration, operation sequence, alternative process plans for part types, machine and worker capacity, workload balancing, cell size limit and lot splitting. The objective is to minimize the total sum of the manufacturing cost, material handling cost, subcontracting cost, inventory holding cost, internal production cost and needed cost of cross-training for workers over the planning horizon. The proposed model is NP-hard and may not be solved to optimality or near optimality using of-the-shelf optimization packages. To this end, we developed a heuristic method based on the Particle swarm optimization algorithm so-called LPEPSO to solve the proposed model. During the course of the search, the Particle swarm optimization algorithm uses the simplex algorithm interactively to solve a linear programming sub-problem corresponding to each integer solution visited in the search process. The obtained results show that LPEPSO can provide a good solution in a negligible time where the average gap between the quality of the solution found by LPEPSO and the best solution found by the branch and bound (B&B) method is nearly 0.77%. The formulated mathematical is still open for future research with considering other issues such as incorporating virtual cellular manufacturing into supply chain design, considering individual learning and forgetting characteristics in workforce grouping for improving system productivity and considering product quality in the design of VCMS.

References

[1] Askin, R. G., Huang, Y., (2001). Forming effective worker teams for cellular manufacturing. International Journal of Production Research, 39(11), 2431–2451.

[2] Balakrishnan, J., cheng, C.H., (2005). Dynamic cellular manufacturing under multi period planning horizons. Journal of manufacturing technology management 16 (5), 516-530.

[3] Baykasoglu, A., (2003). capability-based distributed layout approach for virtual manufacturing cells. International journal of production research 41(11), 2597-2618.

[4] Clerc, M., Kennedy, J., (2002). The particle swarm explosion, stability, and convergence in a multidimensional complex space, IEEE Transaction on Evolutionary Computation 6, 58–73.

[5] Drolet, J., Abdulnour G., Rheault M., 1996. The cellular manufacturing evolution. Computers & industrial engineering 31(1), 139-142.

[6] Eberhart, R.C. , Shi, Y., 1998. Evolving artificial neural networks, in: Proceedings of the International Conference on Neural Networks and Brain, (1998), pp. PL5–PL13.

[7] Irani, S.A., Cavalier, T.M., Cohen, P.H., (1993). Virtual manufacturing cells: exploiting layout design and inter cell flows for the machine sharing problem. International journal of production research 31(4), 791-810.

[8] Kennedy, J., Eberhart, R.C., (1995). Particle swarm optimization, in: Proceedings of the IEEE International Conference on neural Networks, vol. IV, 1995, pp. 1942–1948.

[9] Kennedy, J., Eberhart, R.C., (1997). A discrete binary version of the particle swarm algorithm. Proc world multi conference on Systemics, Cybernetics and Informatics. NJ: Piscatawary, 4104–4109

[10] Kennedy, J., Eberhart, R.C., Shi, Y., (2001). Swarm intelligence. San Francisco, CA: Morgan Kaufmann

[11] Mak, K.L., Lau, J.S.K., Wang, X.X., (2005). A genetic scheduling methodology for virtual cellular manufacturing systems: an industrial application. International Journal of production research 43(12), 2423–2450.

[12] Mak, K.L., Wang, X.X., (2002). Production scheduling and cell formation for virtual cellular manufacturing systems. International Journal of Advanced Manufacturing Technology 20(2), 144–152.

[13] Min, H., Shin, D., (1993). Simultaneous formation of machine and human cells in group technology: A multiple objective approach. International Journal of Production Research, 31(10), 2307–2318.

[14] Montreuil, B., Drolet, J., Lefrancois, P., (1992). The design and management of virtual cellular manufacturing systems. In Proceedings of American Production & Inventory Control Society Conference, 4–5 October, Quebec, Montreal, pp. 410–414.

[15] Nomden, G., Slomp, J., Suresh, N.C., (2006). Virtual manufacturing cells: a taxonomy of past research and identification of future research. International journal of flexible manufacturing systems. 17, 71-92.

[16] Nomden, G., Van der zee, D-J., (2008). virtual cellular manufacturing: configuring routing flexibility. International journal of production economics, doi: 10.1016/j.ijpe.

[17] Norman, B. A., Tharmmaphornphilas, W., Needy, K. L., Bidanda, B., Warner, R. C., (2002). Worker assignment in cellular manufacturing considering technical and human skills. International Journal of Production Research, 40(6), 1479–1492.

[18] Onwubolu, G.C., (2002). Emerging optimization techniques in production planning and control. London: Imperial College Press

[19] Ratchev, S.M., (2001). Concurrent process and facility prototyping for formation of virtual manufacturing cells. Integrated Manufacturing Systems 12(4), 306–315,

[20] Rheault, M., Drolet, J.R., Abdulnour, G., (1995). Physically reconfigurable virtual cells: a dynamic model for a highly dynamic environment. Computers & Industrial Engineering 29(4), 221–225.

[21] Saad, S.M., Baykasoglu, A., Gindy, N.N.Z., (2002). An integrated framework for reconfiguration of cellular manufacturing systems using virtual cells. Production Planning & Control 13(4), 381–393.

[22] Sarker, B.R., Li, Z., 2001. Job routing and operations scheduling: a network-based virtual cell formation approach. Journal of the Operational Research Society 52(6), 673–681.

[23] Shigenori, N., Takamu, G., Toshiku, Y., Yoshikazu, F., (2003). A hybrid particle swarm optimization for distribution state stimation, IEEE Transaction on Power Systems 18 (2003) 60–68.

[24] Shi, Y., Eberhart, R.C., (1998). A modified particle swarm optimizer. Proc IEEE Congress on Evolutionary Computation, NJ: Piscataway, 69–173

[25] Simpson, J.A., Hocken, R.J., Albus, J.S., (1982). The Automated Manufacturing Research Facility of the National Bureau of Standards. Journal of manufacturing systems, 1 (1), 17-32.

[26] Slomp, J., Chowdary, B.V., Suresh, N.C., (2004). Design and operation of virtual manufacturing cells. In Proceedings of FAIM Conference, Tampa, FL.

[27] Slomp, J., Chowdary, B.V., Suresh, N.C., (2005). Design of virtual manufacturing cells: a mathematical programming approach. Robotics and Computer Integrated Manufacturing 21(3), 273–288.

[28] Slomp, J., Bokhorst, J.A.C., Molleman, E., (2005). Cross-training in a cellular manufacturing environment. Computers & Industrial engineering, 48(3), 609-624.

29) Subash, B.A., Nandurkar, K.N., Thomas, A., 2000. Development of virtual cellular manufacturing systems for SMEs. Logistics Information Management 13(4), 228–242.

[30] Suer, G. A., (1996). Optimal operator assignment and cell loading in labor-intensive manufacturing cells. Computers and Industrial Engineering, 26(4), 155–159.

[31] Suresh, N. C., Slomp, J., (2001). Labor assignment and grouping in cellular manufacturing: A multi-objective methodology. International Journal of Production Research, 39(18), 4103–4131.

[32] Tandon, V., (2000). Closing the gap between CAD/CAM and optimized CNC end milling, Master thesis, Purdue School of Engineering and Technology, Indiana University, Purdue University, Indianapolis.

[33] Teghem, J., Pirlot, M., Antoniadis, C., (1995). Embedding of linear programming in a simulated annealing algorithm for solving a mixed integer production planning problem. Journal of Computational and Applied Mathematics 64, 91–102.

[34] Thomalla CS (2000) Formation of virtual cells in manufacturing systems. In Proc. Group Technology/Cellular Manufacturing World Symposium, San Juan, Puerto Rico, pp. 13–16.

[35] Vakharia, A.J., Moily, J.P., Huang, Y., (1999). Evaluating virtual cells and multistage flow shops: An analytical approach. International Journal of Flexible Manufacturing Systems 11(3), 291–314.

[36] Yoshida, H., Kawata, K., Fukuyama, Y., Nakanishi, Y., (1999). A particle swarm optimization for reactive power and voltage control considering voltage stability, in: Proceedings of the International Conference on Intelligent System Application to Power Systems, 1999, pp. 117–121.

Authors Biography

Hassan Rezazadeh

He received his PhD in industrial engineering from Iran University of Science & Technology, Iran in 2008. He holds an MSc in Industrial Engineering and Systems from Iran University of Science & Technology, Iran and a BSc in Applied Mathematical from Tabriz University, Iran. Currently, he serves as a part-time lecturer at Tabriz University. His research and teaching interests are in application of meta-heuristics, in particular evolutionary algorithms in design and operation of production systems, production scheduling and supply chain management.

Mehdi Ghazanfari

He Is an Associate Professor of industrial engineering at Iran University of Science & Technology, Iran. He received his PhD in industrial engineering from University of New South Wales, Australia, in 1995. He also holds an MSc in industrial engineering from Industrial Engineering at Iran University of Science & Technology, Iran. His research and teaching interests are in production and operations management with emphasis on JIT and manufacturing strategy, operations research and supply chain management.

Seyed Jafar Sadjadi

He is an Associate Professor of Industrial Engineering at Iran University of Science & Technology, Iran. He received his PhD in industrial engineering from University of Waterloo, Canada, in 1998. He also holds an MSc in Industrial Engineering from Industrial Engineering at Iran University of Science & Technology, Iran. His research and teaching interests are in production planning, operations research, investment and finance, operations research modeling, preventive methods, maintenance engineering, facility planning and econometrics.

Mir-Bahador-Goli, Arianezhad

He is an Associate Professor of Industrial Engineering at Iran University of Science & Technology, Iran. He received his PhD in industrial engineering from University of Waterloo, Canada, in 1998. He also holds an MSc in Industrial Engineering from Industrial Engineering at Iran University of Science & Technology, Iran. His research and teaching interests are in production planning, operations research, investment and finance, operations research modeling, preventive methods, maintenance engineering, facility planning and econometrics.

Ahmad Makui

He is an industrial engineering Associate Professor at Iran University of Science & Technology, Iran. He received his PhD in industrial engineering from Iran University of Science & Technology, Iran in 1999. He also holds an MSc in industrial engineering from Industrial Engineering at Iran University of Science & Technology, Iran. His research and teaching interests are in operations research, production planning and decision analysis.