DYNAMIC VIRTUAL CELLULAR RECONFIGURATION FOR CAPACITY PLANNING OF MARKET-ORIENTED PRODUCTION SYSTEMS

ZHENGMIN ZHANG, ZAILIN GUAN AND WEIKANG FANG

Huazhong University of Science and Technology
1037 Luoyu Road, Wuhan city, Hubei province, China

LEI YUE*

School of Mechanical and Electrical Engineering, Guangzhou University
Guangzhou city, China

(Communicated by Ada Che)

ABSTRACT. Market-oriented production systems generally have the characteristics of multi-product and small-batch production. Dynamic virtual cellular manufacturing systems create virtual manufacturing cells periodically in a planning horizon to respond to changing demands flexibly and quickly, and thus are suitable for production planning problems of market-oriented production systems. In the current research, we propose a dynamic virtual cell reconfiguration framework under a dynamic environment with unstable demands and multiple planning cycles. In this framework, we formulate a two-phase dynamic virtual cell formation (DVCF) model. In the first phase, the proposed model aims to maximize processing similarity and balance the workload in the system. In the second phase, we consider the objective of reconfiguration stability based on the first phase model. To address the proposed model, we design a hybrid metaheuristic named Lévy-NSGA-II, and perform various computational experiments to examine the performance of the proposed algorithm. Results of experiments indicate that the proposed Lévy-NSGA-II based algorithm outperforms multi-objective cuckoo search (MOCS) and NSGA-II in solution quality and optimal solution size.

1. Introduction. Cellular manufacturing has received lots of attention in the past few decades [2], [11], [12]. It organizes production equipment into machine cells, where each cell specializes in the production of a part family [12]. It is critical to keep the stability of the product mix to guarantee the performance and the efficiency of cellular manufacturing systems [28]. However, more customized goods with shorter processing cycles are demanded, which increases manufacturing costs. Conventional cellular manufacturing systems are not flexible enough to reconfigure the manufacturing cell dynamically and cope with frequent demand changes. In recent years, Industry 4.0 has increased the trend of manufacturing companies to reconfigure
their systems dynamically. Thus, dynamic virtual cellular manufacturing systems play a critical role in mass production scenarios to meet the current needs of manufacturing industries. In dynamic environments, a multi-period planning horizon is usually considered and each period has different product mixes and demands. Therefore, the formed cells in a period may not be optimal or well-organized for the next period. When the product mix and technological requirements of manufacturing enterprises are made changes, the existing virtual manufacturing cells can’t meet the production requirements. That is, the changes in volumes and mix of demands may lead to insufficient capacity in manufacturing cells, and production can only be maintained by borrowing the capacity of other cells frequently. Thus, manufacturing enterprises need to reconfigure manufacturing resources, the frequency of reconfiguration depends on new tasks to cause changes in volumes and mix of demand. The implementation of dynamic virtual cellular manufacturing can improve the performance of manufacturing systems by creating logical manufacturing cells periodically, which can reduce the physical movements of machines in the production system.

Dynamic virtual cell formation (DVCF) is the first step in a dynamic virtual cellular manufacturing system [25], as well as a critical process in the entire manufacturing system. The DVCF problem attempts to classify parts and machines to create virtual cells dynamically. Unlike the cell formation problem, machines in DVCF are not transported into the physical position, instead virtual cells are usually formed logically and periodically according to changing demands. Moreover, DVCF allows a tradeoff between the independence of each cell and the machine share between cells.

The DVCF problem has attracted increased attention from researchers in recent years. Nevertheless, we identify the following literature gaps:

1. Most literature for DVCF is difficult to apply in real-world production systems with unstable demands. In the multi-period DVCF problem, parts and machines are grouped under the assumption that multi-period demands are known in advance [8], [26], [28]. However, it is difficult for us to predict the demand for multiple periods accurately. Thus, there may exist a gap between the theoretical solution and the actual implementation of the multi-period DVCF problem.

2. Few dynamic virtual cell reconfiguration frameworks consider both medium- and long-term planning periods [13]. Existing research on DVCF problems mainly focuses on the reconfiguration of medium- or short-term planning periods with small demand changes or capacity fluctuations. For instance, some studies propose virtual cellular reconfiguration models using a single reconfiguration period as an example [15], [24]. However, products are constantly updated in real production systems and the capacity of production systems requires significant adjustments accordingly. When changes in variety and demand of products are frequent, the formulated model may no longer be effectively applicable, and the efficiency of the cell formation solution is hard to guarantee.

Therefore, the research question of this paper is: how to establish an implementable virtual cell reconfiguration framework for both long- and short-term planning periods and implement this framework to solve capacity planning problems efficiently? To address this problem, we propose a new dynamic virtual cell reconfiguration method to adjust and reconfigure the production system by creating virtual cells in a market-oriented production system. Considering the long-term planning issue of real-world manufacturing systems, we establish a dynamic virtual
cell reconfiguration framework under multiple planning cycles. The multiple planning cycles have huge timelines which include multiple numbers of planning periods, termed as a phase. The duration of one planning cycle is usually 3-5 years. During a planning cycle, the formation model is divided into two phases. In the proposed model, the first phase is the initial virtual cell formation period. We address the first phase model with objectives of maximizing the processing similarity and the workload balance, simultaneously. In the second phase, we perform the reconfiguration by considering demand changes at each phase in a cyclic way. When the product and capacity demand of manufacturing systems reach the critical value of the guaranteed performance, the next planning cycle of the formation model is renewed and the new formation model is built. In this model, we only need to know the demand of the next phase because the proposed framework can update the plan for each phase iteratively. To address the above two-phase DVCF model, we present a new algorithm comprising basic NSGA-II based on a discrete Lévy flight search strategy (Lévy-NSGA-II). Moreover, we use the proposed algorithm to solve the multi-objective formation problems in the proposed model.

The main contributions of this research are as follows: (1) We form a dynamic virtual cell reconfiguration implementation framework under multiple planning cycles. The proposed framework ensures the feasibility of formation and reconfiguration plans. (2) We formulate a new two-phase DVCF model for a single planning cycle, and consider the objective of reconfiguration stability for this model. (3) We propose a hybrid metaheuristic to solve the proposed model, and verify the proposed method has a greater improvement than multi-objective cuckoo search (MOCS) and NSGA-II in solution quality and optimal solution size.

2. Literature review. Virtual cellular manufacturing plays an important role in the implementation of rapid response manufacturing, lean production and agile manufacturing [10]. The cell formation problem is the core issue of a virtual cellular manufacturing system. A considerable amount of literature has been published on virtual cell formation and virtual cell reconfiguration [9], [34], [40]. Some research on cell formation problems also promote the development of virtual cell formation problems. Literature addresses cell formation problems or virtual cell formation problems with different objective functions. For example, Rezaeian et al. [35] present a new nonlinear programming model to minimize the total material handling cost in a dynamic environment. Mahdavi et al. [25] study the consideration of demand and part mix variation over a multi-period planning horizon. Some researchers also solve the cell formation problem to minimize the associated costs by considering dynamic product requirements [5], [31], [33]. Mohammadi and Forghani [27] consider the importance of machine layout and quantity of raw material purchased from qualified suppliers.

In recent research, most scholars use economic indicators [30] as objective functions, which can simplify multi-objective problems. Meanwhile, many studies also use efficient indicators as objective functions to describe problems more appropriately [17], [24], [33]. Generally, these indicators are evaluation criteria under a single cycle, which can improve the solution in a single planning cycle to a certain extent. However, limited research considers the reconfiguration between two formation cycles. For example, Li et al. [20] consider the penalty cost of inheritance in the multi-objective to solve dynamic virtual cell formation problems. Han et al. [14]
study the stability reconfiguration problem of virtual cellular manufacturing system with random arrival of orders and time window.

Besides, many studies also consider routing flexibility in virtual cellular manufacturing systems. This problem is first nominated by Andrew [19] in the generalized cellular manufacturing system and is studied by many researchers. Kia et al. [18] present a mathematical programming model for the intra-cell layout design of a dynamic cellular manufacturing system in the presence of alternative process routings and duplicate machines. Eguía et al. [6] center their studies on cellular reconfigurable manufacturing systems in the existence of alternative paths and multiple periods.

Nonetheless, most of the literature focuses on the virtual cellular manufacturing system with less focus given on the demand changes in different planning cycles. In a real manufacturing environment, the production targets are set ahead of the planning cycle. An efficient virtual cellular manufacturing system needs to consider the demand or capacity changes to make the virtual cellular manufacturing system more realistic. However, few researchers give effective methods to overcome the shortcomings of the data accuracy requirement of multi-period DVCF. Besides, few studies pay attention to virtual cell formation problems with severe demand/capacity changes.

3. Dynamic virtual cell reconfiguration framework under multiple planning cycles. Motivated by real-world industrial applications, Figure 1 demonstrates the dynamic virtual cell reconfiguration framework under multiple planning cycles. There are two planning cycles in Figure 1, and each planning cycle comprises multiple planning periods.

The proposed framework performs cell formation in each planning period considering two phases of the DVCF model. The relevant capacity data of the next planning period are used as the input of the two-phase dynamic virtual cell formation model. In each planning cycle, the first phase model is solved to form the initial formation solution of the current formation period. Then, according to the demand variations, the previous period formation solution can be reconfigured logically, and
a new reconfiguration scheme can be further obtained. When the reconfiguration plan of the current planning period is difficult to cope with the production status, we restart the process of the previous planning cycle from the beginning of a new planning cycle.

4. Model formulation. We present a two-phase DVCF model for a single planning period of the rolling framework to solve the DVCF problem. Compared with other DVCF models, the proposed model can meet the needs of actual production scenarios, especially in enterprises with frequent demand changes. We consider the following assumptions in this model:

(1) The demand for each type of part is known and constant in one phase.
(2) The alternative process routings, the operation sequences and processing times are known in advance.
(3) Only one routing is selected for the production of each part at a time.
(4) One process corresponds to one type of machine.
(5) The capacity of each machine is known and does not change during the planning cycle.
(6) Each type of part/machine can be divided into one virtual cell but the machines can be shared.
(7) Each part can be processed on one machine at a time and cannot be interrupted while processing.
(8) Each machine can only process one part at a time.
(9) The upper and lower bound of the virtual cell size is known and remains fixed over a planning cycle.

As shown in Table 1, we list the notations used in the proposed two-phase DVCF model. The two-phase DVCF model comprises a dual-objective and a three-objective optimization model. In the first phase, the model aims to maximize the processing similarity and balance the workload of machines. In the second phase, we add the reconfiguration stability objective of minimizing the number of machine changes based on the above two objectives.

(1) The formation efficiency objective
This model considers typical production efficiency objectives (indicated by $f_1$ and $f_2$, respectively), which are useful to maximize resource utilization.

The part clustering operations can maximize the similarity in each part family and minimize the similarity between part families. To unify the objectives being optimized, the total dissimilarity is reduced instead of maximizing the overall similarity.

$$f_1 = \min \left( \sum_{c=1}^{C} \sum_{j=1}^{J} \sum_{r=1}^{R_j} \sum_{j'=1}^{J} \sum_{r'=1}^{R_j'} S_{j,j',r,r'} \cdot X_{j,r,c,t} \cdot X_{j',r',c,t} \right), \forall t$$  (1)

$$S_{j,j',r,r'} = 1 - \frac{\sum_{m=1}^{M} \alpha_{j,r,m} \alpha_{j',r',m}}{\sum_{m=1}^{M} (\alpha_{j,r,m} + \alpha_{j',r',m} - \alpha_{j,r,m} \alpha_{j',r',m})}, \forall j, r$$  (2)

Equation (1) and equation (2) calculate the dissimilarity coefficient ($f_1$), where $S_{j,j',r,r'}$ represents the dissimilarity coefficient of two parts in the same cell. Workload balancing of machines can increase machine utilization and reduce queuing time by avoiding long-term processing or long-term idleness of machines. Equation (3) gives the workload balance objective function $f_2$. We use $f_2$ to minimize the workload difference of different machine types. $L_m$ and $L$ are given in equation (4)
Table 1. Notations used in the DVCF model

| Indices |
|---------|
| $j$     | part types, $j = 1, 2, \cdots, J$ |
| $r$     | process routings, $r = 1, 2, \cdots, R_j$ |
| $m$     | machine types, $m = 1, 2, \cdots, M$ |
| $c$     | virtual cells, $c = 1, 2, \cdots, C$ |
| $t$     | formation periods |

| Input parameters |
|------------------|
| $J_t$            | number of part types in period $t$ |
| $R_{j}$          | number of process routings for part type $j$ |
| $M$              | number of machine types |
| $N_m$            | number of machines included in machine type $m$ |
| $B_U$            | upper bounds of virtual cells |
| $B_L$            | lower bounds of virtual cells |
| $D_{j,t}$        | demand for part type $j$ in period $t$ |
| $A_m$            | production capacity of each machine of type $m$ |
| $\alpha_{j,r,m}$| 1, if $r$-th process routing of part type $j$ needs to use the machine type $m$, 0 otherwise |
| $T_{j,r,m}$      | processing time of $r$-th process routing of part type $j$ at machine type $m$ |
| $S_{j,j',r,r'}$  | the similarity coefficient between $r$-th process routing of part type $j$ and $r'$-th process routing of part type $j'$ |
| $Z_{m,c,t-1}$    | number of machines of type $m$ assigned to virtual cell $c$ in period $t-1$ ($t > 1$) |

| Variables |
|-----------|
| $C_t$     | number of virtual cells in period $t$, $B_L \leq C \leq B_U$ |
| $X_{j,r,c,t}$ | 1, part type $j$ to be assigned to routing $r$ and to be assigned to virtual cell $c$ in period $t$, 0 otherwise |
| $Y_{m,c,t}$ | number of machines of type $m$ assigned to virtual cell $c$ in period $t$ (real number) |
| $Z_{m,c,t}$ | number of machines of type $m$ assigned to virtual cell $c$ in period $t$ (integer) |

and equation (5), represent the average load of machine type $m$ and the average capacity of all machines in the period, respectively.

\[
f_2 = \min \left( \sum_{m=1}^{M} |L_m - \bar{L}| \right)
\]  

\[
L_m = \sum_{j=1}^{J} \sum_{r=1}^{R_j} \sum_{c=1}^{C} X_{j,r,c,t} \alpha_{j,r,m} D_{j,t} / A_m N_m, \forall t
\]  

\[
\bar{L} = \sum_{j=1}^{J} \sum_{r=1}^{R_j} \sum_{m=1}^{M} \sum_{c=1}^{C} X_{j,r,c,t} \alpha_{j,r,m} T_{j,r,m} D_{j,t} / \sum_{m=1}^{M} A_m N_m, \forall t
\]  

(2) Reconfiguration stability
Frequent large-scale reconfiguration adjustment increases management difficulty because it requires more capital investment. Therefore, it is necessary to ensure the stability of the production system by minimizing the variation of resource allocation during the reconfiguration period. We define this objective as reconfiguration stability. Equation (6) gives the corresponding indicator \( f_3 \), which is used to minimize the number of machines that needs to be moved from one to another virtual cell between two reconfiguration periods.

\[
f_3 = \min \left( \sum_{c=1}^{C} \sum_{m=1}^{M} |Z_{m,c,t} - Z_{m,c,t-1}| \right), t > 1
\]

Equation (7) represents the objectives of different phases in planning period \( t \). In the initial phase of the planning cycle (the first phase, \( t = 1 \)), the initial formation process needs to be re-integrated with renewed capacity and order data, instead of adjusting the scheme of the previous period. Therefore, the reconfiguration stability objective is excluded from the first phase model. In the second phase of the planning cycle (\( t > 1 \)), the reconfiguration scheme is adjusted according to the previous period.

The constraints for the proposed model are as follows:

\[
Y_{m,c,t} = \frac{\sum_{j=1}^{J} \sum_{r=1}^{R_j} X_{j,r,c,t} \alpha_{j,r,m} T_{j,r,m} D_{j,t}}{A_m}, \forall m, c, t
\]

\[
\sum_{c=1}^{C} Y_{m,c,t} \leq N_m, \forall m, t
\]

\[
Y_{m,c,t} \leq Z_{m,c,t} < Y_{m,c,t} + 1, \forall m, c, t
\]

\[
B_L \leq C \leq B_U
\]

\[
\sum_{c=1}^{C} \sum_{r=1}^{R_j} X_{j,r,c,t} = 1, \forall j, t
\]

\[
X_{j,r,c,t} \in \{0, 1\}, \forall j, r, c, t
\]

\[
Y_{m,c,t} \geq 0, \forall m, c, t
\]

\[
Z_{m,c,t}, Z_{m,c,t-1} > 0 \text{ and integer}
\]

Constraint (8) indicates the minimum real number of machines with type \( m \) to be allocated for each cell. Constraint (9) implies that the actual production time of machines with type \( m \) cannot exceed the total production capacity of the machine. Constraint (10) indicates the minimum integer of the machine with type \( m \) to be allocated to each cell. Constraint (11) ensures that the number of virtual cells cannot below the lower limit \( B_L \) or exceed the upper limit \( B_U \). Constraint (12) guarantees that only one routing is assigned to a part, and the part only belongs to one virtual cell. Constraint (13) restricts that \( X_{j,r,c,t} \) is a 0-1 variable. Constraint (14) ensures that \( Y_{m,c,t} \) is non-negative, and constraint (15) represents \( Z_{m,c,t} \) and \( Z_{m,c,t-1} \) are integers. We provide a detailed example of the proposed two-phase DVCF model in Appendix of this research.
5. **NSGA-II based on discrete Lévy flight search strategy.** Since cell formation problems belong to the class of NP-hard problems, most researchers pay attention to designing heuristics, metaheuristics or hybrid algorithms to solve these problems [7], [16], [17], [24], [25]. The proposed model is difficult and time-consuming to be solved with a solver because of multiple objectives and decision variables. Thus, we consider developing an effective metaheuristic to solve the problem.

5.1. **Hybrid Lévy-NSGA-II algorithm.** NSGA-II, proposed by Deb et al. [4], is a typical evolutionary algorithm based on GA [30]. It has been commonly used in multi-objective problems due to its powerful global exploration ability and robustness. However, due to the unique individual fitness value distribution mechanism and the individual selection operator in NSGA-II, the repeated individuals in the new population lead to a low searching efficiency.

Some hybrid intelligent algorithms combined with NSGA-II have been suggested to improve its performance. We combine the discrete Lévy flight search strategy with NSGA-II to reduce the probability of repeating individuals and improve the population diversity. Lévy flight, represents a model of random walks characterized by their step lengths that obey a power-law distribution [39]. The short jump and occasional long jump are cross-executed in the Lévy flight (Figure 2), which increases the probability of jumping out of local optimum and expands the search range of particles.

Figure 3 demonstrates the detailed flow chart of the proposed Lévy-NSGA-II. A primary improvement of the proposed Lévy-NSGA-II algorithm is using the discrete Lévy flight search strategy to perform a random search process for offspring individuals.

5.1.1. **Scheme for coding.** The structure of a chromosome (Figure 4) comprises four layers as follows:

1. The cell layer comprises genes relating to the assignment of the parts to the virtual cells in period $t$ while $C$ refers to equation (11). For instance, the term “$C^1 = 1$” means that ‘part 1’ is assigned to virtual cell 1. The cell layer is a dominant gene in the chromosome.

2. The routing layer comprises genes relating to the assignment of the part routings while $R^j$ is no more than $R_j$. For instance, the term “$R^2 = 1$” means...
that ‘routing 1’ is chosen for part 2. The routing layer is a dominant gene in the chromosome.

3. The machine layer comprises genes relating to the processing machine under the selected routing of part \(j\). \(M^j\) is a sequence of processing machines. For instance, the routing \(R_1\) of \(P_1\) is \((M_1, M_8, M_6, M_2)\), thus \(M^1 = \{1, 8, 6, 2\}\). The machine layer is a recessive gene in the chromosome, and it changes according to routing layer.
4. The time layer comprises genes relating to the processing time of each processing machine under the selected routing of part \( j \). \( T^j \) is a sequence of processing time. For instance, the processing time under \( R1 \) of \( P1 \) is \( (360, 90, 160, 180) \), thus \( T^1 = \{360, 90, 160, 180\} \). The time layer is a recessive gene in the chromosome.

Based on the above encoding scheme, we can obtain the detailed information of job types, numbers, and routings according to the cell layer and routing layer of a chromosome. Further, machines can be allocated according to the required capacity of each cell. For a determined planning period \( T \), we calculate the number of each type of machine in each cell using equation (16).

\[
Z_{m,c,T} = \frac{\sum_{j=1}^{J} \sum_{r=1}^{R_j} X_{j,r,c,T} A_{j,r,m} T_{j,r,m} D_{j,T}}{A_m} + 1, \forall m, c \tag{16}
\]

In a chromosome, we use \( N^c \) (see in Figure 5) to represent the number of allocations for each machine type in cell \( c \). Thus, \( N^c = \{Z_{1,c,T}, ..., Z_{m,c,T}, ..., Z_{M,c,T}\} \) in period \( T \). For instance, \( N^1 = \{3, 3, 0, 4, 2, 3, 1, 3\} \) means, in cell 1, the machine number of machine type 1-8 are 3,3,0,4,2,3,1,3, respectively.

The pseudocode of the calculation method of \( N^c \) is as follows:

**Algorithm 1 The Calculation Method of \( N^c \)**

1: **Input:** Cell number \( C \) and \( X_{j,r,c} \) of the current solution.
2: **for each cell \( c \) do**
3:     **for each machine type \( m \) do**
4:         *Calculate the capacity requirements of machine \( m \)*
5:     *Based on the capacity result calculate the minimum number \( (Y_{m,c}) \)*
6:         \( N^c(m) \leftarrow Z_{m,c,T} \)
7: **end for**
8: **end for**
9: **Output:** \( N^c \)

5.1.2. Crossover and mutation operations. Lévy-NSGA-II performs crossover and mutation operations [38] between two selected parent individuals. After calculating the non-domination and crowding distance values, Lévy-NSGA-II applies the binary tournament selection method proposed by Blickle and Thiele [1] to form a mating pool.
To improve population diversity and expand search space, we present a double-layer crossover strategy for dominant genes. The cell layer and routing layer are operated separately to expand the search space. Figure 6a depicts the chromosomes (Parent 1 and Parent 2) selected for crossover. The random segments of Parent 1 and Parent 2 are exchanged at the cell layer, and the same operation happens at the routing layer. Similarly, we perform the mutation operation (Figure 6b) according to the double-layer mutation strategy with a low probability.

5.1.3. Discrete Lévy flight search strategy. Lévy flight has a random walking direction, and the motion steps include multiple small steps and occasionally large steps [41]. This feature can improve the global search capability of the algorithm. To apply the Lévy flight search to discrete problems, Karoum and Elbenani [17] propose an effective local and global search strategy. We modify their strategy and apply it to the current multi-layer chromosome optimization problem.

1. Local search operation

Large-scale adjustment of sequencing may destroy the existing superior segments in the chromosome. Therefore, we employ local adjustment methods to update feasible solutions iteratively. Figure 7 presents the local adjustment of the cell layer as an example. Consider \( J = \{ J_1, J_2, ..., J_n \} \) as an available solution for the optimization problem and \( s \) is an input parameter. The feasible solution \( J \) is divided into \( \lfloor n/s \rfloor \) segments, with \( k = n\%s \) elements remaining. The selected two elements in each of \( \lfloor n/s \rfloor \) sections are exchanged according to probability \( p \). If \( k \geq 2 \), then two elements are randomly selected for exchange in the remaining segment according to probability \( p \). The above process enables a small step movement similar to the movement in Lévy flight.

2. Global search operation

The discrete Lévy flight search strategy also includes a low probability global search, which guarantees the global search capability. Figure 8 presents an example...
of the global adjustment of the cell layer. Similar to the local search operation, the feasible solution $J$ is divided into $\lfloor n/s \rfloor$ segments. The selected two segments are exchanged to achieve a large step movement. The routing layer adjustment process is the same as Figure 6b. The individual after discrete Lévy flight search strategy is compared with the original individual, and the replacement of the high-quality individual is made to form a new offspring.

5.2. Implementation of the proposed hybrid Lévy-NSGA-II algorithm. We apply the proposed hybrid Lévy-NSGA-II to address the proposed two-phase DVCF model. Figure 9 illustrates the relationship between the two-phase formation model and the two-phase algorithm. In any planning cycle $R_i$, the input parameters of the first phase ($t = 1$) include not only production requirements, but also the updated resources and capacity information. The first phase considers formation efficiency objectives, and does not need to consider reconfiguration stability. In the second phase ($t > 1$) of the planning cycle, the input of the model is updated according to the demand. Additionally, the model further considers the reconfiguration stability based on formation efficiency objectives to meet the production requirements with minimal reconfiguration fluctuation.

6. Computational results and performance analysis. In this section, we validate the proposed Lévy-NSGA-II by comparing it with multi-objective cuckoo search (MOCS) and NSGA-II. MOCS and NSGA-II are popular stochastic search techniques and regarded as the classical approaches to neighbor search and population search, respectively [24]. These approaches have been used to solve many optimization problems [3]. We conduct experiments with different scales to compare the performance of Lévy-NSGA-II, MOCS and NSGA-II. We use MATLAB R2017a software to code the algorithms from scratch, and complete the numerical tests on a personal computer equipped with Intel® CoreTM2 Quad 1.6 GHz CPU and 8G RAM.

6.1. Numerical experiments. We generate experimental data in test problems based on the manufacturing data from an aircraft shaft parts workshop. We simplify
Figure 9. The applications of Lévy-NSGA-II in the two-phase DVCF model

Table 2. Pattern of data generation

| Parameter | Generation pattern | Parameter | Generation pattern |
|-----------|--------------------|-----------|--------------------|
| \(M\)     | 8                  | \(R_i\)   | U \([1,3]\)         |
| \(\sum_{m=1}^{M} N_m\) | U \([2J, 3J]\) | \(\sum_{m=1}^{M} \alpha_{j,r,m}\) | U \([2, 6]\) |
| \(B_U\)  | Random\(\{3, 4\}\) | \(T_{j,r,m}\) | U 10 * \([4, 32]\) |
| \(B_L\)  | Random\(\{2, 3\}\) | \(A_m\)   | U 100 * \([10, 30]\) |
| \(D_j\)  | U \([0, 10]\)     |           |                    |

Table 3. Type and dimension of test problems

|       | Size1 | Size2 | Size3 | Size4 | Size5 | Size6 |
|-------|-------|-------|-------|-------|-------|-------|
| \(T = 1\) | A1    | A2    | A3    | A4    | A5    | A6    |
| \(T > 1\) | B1    | B2    | B3    | B4    | B5    | B6    |

the actual manufacturing data because of the confidentiality of the workshop, but still retain key features such as multiple process paths and parallel machines in this case. Specifically, we define the following parameters: number of machine types, machine capacities, number of parallel machines, available machines for each process, demand for each planning period, alternative routings for each product, and the boundary of cell numbers. To evaluate the performance of the proposed Lévy-NSGA-II, we generate 6 sets of samples \((J = 15, 18, 21, 24, 27, 30)\) according to the parameter settings in Table 2. As shown in Table 2, we represent these test problems by various codes. For instance, A2 denotes the first phase problem of scale 3 \((j * m * r : 18 * 49 * 33)\). B3 represents the second phase problem of size 4 \((j * m * r : 21 * 55 * 39)\).

The efficiency of algorithms may greatly depend on their parameter values. Thus, we calculate the good value of the input parameters of the proposed algorithm using...
Table 4. The Pareto-optimal solutions for problems A3

| Solution number | Lévy-NSGA-II Dissimilarity coefficient | Workload balance | MOCS Dissimilarity coefficient | Workload balance | NSGA-II Dissimilarity coefficient | Workload balance |
|-----------------|----------------------------------------|-----------------|-------------------------------|-----------------|----------------------------------|-----------------|
| 1               | 7.533333                               | 2.022256       | 1                             | 7.939286       | 1.976695                         | 2.074396       |
| 2               | 7.719048                               | 1.921483       | 2                             | 8.025000       | 1.830073                         | 1.976695       |
| 3               | 7.833333                               | 1.855261       | 3                             | 8.092063       | 1.802253                         | 1.918146       |
| 4               | 7.901190                               | 1.851695       | 4                             | 8.177778       | 1.629786                         | 1.830073       |
| 5               | 7.93730                                | 1.835300       | 5                             | 8.444444       | 1.503174                         | 5               |
| 6               | 7.954762                               | 1.800373       | 6                             | 8.563492       | 1.457544                         | 6               |
| 7               | 8.005159                               | 1.694708       | 7                             | 8.683730       | 1.391009                         | 7               |
| 8               | 8.254762                               | 1.629786       | 8                             | 8.790476       | 1.266009                         | 8               |
| 9               | 8.282143                               | 1.490863       | 9                             | 9.265476       | 1.224086                         | 9               |
| 10              | 8.560714                               | 1.497544       | 10                            | 9.383333       | 1.167368                         | 10              |
| 11              | 8.711111                               | 1.391009       | 11                            | 9.487052       | 1.137268                         | 11              |
| 12              | 8.717063                               | 1.297040       | 12                            | 8.727778       | 1.266009                         | 12              |
| 13              | 8.727778                               | 1.266009       | 13                            | 9.085556       | 1.224086                         | 13              |
| 14              | 9.059524                               | 1.224086       | 14                            | 9.364286       | 1.137268                         | 14              |
| 15              | 9.255952                               | 1.167368       | 15                            | 9.267857       | 1.137268                         |                 |
| 16              | 9.267857                               | 1.137268       |                               |                 |                                  |                 |

the Taguchi method. According to the evaluation of the Taguchi method, we provide the optimal parameters settings as follows: the population size (NPop) is 300, the number of iterations for algorithms to find the best Pareto frontier (MaxIt) is 200, the crossover rate (CrR) is 0.9, the mutation rate (MuR) is 0.1, and the random search rate in Lévy flight search strategy (RaR) is 0.005.

6.2. Experimental results. We solve the first phase problems of the model by MOCS, NSGA-II, and Lévy-NSGA-II (both with NPop = 300), respectively. Figure 10 represents the Pareto-optimal solutions for problems A1-A6. We notice that the proposed algorithm has a better distribution and convergence effect than compared methods, especially when the problem size increases.

Table 4 illustrates the output data of Pareto-optimal solutions for problems A3. Each solution number corresponds to an optimal solution, and the total number represents the number of optimal solutions on the Pareto frontier. The solution number with a red mark belongs to a solution non-dominated by the reference set $S^p$. The reference set $S^p$ denotes the non-dominant solutions. $S^p$ is obtained by combining all the non-dominant solutions obtained by the comparison algorithms and sorting the non-dominant solution of the combined set. We notice that MOCS and NSGA-II contain 1 and 4 non-dominated solutions, respectively, while Lévy-NSGA-II has 14 non-dominated solutions, far more than the compared algorithms.

We solve the second phase cases of the two-phase DVCF model under the same parameters. When the second phase reconfiguration problems are solved by different algorithms, the formation schemes of the previous period should be set to the same to ensure the accuracy of the experiments. Figure 11 demonstrates the Pareto-optimal solutions for problems B1-B6.

As Figure 11 shows, the proposed algorithm is still more convergent than MOCS and NSGA-II when solving three-objective optimization problems. However, the comparison of algorithm performance is not intuitive in Figure 11. To show the efficiency of the proposed algorithm accurately, we compare the solutions using several metrics in the next section.

6.3. Comparisons metrics. To measure and validate the performance of the proposed Lévy-NSGA-II algorithm, we consider the three performance metrics that
Figure 10. Pareto-optimal solutions for A1-A6

were studied in Pan et al. [32], Li et al. [21], [22], and Li et al. [23], i.e., the average Pareto distance $V_{pd}$, the number of the non-dominated solutions $V_{np}$, and the ratio of the non-dominated solutions $V_{rd}$.

Let $S^j (j = 1, 2, 3)$ represent the non-dominated solution set obtained by the algorithm $j$ participating in the comparison. $S^p$ denotes the non-dominant solutions. Therefore, $S^p = \bigcup S^j$. The specific description of three performance metrics is as follows:

1. The average Pareto distance ($V_{pd}$)

2. The number of non-dominated solutions ($V_{np}$)

3. The ratio of non-dominated solutions ($V_{rd}$)
Figure 11. Pareto-optimal solutions for B1-B6

We use the metric $V_{pd}$ to compute the distribution of the obtained non-dominated solution set. The $V_{pd}$ of algorithm $j(V_{pd}^j)$ is obtained by equation (17) and equation (18), where $f_{i}^{max}$ and $f_{i}^{min}$ are the maximum and minimum values of the $i$-th objective in the reference set $S^p$, respectively. $f_i(x)$ is the $i$-th objective function value of the corresponding solution ($I$ objectives in total). A smaller $V_{pd}^j$ shows a better distribution and better approximation of the obtained solution set of algorithm $j$ to the reference set $S^p$. $V_{pd}^j$ equals 0 means that the obtained solution set of algorithm $j$ is equal to the reference point set.

$$
V_{pd}^j = \frac{1}{|S^p|} \sum_{y \in S^p} d_y(S^j) \quad (17)
$$

$$
d_y(S^j) = \min_{x \in S^j} \left\{ \sum_{i=1}^{I} \left( \frac{f_i(x) - f_i(y)}{f_i^{max} - f_i^{min}} \right)^2 \right\}, y \in S^p \quad (18)
$$
Table 5. Comparisons of 6 sets of test problems

| Problem | Pareto distance $V_{pd}$ | Pareto distance $V_{np}$ | Pareto distance $V_{rd}$ |
|---------|--------------------------|--------------------------|--------------------------|
| No      | MOCS NSGA-II Lévy-NSGA-II | MOCS NSGA-II Lévy-NSGA-II | MOCS NSGA-II Lévy-NSGA-II |
| A1      | 0.1016 0.0348 0.0158       | 6 8 10                   | 0.0367 0.0333 0.0333      |
| A2      | 0.0754 0.0346 0.0286       | 2 6 10                   | 0.0367 0.0400 0.0333      |
| A3      | 0.1460 0.0644 0.0030       | 1 4 14                   | 0.0367 0.0467 0.0533      |
| A4      | 0.2443 0.5967 0.0014       | 0 1 10                   | 0.0367 0.0267 0.0400      |
| A5      | 2.0610 2.3134 0.0001       | 0 2 18                   | 0.0367 0.0233 0.0633      |
| A6      | 2.3025 2.7586 0.0000       | 0 0 16                   | 0.0500 0.0500 0.0533      |
| B1      | 0.4266 1.5974 0.0506       | 32 27 42                 | 0.1333 0.1000 0.1467      |
| B2      | 1.0217 2.0321 0.0123       | 18 18 51                 | 0.1067 0.1167 0.1800      |
| B3      | 0.3373 0.4506 0.2616       | 50 67 107                | 0.2600 0.3000 0.3900      |
| B4      | 0.7522 1.6053 0.5043       | 48 20 67                 | 0.3600 0.3633 0.2800      |
| B5      | 1.5456 3.0284 0.4457       | 26 43 42                 | 0.2267 0.2233 0.2600      |
| B6      | 0.8931 2.0602 0.3225       | 29 16 100                | 0.2700 0.2400 0.4267      |

(2) The number of non-dominated solutions ($V_{np}$)

The metric $V_{np}$ is the number of obtained solutions set that are not dominated by the reference set $S^p$. The $V_{np}$ of algorithm $jV_{np}^j$ is obtained by equation (19). A larger $V_{np}^j$ represents that there are more non-dominated solutions of algorithm $j$ in the obtained solution set $S^j$.

$$V_{np}^j = |S^j - \{x \in S^j | \exists y \in S^p : y < x\}|$$

(3) The ratio of non-dominated solutions ($V_{rd}$)

The metric $V_{rd}$ is used to evaluate the ratio of non-dominated solutions in the obtained solution set $S^j$. The computational process of $V_{rd}^j$ is formulated according to equation (20). A larger $V_{rd}^j$ reveals a solution set of algorithm $j$ with a higher probability for the obtained solution set to be a non-dominated solution. The $V_{rd}^j$ closes to 1 means almost all of the solutions of algorithm $j$ are equal to or near non-dominated solutions.

$$V_{rd}^j = \frac{V_{np}^j}{|S^j|}$$

The computational results consider $V_{pd}$, $V_{np}$, and $V_{rd}$ for the two-phase optimization problems which are summarized in Table 5 and illustrated in Figure 12a-12c. From Figure 12a, the Lévy-NSGA-II algorithm has better distribution than MOCS and NSGA-II. Furthermore, it guarantees stable distribution characteristics when solving different scales of problems. As shown in Figure 12b, the proposed algorithm can achieve the largest number of non-dominated solutions with better quality than compared methods. Thus, the convergence of the Lévy-NSGA-II algorithm is stronger than the compared methods. According to Figure 12c, the proposed non-dominated solution ratio of the Lévy-NSGA-II algorithm is slightly better than MOCS and NSGA-II. Figure 12d gives the computational time of three algorithms when solving different scales of problems and shows that the proposed algorithm takes less than 30% extra computational cost than MOCS and NSGA-II.

6.4. Management implications. We combine virtual cellular manufacturing to develop a dynamic virtual cell reconfiguration framework under multiple planning cycles in a multi-variety small-batch production environment. Some enterprises consider purchasing new equipment only when capacity demands continue to be insufficient. Meanwhile, equipment usually needs a long purchasing cycle in these
Motivated by the above real-world manufacturing features, this proposed framework carries out equipment procurement and capacity adjustment ahead of the following planning cycle and does not consider capacity changes of machines under this planning cycle.

Based on this framework, we think the proposed two-phase DVCF model can conform to real-world manufacturing environments. By executing the two phases of the DVCF model in sequence, implementers do not need to know all the manufacturing orders at the beginning of production, but only need to perform virtual cell formation according to the orders of the next formation period. In virtual cell reconfiguration phases, we take the reconfiguration stability into account to avoid the splitting of original cells. The proposed model is a multi-objective model and may provide multiple alternative solutions. Managers could choose the most suitable plan according to their preferences.

Moreover, quantitative evaluation for manufacturing cells plays an undeniably important role in virtual cellular manufacturing systems. In general, two categories of methods are usually used to evaluate the suitability and adaptability of manufacturing cells. The first category is static performance evaluation. It is usually conducted through mathematical models. These methods focus on quantitative evaluation for the effect of manufacturing cells themselves, including Total Bond Energy (TBE), Proportion Exceptional Element (PE), Machine Utilization (MU), Group Efficiency (GE), Shared Machine Number (SMN), etc. Sarker et al. [37] and Rogers et al. [36] analyzed the quantitative evaluation of manufacturing cells in their works.

The second category is dynamic performance evaluation. It is usually conducted through discrete event simulation models. These methods measure the adaptability
of manufacturing cells in terms of operational performance. For instance, minimum transport between cells, minimum investment cost of new machines, maintain certain utilization rates of machines, minimum production cycles, minimum WIP, etc. Morris et al. [29] analyzed the adaptability of manufacturing cells in terms of operational performance in their works.

In our paper, we focus on the research of virtual cell formation. Thus, we only choose to use a simple indicator (the shared machine numbers) to evaluate the adaptability of manufacturing cells. To objectively evaluate the adaptability of a manufacturing cell in realistic manufacturing scenarios, the evaluation process is suggested as follows:

1. Develop static evaluation indicators (such as PE, MU, GE) for virtual cell formation based on existing studies.
2. Conduct dynamic evaluation of the candidate solutions for virtual cell formation (such as transport between cells, production cycles, WIP, utilization rates of machines, etc.)
3. Select the optimal plan comprehensively according to requirements.

7. Conclusion and future research. In this paper, we propose a dynamic virtual cell reconfiguration framework with multiple planning cycles, and establish a new two-phase dynamic virtual cell formation (DVCF) model for a single planning cycle. To address the complex problem of the two-phase optimization and find better Pareto-optimal solutions, we propose a hybrid NSGA-II combined with a discrete Lévy flight search strategy. We use 6 sets of test problems with various scales to evaluate the performance of the proposed Lévy-NSGA-II algorithm. We compare the computational results of Lévy-NSGA-II with that of MOCS and NSGA-II. The comparison results justify the excellent performance of the proposed method regarding the solution quality and the convergence speed.

For future research, the proposed framework and model will be extended by integrating other assumptions or constraints such as worker training and preventive maintenance. Also, it could be interesting to investigate the performance of Lévy-NSGA-II in other engineering optimization problems such as the multi-objective cell formation problem and the multi-period dynamic virtual cell formation problem.

Appendix: A numerical example of virtual cell reconfiguration. For further clarification of the proposed model, we generate a small-scale case randomly. This problem comprises 12 part types, 33 machines, and 22 process routings. Table 6 presents the parts, their routes, the operation sequence, demands, and operation time to each type of part in each period. The formation period is represented by T. Table 7 gives the process-machine incidence matrix, number of machines included in each machine type, and the corresponding available processing time of a single machine.

We solve the proposed model using the proposed method. Figure 13 presents the solutions obtained for the formation of the first period, the abscissa represents part dissimilarity and the ordinate represents workload balance index. In Figure 13, we fit 9 optimal solutions to the Pareto frontier for further analysis. It shows a strong negative correlation between the two objectives, which indicates the rationality of objective setting.
In Figure 13, solution A presents one of the solutions in the Pareto frontier, with the total dissimilarity objective value is 4.0190, and the workload balance objective value is 0.5939. Table 8 presents the details of solution A.

Based on the formation result, Eq. (8) provides the minimum real number of machines \( Y_{m,c} \) for each virtual cell. Since the capacity of a single machine is not detachable, the real number \( Y_{m,c} \) needs to be rounded up. Thus, the demand quantity for one machine type may exceed the total quantity of that type. Taking M7 as an example, there are three M7 machines in total. According to Table 8, five M7 machines are allocated to three cells, so two M7 machines are shared between virtual cells.

Theoretically, all the results in the Pareto frontier are optimal. However, we also need to consider the autonomy and independence of virtual cells. Thus, we

### Table 6. Parts information for the numerical example

| Parts | Routes | Operation | Demand | Time |
|-------|--------|-----------|--------|------|
| P1    | R1     | 1         | 360    | 90   |
|       |        | 5,7       |        |      |
|       | R2     | 5         | 90     | 170  |
|       |        | 180       |        |      |
| P2    | R1     | 6         | 80     | 2    |
|       |        | 99        |        |      |
|       | R2     | 7         | 90     | 180  |
|       |        | 90        |        |      |
| P3    | R1     | 3         | 90     | 120  |
|       |        | 120       |        |      |
| P4    | R1     | 2         | 50     | 10   |
|       |        | 130       |        |      |
| P5    | R1     | 5         | 100    | 60   |
|       |        | 60        |        |      |
|       | R2     | 6         | 120    | 120  |
|       |        | 60        |        |      |
| P6    | R1     | 1         | 120    | 90   |
|       |        | 60        |        |      |
|       | R2     | 2         | 120    | 7    |
|       |        | 80        |        |      |
| P7    | R1     | 1         | 90     | 90   |
|       |        | 100       |        | 60   |
|       | R2     | 2         | 120    | 90   |
|       |        | 80        |        | 5    |

In Figure 13, solution A presents one of the solutions in the Pareto frontier, with the total dissimilarity objective value is 4.0190, and the workload balance objective value is 0.5939. Table 8 presents the details of solution A.
Table 7. Process-machine incidence matrix for the numerical example

| Operation | Machine type | Machine name            | Machine number | Available processing time /min |
|-----------|--------------|-------------------------|----------------|------------------------------|
| 1         | M1           | CNC lathes              | 5              | 3000                         |
| 2         | M2           | Ordinary lathes         | 5              | 3000                         |
| 3         | M3           | Slotting machines       | 2              | 1700                         |
| 4         | M4           | CNC slotting machines   | 5              | 1600                         |
| 5         | M5           | Grinders                | 6              | 1200                         |
| 6         | M6           | Grinding machines       | 4              | 1200                         |
| 7         | M7           | Gun Drill               | 3              | 1600                         |
| 8         | M8           | Drilling machines       | 3              | 1200                         |

Figure 13. The Pareto-optimal solutions for the first phase

Table 8. One of the schemes for the first period of the numerical example

| Cell | Part(routing) | Number of each machine type in each cell | f1    | f2    |
|------|--------------|------------------------------------------|-------|-------|
|      | M1           | M2 | M3 | M4 | M5 | M6 | M7 | M8 |     |     |
| 1    | 1(2),4(1),8(1),9(2),12(1) | 3 | 3 | 0 | 4 | 2 | 3 | 1 | 3 |     |     |
| 2    | 2(2),5(2),11(1)     | 2 | 2 | 2 | 1 | 5 | 0 | 3 | 0 | 4.0190 | 0.5939 |
| 3    | 6(1),7(2)       | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 1 |     |     |

calculate the number of shared machines for all the solutions of the Pareto frontier. The numbers of shared machines for all the optimal solutions are 2, 2, 3, 3, 4, 2, 4, 3, 5, respectively (Solutions from left to right in Figure 13). Solutions with the minimum number of shared machines are suggested if virtual cells need to keep high independence. Solutions with the minimum result of workload difference objective are suggested if users hope to reduce the accumulation of work-in-process.

The DVCF requires rolling and iterative execution. That is, we need to select the scheme of the current period before solving the next formation period reconfiguration problem. We suggest choosing the scheme based on target preferences and production status. In this case, the first period problem selects solution A as the final scheme.

The second phase of the model adds the objective of reconfiguration stability, leading to a three-objective problem. Based on the solution A, we use the proposed method to solve the second-phase three-objective model, and obtain the Pareto-optimal solutions (Figure 14). Solution B presents one of the solutions in the Pareto
Table 9. One of the schemes for the second period of the numerical example

| Cell | Part (routing) | M1 | M2 | M3 | M4 | M5 | M6 | M7 | M8 | f1     | f2     | f3 |
|------|----------------|----|----|----|----|----|----|----|----|--------|--------|----|
| 1    | 1(2), 8(1), 9(2), 12(1) | 3  | 3  | 0  | 4  | 2  | 3  | 1  | 3  | 3.3619 | 0.6384 |    |
| 2    | 2(2), 5(2), 7(1), 10(3) | 1  | 1  | 1  | 1  | 3  | 0  | 3  | 0  |        |        | 6  |
| 3    | 3(1)            | 1  | 1  | 0  | 0  | 1  | 0  | 1  |    |        |        |    |

Table 10. Detailed machine changes between two formation periods

| Cell | M1 | M2 | M3 | M4 | M5 | M6 | M7 | M8 |
|------|----|----|----|----|----|----|----|----|
| 1    | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |
| 2    | -1 | -1 | -1 | 0  | -2 | 0  | 0  | 0  |
| 3    | +1 | 0  | 0  | 0  | 0  | 0  | -1 | 0  |

Figure 14. The Pareto-optimal solutions for the second period

frontier, with the total dissimilarity objective is 3.3619, the workload balance objective is 0.6384, and machine adjustment quantity is 6. Table 9 presents the detailed solution, including the part families and the number of each machine type in each cell. Compared with the upper period, P4, P6, P11 are no longer needed, while P3, P10 are demanded. Based on the schemes of two periods, Table 10 gives detailed machine changes.

The principle of machine adjustment is to adjust the machines that are already in use. Taking M1 as an example, Cell 2 reduces one machine and Cell 3 adds one,
so the machine in Cell 2 is directly transferred to Cell 3. Therefore, only one M1 is adjusted between these two formation periods.

REFERENCES

[1] T. Blickle and L. Thiele, A comparison of selection schemes used in evolutionary algorithms, *Evolutionary Computation*, 4 (1996), 361–394.

[2] M. Bortolini, E. Ferrari, F. G. Galizia and A. Regattieri, An optimisation model for the dynamic management of cellular reconfigurable manufacturing systems under auxiliary module availability constraints, *Journal of Manufacturing Systems*, 58 (2021), 442–451.

[3] C. W. Chou, C. F. Chien and M. Gen, A multiobjective hybrid genetic algorithm for tft-lcd module assembly scheduling, *IEEE Transactions on Automation Science and Engineering*, 11 (2014), 692–705.

[4] K. Deb, A. Pratap, S. Agarwal and T. Meyarivan, A fast and elitist multiobjective genetic algorithm: Nsga-ii, *IEEE Transactions on Evolutionary Computation*, 6 (2002), 182–197.

[5] K. Deep and P. K. Singh, Design of robust cellular manufacturing system for dynamic part population considering multiple processing routes using genetic algorithm, *Journal of Manufacturing Systems*, 35 (2015), 155–163.

[6] I. Eguia, J. C. Molina, S. Lozano and J. Racero, Cell design and multi-period machine loading in cellular reconfigurable manufacturing systems with alternative routing, *International Journal of Production Research*, 55 (2017), 2775–2790.

[7] B. Erfani, S. Ebrahimnejad and A. Moosavi, An integrated dynamic facility layout and job shop scheduling problem: A hybrid nsga-ii and local search algorithm, *J. Ind. Manag. Optim.*, 16 (2020), 1801–1834.

[8] J. Fan and D. Feng, Design of cellular manufacturing system with quasi-dynamic dual resource using multi-objective ga, *International Journal of Production Research*, 51 (2013), 4134–4154.

[9] H. Feng, W. Da, L. Xi, E. Pan and T. Xia, Solving the integrated cell formation and worker assignment problem using particle swarm optimization and linear programming, *Computers & Industrial Engineering*, 110 (2017), 126–137.

[10] R. Y. Fung, F. Liang, Z. Jiang and T. Wong, A multi-stage methodology for virtual cell formation oriented agile manufacturing, *The International Journal of Advanced Manufacturing Technology*, 36 (2008), 798–810.

[11] H. Guo, M. Chen, K. Mohamed, T. Qu, S. Wang and J. Li, A digital twin-based flexible cellular manufacturing for optimization of air conditioner line, *Journal of Manufacturing Systems*, 58 (2021), 65–78.

[12] W. Hachicha, F. Masmoudi and M. Haddar, Formation of machine groups and part families in cellular manufacturing systems using a correlation analysis approach, *The International Journal of Production Research*, 36 (2008), 1157–1169.

[13] M. Hamedi, G. Esmaeilian, N. Ismail and M. Ariffin, A survey on formation of virtual cellular manufacturing systems (vcms) and related issues, *Scientific Research and Essays*, 7 (2012), 3316–3328.

[14] W. Han, Y. Yu, L. Gao, J. Fang and Z. Li, Virtual cellular inheritance reconfiguration driven by random arrival orders and time window, *Jisuanji Jiexing Zhixao Xitong/Computer Integrated Manufacturing Systems*, CIMS, 24 (2018), 1317–1326.

[15] A. Hosseini, M. M. Paydar, I. Mahdavi and J. Jouzdani, Cell forming and cell balancing of virtual cellular manufacturing systems with alternative processing routes using genetic algorithm, *Journal of Optimization in Industrial Engineering*, 9 (2016), 41–51.

[16] M. Imran, C. Kang, Y. H. Lee, M. Jahanzaib and H. Aziz, Cell formation in a cellular manufacturing system using simulation integrated hybrid genetic algorithm, *Computers & Industrial Engineering*, 105 (2017), 123–135.

[17] B. Karoum and Y. B. Elbenani, Optimization of the material handling costs and the machine reliability in cellular manufacturing system using cuckoo search algorithm, *Neural Computing and Applications*, 31 (2019), 3743–3757.
[18] R. Kia, N. Javadian, M. M. Paydar and M. Saidi-Mehrabad, A simulated annealing for intra-cell layout design of dynamic cellular manufacturing systems with route selection, purchasing machines and cell reconfiguration, Asia-Pac. J. Oper. Res., 30 (2013), 1350004, 41 pp.
[19] A. Kusiak, The generalized group technology concept, International journal of production research, 25 (1987), 501–509.
[20] J. Li, A. Wang and C. Tang, Production planning in virtual cell of reconfiguration manufacturing system using genetic algorithm, The International Journal of Advanced Manufacturing Technology, 74 (2014), 47–64.
[21] J. Q. Li, Q. K. Pan and M. F. Tasgetiren, A discrete artificial bee colony algorithm for the multi-objective flexible job-shop scheduling problem with maintenance activities, Appl. Math. Model., 38 (2014), 1111–1132.
[22] J. Q. Li, Q. K. Pan and F. T. Wang, A hybrid variable neighborhood search for solving the hybrid flow shop scheduling problem, Applied Soft Computing, 24 (2014), 63–77.
[23] J. Q. Li, H. Y. Sang, Y. Y. Han, C. G. Wang and K. Z. Gao, Efficient multi-objective optimization algorithm for hybrid flow shop scheduling problems with setup energy consumptions, Journal of Cleaner Production, 181 (2018), 584–598.
[24] C. Liu, J. Wang and M. Zhou, Reconfiguration of virtual cellular manufacturing systems via improved imperialist competitive approach, IEEE Transactions on Automation Science and Engineering, 16 (2018), 1301–1314.
[25] I. Mahdavi, A. Aalaei, M. M. Paydar and M. Solimanpur, Multi-objective cell formation and production planning in dynamic virtual cellular manufacturing systems, International Journal of Production Research, 49 (2011), 6517–6537.
[26] P. M. Mahdi and S.-M. Mohammad, A hybrid genetic algorithm for dynamic virtual cellular manufacturing with supplier selection, The International Journal of Advanced Manufacturing Technology, 92 (2017), 3001–3017.
[27] M. Mohammadi and K. Forghani, A hybrid method based on genetic algorithm and dynamic programming for solving a bi-objective cell formation problem considering alternative process routings and machine duplication, Applied Soft Computing, 53 (2017), 97–110.
[28] M. Moradgholi, M. M. Paydar, I. Mahdavi and J. Jouzdani, A genetic algorithm for a bi-objective mathematical model for dynamic virtual cell formation problem, Journal of Industrial Engineering International, 15 (2019), 25–40.
[29] V. Rahimi, J. Arkat and H. Farughi, A vibration damping optimization algorithm for the integrated problem of cell formation, cellular scheduling, and intercellular layout, Computers & Industrial Engineering, 143 (2020), 106439.
[30] J. Rezaeian, N. Javadian, R. Tavakkoli-Moghaddam and F. Jolai, A hybrid approach based on the genetic algorithm and neural network to design an incremental cellular manufacturing system, Applied Soft Computing, 11 (2011), 4195–4202.
[31] D. Rogers and S. Shafer, Measuring cellular manufacturing performance, In Manufacturing Research and Technology, 24 (1995), 147–165.
[37] B. Sarker, Grouping efficiency measures in cellular manufacturing: A survey and critical review, International Journal of Production Research, 37 (1999), 285–314.

[38] G. Syswerda, Uniform crossover in genetic algorithms, In Proceedings of the Third International Conference on Genetic Algorithms, Morgan Kaufmann Publishers, (1989), 2–9.

[39] G. M. Viswanathan, S. V. Buldyrev, S. Havlin, M. Da Luz, E. Raposo and H. E. Stanley, Optimizing the success of random searches, Nature, 401 (1999), 911–914.

[40] G. Xue and O. F. Offodile, Integrated optimization of dynamic cell formation and hierarchical production planning problems, Computers & Industrial Engineering, 139 (2020), 106155.

[41] X. S. Yang and S. Deb, Cuckoo search via levy flights, In 2009 World Congress on Nature & Biologically inspired computing (NaBIC), Ieee, (2009), 210–214.

Received July 2021; revised October 2021; early access January 2022.

E-mail address: hust_zzm@hust.edu.cn
E-mail address: zlguan@hust.edu.cn
E-mail address: fangweikang@hust.edu.cn
E-mail address: leileiyok@gzhu.edu.cn