A Novel Integrated Parallel Collaborative Scheduling Model for Fabrication and Assembly in Cellular Manufacturing

Dongfang ZHAO[a] *, Xiaodong ZHANG[b], Yiqi WANG[c], Hongli ZHOU[d]

Donlinks School of Economics and Management, University of Science and Technology Beijing, Beijing100083, P. R. China

[a]zhdf88@163.com, [b]xdzhang@manage.ustb.edu.cn, [c]zhouhongli006@163.com, [d]yqwang_17@163.com

Abstract **— This paper examines the issues of scheduling in fabrication and assembly in multi manufacturing cells, where each cell contains some machines or workstations of different functionality. The parts that are produced in various cells are the component of the product in the assembly cell. We consider the collection of these cells as a manufacturing system (MS). In order to minimize the manufacturing span, an integrated parallel collaborative scheduling model of fabrication and assembly cells is presented by considering the transport time, changeover time, and debugging time. In order to consider the efficiency of the model, a parallel collaborative genetic algorithm (GA) is also proposed. In this, a two-layer hierarchical scheme strategy suggests the GA encoding, which is carried out by a segmented parallel collaborative crossover and mutation of multi-cellular manufacturing (CM). Then the model is tested in a complex electromechanical products workshop which has fabrication and assembly cells. The results indicate that parallel collaborative scheduling by considering the transport time, changeover time and debugging time has a significant positive effect on production. It can reduce the manufacturing span and enhance the machine and operator utilization ratio.**

Keywords-parallel collaborative scheduling; segment parallel collaborative genetic algorithm; cellular manufacturing

I. INTRODUCTION

A cellular manufacturing (CM) layout and logistics are typically designed as product family, which is divided as group technology (GT) [1,2,3]. As an important and flexible method to improve efficiency, CM could meet multi-varieties and small-batch production needed [4]. It means CM needs to produce lots of products at the same time where scheduling is very crucial in production. Scheduling is concerned the products sequence and determined products manufactured in specific machine by process route, determined products start till the end time in each machine. Therefore, high-quality schedules improve the production efficiency to match the customer demand and dynamic environment, the solution method always to shorten the manufacturing cycle time and reduce the manufacturing cost [5,6]. However, in the traditional scheduling studies, some previous researchers were focused on single CM or job-shop scheduling to minimize the total production cycle time or total cost [7-13]. Pajoutan M et al. [12] addressed the CM scheduling problem, and then proposed a mathematical model that minimized the production time by considering material handling time and routing flexibility. Furthermore, researchers developed a simulated annealing-based heuristic to solve the model. Naderi B et al. [13] believed that the scheduling of a flexible machine (MC) could be paralleled machine process. Then proposed a mixed integer linear programming model which minimized production time.

In industrial practice, the product is made up of many parts which are needed to produce and assembled in several CMs to become the final product. These CMs are to be a manufacturing system (MS). Although the scheduling research is optimal in single CM, the MS efficiency is poor, and it means the cycle time of MS might be shortened. The main reasons of situation include: (1) the part which is produced in CM is not needed in assembled CM at the same time, so that the production between CMs is not collaborative, which cannot meet the assembly just in time (JIT). (2) CMs are not parallel collaborating production so that the MS efficiency is reduced.

Because of this, the fabrication and assembly scheduling problem of multi-CMs has received significant attention in nowadays. Consequently, various solution methods [14-25] were proposed to solve this issue. Yu Xiaoyi et al. [15] studied collaborative scheduling planning of multi-assembly workshops based on assembly theory of constraint (TOC), and presented a model that minimizes production cycle time. At last, a parallel collaborative evaluation genetic algorithm (GA) proposed to solve the model. However, this research needed all the sub-parts should be completed and then begin to be assembled. Xiong Fuli et al. [16] studied a mixed-batch job-shop and a mixed-model assembly workshop, which are series connection together and presented two-stage integrated optimization model of production planning and scheduling, based on nonlinear mixed integer programming (NMIP) which considered machine capacity, etc. and solved the model with the alternant iterative method by hybrid GA. Although, this research examined the process and assembled scheduling, the two-stage presented model did not parallel collaborative considering the processing and assembly operation. Liang Yanjie et al. [17] studied the integrated optimization scheduling problem of processing and assembly flow shop, presented a mathematical model to minimize the maximum completion time of line and solved the model through the GA. Through this method they can reduce the work-in-process (WIP) inventory, however, in this research the products moved by the whole batch, i.e. when the number of same products produced on a machine was finished then whole products transferred to the next machine. Halat K et al. [18] studied the part which provided in different CMs, i.e. some of the processes of one part were produced in one CM, other processes of this part were generated in another. Then they presented a mathematical model that minimized manufacturing cycle time in multi-CMs. In the model, they considered the processing sequence, preparation time, and transport time. At last, they solved the model by GA. Elmi A et al. [19] studied the multi-CMs scheduling problem by considering transport time of part between CMs, then proposed an integer linear programming (ILP) model to minimizing makespan, and solved it by simulated annealing (SA). Salmasi N et al. [20] studied the flow shop scheduling problem which considered setup time based on the sequence-dependent group and proposed a model to minimizing total flow time. Al-Anzi F S et al.[21] studied the two-stage assembly flow shop scheduling problem, which considered the setup times. The first stage has many machines while the second stage has only one machine. Then a model was proposed to minimizing completion time of all parts and solved it by particle swarm optimization (PSO). Sung C S et al. [22] studied two-stage assembly scheduling problem, in each stage the shop had two machines. In the first stage the machine produced components, in the second stage, the machine assembles the components to the product. The core objective of this research was to minimizing the sum of completion times, then to solve the model by the branch-and-bound and heuristic algorithm (BBHA). However, sequence of operation and transport time were ignored. Besides, products in a shop have less than four processes is considered. Allahverdi A et al. [23] also studied two-stage assembly scheduling problem, in the first stage has m machines, in the second stage has only one machine for the assembly operation. However, when the job completed in the first stage, then to produce in the second stage. In order to minimize makespan with setup times, heuristics is proposed to solve the scheduling problem. Liao C J et al. [24] also studied the assembly scheduling problem in two-stage, in the first stage components cell was produced while in the second stage assembly cell. There is only one machine in each stage. At last, they proposed a mathematic model of minimizing makespan by considering the batch setup times. Xu J et al. [25] studied the assembly scheduling problems in work centers (i.e. job-shop), presented a mixed integer linear programming model to minimize makespan by considering the tree-structure precedence constraints, then solved the model by Lagrangian relaxation (LR) approach. But neglected the preparation time and transport time.

As addressed by above researchers, most of the studies provided various fabrication and assembly solutions for multi-CMs or job-shop, but the researchers focused on flow-shop cell and assembled the cell. Few studies have investigated flexibility cell and flexibility assembly cell. Furthermore, very few studies have investigated cell scheduling problem by considering changeover time and debugging time. On fabrication and assembly scheduling, changeover time and debugging time have significant effects on production. Although most of the researchers focused on cell scheduling problem which considered the preparation time and setup time, these times were fixed value. MC contains many flexible machines to date which could be produced different family products where changeover time of each part is not fixed value and depends on the previously process of another part on that machine. For example, if two parts belong to the same family, they produced one after another on the same machine, when one part finished production and another part begin to produce, the machine does not change over production tooling, it only to do debugging production; so the machine does not have change over time, and it has only debugging time. To the best of our knowledge, no researchers have presented a mathematics programming model to solve scheduling problem regarding change over time, debugging time, transport time. In addition, most of the studies considered the fabrication and assembly scheduling by two stage, which could not integrate considering scheduling problem in fabrication and assembly cellular. Today's multi-varieties and small-batch production environment, integrated parallel collaborative between different fabrication and assembly cells become important for reducing makespan and improve the efficiency. The best production strategy in assembly cell is to transport the part to assembly process just in time (JIT). Considering these factors necessitate integrated parallel collaborative scheduling in MS.

In this paper, fabrication and assembly multi-CMs regarded as a system, an integrated collaborative scheduling model based on parallel processing by considering the part transport time, changeover time and debugging time is proposed to minimizing makespan. Also, we designed a segment parallel collaborative evaluation genetic algorithm (SPCEGA) to solve the model. The remainder of this paper is organized as follows: In Sect. 2, the detail scheduling problem is described, and assembly node definition is presented, then the integrated parallel collaborative scheduling model of fabrication and assembly multi-CMs is proposed. In Sect. 3, integrated multi-CMs GA solution method is designed for parallel and collaborative scheduling model. In Sect. 4, the scheduling case study and comparison is applied in complex electromechanical products workshop, Sect. 5 is dedicated to the conclusions of the research.

II. PARALLEL COLLABORATIVE SCHEDULING MODEL

A. Problem Description

The product has several kinds of parts which are produced in different CMs. These CMs contains fabrication and assembly process. In fabrication CM, parts produced on the different machines, when the part is completed, transport it to assembly process of assembly cell just in time. At last, when the part is assembled in assembly cell, finally a new product is arrival. Fabrication and assembly CMs has many machines, at the same time they can parallel produce a lot of parts in each CM. It means that multi-CMs should be collaborative and parallel produced to complete product on time. According to this characteristic, all

CMs involved in production will be as an integrated MS, which could be parallel collaborative produce a number of products at the same time, the MS production processing model of multi-CMs is shown in Fig.1. In MS, some CMs include flexible computer numerical control (CNC) machines. In production, changeover time and debugging time should be considered when machine finished one batch products to produce another. Some CMs include the workstation when beginning to produce bath products preparation time should be considered, such as the manual workstation. Because of MS consist of multi-CMs and multi- products, we should be regarded as parallel and collaborative scheduling problem in actual production. Thus, the paper scheduling problem can be described: an MS consist of *g* CMs in which *s* products is produced. In order to achieve parallel collaborative production and get minimum cycle time of MS, we should determine production sequence of the parts *i*, *i'* according to *the* f^{th} product by considering the debugging time, changeover time and transport time. In the study, assuming that all CMs are located in the one workshop, the transport time of parts is negligible in CM, but transport time is fixed value between CMs. Besides, the changeover time depends on the previously process of another part on the machine. To better understand the researching of multi-CMs scheduling, as addressed in the literature[17]. Furthermore, the assembly node is defined as follows:

Definition 1: In MS, the process of one part which is assembled by another completion part is defined as assembly node. If the process of the part is not an assembly node, all CMs parallel and collaborative produce. If the part has assembly node, another assembled part must be completed and transported to assembly process before it starts to operation.

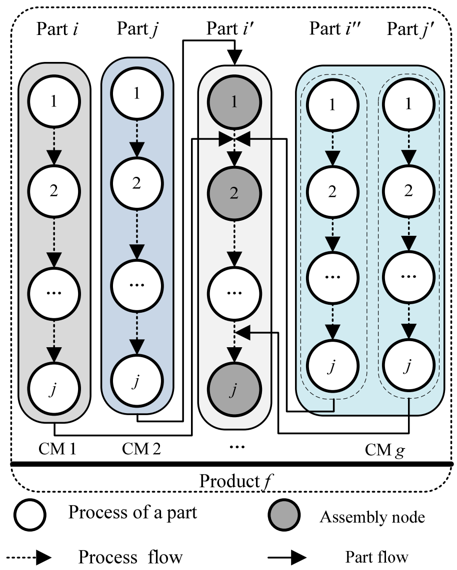


Figure 1. The MS production processing model of multi-CMs

B. *Parallel and Collaborative Scheduling Models*

To illustrate the problem, the following parameters notation, decision variables, and describing will be used.

TABLE 1. PARAMETERS NOTATION AND DESCRIPTION

Parameters notation	Description
g, g'	index for number of CM, *g, g'*=1, 2,…, *b* (*b* is the total number of CMs)
f, f '	index for number of product, *f, f '*=1, 2,…, *s* (*s* is the total number of products)
i, i'	index for number of parts, *i, i'*=1, 2,…, *a* (*a* is the total number of parts)
j, j'	index for number of process of a part (*c* is the total number of a part process)
m, m'	index for number of machine or workstation, *m, m'*=1, 2,…, *M* (*M* is the total number of machines and workstation in MS)
O_{ij}	the j^{th} process of part i
Q_i	the demand of part *i*
Q_f	the demand of product *f*
d_{fi}	the demand of part *i* when produce a product *f*
t_{gijm}	processing time of process *j* of part *i* produced on machine or workstation *m* in CM *g*
pt_{gijm}	preparation time of process *j* of part *i* produced on workstation *m* in CM *g*
st_{gijm}	changeover time of process *j* of part *i* produced on machine *m* in CM *g*
dt_{gijm}	debugging time of process *j* of part *i* produced on machine *m* in CM *g*
$ct_{igg'}$	transport time of part *i* from CM *g* to CM *g'*
ST_{gijm}	start time of process *j* of part *i* produced on machine or workstation *m* in CM *g*
ET_{gijm}	completion time of process *j* of part *i* produced on machine or workstation *m* in CM *g*
CT_{gijm}	start changeover time of process *j* of part *i* created on machine *m* in CM *g*
ET_{gi}	completion time of part *i* produced in CM *g*
PT_{gijm}	start preparation time of process *j* of part *i* produced on workstation *m* in CM *g*

TABLE 2. DECISION VARIABLES AND DESCRIPTION

Decision variables	Description
X_{gijm}	1 if process *j* of part *i* produced on machine or workstation *m* in CM *g*, 0 otherwise
Y_{fi}	1 if part *i* assembled in product *f*, 0 otherwise
$Z_{gii'}$	1 if part *i* and *i'* belongs to the same part family, 0 otherwise
V_{ig}	1 if part produced in CM *g*
$W_{gig'i'j'}$	1 if part *i* completed in CM *g* is assembled on process *j'* of part *i'* in CM *g*, 0 otherwise

Because of the complexity of the MS, the manufacturing cycle time is longer than single CM. In order to reduce the makespan, the problem could be formulated as the following model:

$$\min T = \max(ET_f) \tag{1}$$

Subject to:

$$\sum_{m=1}^{M} X_{gijm}=1,\ \forall g,\ i,\ j \tag{2}$$

$$\sum_{j=1}^{c} X_{gijm}=1,\ \forall g,\ i,\ m \tag{3}$$

$$\max(ET_{gijm}+ct_{igg'}) \le ST_{g'i'j'm'},\ \forall \sum_{\substack{i=1\\ i\ne i'}}^{a} W_{gig'i'j'} \ge 1 \tag{4}$$

$$Q_i = Q_f d_{fi} Y_{fi} \tag{5}$$

$$ET_{gijm} \ge ET_{gi(j-1)m'} + t_{gijm},\ \ m \ne m' \tag{6}$$

$$ST_{gijm} \ge ST_{gi(j-1)m'} + t_{gi(j-1)m'}\ ,\ \ m \ne m' \tag{7}$$

$$ET_{gijm} = CT_{gijm} + \sum_{m=1}^{M}(Q_i t_{gijm} X_{gijm} + Z_{gii'} st_{gijm} X_{gijm} + dt_{gijm} X_{gijm}) \tag{8}$$

$$ET_{gijm} = PT_{gijm} + \sum_{m=1}^{M}(Q_i t_{gijm} X_{gijm} + pt_{gijm} X_{gijm}) \tag{9}$$

$$ST_{gijm} = CT_{gijm} + \sum_{m=1}^{M}(Z_{gii'} st_{gijm} X_{gijm} + dt_{gijm} X_{gijm}) \tag{10}$$

$$ST_{gijm} = PT_{gijm} + \sum_{m=1}^{M}(pt_{gijm} X_{gijm}) \tag{11}$$

$$ET_{gi} = ST_{gicm} + \sum_{m=1}^{M}(Q_i t_{gicm} X_{gicm}) \tag{12}$$

$$ET_f = \max(ET_{gi}) \tag{13}$$

$$\sum_{g=1}^{b}\sum_{i=1}^{a}\sum_{j=1}^{c} V_{ig} O_{ij} = \sum_{g=1}^{b}\sum_{i=1}^{a}\sum_{j=1}^{c}\sum_{m=1}^{M} X_{gijm} \tag{14}$$

The objective function (1) is the minimization of production cycle time of MS. Constraint (2) and (3) make sure that only one process is produced on one machine and only one machine is produced one process during the processing time. Constraint (4) ensures assembly part should be completed and transported to assembly node before it producing. Constraint (5) is the demand of part *i*. Constraint (6) and (7) to start to produce process j^{th}e of part *i* when its *j*-1 process has been completed. Constraint (8) and (9) are the whole processing time of process j^{th} of part *i*. Constraint (10) and (11) are the start time of process j^{th} of part *i*. Constraint (8) and (10) are CNC machine producing time, Constraint (9) and (11) are workstation processing time. Constraint (12) is completion time of part *i* when it is produced in CM *g*. Constraint (13) is completion time of product *f*. Constraint (14) ensures total process equates to the overall operation.

III. INTEGRATED MULTI_CMS GA SOLUTION METHOD DESIGN

A. *Segment Parallel Collaborative Scheduling Algorithm Process*

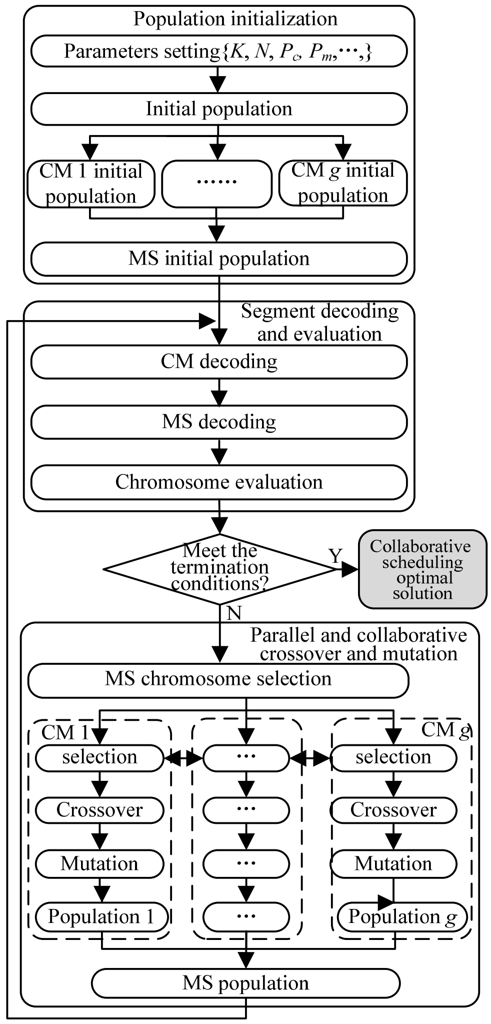


Figure 2. The algorithm process structure framework of parallel collaborative scheduling.

According to the characteristics of parallel collaborative production of CMs, an improved SPCEGA is used for solving scheduling model, as shown in Fig. 2. The main idea of Fig. 2 is to set the parameters *K*, *N*, *Pc*,

DOI 10.5013/ IJSSST.a.17.29.36 ISSN: 1473-804x online, 1473-8031 print

etc., generate CM's initial population. Then, an initial population of MS is generated based on an initial population of CMs. Thirdly, the CMs chromosomes and MS chromosomes are decoded by stages, and all of them are evaluated. Finally, in order to generate the new population of MS, apply the parallel crossover and mutation in each CM.

The whole algorithm process structure framework of parallel collaborative scheduling is as follows:

Step 1. Parameters setting. In the initial stages of parallel cooperative scheduling, the population sizes K, evolutionary generations N, crossover probability P_c, mutation probability P_m etc. are stetted.

Step 2. Initial population. According to the encoding regular, parallel generate initial population size of each CM, which is combined into MS initial population.

Step 3. Decoding. The decoding process is divided into CM decoding and MS decoding. The CM decoding determines the production sequence and processing time of each part in CM. MS decoding determines the processing time of assembly node, so as to determine the cycle time of MS.

Step 4. Individual evaluation. Evaluate each chromosome according to the fitness value, if it reaches the termination conditions, then the scheduling plane is to be output. Or else the process goes to step 5.

Step 5. Parallel and collaborative evolution. To generate the new chromosome, respectively apply the selection, crossover, and mutation to each CM population, and then all of the CM chromosomes combined into MS chromosomes.

Step 6. After generating new population, go back to step 3, and continue the decoding process.

B. *Encoding Scheme of Chromosome*

A two-layer scheme for chromosome encoding is proposed to illustrate the problem where each of chromosomes consists of two sections. The length of the first layer is equal to the strength of the second layer, length of each layer is $\sum_{j=1}^{c} 2*Z_{ig}O_{ij}$, and the first section length is equal to the second section, as illustrated in Fig. 3. In the first layer, the first section is processed, which indicates the produce numbers of a part and production sequence in the machine or workstation. In process section, one part has the same genes notation, and numbers of genes of a part are a total process of the part. The second section is machine or workstation, which are one-to-one correspondence with process encoding and indicates the process produced on the corresponding machine or workstation. In the second layer, the first section is the assembly encoding, which means the part is assembled to the process of index part. The second section is CM encoding, which indicates the machine, workstation and part located in which CM. For example, a chromosome is $\begin{bmatrix} 1113334421364587 \\ 0000140011122233 \end{bmatrix}$, in the first layer, it shows that part 1-3 have 3, 3, and 2 processes; each process 1, 2, and 3 of part 1 is produced on machine 2,1,3, others like this. In the second layer, part 1is assembled to process 2 of part 3, part 4 is assembled to process 3 of part 3; part 1 is produced in CM1, part 3 is produced in CM2, and part 4 is produced in CM3.

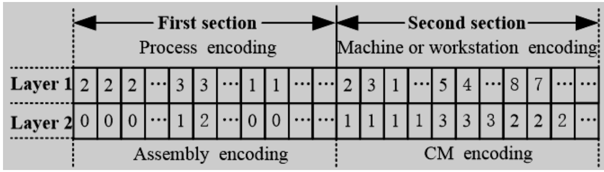


Figure 3. Two-layer scheme for chromosome representation

C. *Segment Decoding Scheme of Chromosome*

First of all, in first stage, to solve the CM decoding, the parallel production time is obtained in each CM. Later, for MS decoding, the collaborative production time was obtained, according to the assembly node, the MS cycle time is to be output.

(1) The CM decoding. There are two situations in CM decoding. On the one hand, the chromosomes apply decoding in CM if the CM does not have assembly node; on the contrary, if the CM has assembly node, we should take decoding before assembly node in CM. The detail process is as follows:

Step 1. Take out one gene from a chromosome, and convert it to be produced process j of part i, machine or workstation m, CM g and assembly relation $W_{gig'i'j'}$.

Step 2. Obtain the scheduling condition of process j-1 of part i and machine or workstation m, m'; ensure $Z_{gii'}$, i.e. part i and i' if in the same family.

Step 3. Obtain the completion time $ET_{gi'j'm}$ of process j' of part i' which was produced on machine m or workstation m before process j of part i. If the process produced on machine m, obtain the changeover time st_{gijm} and debugging time dt_{gijm} of process j of part i. If the process produced on workstation m, obtain the preparation time pt_{gijm} of process j of part i. Besides, if $j\neq1$, obtain the first one completion time of process j of part i, and get the completion time $ET_{gi'j'm}$ of process j' of part i'.

Step 4. The earlier start time of process j of part i is as follows:

1) If j=1 and the process is produced on machine m, the start time of process j of part i is shown as Formula (15),

$$ST_{gijm} = ET_{gi'j'm} + Z_{gii'}st_{gijm} + dt_{gijm} \tag{15}$$

If $j\neq1$ and the process are produced on workstation m, the start time of process j of part i is shown as Formula (16),

$$ST_{gijm} = ET_{gi'j'm} + pt_{gijm} \tag{16}$$

2) If $j\neq1$, the Δt is calculated by Formula (17),

$$\Delta t = ST_{gi(j-1)m'} + t_{gi(j-1)m'} - ET_{gi'j'm} \tag{17}$$

Here, Δt is the value of completion time $ET_{gi'j'm}$ of process j' of part i' minus the first one completion time of process j of part i. If $\Delta t > 0$, the start time of process j of part i is shown as Formula (18). If $\Delta t < 0$ and the process is produced on machine *m*, the start time of operation *j* of part *i* is shown as Formula (19). If $\Delta t < 0$ and the process produced on workstation *m*, the start time of operation *j* of part *i* is shown as Formula (20).

$$ST_{gijm} = ST_{gi'j'm} + t_{gijm} \tag{18}$$

$$ST_{gijm} = ET_{gi'j'm} + Z_{gii'}st_{gijm} + dt_{gijm} \tag{19}$$

$$ST_{gijm} = ET_{gi'j'm} + pt_{gijm} \tag{20}$$

Step 5. If all chromosome ended decoding, the CM decoding should stop, or else the algorithm will go back to step1.

(2) The MS decoding. The MS decoding is to determine the product assembly time based on assembly node, the process is presented as below:

Step 6. Get the assembly relationship $W_{igji'g'j'}=1$ of part *i* in CM *g* and part *i'* in CM *g'*, if $\sum_{i=1, i\neq i'}^{a} W_{gig'i'j'} = 1$, the start time of process of *j'* of part *i* is shown as Formula (21); if $\sum_{i=1, i\neq i'}^{a} W_{gig'i'j'} > 1$, the start time of process of *j'* of part *i* is shown as Formula (22).

$$ST_{g'i'j'm'} = ET_{gijm} + pt_{g'i'j'm'} + ct_{igg'} \tag{21}$$

$$ST_{g'i'j'm'} = \max(ET_{gijm} + pt_{g'i'j'm'} + ct_{igg'}) \tag{22}$$

Step 7. When the part assembled, if the next process is not assembly node, the algorithm go back to step 2. Otherwise, the algorithm goes back to step 6.

Step 8. If all of the chromosome which in assembly node CM has ended decoding, the algorithm should stop, or else the algorithm goes back to step 2.

D. Segment Parallel Crossover and Mutation

Because the processing sequence cannot be changed and to illustrate chromosome crossover along with mutation, gene block is defined as below:

Definition 1: A gene block is a number of genes of the chromosome. In order to not change the production sequence of the part, the genes of a part in one chromosome as a gene block, and then apply the crossover and mutation to these gene blocks.

As characteristic of parallel and collaborative production, the production sequence is the primary constraint on MS cycle time. In turn, the location of CM encoding cannot influence the MS cycle time. Therefore, the first layer and the second section of the second layer apply the parallel crossover and mutation, it means that the gene blocks implementing the parallel crossover and mutation in each CM. Because the crossover and mutation do not consider the CM location, so the CM encoding do not apply crossover and mutation.

(1) Segment parallel crossover of multi-CMs

As a significant operation in GA, the crossover can effect on population diversity. So, according to chromosome structure, the paper adopts a strategy of multiple point crossovers (MPC), with the method: according to parts number *a'* of CM *g*, *a'* numbers of b_s integer is randomly generated. The b_s is belonged to $\{b_s \in [0,1] | b_s \in N\}$. Then b_s are corresponding to the gene blocks of chromosome of CM *g* which the numbers are *a'*. At last, if b_s=0, the corresponding gene blocks of father chromosome is exchanged with each other to generate the offspring. When crossovers, the genes sequence is not changed in gene block.

(2) Segment parallel mutation of multi-CMs

For keeping the population diversity, the chromosome applies parallel mutation in each CM. We take the two gene block with random selection in order to exchange in CM to achieve mutation, Hence, it generate the new population.

IV. CASE STUDY

In this section, we take the electromechanical products workshop scheduling as an example, the results of fabrication and assembly multi-CMs experiment are described. The experiment is to compare the efficiency of the different scheduling model, which one is integrated parallel collaborative scheduling model proposed by this paper considered change over time, debugging time, transport time; another model is considered transport time, preparation time and setup time.

A. Basic Information of Workshop

The workshop produces multi-varieties and small-batch products, which has 3 CMs, with g1-g3 index. In *g*1, there are 7 CNC machines (index is *M*1-*M*7) that produce mechanical products. In *g*2, there are 2 CNC machines (index is *M*8-*M*9) and 3 manual workstations (index is *M*10-*M*12) which produce electronic products. In *g*3, there are 2 manual workstations (index is *M*13 and *M*12) and 2 CNC machines (index is *M*14 and *M*16). The *g*3 is a assemble CM, many parts which come from *g*1 and *g*2 assembled together. To illustrate the experiments, 6 typical products (index is *P*1-*P*6) is to be researched. Frequent changeover characterizes these products; preparation time is longer, and demand is larger. *P*1-*P*6 are consist of 21 parts (index is *E*1-*E*21). *P*1 is produced by *E*1, *E*9 and *E*16; *P*2 is produced by *E*2, *E*10 and *E*17; *P*3 is produced by *E*3, *E*11, *E*12 and *E*18; *P*4 is produced by *E*4, *E*5, *E*13 and *E*19; *P*5 is produced by *E*6, *E*7, *E*14 and *E*20; *P*6 is produced by *E*8, *E*15 and *E*21. The above-mentioned parts, *E*1, *E*2 and *E*3 are one product family; *E*4 and *E*5 are one product family; *E*9, *E*12 and *E*15 are one product family; *E*11 and *E*13 are one product family; *E*16 and *E*17 are one product family; *E*18 and *E*20 are one product family. The detail demand of products and product component is shown in table 3. In table 4 and table 5, parts changeover time and debugging time are shown. The assembly information and transport time are shown in table 6. The processing sequence and processing time of each part is listed in Table 7.

TABLE 3. PRODUCT COMPOSITION, D_{FI} AND PRODUCTION QUANTITY

Product	Q_f	Part	d_{fi}	Q_i	Product	Q_f	Part	d_{fi}	Q_i
*P*1	12	*E*1	1	12	*P*2	12	*E*2	1	12
		*E*9	1	12			*E*10	2	24
		*E*16	1	12			*E*17	1	12
*P*3	13	*E*3	1	13	*P*4	15	*E*4	2	30
		*E*11	2	26			*E*5	1	15
		*E*12	1	13			*E*13	1	15
		*E*18	1	13			*E*19	1	15
*P*5	13	*E*6	1	13	*P*6	8	*E*8	1	8
		*E*7	1	13			*E*15	1	8
		*E*14	1	13			*E*21	1	8
		*E*20	1	13			-	-	

TABLE 4. PREPARATION TIME, CHANGEOVER TIME AND DEBUGGING TIME OF PART (MIN)

Part	CM *g*1						
	*M*1	*M*2	*M*3	*M*4	*M*5	*M*6	*M*7
*E*1	30/19	18/12	-	15/12	-	-	-
*E*2	-	20/14	-	-	41/29	52/31	34/28
*E*3	-	-	-	-	39/25	-	31/27
*E*4	-	19/17	-	-	38/24	-	30/25
*E*5	27/16	-	34/24	14/12	-	-	-
*E*6	31/18	-	43/27	21/13	-	-	-
*E*7	-	-	31/12	-	28/11	34/13	-
*E*8	-	-	-	20/11	24/10	21/12	-

//// TABLE 5. PREPARATION TIME, CHANGEOVER TIME AND DEBUGGING TIME OF PART (MIN)

Part	CM *g* 2					part	CM *g* 3			
	*M*8	*M*9	*M*10	*M*11	*M*12		*M*13	*M*14	*M*15	*M*16
*E*9	40/12	51/13	-	21	15	*E*16	12	16	32/12	29/8
*E*10	-	32/12	13	10	12	*E*17	8	34/13	9	34/12
*E*11	36/9	-	11	9	-	*E*18	12	33/10	12	35/10
*E*12	46/9	38/11	9	-	11	*E*19	10	42/11	11	34/16
*E*13	23/9	32/14	16	10	11	*E*20	38/11	42/9	10	12
*E*14	35/11	36/12	-	11	9	*E*21	18	17	42/21	41/9
*E*15	17	43/19	14	46/16	-	-	-	-	-	-

In table 4 and table 5, if an operation on the machine, before the "/" changes over time, after the "/" is debugging time, the preparation time and setup time is changeover time and debugging time. If the operation on a workstation, there just have preparation time.

TABLE 6. PRODUCT ASSEMBLY INFORMATION AND TRANSPORT TIME (MIN)

$W_{giq'i'j'}/t_{iqq'}$										
*E*18	*E*3	*E*11	*E*12	*E*16	*E*1	*E*9		*E*17	*E*2	*E*10
	0	0	0		0	0			0	0
	1/8	0	0		1/12	0			1/12	0
	0	1/5	0		0	1/10			0	1/5
	0	0	1/6		0	0			0	0
*E*19	*E*4	*E*5	*E*13	*E*20	*E*6	*E*7	*E*14	*E*21	*E*8	*E*15
	0	0	0		0	0	0		0	0
	1/7	0	0		0	1/10	0		0	0
	0	0	1/6		1/12	0	0		1/21	0
	0	1/15	0		0	0	1/14		0	1/22

In table 6, *E*1-*E*15 are assembly parts. According to *E*16-*E*21, the *E*1-*E*15 are assembled in assembly node process. For example, *E*1 is assembled to the second process of *E*16, *E*9 is assembled to *the* third process of *E*16, and transport time of *E*1and *E*9 to *E*16 is 12 minute and 10 minute. This can illustrate by assembly node; the above example is $W_{113(16)2}$ and $W_{293(16)3}$.

TABLE 7. PROCESSING TIME OF PART (MIN)

CM	machine	*E*1	*E*2	*E*3	*E*4	*E*5	*E*6	*E*7	*E*8
*g*1	*M*1	29	-	-	-	27.5	29.8	-	-
	*M*2	28.4	48	-	29	-	-	-	-
	*M*3	-	-	-	-	27.5	29	34.5	-
	*M*4	28	-	-	-	27	30	-	37.5
	*M*5	-	-	30	30	-	-	35	38
	*M*6	-	49	-	-	-	-	34.2	37
	*M*7	-	48.6	30.5	29.5	-	-	-	-
CM	machine	*E*9	*E*10	*E*11	*E*12	*E*13	*E*14	*E*15	-
g 2	*M*8	32	-	23	43	37	28	36	-
	*M*9	31	28	-	42	36	26	35	-
	*M*10	-	32	11	35	35	-	37	-
	*M*11	45	31	9	-	36	31	36	-
	*M*12	43	30	-	36	35	30	-	-
CM	machine	*E*16	*E*17	*E*18	*E*19	*E*20	*E*21	-	-
g 3	*M*13	32	28	29	35	15	18	-	-
	*M*14	36	30	30	34	20	18	-	-
	*M*15	35	29	30	35	14	17	-	-
	*M*16	37	37	28	34	21	20	-	-

B. Scheduling Results Analysis

To compare different scheduling results, there are two schemes are designed. Scheme 1 is solved by the integrated parallel collaborative scheduling model by considering change over time, debugging time, and transport time as proposed by current paper. Collaborative scheduling model solved Scheme 2 by considering preparation time, setup time and transport time, this method is mentioned in the Sect.1 of the literature Review.

Based on the above information, we used the table 1-talbe 6 as the input, but there is one different setting in two schemes. In Scheme 1, the changeover time is depending on previously part on that machine. In Scheme 2, the preparation time is debugging time in table 4, setup time is changeover time in table 4, these preparation time and setup time are fixed value. Based on the above information, we use the table 1-talbe 6 as the input, the model being developed in Matlab programming. In the model, setting the population size is 100, evolution generation is 300, crossover probability is 0.9, and mutation is 0.05. The scheduling results are shown in Fig.4 and Fig.5.

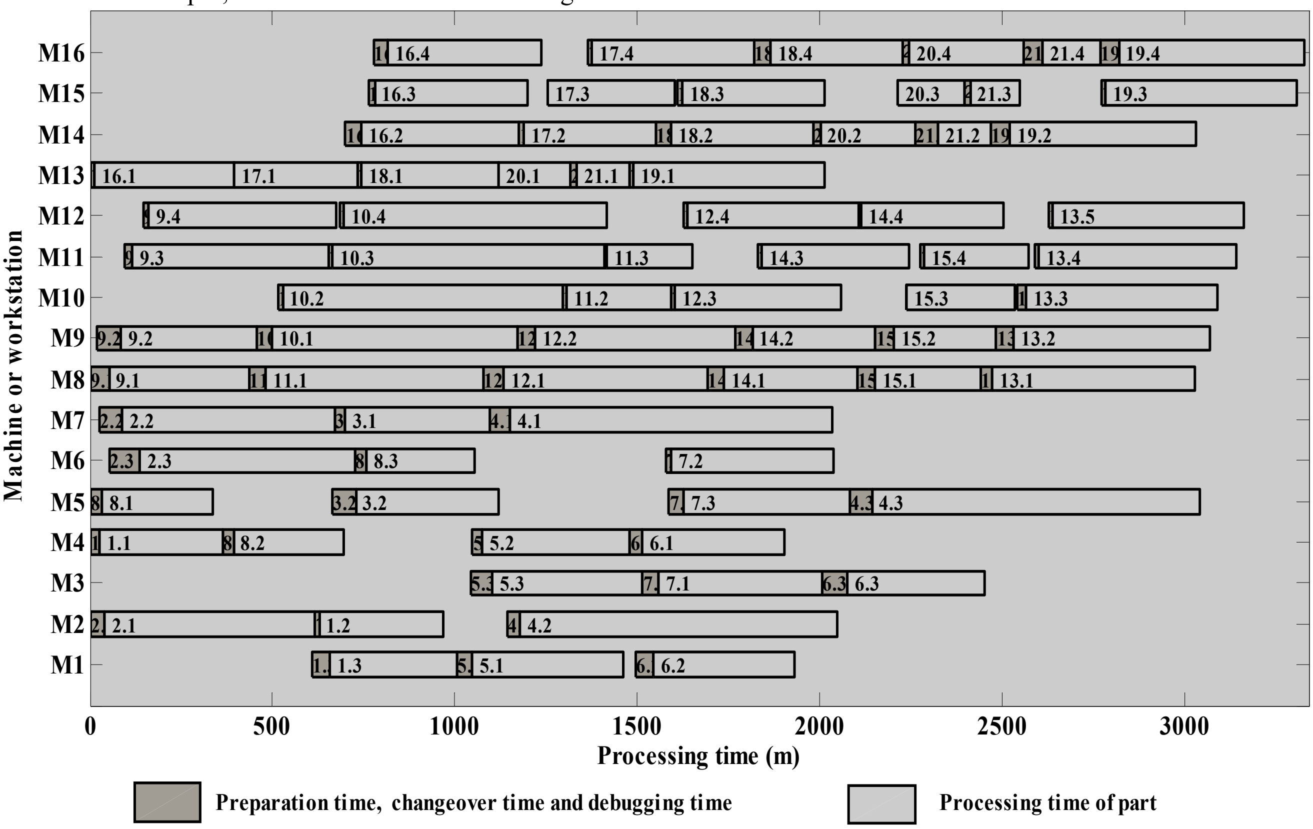


Figure 4. Scheme 1 scheduling result

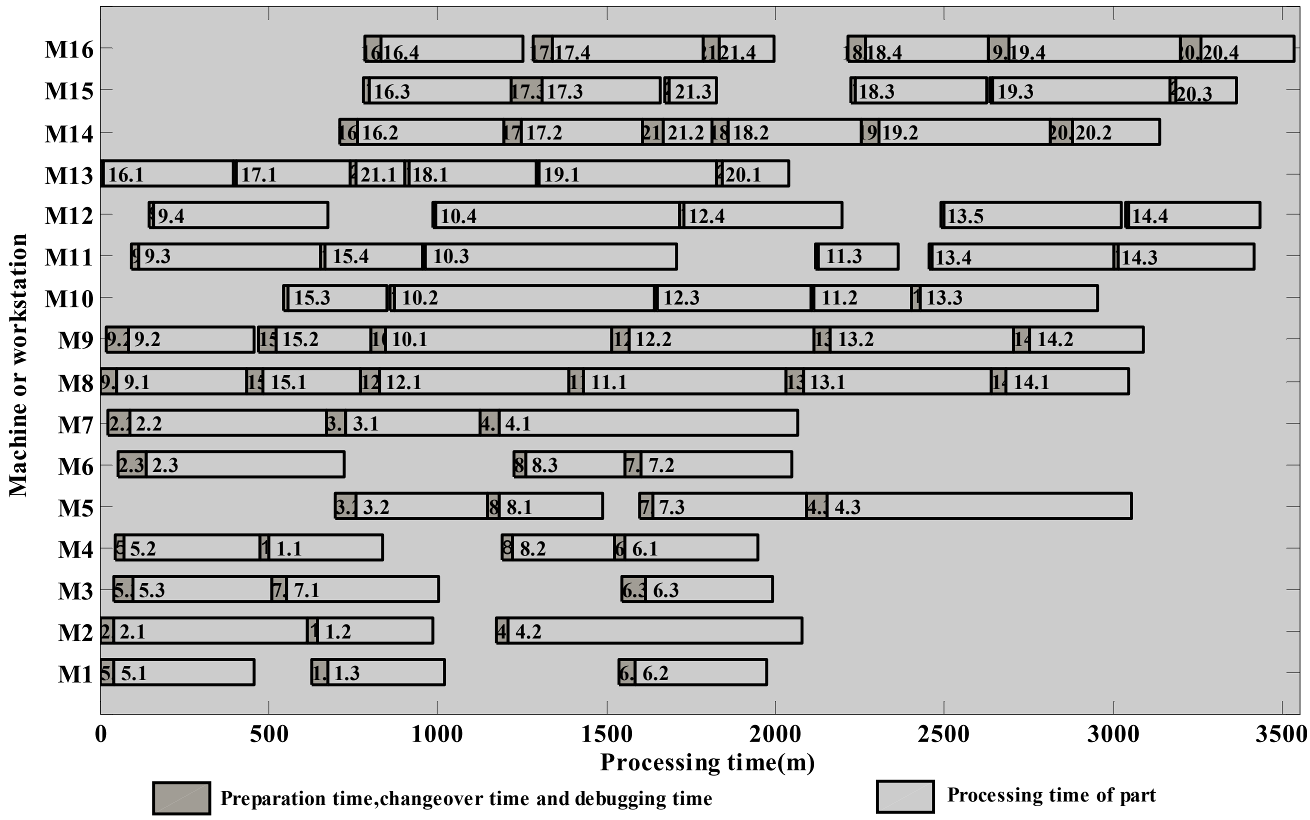


Figure 5. Scheme 2 scheduling result

In Fig.4 and Fig.5, the index of 1.1 indicates the first process of part $E1$, the index of 1.2 indicates the second process of part $E1$, others like this. Besides, the product should be produced by sequence, i.e. when the process j-1 of part i is completed, the process j of part i could begin to produce. Fig.4 is scheduling result of scheme 1, the minimal scheduling time of MS is 3329 minutes. Fig.5 is scheduling result of scheme 2, the minimal scheduling time of MS is 3596 minute. Compare with two scheduling models, and the scheme 1 reduces 267 minutes; productivity of scheme 1 is improved about 7.42%.

It can be seen, in scheme 2, the completion time of parts in CM $g1$ and CM $g2$ is longer than the completion time of parts in CM $g1$ and CM $g2$ in scheme 1, the MS scheduling time is shorter in Scheme 1. This indicates paper proposed scheduling model is better. In Scheme 2, because of preparation time of the changeover time is fixed value, thus the production time is longer. In addition, production efficiency is a significant concern in the workshop. In order to analyze production efficiency, the machine and operator utilization ratio, average machine utilization ratio and operator utilization ratio are compared in scheme1 and scheme 2, thus to check which scheme has the best efficiency, comparison results are shown in Fig.6 and Fig.7. In Fig. 6, $M1$-$M9$, $M14$, and $M16$ are the machine utilization ratio; $M10$-$M13$ and $M15$ are the operator utilization ratio.

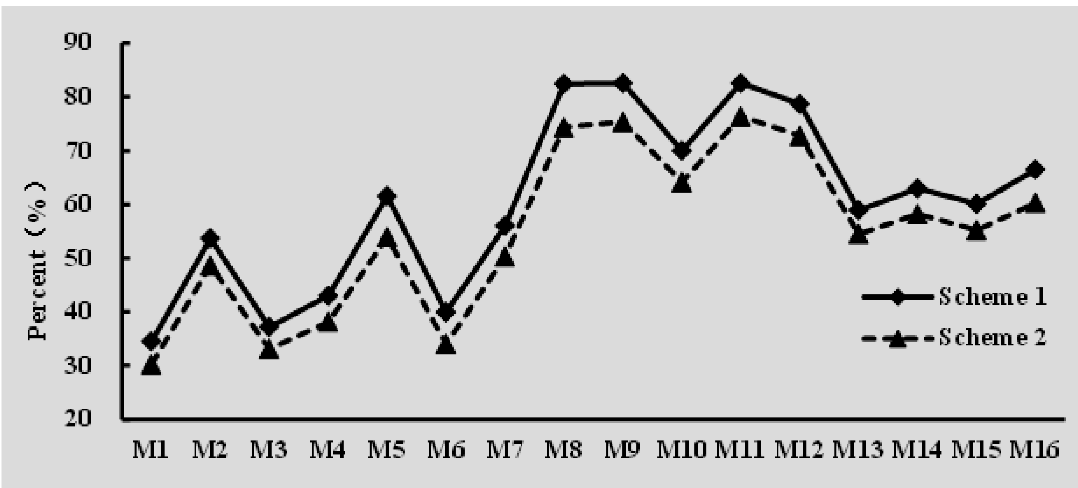
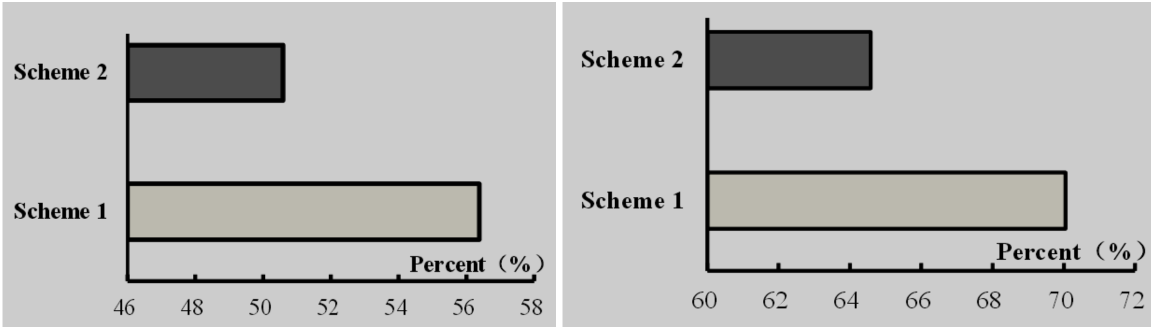


Figure 6. Comparison result of machine and operator utilization ratio



(a) The average machine utilization ratio (b) The average operator utilization ratio

Figure 7. The average utilization ratio compares with scheme 1 and scheme 2

Because production time is same in two schemes, therefore, the ratio of use trend is same in scheme 1 and scheme 2. By comparison, the parallel collaborative scheduling model by considering change over time and debugging time can reduce the MS cycle time, and improve the machine and operator utilization ratio. The practical and operative decisions could be made for the scheduling management of the company.

V. CONCLUSIONS

The optimal parallel collaborative production is crucial in order to reduce the manufacturing system makespan. This paper revealed an integrated parallel collaborative scheduling method of fabrication and assembly multi-cellular manufacturing to improve the production efficiency of the workshop. By considering the transport time, change over time, and debugging time; a parallel collaborative scheduling model of multi-CMs is proposed as solved by parallel and collaborative genetic algorithm (GA). According to the actual characteristics of the production, a two-layer hierarchical scheme is adopted based on two sections for chromosomes encoding. At last, the experiment is designed to compare the scheduling results with other GA scheduling model in a complex electromechanical products workshop, which has 3 CMs. The results indicate that parallel collaborative scheduling by considering the transport time, debugging time and changeover time have a significant effect on the production, it could reduce the manufacturing makespan as well as could enhance the machine and operator utilization ratio.

Compared with existing references, the proposed method is developed for parallel collaborative scheduling and has the following highlights: (1) multi-CMs regarded as a system. In this way, all of the CMs considered as an integrated MS which carry out parallel collaborative scheduling. (2) Proposed a parallel collaborative scheduling model by considering the change over time, debugging time and transport time, these times have a relationship with parts family produced the sequence. Through this way, we could achieve the single minute exchange of die (SMED), and quick response production system.

Although, the case study shows that proposed method of parallel collaborative scheduling could improve the MS performance, there are some limitations of the approach. Perhaps, in the current parallel collaborative scheduling model, we neglect the workers' cooperation that can reduce the operation time when they work in the workstation. Moreover, in our study, the model considers just cycle time of production. In future transport cost, operation cost, and changeover cost might be considered with the operator's skill.

ACKNOWLEDGMENTS

This research is supported by the National Natural Science Foundation of China under Grant No. 70971146 and No. 71231001.

REFERENCES

[1] C. R. Shiyas and V. Madhusudanan Pillai, "A mathematical programming model for manufacturing cell formation to develop multiple configurations," Journal of Manufacturing Systems, vol. 33, pp. 149-158,2014.

[2] F. K. M. Mohammadi and V. Ghezavati, "Integrated cell formation and layout problem considering multi-row machine arrangement and continuous cell layout with aisle distance," The International Journal of Advanced Manufacturing Technology, vol. 78, pp. 687-705, 2015.

[3] A. Aydin, et al, "A multi-objective mathematical model for cellular manufacturing systems design with probabilistic demand and machine reliability analysis," The International Journal of Advanced Manufacturing Technology, vol. 75, pp. 755-770, 2014.

[4] G. A. Süer, et al, "Minimizing total tardiness subject to manpower restriction in labor-intensive manufacturing cells," Mathematical and Computer Modelling, vol. 57, pp. 741-753, 2013.

[5] O. Durán, N. Rodriguez and Luiz Airton Consalter, "Collaborative particle swarm optimization with a data mining technique for manufacturing cell design," Expert Systems with Applications, vol. 37, pp. 1563-1567, 2010.

[6] Mohammadi, Mohammad, and Kamran Forghani, "A novel approach for considering layout problem in cellular manufacturing systems with alternative processing routings and subcontracting approach," Applied Mathematical Modelling, vol. 38, pp. 3624-3640, 2014.

[7] M. B. Aryanezhad, J. Aliabadi and R. Tavakkoli-Moghaddam, "A new approach for cell formation and scheduling with assembly operations and product structure," International Journal of Industrial Engineering Computations, vol. 2, pp. 533-546, 2011.

[8] I. Mahdavi, A. Aalaei, M. M Paydar, et al, "Multi-objective cell formation and production planning in dynamic virtual cellular manufacturing systems," International Journal of Production Research, vol. 49, pp. 6517-6537, 2011.

[9] A. Mojtaba and J. Rezaeian, "Resource-constrained unrelated parallel machine scheduling problem with sequence dependent setup times, precedence constraints and machine eligibility restrictions," Computers & Industrial Engineering, vol. 98, pp. 40-52, 2016.

[10] Y. Li, X. Li and J. N. D. Gupta, "Solving the multi-objective flowline manufacturing cell scheduling problem by hybrid harmony search." Expert Systems with Applications, vol. 42, pp. 1409-1417, 2015.

[11] A.J. Liu, Y. Yang, Q. Xing, et al, "Dynamic scheduling on multi-objective flexible Job Shop," Computer Integrated Manufacturing Systems, vol. 17, pp. 2629-2637, 2011.

[12] M. Pajoutan, A. Golmohammadi and M. Seifbarghy, "CMS scheduling problem considering material handling and routing flexibility". The International Journal of Advanced Manufacturing Technology, vol. 72, pp. 881-893, 2014.

[13] B. Naderi and A. Azab, "Modeling and scheduling a flexible manufacturing cell with parallel processing capability," CIRP Journal of Manufacturing Science and Technology, vol. 11, pp. 18-27,2015.

[14] H. Na and J. Park, "Multi-level job scheduling in a flexible job shop environment," International Journal of Production Research. vol. 52, pp. 3877-3887, 2014.

[15] X. Y. Yu, S. D. Sun and W. Chu, "Parallel collaborative evolutionary genetic algorithm for multi-workshop planning and scheduling problems," Computer Integrated Manufacturing Systems. vol. 14, pp. 991-1000,2008.

[16] F.L.Xiong and H.S. Yan, "Integrated production planning and scheduling of multi-stage workshop ased on alternant iterative genetic algorithm," Journal of Southeast University (Natural Science Edition), vol. 42, pp. 183-187,2012.

[17] Y.J. Liang, M.S. Yang, X.Q. Gao, et al, "Multi-objective optimizing model for solving mixed model shop of fabrication and assembly," Computer Engineering and Applications, vol. 52, pp. 247-253,2016.

[18] K. Halat and R. Bashirzadeh, "Concurrent scheduling of manufacturing cells considering sequence-dependent family setup times and intercellular transportation times," The International Journal of Advanced Manufacturing Technology, vol. 77, pp. 1907-1915, 2015.

[19] A. Elmi, M. Solimanpur, S. Topaloglu, et al, "A simulated annealing algorithm for the job shop cell scheduling problem with intercellular moves and reentrant parts," Computers & industrial engineering, vol. 61, pp. 171-178, 2011.

[20] N. Salmasi, R. Logendran and M. R. Skandari, "Total flow time minimization in a flowshop sequence-dependent group scheduling problem," Computers & Operations Research, vol. 37, pp. 199-212, 2010.

[21] F. S. Al-Anzi and A. Allahverdi, "A self-adaptive differential evolution heuristic for two-stage assembly scheduling problem to minimize maximum lateness with setup times," European Journal of Operational Research, vol. 182, pp. 80-94, 2007.

[22] C. S.Sung and H. A. Kim, "A two-stage multiple-machine assembly scheduling problem for minimizing sum of completion times," International Journal of Production Economics, vol. 113, pp. 1038-1048, 2008.

[23] A. Allahverdi and F. S. Al-Anzi, "Al-Anzi. The two-stage assembly scheduling problem to minimize total completion time with setup times," Computers & Operations Research, vol. 36, pp. 2740-2747, 2009.

[24] C. J. Liao, C. H. Lee and H. C. Lee, "An efficient heuristic for a two-stage assembly scheduling problem with batch setup times to minimize makespan," Computers & Industrial Engineering, vol. 88, pp. 317-325, 2015.

[25] J. Xu, R and Nagi, "Solving assembly scheduling problems with tree-structure precedence constraints: a Lagrangian relaxation approach," IEEE Transactions on Automation Science and Engineering, vol. 10, pp. 757-771, 2013.