Batch splitting activity in scheduling of virtual cell with capacity constraints based on bi-level mathematical model

Wenmin Han¹, a, Juan Chen¹,b and Xiangzun Bu¹,c

¹School of Economics and Management, Jiangsu University of science and Technology, 2 Mengxi Road, Zhenjiang, Jiangsu, 212003, China

a wlmh63@163.com, b chenjuan131988@163.com, c bill2009start@126.com

Keywords: theory of Constraints (TOC), scheduling of VMC, Batch splitting activity, bi-level mathematical model

Abstract. Although there has been some researches about virtual cell manufacturing system, the existing literature lack of discussion about the scheduling model that considering with bottleneck machines in the virtual cells. In view of this deficiency and the new characteristics of the batch splitting problem, this paper considered the batch splitting (or lot splitting) problem in scheduling of virtual manufacturing cells with bottlenecks and multiple machine types, and each of which has several identical machines. In consideration of the hierarchical decision structure of the problem, we developed a bi-level multi-objective mathematical model. Scheduling results and batch splitting strategies of both bottleneck and non-bottleneck machines are given in separate decision levels and additional scheduling objectives are improved in the second model level, while maintaining the maximum use of the bottleneck machine ability. In order to demonstrate how this approach works, application example was shown in this paper.

Introduction

In order to obtain the benefits of virtual manufacturing cells system (VMCS), one of the most important problems is the implementation of VCMS scheduling system. In view of the defect that considers batch and scheduling problem separately, virtual cell scheduling process must take batch problems into account. In a virtual manufacturing cell, batch splitting is the key element that must be considered in management activities. Previous researches pay more attention to the batch splitting problem that caused by bottleneck machine and move pattern of the processing objects, but ignore a significant aspect. As we all know, in the single production system with features of large-scale and complex, batch splitting can't be ignored when discussing virtual cell scheduling because of the widely existing phenomenon of “Master many skills while specializing in one” of machine and identical machines in these enterprises. In addition, in the production system with bottleneck resources, isolated consider bottleneck position and batch will lead to deviation of the production operation plan in the process of implementing. Therefore, in view of the characteristics of the virtual cell batch splitting problem, this article study the batch splitting activity in scheduling of virtual manufacturing cells with capacity constraints. This problem has a hierarchical decision structure, so this paper attempts to decompose and simplify the problem base on the existing researches, and propose a bi-level multi-objective mathematical model to solve the scheduling strategy.

Batch splitting or lot splitting is a concept in which a large production lot is split into smaller sublots so that its operations at successive stations can be overlapped and its progress accelerated. Fig 1 is the schematic diagram of the lot splitting activity caused by identical machines. Batch splitting method is first introduced by Reiter[2], later, many scholars give further research to this method.

Mohamad[3], Defersha[4], Pan[5], Pan[6] et al discuss the lot splitting in flow-line production system. Chinyao[7], Rahime[8], Taofeng Ye[9] et al investigate the lot splitting in a jobs hop production system. Saricicek[10]focuses on the problem of scheduling n independent jobs to be processed on m identical parallel machines with the aim of minimizing the total tardiness of the jobs considering a job splitting property. As the lot splitting concept has been used widely to improve the
makespan in production system, most researches have investigated the flow shop or job shop production systems; however, cell especially virtual cell manufacturing systems have received much less attention, relatively. Suer [11], Lockwood [12] and Defersha [13] et al study the design problem of cell manufacturing systems with lot splitting. Ahkioon et al. [14] formulated the integrated approach to CMS design as the non-linear mixed-integer programming model incorporating production planning and system reconfiguration decisions with the presence of alternate process routings, operation sequence, duplicate machines, machine capacity and lot splitting. Saadettin [15] developed two mixed integer linear programming (MILP) formulations with and without batch splitting to research the virtual cell scheduling problem. Kia [16] solving a group layout design model of a dynamic cellular manufacturing system with alternative process routings, lot splitting and flexible reconfiguration by simulated annealing.

From the above analysis of literature, we know that, in different manufacturing system, the causes of batch splitting is different, such as the batch splitting in parallel production line, under different delivery date, and those caused by system bottlenecks. But in the virtual cell production organization mode, there are multiple machine types, each of which has several identical machines, so each operation can have not only one choice, which is to say, when discussing virtual cell scheduling problem, the batch splitting problem must be taken into consideration. But most of the existing researches about virtual cell scheduling problem consider the production batch as known parameters, and there are few study of the batch splitting problem caused by bottleneck machine and identical machines, because the split of the batch greatly increased the scheduling difficulty. However, in practical production, scheduling problem with the feature of lot splitting is common. In addition, without considering the resources capacity constraints of the bottleneck and the batch splitting caused by the bottleneck machine, system load balance and the production efficiency will be affected greatly, so researches must consider using bottleneck management to optimize the virtual manufacturing cell scheduling process.

**Problem Description**

The motivation to propose bi-level mathematical model is three fold. Firstly, in order to achieve the steps in the TOC procedure, two machine types must be managed, bottleneck and non-bottleneck machines, in separate decision levels. Slack the capacity constraints of the bottleneck machining, scheduling the hardest bottleneck machines, then arrange the non-bottleneck machines, to gain the feasible scheduling scheme. Secondly, batch strategies in bottleneck and non-bottleneck machines are different, and this should be solved by a hierarchical decision model. For bottleneck machines, processing batch size should as big as possible, so as to improve the utilization, and increase output. For non-bottleneck machines, processing batch size should as small as possible, in order to shorten the lead time, and reduce the work in process. It also leads to the different model objectives of the two model levels. Thirdly, although using single model to describe the comprehensive integration problems of multi-level production plan can quite comprehensively consider all sorts of constraint, the complexity of the model is very high. If integrate the batch plan and operation sequence together, the solution difficulty will be higher. Therefore this paper discusses the batch splitting activity in
scheduling of virtual manufacturing cells with capacity constraints using bi-level multi-objective mathematical model based on the concept of multistage decomposition and coordination model. From the above causes, it is not difficult to discover that the first level of the model aims at minimizing idle time on the bottleneck machine so as to generate the initial schedule scheme. Because the makespan and travelling distance performances is particularly important in virtual cell scheduling, therefore, by applying the batch splitting technique, the second level model minimize the sum of the weighted makespan and weighted total travelling distance, while maintaining the bottleneck sequence obtained from the first level decision. In addition, there are other goals, and we can discretely take them as the model objectives according to the situation. By using the concept of transfer lot, the waiting time on each machine is reduced because of the existence of the overlapped operations. Each job type can visit different machine types so operation sequences are vary. Every machine type consists of more than one identical machine. So for each operation of any job type, there exists not only one alternative. Since there are several routes for each machine type, all parts in any job type want to take the nearest total travelling distance route, but this will deteriorate the makespan performance. So, when consider batch splitting, it is sensible to say that there is a conflict between two different objectives. Batch splitting also increases the total number of operations since one operation of any job may be divided into more than one. So the second level gives the complete scheduling solution, ensures the maximize use of the bottleneck machines and the whole system through time, it must decide the following things: which individual machine is selected in machine type, how many parts in any job are processed on any specific machine, and when the operations should start.

Model Building

The Proposed Mathematical Method. With the hierarchical decision structures, many planning and scheduling problems can be modeled using a multi-level programming model. Different levels are interdependent and there exist conflict objectives. Bi-level programming model has two decision levels, and they are relying on each other. The general formulation of the bi-level programming model is given as Eq.1 (Colson[17]).

This text develops a bi-level programming model for the virtual cell scheduling problem with the feature of batch splitting. On the first level, the objective is to minimize the idle time of the bottleneck machines by using the concept of transfer lot to allow overlapped operations. The result of this step is to find the processing order that has the minimizing idle time of the bottleneck machine. The lower level is similar with the upper level model, the changes are as follows:

1) Additional restrictions on the bottleneck to guarantee the processing order;
2) Consider different objective functions.

\[
\begin{align*}
\text{Min: } & F(X, Y) \\
\text{s.t. } & G(X, Y) \leq 0 \\
\text{Min } & f(X, Y) \\
\text{s.t. } & g(X, Y) \leq 0
\end{align*}
\]

Model Assumption. Model assumptions are as follow: (1) the bottleneck machines are known, (2) all the parameters are certain and known, (3) the virtual cell formation has been finished. (4) Transfer lot size of each job is constant, (5) jobs to be scheduled are ready at time zero and no new part arrivals are accepted until the completion of the scheduling, (6) no re-entrant job, (7) each device can only process a task in any time, (8) no interrupt is allowed when the task start.
Model Formulation: Bi-level Mathematical Model for VC Scheduling.

Indices:
- \( j, u \) job number \((j, u = 1, \ldots, n)\)
- \( i, k, o \) machine type \((i, k, o = 1, \ldots, m)\)
- \( s(i) \) individual machine \(s\) belonging to machine type \(i\)
- \( b \) bottleneck machine \((b=1, 2, \ldots, B)\)

Parameters:
- \( t_j \) transfer lot size of job \(j\)
- \( N_j \) batch size for job \(j\)
- \( P_{ji} \) unit processing time of job \(j\) on machine type \(i\)
- \( D_{s(i)s(k)} \) unit transportation distance for each job to be carried from machine \(s(i)\) to \(s(k)\)
- \( M \) sufficiently large number
- \( W_1 \) weight of the makespan \(C_{\text{max}}\)
- \( W_2 \) weight of the total travelling distance

Decision variables:
- \( C_{\text{max}} \) completion time of the last job to leave the system (makespan)
- \( t_{j,s(i)s(k)} \) starting time of job \(j\) on individual machine \(s(k)\) such that job \(j\) is sent from individual machine \(s(i)\) to \(s(k)\)
- \( V_{j,s(i)s(k)} \) number of parts in job \(j\) sent from \(s(i)\) to \(s(k)\)
- \( q_{s(i)s(k)} \) 1 if parts for job \(j\) coming from machine \(s(i)\) are processed before those of machine \(s(i)\) on machine \(s(k)\), 0 otherwise
- \( r_{s(k)s(j),u} \) 1 if job \(j\) comes before job \(u\) on machine \(s(k)\), 0 otherwise

The first level mode:

\[
\min \text{imise} \sum_{b=1}^{B} \left[ \max_{j, b} \left\{ t_{j,s(i)b} + N_j P_{jb} \right\} - \sum_{j=1}^{n} N_j P_{jb} \right] 
\]

Subject to:

\[
t_{j,s(i)s(k)} + t_j P_{ji} \leq t_{j,s(i)s(k)} 
\]

\[
t_{j,s(i)s(k)} + N_j P_{jk} \leq t_{j,s(i)s(k)} + M(1-r_{s(i)s(k)}) 
\]

\[
t_{u,s(i)s(k)} + N_j P_{jk} \leq t_{j,s(i)s(k)} + M(r_{s(k)s(i)}) 
\]

\[
I_{j,s(i)s(k)} = P_{jk} \left\{ \begin{array}{ll} 0, & P_{ji} \leq P_{jk} \\ (P_{jk} - P_{ji}) N_j & P_{ji} > P_{jk} \end{array} \right. 
\]

\[
t_{j,s(i)s(k)} \geq t_{j,s(i)s(i)} + P_{ji} t_j + I_{j,s(i)s(k)} 
\]

\[
l_{j,s(i)s(k)} \geq 0 
\]

\[
r_{s(k)s(j),u} \in \{0,1\} 
\]

The second level model:

\[
\min \text{imise} f = W_1 C_{\text{max}} + W_2 \sum_{j} \sum_{s(i), s(k)} D_{s(i)s(k)} V_{j,s(i)s(k)} 
\]

In order to ensure the processing orders on the bottleneck machine, the second level model should first meet constraint (10) and (11).

\[
t_{j,s(i)s(k)} \leq t_{j,s(i)b} \quad \text{when} \quad s(k)=b 
\]

\[
r_{s(k)s(j),u} \leq r_{s(j)s,u} \quad \text{when} \quad s(k)=b 
\]

\[
t_{j,s(i)s(k)} - t_{j,s(o)s(i)} \geq V_{j,s(i)s(k)} P_{ji}, \forall o, i, j: (j,o) \rightarrow (j,i) \rightarrow (j,k) 
\]
The first level objective, Eq. 2 is to minimize the idle time of the bottleneck machines. That is the completion time of the bottleneck minus the processing time of the bottleneck machine, and the goal is to maximize the use of the bottleneck machine ability. Eq. 3 precedence constraints, Eq. 4 machine conflict constraints. Eq. 5 is the sum of the waiting time between different transfer lots, and when the processing time of the preceding process is less than the succeeding process, it is zero; otherwise, the calculation method of it is shown in Fig. 2 below. Eq. 6 is the earliest starting time constraint when using transfer lot. Non-negativity constraints and binary constraints are shown in Eq. 7 and Eq. 8, respectively.

The second level objective, Eq. 9 is to minimize the total weighted completion time and weighted transportation distance. Eq. 10 and Eq. 11 ensure the processing order on the bottleneck machine that obtained in the first level. Eq. 12 indicates that jobs in the prior stage must be finished before the start of the operation in the next stage. Eq. 13 is either-or constraint; Parts coming from the previous machine must be processed in the free time of the next machine. Eq. 14 requires that the makespan is equal or great than the completion time of all the work in the system. Eq. 15 is either-or constraint too; at any time there always exist one constraint that is established, so as to ensure that different works can be processed at different time. Eq. 16 requires that the sum of jobs processed in the previous machine types of any job must be equal to the batch size. Eq. 17 guarantees the number of jobs coming to the machine must be equal to the number of jobs processing on the machine. Eq. 18, Eq. 19 and Eq. 20 are non-negativity constraints. Eq. 21 and Eq. 22 are binary constraints.

![Fig.2 The calculation method of $I_{j,s(i),s(k)}$](image-url)
Model Application

Illustrative Example. To test the validity of the model, the outfitting work of a shipyard in Mainland China is taken for case study. Considering the data confidentiality, the paper briefly expresses the machine and process names. Weight determination is a Multi-objective decision problem, which can be calculated according to experience or the existing method of weight determination. In this article, we do not talk about it. From the viewpoint of concise, the weight $W_1$ of the makespan $C_{max}$ is provisional for 0.95, and the weight $W_2$ of the total travelling distance is Provisional for 0.05. There are four types of machines in the virtual cell, and the specific situation is shown in Table 1, the first column listed as machine types, the second column listed as the number of the individual machine belonging to this machine type. There are six jobs in the virtual cell, and the processing sequence, processing time, batch size are shown in Table 2. Transfer lot size of each job is set to one. Table 3 shows the travelling distance between each pair of machines which is equal in both directions. Machine 4(1) is the bottleneck machine.

Computational Results. This article uses a General Algebraic Modeling System (senior modeling system for solving mathematical programming) to calculate the proposed model. The Gantt chart of the first level’s scheduling result is shown in Fig. 3, idle time of the bottleneck machine is 30; job sequence on the bottleneck machine is J3-J5-J2; the $C_{max}$ equals to 360; the total travelling distance is 3106. In order to guarantee the maximum use of the bottleneck machine ability in the second level model, the starting time constraints of jobs on the bottleneck machine are $t_{35(1)} \leq 30$, $t_{35(1)} \leq 78$, $t_{2(1)} + t_{4(1)} \leq 146$, and the job sequences constraints are $r_{4(1)35} = 1$, $r_{4(1)32} = 0$, $r_{4(1)35} = 0$, $r_{4(1)32} = 1$, $r_{4(1)32} = 0$, $r_{4(1)25} = 0$. Idle time of the bottleneck machine is still 30, but the value of $C_{max}$ is 272; the total travelling distance equals to 3756. The maximum completion time of the second level model is reduced by 88. The Gantt chart of the second level’s scheduling result is shown in Fig. 4. From Fig. 4, we can clearly see that after batch splitting, the utilization rate of the machines improved significantly. The specific process number of each job and the new process route after batch splitting are shown in Fig. 5.

<table>
<thead>
<tr>
<th>Table 1 Machine Information</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>machine type</strong></td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>2</td>
</tr>
<tr>
<td>3</td>
</tr>
<tr>
<td>4</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 2 Job Information</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Job number</strong></td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>2</td>
</tr>
<tr>
<td>3</td>
</tr>
<tr>
<td>4</td>
</tr>
<tr>
<td>5</td>
</tr>
<tr>
<td>6</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 3 Travelling Distance Matrix</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Machine</strong></td>
</tr>
<tr>
<td>1(1)</td>
</tr>
<tr>
<td>1(2)</td>
</tr>
<tr>
<td>1(3)</td>
</tr>
<tr>
<td>2(1)</td>
</tr>
<tr>
<td>2(2)</td>
</tr>
<tr>
<td>2(3)</td>
</tr>
<tr>
<td>3(1)</td>
</tr>
<tr>
<td>3(2)</td>
</tr>
<tr>
<td>4(1)</td>
</tr>
</tbody>
</table>
Conclusions

This paper mainly discusses bottleneck management and batch splitting problem in virtual cell scheduling system. We consider the situation that there are multiple jobs with different processing sequences and there are various machine types, each of which has more than one individual machine and there are bottleneck resources in the virtual cell. Based on the concept of TOC, we develop a bi-level mathematical programming model, so this article compensates for the lack of the theory that the existing literature lack of discussion about the scheduling model that considering with bottleneck machine and batch splitting problem in the VCMS.

The results of the case study show that the proposed bi-level mathematical model can give the scheduling strategies of the bottleneck and non-bottleneck machines, in separate decision levels. In addition, it also shows that in virtual cell manufacturing system, discussion of batch splitting problem is very necessary. Because that in consideration of the system bottlenecks, model with batch splitting can obtain more satisfactory results, and scheduling solution that given by this method is more actual.

As batch splitting makes the scheduling activities more complicated because the total number of operations that must be considered increases, some metaheuristic approaches such as genetic algorithm, ant colony optimization, tabu search, etc. can be applied to the problem, to obtain near optimal results in a shorter computational time or provide solutions for both decision levels simultaneously.
Acknowledgements

This research was supported by the National Natural Science Foundation of China (71271105) and the Ministry of Education Humanities Social Sciences Planning Foundation of China (12YJA630036).

References


