The Method of Optimization for Control of Flexible Manufacturing Systems

ADRIANA FOTA, SORIN ADRIAN BARABAS
Department of Economical Engineering and Production Systems
Transilvania University of Brasov
Bd. Eroilor no 29, 500036 Brasov
ROMANIA
fota.a@yahoo.com, ab.sorin@gmail.com, http://www.unitbv.ro

Abstract: - The computerized nature of the flexible manufacturing systems (FMS) makes it readily adaptive to the web-based information system. However, processes run by the FMS may not be fully automatic because of potential resource conflict, i.e. a floating characteristic relationship between system facility and production orders. Since coordination between system facility and unpredictable orders is difficult, this paper will present an off-line simulation approach to reveal the embedded relationship and then avoid the conflicts on-line. The method employs three dispatching rules individually to direct the process flows inside a flexible manufacturing cell, and acquires potential deadlock patterns of part processing sequence from an off-line simulation. Then an on-line matching/reordering process is used to keep the incoming orders dissimilar to the deadlock patterns. Two major advantages have been achieved by the proposed method: it provides an effective routing mechanism for deadlock-free production on randomly arrived orders, and it improves the feasibility of any planned schedules by removing the potential of resource deadlock. This research uses timed Petri nets to simulate the flexible manufacturing cell. Three dispatching rules, which generate pull tendency at cell exit, are employed and compared to demonstrate the routing mechanism.

Key-Words: - flexible manufacturing systems, modelling, control.

1 Introduction
The time for electronically connecting business and manufacturing has come. Advancements in e-commerce have requested that business decisions on web be the direct driving info to the corresponding manufacturing activities without human interventions. Since orders from the web may arrive in random sequence and with a range of specifications, flexible manufacturing systems (FMS) may be the proper automata for the basis of e-manufacturing. An FMS is designed to handle orders having moderate variations in both part style and quantity.

In the course of manufacture automation, many efforts have been made to develop different aspects and levels of mechanism both in hardware and software. Intrinsically, flexible manufacturing possesses the most sophisticated control given the subtle requirements it is designated. In an intelligent flexible manufacturing system (FMS), the ultimate goal is to conduct humanly performance on coordinating hardware diversity without human intervention.

A smooth process demands not only on the correctness of individual operations, but also on overall shop floor fluency. Most research of FMS had focused on hardware configurations [5], schedule efficiency [1, 3], and system flexibility, and most of them assumed a perfect compliance among system elements. However, a potential conflict of resource deadlock may hinder the focus from a smooth application. Resource deadlocks can be caused by many factors inside the system, mostly malfunction and/or disharmony. Since facility maintenance is relatively more manageable, coordination management is of the interest in this paper.

An FMS is designated to handle a variety of part configurations in any sequence; therefore, the resource disharmony, or traffic bottlenecks, among the machine workstations are almost unpredictable. Nevertheless, something about the bottleneck is for sure: it is inherent within the workstations and parts, it is floating depending on the in-process parts and their processing sequence and it is implicit. The purpose of this research is to expose the potential deadlocks and avoid them with an on-line re-routing mechanism. A flexible manufacturing cell (FMC) is a subsystem of FMS, designed to process parts that require similar machine tools. The physical arrangement of the facility in an FMC is planned based on the characteristics of the machine tools as well as the parts using Group Technology classification, and thus will not change lightly once defined [2]. Parts are assigned to FMCs according to their process or machine requirements, regardless of the processing sequence. Therefore, it is
possible that some parts have fixed processing sequence, or single routing, while others may allow alternative routings. In addition, certain parts may block the path of one another during the process because of the competition on the manufacturing resources. In general, orders for different part type may arrive randomly; so it is difficult to schedule smooth production in ahead to maximize overall benefit [6], and thus making the dispatching of manufacturing resources complicated. In which case, two potential problems ought to be resolved before a fully automatic flow control can be achieved. One is the deadlock of process flow due to unpredictable part arrival and limited shop floor resource [4]; another is the dispatching of machines [2, 6].

Petri nets have been used extensively on the simulation of all sorts of discrete events/processes environments, especially on the FMS, because the encapsulated structure of the nets can hierarchically model a manufacturing system in all levels of details [2, 3]. From a Petri net’s point of view, deadlocks occur when no transition can be fired from certain critical markings, or the firing of transitions has run into an endless loop. The first case indicates all manufacturing resources are held in static, and the second implies that no manufacturing process can advance in a loop. Analytically, critical markings and endless loops (if exist) of a Petri net can be spot by searching the reach ability tree of the net [3]. However, the search is somewhat impractical because not all states in the tree are meaningful to the manufacturing process; only a fraction of the tree is visited along the process. Besides, searching a tree can be computationally expensive when the tree represents a complex reality. On the other hand, the patterns of the in-process parts that lead to deadlock would be a much smaller subset compare to the states on the tree, and are more useful in determining inappropriate intake orders. Some methods have employed infinite or large number of buffers in the system to absorb part overflow caused by waiting. The loose definition of buffer size may be effective but is impractical in limited space.

In the area of FMS scheduling, the objective was on the overall efficiency of time or capital given a range of system configurations. Similar research on rescheduling was aiming at fitting in unexpected orders. However, the scheduled or rescheduled processes may still be blocked potentially by the competitions on resources [4]. On the other hand, advancements in e-commerce have requested that business decisions on web be the direct driving info to the corresponding manufacturing activities.

Process flaw of resource deadlock could be a fatal problem for such purpose. Therefore, a mechanism that automatically avoids resource deadlock is quite essential every aspect. The following procedure will develop a routing scheme to evade the blocks, and make the routing control man less.

2 Methodology

Since facility setup is fixed in an FMC and the newly assigned parts could prone to resource deadlock, screening the assignments and rule out potentially inappropriate parts at cell entrance may be a practical tactic to prevent the conflicts. This method will keep the net markings dissimilar to the critical markings.

The rejected parts will be re-evaluated once the states of the FMC have proceeded to make sure that all orders are handled. Figure 1 outlines the course of the method. The previous approach has superficially classified the incoming orders into two types: process it now and process it later. Indeed, the critical markings did represent a characteristic relationship between the cell machines and the parts. However, given an FMC and parts, there are more independent factors that may influence the relationship, such as processing times, resource dispatching rules, numbers of buffers, types and time interval of the arrivals, machine breakdowns, and routing alternatives. Small variations on any factors might tremendously increase the complexity of the problem, not to mention the combinatorial explosion. Nevertheless, some practical assumptions will confine the problem and make it manageable.

Orders arrive in a random fashion. All equipments in the cell are maintained in a regular basis, so there is no machine or tool breakdown. Each FMC is equipped with exactly one robot for in-cell transportation, and every loading and unloading position in the cell is accessible by the robot.

All machines in the FMC are CNC tools so the processing times are fixed. All part routings are known to the FMC and are pre-defined. After the assumptions, only two controllable factors remained to manipulate the deadlocked resources: the sequence of the line-up orders, and the rules of resource dispatching. Details of the method are described in the following with an example.
3 Entrance control
In this section, a Petri net that emulates the status of all machines inside the FMC is built, and then a specimen of line-up parts are fetched to the net for testing. If deadlock occurs during the test, then record the specimen. The test is to repeat on all possible line-up combinations to determine the sequences that cause deadlock.

The deadlocked specimens are then refined to subset patterns that are later used to screen the intake orders on-line. The whole procedure is elaborated below.

3.1 Generate Petri net of the FMC
The cell net is a synthetic of individual part nets, and the part nets are created by connecting the processing machines in a proper sequence.

Figure 2 shows a sample part net, and Table 1 defines the net. The processing times can be set into the net transitions to simulate the real process. For a more realistic simulation on processing times, stochastic Petri net can be used to model a system.

Part nets are synthesized at the identical resources (places) to form the cell net [2].

![Fig. 2. Sample part net](image)

<table>
<thead>
<tr>
<th>Definition of places</th>
<th>Definition of transition</th>
</tr>
</thead>
<tbody>
<tr>
<td>p1</td>
<td>part-order arrived</td>
</tr>
<tr>
<td>p2</td>
<td>M1 ready</td>
</tr>
<tr>
<td>p3</td>
<td>M1 done</td>
</tr>
<tr>
<td>p4</td>
<td>part in buffer</td>
</tr>
<tr>
<td>p5</td>
<td>M2 ready</td>
</tr>
<tr>
<td>p6</td>
<td>M2 done</td>
</tr>
<tr>
<td>p7</td>
<td>part left cell</td>
</tr>
<tr>
<td>p8</td>
<td>robot available</td>
</tr>
<tr>
<td>p9</td>
<td>M1 available</td>
</tr>
<tr>
<td>p10</td>
<td>buffer available</td>
</tr>
<tr>
<td>p11</td>
<td>M2 available</td>
</tr>
</tbody>
</table>

The deadlocked specimens are then refined to subset patterns that are later used to screen the intake orders on-line. The whole procedure is elaborated below.

3.2 Generate specimens
A specimen is a set of orderly parts that is to fetch to the cell net for resource coordination test, i.e. the deadlock test. The length of a specimen is set to the smallest amount that will effectively fill the cell facility. Assuming that each machine can hold at most one part at a time, then the amount is the sum of machines plus buffers plus 1, to ensure fullness.

For a cell with $m$ positions and $n$ part types, the length of a specimen is $(m+1)$, and the total number of specimens can be as high as: $(n+1)^{m+1}$, where the $(n+1)$th part type is a dummy part called “Bubble”. It simply introduces time delay without taking any equipment. The bubble is used to simulate vacancy between arrivals. Therefore, in order to simulate all possible line-up sequences, a specimen $S_i$ can be defined as follow:

$$S_i: \{P_j\}$$

where $P_j$ is a part type of the cell, $P_j = 0, 1, 2, \ldots, n$, “0” is the bubble,
By default, the elapsed time of a bubble is \( t \), which is set equal to the longest processing time of the cell parts, and can be adjusted for further testing. Technically, specimens that have bubble at head or tail will be removed from the test for a head bubble is void and a tail bubble cannot fill the cell. This will provide some release to the computational loads in the following steps.

3.3 Collect deadlock patterns

After all specimens are tested, the ones that caused deadlock are defined as Deadlocked Specimens. The portion of a Deadlock Specimen that has entered the cell when deadlock occurred was recorded and called a Deadlock Pattern \( D_k \). These patterns can be regarded as the implicit characteristic of the FMC.

For the repeated deadlock patterns, only one is kept
to the set of $D_k$. The remaining deadlock patterns are classified into groups according to their lengths.

This procedure (Re-order line-up sequence at the entrance) will evaluate the first part at the line-up to determine whether it is allowed to enter the cell. The end portion parts in the cell plus the first line-up are defined as a test pattern, and the test pattern is to compare with the deadlock patterns of the shortest length. If the test pattern matches any $D_k$ in that group, then there is a potential deadlock and the line-up will be re-ordered. If not matched, then extend the test pattern and move on testing the next length of group. If there is no match to any $D_k$ in all groups, then there is no potential deadlock for the test pattern and the first line-up is allowed to enter the cell. It is noted that part types and their positions must be identical to call a match. Figure 4 demonstrates the pattern matching process.

In order to have the re-ordered part processed as early as possible, the part will move back only one position. If the second line-up matches again, then both parts will move back one position, and so on. If there are $n$ consecutive matches, then the mechanism will insert a bubble at the entrance. If one bubble still causes deadlock, then insert another, and so on, until no matches happen. The number of $n$ can be determine by the product due day, and will not be elaborated in this paper. Evidently, the more insertion of bubbles, the longer overall delay will be expected. This is also part of the characteristic relationship between the cell and parts. The following section will introduce other dispatching rules to improve the relationship. The aim is to cut down total number of deadlock patterns, so that the probability of hitting a match is minimized.

### 3.4 Rules
Since the number of buffers is limited, and there is a push forward tendency at the cell entrance by the arrivals, the principle of dispatching will aim at moving the requested parts that are closest to the cell exit, i.e. job done, first, to create a pull tendency at the exit. Based on the idea, two types of dispatching rules are defined. Upon moving requests, the first rule (Figure 5) will clear all finished parts according to the remaining processing time of their machine successor, and then move the rest according to their remaining processing time when no finished part presents.

The second will create a list for the finished parts and a list for the rest. The finished parts are sorted in the order of their finishing time, and the rest parts are sorted in the order of their remaining processing time. Then the mechanism will alternatively move the first part on the lists starting from the finished list. If there is no part in either list, then there will be no waiting and the moving will shift to the other list. Both rules will be employed and programmed into the FMC’s Petri net for evaluation to compare with the FCFS rule.

### 4. Implementation
The proposed method is tested on an FMC with four parts, five machines, one robot, and no buffer. The process plans of the parts are shown in Table 2, where two are of single routing and two are of multiple routing. Several observations were obtained during the implementation. Some have provided strong supports to the proposal, and rooms for improvement were also revealed.

<table>
<thead>
<tr>
<th>Part type</th>
<th>Process plan</th>
<th>Route type</th>
</tr>
</thead>
</table>

![Fig. 4 Pattern matching](image)

![Fig. 5 Rule I](image)
These issues are discussed below. A routing strategy is more like the traffic lights that guide the traffic inside the cell for a deadlock-free passage. Given the process plans of the parts, the interactions between parts and machines are like traffic and lights. Since routes are pre-defined and the time or speed is confined, the traffics can only be directed by switching lights, i.e. the dispatching rules.

The method proposed in this paper uses off-line simulations to support on-line detections. For cells having more machines and part types, on-line deadlock simulations are recommended at the cell entrance. The critical numbers of part types and machines can be determined by extensive checks on simulation time.

The total number of deadlock patterns in the three experiments is around 30% of the number of the total specimens. It is a large number because no buffer was used in the FMC. Intuitively, introducing more buffers to the cell will help cut down deadlocks. However, the experiments have explicitly shown the part/machine coordination relationship. If more buffers are used, then more waiting process will be inside the cell. If no buffer is used, the waiting will be at the entrance of the cell or has been replaced by bubbles during the re-line-up process. The reason why deadlocks were not tremendously cut down by alternating dispatching rules is that, although providing better flexibility, parts of multiple routing tend to jam the cell because of their shorter waiting times, and multiple-routed parts will compete manufacturing resources with the single routed parts at an equal basis principle which has not been thoroughly investigated. Since more deadlock patterns mean higher possibility of bubble insertions, there is a balance between limiting alternative routes and reducing overall processing time. Since resource dispatching has made easy by coding rules into the corresponding transitions of the Petri net, more variations in dispatching rules can be employed to improve the proposed method.

From the machine’s point of view, blanks between two arrivals were hard to predict. This phenomenon has been effectively emulated in the methodology by using bubbles to fill the blanks. Therefore, each bubble is also tested before fetch to the FMC. The reason why only one part, i.e. the first line-up, is tested at the cell entrance is that dissimilarity from the deadlock patterns can be easily acquired by controlling the first line-up. This method also helps increasing the possibility of test approval. Therefore, each pattern will represent an independent instance. In general, getting match with a shorter pattern has higher possibility than a match with a longer pattern; so shorter patterns should be compared first. As a result, deadlock patterns are classified into groups according to their lengths. Although the process records only the parts that have entered the cell when a deadlock occur, there is no missing in the collection of all potential deadlock patterns, for the discarded parts of the deadlocked specimens will also appear in the front portion of other specimens. Analogously, the ways the parts would have been sequenced in a batch production are subset to the set of all specimens. Therefore, the feasibility for any batch production has also been tested within the method.

5. Conclusions

The purpose of scheduling is to provide proper arrangement of the orders for machines, cells, or systems so that the overall efficiency of production is optimized. The proposed method can be used as an on-line reschedule mechanism that automatically fit in the improper scheduled orders. Therefore, this paper has improved the feasibility of any planned schedule.

The method is also effective in arranging deadlock-free production for random arrived orders, which can be extended to simulate the e-commerce activated or web-based, autonomous productions. Thus, it can be used as a link mechanism between e-commerce and e-manufacturing. In a mixed routing manufacturing cell, multiple-routed parts should conditionally surrender their route priority, or flexibility, to the single-routed parts for the latter have no alternative when conflict. The relationships between part types will be the focus of the ongoing research of this paper.

Acknowledgement: This paper is a result of the research from the scientific Contract CNCSIS - IDEI, code PCE_756 / 2008.

References:

| Part | m1→m2→m3; m1→m3→m2; m2→m1→m3; m2→m3→m1; m3→m1→m2; m3→m2→m1; | Multiple |
| Part B | m2→m3→m5; m2→m5→m3; m5→m3→m2; | Multiple |
| Part C | m2→m4→m5; | Single |
| Part D | m4→m3→m1; | Single |
