International Journal of Applied Science and Engineering
2004. 2, 2: 163-176

On-line Deadlock Avoidance for Complex Routing Flexible Manufacturing Cells

Tzong-Ming Cheng

Department of Industrial Engineering and Management,
Chaoyang University of Technology,
Wufeng, Taichung country 413, Taiwan, R.O.C.

Abstract: 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. The computerized nature of the 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 (FMC), 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.

Keywords: deadlock avoidance; e-Manufacturing; flexible manufacturing systems.

1. Introduction

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 shopfloor fluency. Most research of FMS had focused on hardware configurations [11], schedule efficiency [1, 7, 14, 18, 19, 21], and system flexibility

Corresponding e-mail: rtmc@cyut.edu.tw *Accepted for Publication: Apr. 16, 2004*

[16, 20], 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, 10, 17], 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, 9, 16]; another is the dispatching of machines [14, 18, 20].

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 [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 reachability tree of the net [8, 13]. 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 [5]. 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 from every aspect. The following procedure will develop a routing scheme to evade the blocks, and make the routing control manless.

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.

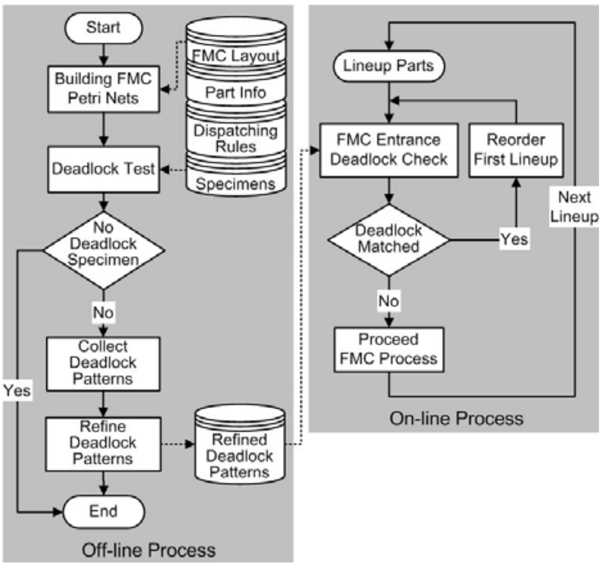

Figure 1. Outline of the method

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. The assumptions are listed below:

1. The amount of buffers in an FMC is not infinite.
2. Orders arrive in a random fashion.
3. All equipments in the cell are maintained in a regular basis, so there is no machine or tool breakdown.
4. 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.
5. All machines in the FMC are CNC tools so the processing times are fixed.
6. 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 lineup 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 lineup 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 lineup 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 [14]. 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.

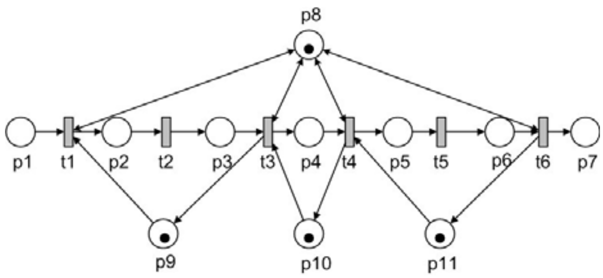

Figure 2. Sample part net

Part nets are synthesized at the identical resources (places) to form the cell net [12, 13].

Figure 3 is a sample of synthetic, where places p8, p9, p10, and p11 are common to both part nets shown on upper and lower portions of the Figure. The transition that has its other inbound tokens ready will have the privilege of the common resource. This particular token distribution process is a First Come First Serve (FCFS) principle. For other dispatching rules, the distribution mechanism should be otherwise programmed.

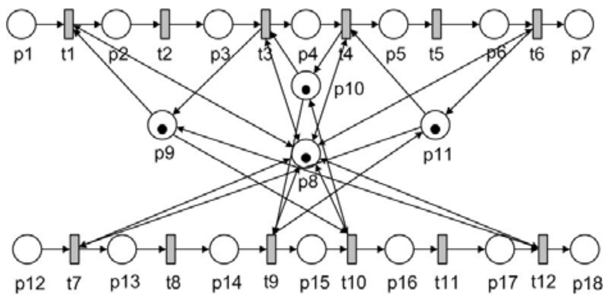

Figure 3. Sample synthetic net

Table 1. Definitions of Figure 2

Definition of places			
p1	part-order arrived	p7	part left cell
p2	M1 ready	p8	robot available
p3	M1 done	p9	M1 available
p4	part in buffer	p10	buffer available
p5	M2 ready	p11	M2 available
p6	M2 done		
Definition of transitions			
t1	load M1	t4	load M2
t2	process M1	t5	process M2
t3	M1 to buffer	t6	unload M2

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 lineup 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, …. n, “0” is the bubble,
$1 \le i \le (n+1)^{(m+1)}$,
j = 1, 2, …, (m+1).

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.

3.4. Refine deadlock patterns

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.

3.5. Re-order lineup sequence at the entrance

This procedure will evaluate the first part at the lineup to determine whether it is allowed to enter the cell. The end portion parts in the cell plus the first lineup 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 lineup 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 lineup 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.

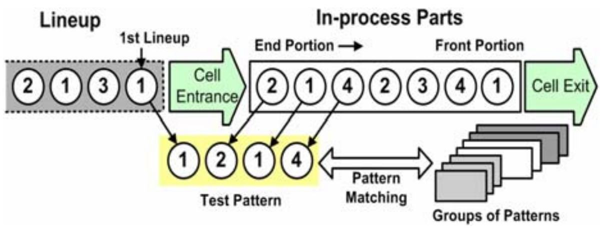

Figure 4. Pattern matching

In order to have the re-ordered part processed as early as possible, the part will move back only one position. If the second lineup 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.

4. Dispatching 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.

4.1. Rule-I

Upon moving requests, the first rule 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 rule is also described in Figure 5.

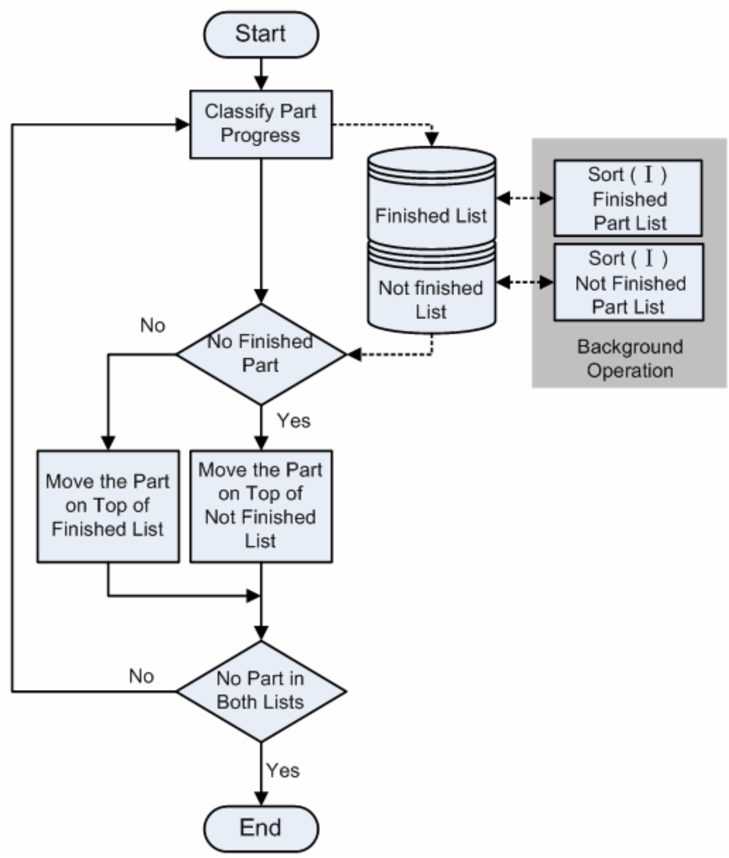

Figure 5. Rule-I

4.2. Rule-II

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. The rule is described in Figure 6. Both rules will be employed and programmed into the FMC's Petri net for evaluation to compare with the FCFS rule.

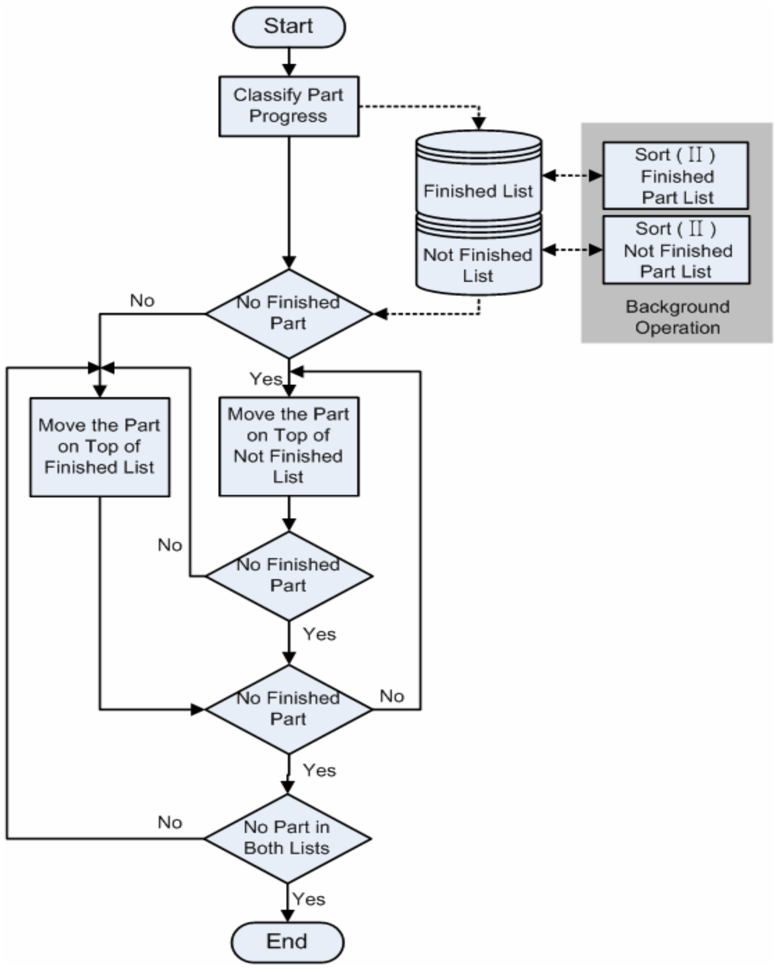

Figure 6. Rule-II

5. 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. The Petri net of the cell is shown in Figure 7, and defined in Table 3. Since there is only one robot in the cell, the dispatching rule will be used to control the robot solely. The place p03 is to collect all tokens for robot releasing. Then the transition t46 will activate an evaluation process according to the dispatching rules to dispatch robot. Since there is only one robot in the cell, all loading and unloading processes will need a token to start. The places p08, p12, p15, p76, and p64 are to collect tokens from machine 1 to 5 respectively for machine releasing. Then the transitions t47, t48, t49, t50, and t51 will each activate a sorting process in the background for the dispatching rules to make decisions. The process stops when there is no part present at cell entrance. Table 4 shows the refined deadlock patterns of the cell in FCFS rule, and Table 5 and 6 are the patterns from Rule-I and Rule-II respectively. In the deadlock patterns, 1 is used to represent Part_A, 2 is Part_B and so on. It is shown that some improvements have been made by employing alternative dispatching rules.

Table 2. Routing plans of an FMC

Part type	Process plans	Route type
Part_A	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

Table 3. Definitions of Figure 7

Definitions of places

Name	Cell Positions								Name	Cell positions								Name	Cell positions							
	m1	m2	m3	m4	m5	in	ex	rb		m1	m2	m3	m4	m5	in	ex	rb		m1	m2	m3	m4	m5	in	ex	Rb
p01						A			p32			O			F		A1R	p63	B3N		B3N		B3N			
p02	O					F		A1R	p33			A1R						p64	V		V		V			
p03								V	p34			A1N						p65							B	
p04	A1R								p35	O		F					A2R	p66	F		F		F		O	BR
p05	A1N								p36		O	F					A2R	p67						C		
p06	F	O						A2R	p37	F	O						A3R	p68		O				F		C1R
p07		A2R							p38	O	F						A3R	p69		C1R						
p08	V								p39		A3N							p70		C1N						
p09		A2N							p40		F					O	AR	p71		F		O				C2R
p10		F	O					A3R	p41	F						O	AR	p72				C2R				
p11			A3R						p42							A		p73				C2N				
p12		V							p43						B			p74				F	O			C3R
p13			A3N						p44		O				F		B1R	p75					C3R			
p14			F				O	AR	p45		B1R							p76				V				
p15			V						p46		B1N							p77					C3N			
p16	F		O					A2R	p47		F	O					B2R	p78					F		O	CR
p17			A2R						p48			B2R						p79					V			
p18			A3N						p49			B2N						p80							C	
p19		O	F					A3R	p50			F		O			B3R	p81						D		
p20		A3R							p51					B2R				p82				O		F		D1R
p21		O				F		A1R	p52					B2N				p83				D1R				
p22		A1R							p53			O		F			B3R	p84				D1N				
p23		A1N							p54			B3R						p85			O	F				D2R
p24	O	F						A2R	p55					V				p86			D2R					
p25	A2R								p56					O	F		B1R	p87	O		F					D3R
p26	A2N								p57					B1R				p88			D2N					
p27	F		O					A3R	p58					B1N				p89	D3R							
p28		F	O					A2R	p59			O		F			B2R	p90	D3N							

continue…

Definitions of places																										
Name	Cell Positions								Name	Cell positions								Name	Cell positions							
	m1	m2	m3	m4	m5	in	ex	rb		m1	m2	m3	m4	m5	in	ex	rb		m1	m2	m3	m4	m5	in	ex	Rb
p29	O		F					A3R	p60					V				p91	F						O	DR
p30	A3R								p61	O		F					B3R	p92							D	
p31	A1N								p62	B3R								p93					B3R			
Definitions of transitions																										
t01	O					F			t16	A3								t31					B1			
t02	A1								t17			O			F			t32		B3						
t03	F	O							t18			A1						t33		F	F		F		O	
t04		A2							t19			F				O		t34		C1						
t05		F	O						t20		F					O		t35		F		O				
t06			A3						t21	F						O		t36				C2				
t07	F		O						t22		B1							t37				F	O			
t08			A2						t23			B2						t38					C3			
t09		O	F						t24			F		O				t39					F		O	
t10		A3							t25					B3				t40				O		F		
t11		O				F			t26		F			O				t41				D1				
t12		A1							t27					B2				t42			O	F				
t13	O	F							t28			O		F				t43			D2					
t14	A2								t29			B3						t44	D3							
t15	O		F						t30					O	F			t45	F						O	
[m1 m2 m3 m4 m5]: machine No.1 2 3 4 5 respectively; **in**: cell entrance; **ex**: cell exit; **rb**: robot																										
M: Move; **F**: From; **O**: tO; **V**: aVailable; **[1 2 3]**: Process_Nmbers; **[A B C D]**: Part_Types **R**: Ready; **N**: doNe																										

Table 4. Deadlock patterns of FCFS

FCFS		
Length	Examples of deadlock pattern	Count
3	(1,1,1)(1,3,1)(2,4,3) (3,1,1)(3,4,4)(4,3,4)(4,4,3) (4,4,4)…	1100
4	(1,0,1,1)(1,1,0,1)(1,1,2,3)(1,1,3,1)(1,1,3,2) (2,1,4,3)…	6180
5	(2,1,1,0,1)(4,1,1,0,1)(1,2,1,0,1)(3,2,1,0,1) (2,3,1,0,1)(3,3,1,0,1)…	10432
6	(3,0,4,1,1,3)(4,0,4,1,1,3)(1,0,4,4,2,2)(3,0,4,4,2,2) (4,0,4,1,0,1)…	8148
7	(1,0,2,4,2,3,4)(3,0,2,4,2,3,1)(1,2,0,4,4,2) (3,2,1,0,4,4,4)…	8819
Total deadlock patterns		**34679**

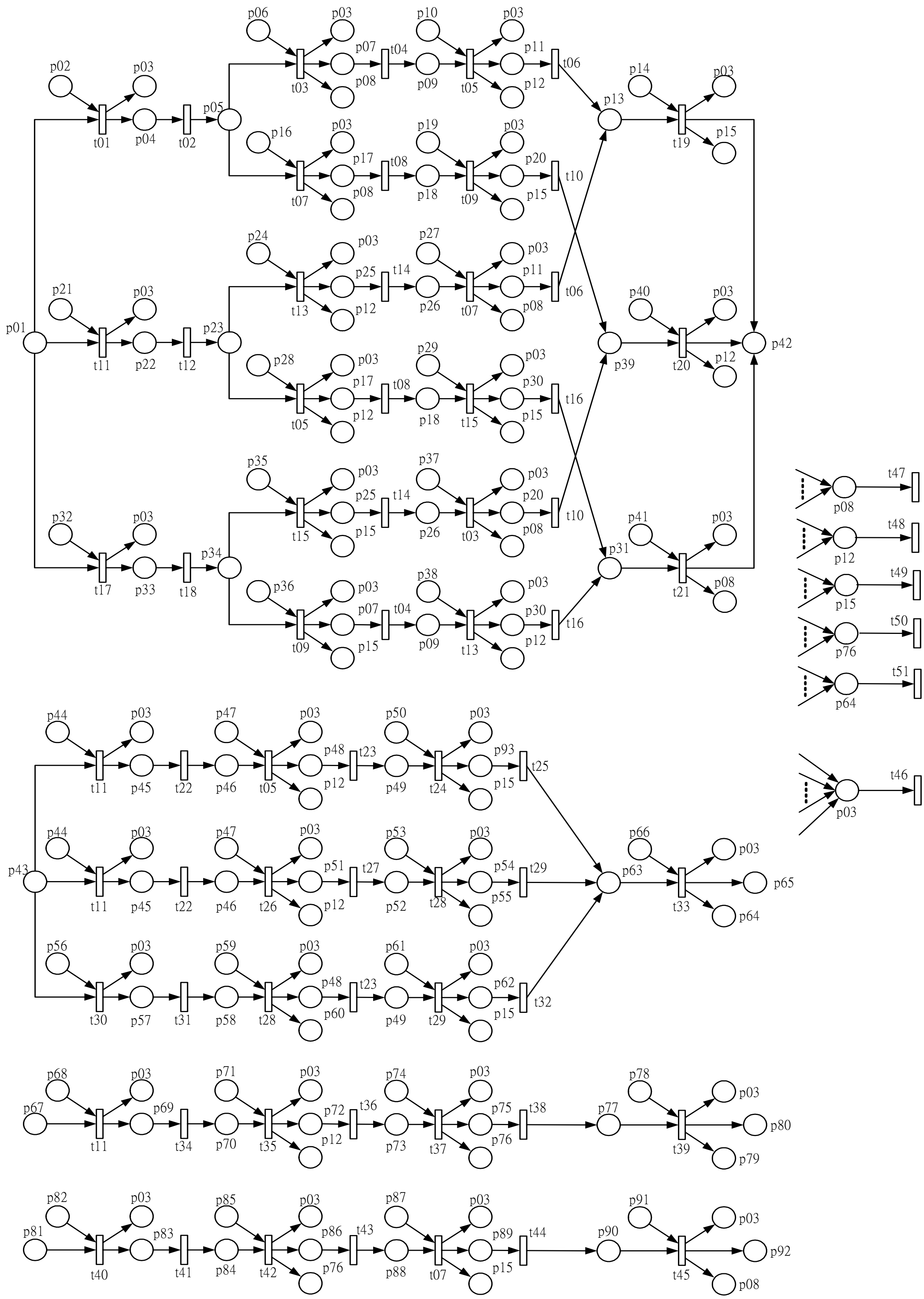

Figure 7. The routing of the implementation FMC

Table 5. Deadlock patterns of Rule-I

Rule-I		
Length	Examples of deadlock pattern	Count
3	(1,1,1)(2,4,3)(4,4,4)(4,4,3)(4,3,4)(3,4,4)…	900
4	(1,0,1,1)(1,1,1,3)(1,1,2,1)(1,1,2,3)(1,1,3,2)(3,1,3,1)…	4799
5	(1,0,1,1,0)(2,3,0,1,1)(1,4,4,2,2)(3,3,4,1,4)(4,4,4,1,3)(4,1,1,3,3)…	7462
6	(0,1,4,4,4)(2,0,2,1,1,3)(2,1,3,0,1,3)(4,3,0,1,3,2)(1,0,1,3,4,1)…	5061
7	(2,0,3,3,1,4,1)(1,0,3,0,4,4,2)(2,4,0,2,4,2,3)(4,1,2,4,1,3,4)…	6383
Total deadlock patterns		**24605**

Table 6. Deadlock patterns of Rule-II

Rule-II		
Length	Examples of deadlock pattern	Count
3	(1,1,1)(3,1,1)(1,3,1)(2,4,3)(4,4,4)(2,4,3)(4,4,3)(4,3,4)…	1100
4	(1,0,1,1)(1,1,0,1)(1,1,2,3)(1,1,3,1)(1,1,3,2)(2,1,4,3)…	6180
5	(2,1,1,0,1)(4,1,1,0,1)(1,2,1,0,1)(3,2,1,0,1)(2,3,1,0,1)(3,3,1,0,1)…	10424
6	(2,3,1,4,0,1)(3,3,1,4,0,1)(4,4,0,2,0,2)(4,0,4,2,0,2)(4,4,0,1,2,3)…	8068
7	(1,0,1,0,1,0,4)(2,4,1,4,1,0,4) (4,3,1,2,2,3,4)(2,4,1,2,2,3,4)…	7983
Total deadlock patterns		**33755**

In the implementation, timed Petri nets are built using PACE™ 3.1 on a PC with Pentium 4-2.0G and Microsoft Windows 2000 Professional. For each dispatching rule, collecting deadlock patterns from all 78125 specimens takes about 140 minutes. For each on-line testing at cell entrance, the process takes one second for every 1685 pattern matching in average. The processing time may be improved with better equipment and programming.

6. Discussions

Several observations were obtained during the implementation. Some have provided strong supports to the proposal, and rooms for improvement were also revealed. 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-lineup process.

The reason why deadlocks were not tremendously cut down by alternating dispatching rules is that, although provide 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 means 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 lineup, is tested at the cell entrance is that dissimilarity from the deadlock patterns can be easily acquired by controlling the first lineup. This method also helps increasing the possibility of test approval.

Tables 4 to 6 show that pattern (1,1,1) had caused deadlock in all three rules. It is obvious that Part_A cannot be processed in a batch of 3 or more. Although other dispatching rules may allow continuous processing of Part_A, but for existing rules, a bubble will be placed after every two entries of Part_A to prevent continuous lineup.

By checking the deadlock patterns in each test, there are no consecutive bubbles in any pattern. Therefore, one bubble should have provided sufficient relief for the traffic congestion in this example. Deadlocks would have happened because of the previously entered parts, not because of the bubbles. Therefore, in order to reduce the sample space of the test, specimens with two or more consecutive bubbles should be removed from the test list. Although a fixed-span bubble has provided operational simplicity to the tests, the elapsed time of a bubble should be a function of the in-cell parts and requires a careful examination for a better overall efficiency.

Another way of cutting down the test specimens is to discard the test specimens whose front portion is identical to any found deadlock pattern, so that the total number of deadlock simulations is reduced.

Since each deadlock pattern is collected when a simulated deadlock occurs and repeated patterns are discarded, shorter patterns will not appear in the front portion of any longer pattern. 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.

7. 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.

Acknowledgment

This research was funded by the National Science Council; project number NSC91-2212-E-324-002.

References

[1] Basnet, C. and Mize, J. H. 1994. Scheduling and control of flexible Manufacturing system: a critical review. *International Journal of Computer Integrated Manufacturing*, 7, 6: 340-355.

[2] Belarmino, A. D., Sebastian, L., Jesus, R., and Fernado, G.. 2001. Machine cell formation in generalized group technology. *Computers and Industrial Engineering*, 41: 227-240.

[3] Chang, K. F. and Gong, D. C. 1999. Introduction to the basic theory of petri met. *Journal of the Chinese Institute of Industrial Engineers*, 16, 1: 63-83.

[4] Huang, H. H. 2001. Matrix Controller Design and deadlock analysis of automated manufacturing systems. Part 1. Designing the matrix-based controller. *The International Journal of Advanced Manufacturing Technology*, 18: 434-447.

[5] Jeong, K. C. and Kim, Y. D. 1998. A real-time scheduling mechanism for a flexible manufacturing system: using simulation and dispatching rules. *International Journal of Production Research*, 36, 9: 2609-2626.

[6] Jonghun, P. and Spyros, A. R. 2001. Deadlock avoidance in sequential resource allocation systems with multiple resource acquisitions and flexible routings. *IEEE Transactions on Automatic Control*, 46, 10:1572-1583.

[7] Lee, D. Y. and DiCesare, F. 1994. Scheduling flexible manufacturing systems using Petri nets and heuristic seaech. *IEEE Transactions on Robotics and Automation*, 10, 2: 123-132.

[8] Maria, P. F., Bruno, M., and Biagio, T. 2000. Comparing digraph and Petri net approaches to deadlock avoidance in FMS. *IEEE Transactions on Systems, Man, and Cybernetics*, 30, 5:783-798.

[9] Maria, P. F., Guido, M., and Biagio, T. 2002. Design of supervisors to avoid deadlock in flexible assembly systems. *The International Journal of Flexible Manufacturing Systems*, 14: 153-171.

[10] Mark, L. and Spyros, R. 2001. Deadlock avoidance for sequential resource allocation systems: hard and easy cases. *The International Journal of Flexible Manufacturing Systems*, 13: 385-404.

[11] Matta, A., Tolio, T., Karaesmen, F., and Dallery, Y. 2001. An integrated approach for the configuration of automated manufacturing systems. *Robotics and Computer Integrated Manufacturing,* 17: 19-26.

[12] Mu, D. J. 1997. A Petri net synthesis theory for modeling flexible manufacturing systems. *IEEE Transactions on Systems, Man, and Cybernetics,* 27, 2:169-183.

[13] Peterson, J. L. 1981. *"Petri Net Theory and the Modeling of Systems"*. Prentice Hall Inc.

[14] Reyes, A., Yu, H., Kelleher, G., and Lloyed, S. 2002. Integrating Petri Nets and hybrid heuristic search for the scheduling of FMS. *Computers in Industry,* 47: 123-138.

[15] Tianlong, G.. and Parisa, A. B. 2002. A survey of Petri net applications in batch processes. *Computers in Industry,* 47: 99-111.

[16] Uzam, M. 2002. An optimal deadlock prevention policy for flexible manufacturing systems using Petri net models with resources and the theory of regions. *The International Journal of Advanced Manufacturing Technology,* 19: 192-208.

[17] Widodo, S. and Mark, A. L. 2001. Deadlock avoidance for manufacturing systems with partially ordered process plans. *IEEE Transactions on Robotics and Automation*, 17, 6:819-832.

[18] Xiong, H. H., Zhou, M. C., and Caudill, R. J. 1996. A hybrid heuristic search algorithm for scheduling flexible manufacturing systems. *Proceedings of the 13th IEEE International Conference on Robotics and Automation*, 2793-2797.

[19] Yim, S. J. and Lee, D. Y. 1996. Multiple objective scheduling for flexible manufacturing systems using Petri nets and heuristic search. *IEEE:* 2984-2989.

[20] Yu, H., Reyes, A., Cang, S., Yu, H., and Lloyed, S. 2003. Combined Petri net modeling and AI-based heuristic hybrid search for flexible manufacturing systems-part II. Heuristic hybrid search. *Computers and Industrial Engineering,* 44: 545-566.

[21] Zhou, M. C., Chiu, H. S., and Xiong, H. H. 1995. Petri net scheduling of FMS using branch and bound method. *IEEE:* 211-216.