

# A PETRI NET-BASED HYBRID HEURISTIC SCHEDULING ALGORITHM FOR FLEXIBLE MANUFACTURING SYSTEM

Xu, S. Z.

Nanjing University of Finance & Economics, Nanjing 210023, China

E-Mail: 9120181051@nufe.edu.cn

## Abstract

Considering the importance and complexity of scheduling to modern manufacturing systems, this paper puts forward a hybrid heuristic scheduling algorithm for flexible manufacturing system (FMS) based on Petri net (PN). Specifically, the PN modelling and scheduling algorithm of the FMS were discussed in details, and the hybrid heuristic scheduling algorithm was proposed based on the reachability graph of PN model. On this basis, a PN-based strategy for discrete event system (DES) modelling was developed, and applied to simulate the scheduling of a production line of shock absorbers. The results show that the FMS scheduling can be simulated effectively through the combination between extended PN and FlexSim.

(Received, processed and accepted by the Chinese Representative Office.)

**Key Words:** Flexible Manufacturing Systems (FMS), Petri Net (PN), Heuristic Scheduling, Discrete Event System (DES)

## 1. INTRODUCTION

The scheduling of flexible manufacturing system (FMS) is an NP-hard problem that cannot be optimized by polynomial algorithm. Over the years, this problem has attracted much attention from the academia, giving birth to various practical scheduling methods [1-7]. The traditional approaches for FMS scheduling include mathematical programming, neighbourhood search, heuristic scheduling and artificial intelligence. However, the manufacturing path in the FMS is too flexible to be scheduled well by these approaches.

One of the promising solutions to FMS scheduling lies in Petri net (PN), a powerful graphical modelling tool [8, 9]. The PN can model the constraints and features of the FMS in a systematic manner, making it possible to derive the optimal solution using optimization algorithm based on model information.

Faced with FMS scheduling, the PN is often integrated with heuristic scheduling, search algorithm and genetic algorithm, aiming to optimize the scheduling result in a proper computing time. Thus, it is possible to prepare the optimal scheduling plan by setting up a scientific PN model that accurately depicts the manufacturing system, which lays the basis for rapid organization of production and reduction of production cost [10].

## 2. LITERATURE REVIEW

With advanced manufacturing equipment, computing technique and information technology, the FMS can satisfy changing consumer demand with flexible manufacturing operations [11].

Recent years has seen the application of PN to the modelling and analysis on FMS scheduling. From the perspective of PN, the scheduling optimization problem is equivalent to the solving of the optimal trigger sequence of PN [12]. Many search methods have been adopted to improve the FMS scheduling. Based on the A\* search algorithm, Lee and DiCesare [13] combined PN and global search into the L1 heuristic search algorithm, which obtains the optimal result through the search for the most promising nodes in the reachable tree with the help of heuristic information. Since then, the L1 algorithm has been improved

repeatedly to ease the workload of the search for reachable graphs. For instance, Baruwa and Piera [14] restricted the extension of reachable graphs by selectively deleting undesirable branches. Luo et al. [15] introduced controllable backtracking (BT) to L1 algorithm, and then proposed two hybrid heuristic search algorithms, namely, L1 front-end BT method and back-end BT method. To minimize the search effort and maximize acceptability, Han et al. [16] put forward a hybrid scheduling algorithm based on PN, and controlled the sub-node generation with a descendent generator.

The previous studies have also shown the importance of heuristic functions to the enhancement of scheduling speed and quality. For instance, Lei et al. [17] minimized the mean flow time of FMS scheduling, taking the mean waiting time and scheduling rule as heuristic functions. Caballero et al. [18] established a resource-cost reachability matrix in light of the features of a special PN model, and derived a heuristic function representing the minimum utilization of machines and the maximum number of concurrent tasks. Li et al. [19] transformed the integer programming problem into matrix computation using PN state equation, and used the total cost of trigger vectors as heuristic function for the search process. In this way, the optimal or suboptimal plans were generated thanks to the sufficient global information in heuristic function.

Due to the changing structure and operation condition of the FMS, there is no scheduling rule that is always better than all the other rules. To solve the problem, the scheduling rules have been integrated into a knowledge base for multi-objective scheduling. For example, Kahloul et al. [20] integrated a knowledge base scheduling system into the FMS design and set up the corresponding PN model. Kim et al. [21] developed a PN-based simulation method that generates priority rules from the knowledge base during changes and conflicts to solve the scheduling problem. Lu et al. [22] proposed an object-oriented PN template, whose logic control contains scheduling and allocation knowledge.

The PN is an excellent descriptor of the special feature of discrete event system (DES) simulation: the system state varies only at discrete time points under the occurrence of some random events. In fact, the DES simulation has gained increased popularity with the development of object-oriented method. Currently, the PN-based DES research covers system modelling, analysis and control. For instance, Liu et al. [23] simulated and analysed the manufacturing system by object-oriented extended-time PN, and thus enhanced the real-time scheduling and execution of the system. Based on fuzzy parameters and stochastic PN, Koriem et al. [24] designed a DES simulation method for flexible manufacturing cell, and implemented the method to simulate a dynamic complex manufacturing system. Cabasino et al. [25] introduced hybrid PN to discrete manufacturing systems, and simulated parallel activities in such systems. Through empirical research, Li et al. [26] acquired relevant data from manufacturing enterprises, and used them to simulate the inventory management of time-limited materials. Van der Vorsta et al. [27] evaluated supply chain planning in a systematic manner, using both supply chain change modelling and DES simulation. Yu and Wright [28] probed deep into the use of PN in workflow model construction, and expounded the reasons for selecting PN to build DES simulation model.

### **3. PN-BASED FMS HYBRID HEURISTIC SEARCH SCHEDULING**

#### **3.1 PN-based FMS modelling**

The states of operations and resources in the PN are described as places. Each place whose operation is in progress is assigned with a token at the most, because the safety of each operation space hinges on the prevention of simultaneous implementation of different operations on the same machine. The token represents the available resources in the place. The caches with a greater-than-one capacity can be depicted as bounded places. If the

transition is enabled, the preconditions of the output place must have been satisfied; if the transition is triggered, the operation must have started or ended. Similar resources can be expressed as a place, in which the number of tokens represents the number of resources. If an operation can only be initialized under certain conditions and resources, the initialization process can be modelled as a transition of several input places. If an operation may release some resources and change some conditional states, then the completion process can be represented as a transition with several output places.

The FMS often consists of robots, machines, materials, buffers, conveying systems and accessories related to special processes. Special attention should be paid to the main operations and their sequential, concurrent, conflicting or cyclic relationships in the modelling phase. These basic relationships and their PN model can be summarized as follows:

**Sequential relationship:** If one operation in the PN follows another operation, then their places and transitions should form a cascade or sequential relationship in the system. As shown in Fig. 1, operation  $t_2$  occurs after operation  $t_1$ .

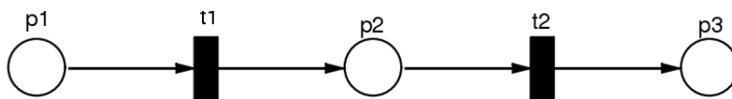


Figure 1: Sequential relationship.

**Concurrent relationship:** If two or more operations in the PN are initialized by the same event, then a parallel structure will form starting from the same transition, that is, two or more places are output places of the same transition. The concurrent relationship is shown as Fig. 2.

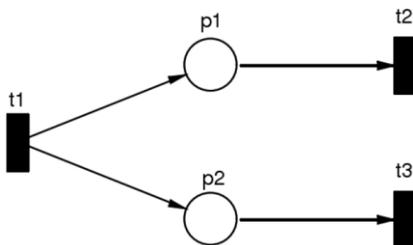


Figure 2: Concurrent relationship.

**Conflicting relationship:** If two or more operations in the PN share the same resource, there will be a competition for resources. This relationship can be described as a place shared by two or more transitions as input places. The conflicting relationship is shown as Fig. 3.

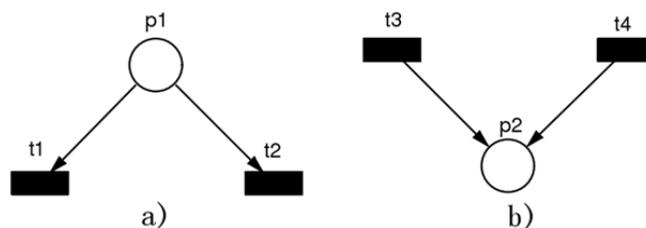


Figure 3: Conflicting relationship.

**Cyclic relationship:** If the operations in a sequence is interconnected, and if the first operation is initialized at the completion of the last one, then these operations will form a loop structure. The cyclic relationship is shown as Fig. 4.

The PN model for the FMS has some basic diagrams. The basic diagrams of several events need to be combined according to the needs of functional modelling features.

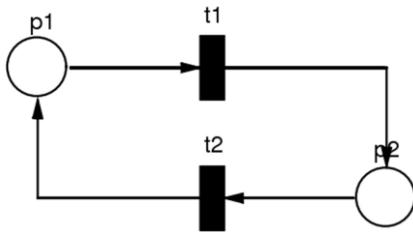


Figure 4: Cyclic relationship.

### 3.2 PN-based FMS scheduling

Considering the importance of operation time to scheduling, the time PN (*TPN*) was adopted and combined with the place factor, forming the timed place PN (*TPPN*). Here, the *TPPN* can be expressed as a six-tuple:  $TPPN = (P, T, F, W, M_0, dt)$ , where  $P = \{p_1, p_2, \dots, p_m\}$  is a limited set of places that represents resources or operations,  $T = \{t_1, t_2, \dots, t_n\}$  is a limited set of transitions that represents events,  $F \subseteq (P \times T) \cup (T \times P)$  is a set of directed arcs,  $W: F \rightarrow \{0, 1, 2, \dots\}$  is a mapping function that represents weight,  $M_0: P \rightarrow \{0, 1, 2, \dots\}$  is the initial identifier that represents the initial state of the system, and  $dt: P \rightarrow R^+$  is the time set associated with the place.

In the FMS, a process sequence needs to be designed according to the technical constraints of the products. The PN model of the process sequence for a 3-machine, 2-job FMS can be illustrated as Fig. 5, where M1~3 are the machines and J1~2 are the jobs.

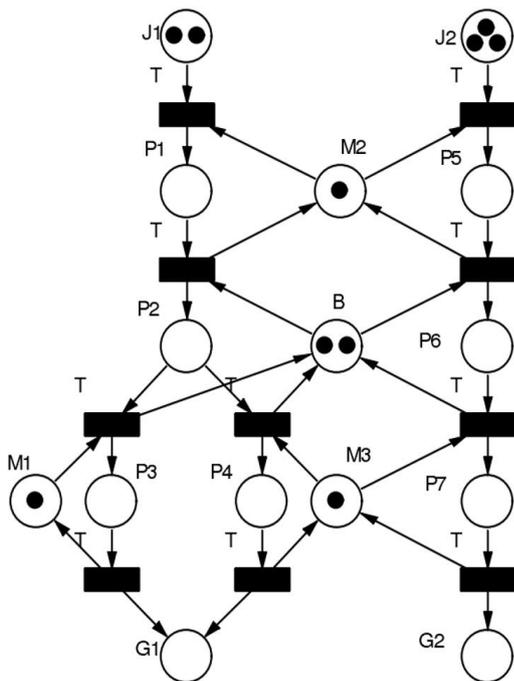


Figure 5: PN model of a FMS example.

In the established PN model of the FMS, the system changes can be reflected by the transition of the corresponding places. Hence, the optimal scheduling can be obtained in two steps: generating the reachability graph and finding the optimal path from the initial identifier to the target identifier. The path is the transition trigger sequence in the PN model.

### 3.3 Hybrid heuristic search algorithm for PN-based FMS scheduling

The numerous transition trigger sequences in the PN from the initial identifier to the target identifier can be represented by the reachable tree. Since the graph scale increased

dramatically with the growth in the number of transitions and places, it is an exponential complexity problem to enumerate all trigger sequences in the graph and traverse the whole reachable tree.

Here, the dynamic window search (DWS) is conceptualized to reduce the scale of the reachable tree and increase the chances of finding the optimal plan. Meanwhile, the best-first (BF) and BT strategies were adopted to search the top and bottom of the reachable graph, shortening the time to converge to the sub-optimal scheduling plan. In this way, a dynamic hybrid heuristic search algorithm was established.

The algorithm begins with the BF search. The search window moves down, when the top layer has maximum number of nodes (*max-node*), or the bottom layer has no node. The largest node of the heuristic function should be replaced with the smallest node, if the number of nodes falls between the top and bottom layers. When the newly generated node is above the top layer, the BT method should be adopted to start the search from the node. If the target node cannot be found, the search window should be returned and the BF method should be used to start the search with the node with the smallest heuristic value as the sink node. The specific steps of the algorithm are as follows:

Step 1: Place the initial identifier  $m_0$  in the *OPEN* list.

Step 2: Terminate the calculation if the *OPEN* list is empty.

Step 3: Remove the first identifier  $m$  from the *OPEN* list and place it in the *CLOSED* list.

Step 4: If  $m$  is the target identifier, construct an optimal path from the initial identifier to the target identifier, and terminate the calculation.

Step 5: Find the enabling transition under identifier  $m$  if  $depth(m) < top$ .

Step 6: For each enabling transition, generate the subsequent identifier, execute the process *Prune*, and delete some identifiers from the *OPEN* list.

Step 7: For each successor  $m'$  of  $m$ , calculate  $h(m')$  and  $f(m')$ . If  $m'$  is already in the *OPEN* list, adjust its pointer to the path producing the minimum  $f(m')$ ; if  $m'$  is already in the *CLOSED* list, adjust its pointer to the path producing the minimum  $f(m')$ . If  $depth(m') < top$ , select the node  $m''$  with the largest  $depth(m')$  layer  $f$  from the *OPEN* list; if  $depth(m') = top$ , place  $m'$  into the *OPEN* list; if  $size(top) = max-top$ , delete the bottom layer nodes from the *OPEN* list, add 1 to the *top* and the *bottom*, and go to Step 2; if  $depth(m') > top$ , go to Step 10.

Step 8: Check the *OPEN* list, add 1 to the *top* and the *bottom* if  $size(bottom) = 0$ .

Step 9: Go to Step 2.

Step 10: Take  $m'$  as the sink node of the BT search and put it into *OPEN0*.

Step 11: If *OPEN0* is empty, go to Step 2.

Step 12: Check the identifier  $m^*$  of the front end in *OPEN0*.

Step 13: If all enabling transitions in  $m^*$  have been triggered, move  $m^*$  from *OPEN0*, and go to Step 11.

Step 14: For an enabling transition that has not been triggered, generate a successor identifier  $m^{**}$ . Through calculation, use  $g(m^{**})$  to place  $m^{**}$  at the top of *OPEN0* and adjust the pointer from the subsequent identifier  $m^{**}$  to  $m^*$ .

Step 15: Extract the  $m^{**}$  if it is the target identifier. Search from  $m^{**}$  to  $m_0$  along the pointer direction, and thus identify the optimal path from the initial identifier to the target identifier. Then, terminate the algorithm.

Step 16: Remove  $m^{**}$  from *OPEN0* if it is a deadlock identifier.

Step 17: Go to Step 11.

#### **4. PN-BASED DES SIMULATION**

The DES is featured by discontinuous variables, random discrete time points and randomly changing state. In such a system, the current state may shift into various other states under different events or due to varied reasons. Here, the DES simulation is defined by stochastic systems, and the dynamic modelling process is explained in logical or mathematical form. In other words, the main modelling method is object-oriented DES simulation, in which the

system components are treated as objects covering such three aspects as message, activity and property.

#### 4.1 DES simulation strategy

The simulation strategy directly bears on the modelling effect and model composition. The common DES simulation strategies fall into three categories.

##### (1) Event scheduling

As the name suggests, event scheduling explores the actual system from the perspective of events, and identifies each event and its impact on the system state. For each event, the relevant logical relations are defined and implemented in time sequence. During model construction, all events are placed in the event table, and the time control part is taken into account. This part picks out the first event from the event table, and adjusts the simulation clock at the time of the event. Then, the event is processed by the corresponding module of the event is called, before returning to the control part. The above selection and processing are repeated until the termination condition is satisfied or the termination event is observed.

##### (2) Activity scanning

Activity scanning considers the system components as activities, each of which has an activation condition. Once the condition is fulfilled, the corresponding activity will be activated. During the simulation, the generation time of an activity is often regarded as a priority condition, and compared with other conditions. The activation conditions are measured in two steps: judging if the generation time of an activity is as required and assessing other conditions. The above scanning process is repeated until the simulation stops.

##### (3) Process interaction

Process interaction controls the simulation clock of the system using two event tables, namely, the current event list involving events being implemented right now and the future event list containing events that will occur in future.

It is difficult to build specific mathematical models for the DES. To overcome the difficulty, the activities of system entities are usually defined in the form of process or network diagram. Hence, the mathematical model of the DES is actually a flow chart of entities belonging to the following three parts: the arrival model of temporary entities, queuing rules, and the service model of permanent entities.

#### 4.2 Extended PN-based DES simulation model

The basic model of the DES is shown in Fig. 6, where  $X$  is the set of inputs,  $Y$  is the set of outputs,  $\delta$  is the processor,  $S$  is the fluctuating system state, and  $t$  is a time-marching function. It can be seen from Fig. 6 that the transfer functions are abstracted into a set of processors  $\delta$ . Therefore, the DES model should focus on the description of the complex processing procedure. Owing to the interaction between discrete events, the system state evolves in an extremely complex manner, and the real-time variation is too subtle to be simulated by the PN-based model. This calls for the abstraction at the logical level is needed before PN-based modelling.

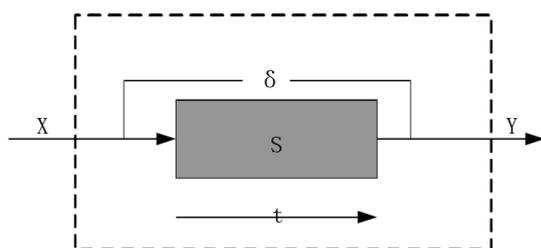


Figure 6: Basic DES model.

The introduction of system concept to the PN enables the hierarchal simulation of the DES, such that simple descriptions can be made at one layer while the target system behaviour can be detailed at another layer. The resulting entrance and exit of the high-layer nodes give the PN a hierarchical structure.

The general DES model is explained in Fig. 7 below, after the DES is abstracted and modelled with extended PN.

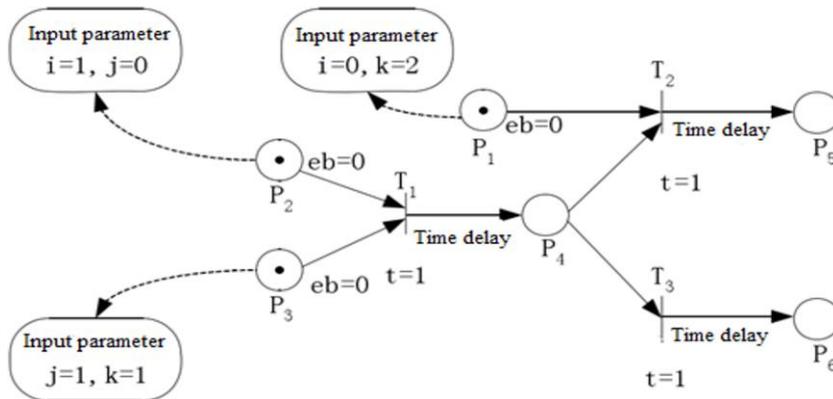


Figure 7: The general DES model.

The PN in Fig. 7 has 6 places, 3 transitions and 8 directed arcs. The letter P denotes the set of places, and T denotes the set of transitions.  $P_1$ ,  $P_2$  and  $P_3$  have initial markers and can be self-excited. Each transition records the value and attribute of the time attribute  $eb$  arrived by the initial input place, and each transition has a time delay.

The simulation was designed through the following steps, after the logical abstraction of DES and the establishment of extended PN model. Firstly, the depth-first tree structure was constructed from the PN topology. Then, the event scheduling by traditional simulation methods was transformed into active objects, i.e. a simulation queue was set up to schedule the active objects in the model. Finally, the arrival time, transition trigger time and transition extension time of each place in the extended PN were calculated in turns.

Let  $\Phi$  be the set of input ports of the model. The simulation algorithm can be defined as:

(1) Scan the whole network to find the transition  $t$  that can be fired, initialize the firing and the timestamp, and add the active object corresponding to the transfer  $t$  to the queue of the model.

(2) Scan the set of input ports  $\Phi$  of model, and thus obtain the minimum clock of the set of input port sets  $\sigma = \min\{T(p), p \in \Phi\}$ .

(3) Execute the simulation object of  $\sigma$ , and add it to the queue via the adjacent active object after depth-first access.

(4) Update the queue and timestamp of the object that communicates with the simulation object as  $T_{out} = T_{in} + \Delta t$ .

(5) Put forward the current simulation clock  $\delta$ , delete the node and all arcs ending with the clocked from the graph, and go to Step (1) until the queue is empty.

## 5. FMS DES SIMULATION MODEL CONSTRUCTION AND ANALYSIS

To evaluate FMS scheduling efficiency, the extended PN was combined with FlexSim to study a typical production line in the FMS, which produces three kinds of shock absorbers, denoted respectively X, Y and Z, through ten workstations. Each workstation needs to find the corresponding FlexSim object to express the actual meaning. The physical design of the production line system and the workstation assignment schedule are presented as Tables I and II, respectively.

Table I: Physical design of the production line system.

Flexsim object	Actual production line system
Source	Placing product orders
Queue	Buffering products
Separator	Decomposing different types of products
Processor	Workstations
Conveyor	Transporting products between workstations
Object	Products to be processed
Sink	Delivery to warehouse after completion

Table II: The workstation assignment schedule.

Workstation	X-type shock absorber		Y-type shock absorber		Z-type shock absorber	
	Standard work time	Runtime	Standard work time	Runtime	Standard work time	Runtime
Processor 1	18	U (12, 24)	19	U (13, 25)	15	U (10, 22)
Processor 2	21	U (15, 27)	16	U (11, 23)	16	U (11, 23)
Processor 3	19	U (12, 25)	20	U (14, 26)	17	U (11, 23)
Processor 4	22	U (16, 28)	18	U (12, 24)	18	U (12, 24)
Processor 5	18	U (12, 24)	19	U (13, 25)	16	U (11, 23)
Processor 6	17	U (11, 23)	20	U (15, 27)	20	U (14, 26)
Processor 7	19	U (12, 25)	20	U (15, 27)	18	U (12, 24)
Processor 8	21	U (15, 27)	20	U (15, 27)	19	U (13, 25)
Processor 9	17	U (11, 23)	15	U (10, 22)	17	U (11, 23)
Processor 10	22	U (16, 28)	21	U (15, 27)	19	U (13, 25)

According to the collected data, the half-year production cycle of the production line system was simulated by the DES simulation model. The results show that the effective time of the system for half a year was about 72,000 mins. Table III shows the simulation results at the production rhythm of 50 mins.

Table III: The simulation results at the production rhythm of 50 mins.

Object	Class	Idle	Processing	Busy	Blocked	Empty	Conveying
Processor 1	Workstation 1	42.16 %	57.84 %	0.00 %	0.00 %	0.00 %	0.00 %
Processor 2	Workstation 2	46.72 %	53.28 %	0.00 %	0.00 %	0.00 %	0.00 %
Processor 3	Workstation 3	45.77 %	54.23 %	0.00 %	0.00 %	0.00 %	0.00 %
Processor 4	Workstation 4	39.21 %	60.79 %	0.00 %	0.00 %	0.00 %	0.00 %
Processor 5	Workstation 5	40.89 %	59.11 %	0.00 %	0.00 %	0.00 %	0.00 %
Processor 6	Workstation 6	38.32 %	61.68 %	0.00 %	0.00 %	0.00 %	0.00 %
Processor 7	Workstation 7	39.78 %	60.22 %	0.00 %	0.00 %	0.00 %	0.00 %
Processor 8	Workstation 8	38.79 %	61.21 %	0.00 %	0.00 %	0.00 %	0.00 %
Processor 9	Workstation 9	47.32 %	52.68 %	0.00 %	0.00 %	0.00 %	0.00 %
Processor 10	Workstation 10	45.39 %	54.61 %	0.00 %	0.00 %	0.00 %	0.00 %
Conveyor 0–9	-	0.00 %	0.00 %	0.00 %	0.00 %	0.00 %	100.00 %

As shown in Table III, none of the conveyor belts were blocked at the production rhythm of 50 mins, indicating that the normal operation of the production line did not cause congestion and overload. The workstations of shock absorbers were ranked in ascending order by idle time, revealing that Processor 9 had the highest idle rate (47.32 %) and Processor 6 had the lowest idle rate (38.32 %). Thus, the proposed method can optimize the target production line, leading to better efficiency and lower cost.

## **6. CONCLUSIONS**

This paper puts forward a PN-based hybrid heuristic scheduling algorithm for the FMS. Firstly, the related theories of the PN and discrete system were elaborated, and the FMS scheduling methods were reviewed. Then, the author proposed a hybrid heuristic scheduling algorithm for the FMS. After that, a PN-based DES modelling strategy was developed through a detailed analysis on FMS modelling and simulation. Finally, the proposed DES modelling was applied to simulate the scheduling of a production line of shock absorbers. The results show that the FMS scheduling can be simulated effectively through the combination between extended PN and FlexSim.

## **ACKNOWLEDGMENT**

The authors acknowledge the financial support from a research fund program about philosophy and social science at the colleges and universities of Jiangsu Provincial Department of Education (2017SJB2056) and an annual subject in 2016 on the Jiangsu Education and Science “13<sup>th</sup> Five-Year Plan” (C-b/2016/01/26).

## **REFERENCES**

- [1] Li, Z. W.; Zhou, M. C. (2004). Elementary siphons of Petri nets and their application to deadlock prevention in flexible manufacturing systems, *IEEE Transactions on Systems, Man, and Cybernetics – Part A: Systems and Humans*, Vol. 34, No. 1, 38-51, doi:[10.1109/TSMCA.2003.820576](https://doi.org/10.1109/TSMCA.2003.820576)
- [2] Ezpeleta, J.; Colom, J. M.; Martinez, J. (1995). A petri net based deadlock prevention policy for flexible manufacturing systems, *IEEE Transactions on Robotics and Automation*, Vol. 11, No. 2, 173-184, doi:[10.1109/70.370500](https://doi.org/10.1109/70.370500)
- [3] Li, B.; Guo, C.; Ning, T. (2018). An improved bacterial foraging optimization for multi-objective flexible job-shop scheduling problem, *Journal Européen des Systèmes Automatisés*, Vol. 51, No. 4-6, 323-332, doi:[10.3166/JESA.51.323-332](https://doi.org/10.3166/JESA.51.323-332)
- [4] Stecke, K. E. (1983). Formulation and solution of nonlinear integer production planning problems for flexible manufacturing systems, *Management Sciences*, Vol. 29, No. 3, 273-288, doi:[10.1287/mnsc.29.3.273](https://doi.org/10.1287/mnsc.29.3.273)
- [5] Yang, X. P.; Gao, X. L. (2018). Optimization of dynamic and multi-objective flexible job-shop scheduling based on parallel hybrid algorithm, *International Journal of Simulation Modelling*, Vol. 17, No. 4, 724-733, doi:[10.2507/IJSIMM17\(4\)CO19](https://doi.org/10.2507/IJSIMM17(4)CO19)
- [6] Tang, M.; Qi, Y.; Zhang, M. (2017). Impact of product modularity on mass customization capability: An exploratory study of contextual factors, *International Journal of Information Technology and Decision Making*, Vol. 16, No. 4, 939-959, doi:[10.1142/S0219622017410012](https://doi.org/10.1142/S0219622017410012)
- [7] Matejic, M.; Tadic, B.; Lazarevic, M.; Misic, M.; Vukelic, D. (2018). Modelling and simulation of a novel modular fixture for a flexible manufacturing system, *International Journal of Simulation Modelling*, Vol. 17, No. 1, 18-29, doi:[10.2507/IJSIMM17\(1\)407](https://doi.org/10.2507/IJSIMM17(1)407)
- [8] Sun, G.; Bin, S. (2018). A new opinion leaders detecting algorithm in multi-relationship online social networks, *Multimedia Tools and Applications*, Vol. 77, No. 4, 4295-4307, doi:[10.1007/s11042-017-4766-y](https://doi.org/10.1007/s11042-017-4766-y)
- [9] Wang, X.-L.; Huang, Q.-G. (2018). Infusion monitoring communication model of smart home based on coloured Petri net, *International Journal BIOAUTOMATION*, Vol. 22, No. 3, 239-252, doi:[10.7546/ijba.2018.22.3.239-252](https://doi.org/10.7546/ijba.2018.22.3.239-252)
- [10] Gong, D.; Liu, S.; Tang, M.; Ren, L.; Liu, J.; Liu, X. (2018). Revenue sharing or profit sharing? An internet production perspective, *Advances in Production Engineering & Management*, Vol. 13, No. 1, 81-92, doi:[10.14743/apem2018.1.275](https://doi.org/10.14743/apem2018.1.275)
- [11] Moro, A. R.; Yu, H.; Kelleher, G. (2002). Hybrid heuristic search for the scheduling of flexible manufacturing systems using Petri nets, *IEEE Transactions on Robotics and Automation*, Vol. 18, No. 2, 240-245, doi:[10.1109/TRA.2002.999652](https://doi.org/10.1109/TRA.2002.999652)

- [12] Ferrarini, L.; Piroddi, L. (2008). Modeling and control of fluid transportation operations in production plants with Petri nets, *IEEE Transactions on Control Systems Technology*, Vol. 16, No. 5, 1090-1098, doi:[10.1109/tcst.2007.906345](https://doi.org/10.1109/tcst.2007.906345)
- [13] Lee, D. Y.; DiCesare, F. (1994). Scheduling flexible manufacturing systems using Petri nets and heuristic search, *IEEE Transactions on Robotics and Automation*, Vol. 10, No. 2, 123-132, doi:[10.1109/70.282537](https://doi.org/10.1109/70.282537)
- [14] Baruwa, O. T.; Piera, M. A. (2014). Anytime heuristic search for scheduling flexible manufacturing systems: a timed colored Petri net approach, *International Journal of Advanced Manufacturing Technology*, Vol. 75, No. 1-4, 123-137, doi:[10.1007/s00170-014-6065-3](https://doi.org/10.1007/s00170-014-6065-3)
- [15] Luo, J. C.; Xing, K. Y.; Zhou, M. C.; Li, X. L.; Wang, X. N. (2015). Deadlock-free scheduling of automated manufacturing systems using Petri nets and hybrid heuristic search, *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, Vol. 45, No. 3, 530-541, doi:[10.1109/TSMC.2014.2351375](https://doi.org/10.1109/TSMC.2014.2351375)
- [16] Han, L.; Xing, K.; Chen, X.; Xiong, F. (2018). A Petri net-based particle swarm optimization approach for scheduling deadlock-prone flexible manufacturing systems, *Journal of Intelligent Manufacturing*, Vol. 29, No. 5, 1083-1096, doi:[10.1007/s10845-015-1161-2](https://doi.org/10.1007/s10845-015-1161-2)
- [17] Lei, H.; Xing, K.; Han, L.; Xiong, F.; Ge, Z. (2014). Deadlock-free scheduling for flexible manufacturing systems using Petri nets and heuristic search, *Computers & Industrial Engineering*, Vol. 72, 297-305, doi:[10.1016/j.cie.2014.04.002](https://doi.org/10.1016/j.cie.2014.04.002)
- [18] Caballero-Villalobos, J. P.; Mejía-Delgadillo, G. E.; García-Cáceres, R. G. (2013). Scheduling of complex manufacturing systems with Petri nets and genetic algorithms: a case on plastic injection moulds, *International Journal of Advanced Manufacturing Technology*, Vol. 69, No. 9-12, 2773-2786, doi:[10.1007/s00170-013-5175-7](https://doi.org/10.1007/s00170-013-5175-7)
- [19] Li, C.; Wu, W.; Feng, Y.; Rong, G. (2015). Scheduling FMS problems with heuristic search function and transition-timed Petri nets, *Journal of Intelligent Manufacturing*, Vol. 26, No. 5, 933-944, doi:[10.1007/s10845-014-0943-2](https://doi.org/10.1007/s10845-014-0943-2)
- [20] Kahloul, L.; Bourekkache, S.; Djouani, K.; Chaoui, A.; Kazar, O. (2014). Using high level Petri nets in the modelling, simulation and verification of reconfigurable manufacturing systems, *International Journal of Software Engineering and Knowledge Engineering*, Vol. 24, No. 3, 419-443, doi:[10.1142/S0218194014500168](https://doi.org/10.1142/S0218194014500168)
- [21] Kim, Y. W.; Suzuki, T.; Narikiyo, T. (2007). FMS scheduling based on timed Petri net model and reactive graph search, *Applied Mathematical Modelling*, Vol. 31, No. 6, 955-970, doi:[10.1016/j.apm.2006.10.023](https://doi.org/10.1016/j.apm.2006.10.023)
- [22] Lu, M.-S.; Tseng, L.-K. (2010). An integrated object-oriented approach for design and analysis of an agile manufacturing control system, *International Journal of Advanced Manufacturing Technology*, Vol. 48, No. 9-12, 1107-1122, doi:[10.1007/s00170-009-2359-2](https://doi.org/10.1007/s00170-009-2359-2)
- [23] Liu, H.; Jiang, Z.; Fung, R. Y. K. (2009). Performance modeling, real-time dispatching and simulation of wafer fabrication systems using timed extended object-oriented Petri nets, *Computers & Industrial Engineering*, Vol. 56, No. 1, 121-137, doi:[10.1016/j.cie.2008.04.006](https://doi.org/10.1016/j.cie.2008.04.006)
- [24] Koriem, S. M.; Dabbous, T. E.; El-Kilani, W. S. (2004). A new Petri net modeling technique for the performance analysis of discrete event dynamic systems, *Journal of Systems and Software*, Vol. 72, No. 3, 335-348, doi:[10.1016/S0164-1212\(03\)00211-5](https://doi.org/10.1016/S0164-1212(03)00211-5)
- [25] Cabasino, M. P.; Giua, A.; Poggi, M.; Seatzu, C. (2011). Discrete event diagnosis using labeled Petri nets. An application to manufacturing systems, *Control Engineering Practice*, Vol. 19, No. 9, 989-1001, doi:[10.1016/j.conengprac.2010.12.010](https://doi.org/10.1016/j.conengprac.2010.12.010)
- [26] Li, Z. W.; Wu, N. Q.; Zhou, M. C. (2012). Deadlock control of automated manufacturing systems based on Petri nets – a literature review, *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, Vol. 42, No. 4, 437-462, doi:[10.1109/TSMCC.2011.2160626](https://doi.org/10.1109/TSMCC.2011.2160626)
- [27] Van der Vorsta, J. G. A. J.; Beulens, A. J. M.; van Beek, P. (2000). Modelling and simulating multi-echelon food systems, *European Journal of Operational Research*, Vol. 122, No. 2, 354-366, doi:[10.1016/s0377-2217\(99\)00238-6](https://doi.org/10.1016/s0377-2217(99)00238-6)
- [28] Yu, B.; Wright, D. T. (1997). Software tools supporting business process analysis and modelling, *Business Process Management Journal*, Vol. 3, No. 2, 133-150, doi:[10.1108/14637159710173096](https://doi.org/10.1108/14637159710173096)