

# Optimization for Manufacturing Process Based on Timed Petri Net and Genetic Algorithm

Li Tingpeng<sup>1a</sup>, Li Yue<sup>1b</sup>, Qian Yanling<sup>1c</sup>, Zeng shuanggui<sup>2d</sup>

<sup>1</sup>Science and Technology on Integrated Logistics Support Laboratory, National University of Defense Technology, Changsha, 410073, China

<sup>2</sup> Unit 72474, HanNan, 572018, China

<sup>a</sup>tovey1987@126.com <sup>b</sup>liyue@nudt.edu.cn <sup>c</sup>ylqian@nudt.edu.cn <sup>d</sup>281487857@qq.com

**Keywords:** Petri Net; Optimization; Genetic Algorithm; Production Scheduling

**Abstract.** In order to improve the optimization efficiency, an optimization approach for Manufacturing Process based the Timed Petri Net (TPN) model and Genetic Algorithm is proposed in this paper. Firstly, the Manufacturing Process is modeled with Timed Petri Net (TPN). Then, an improved Genetic Algorithm (GA) is proposed to get the optimal process. In order to introduce the method in detail, a FMS scheduling problem is shown in this paper.

## Introduction

Petri net is a graphical and mathematical modeling and analyzing tool for systems characterized as concurrent, distributed, asynchronous, parallel and/or stochastic<sup>[1]</sup>. The concept of Petri net was developed and introduced by Carl Adam Petri in 1962<sup>[2]</sup>. From then on, the theory and application of Petri net has a great development and many high-level variants of Petri net are proposed, such as Timed Petri Net (TPN)<sup>[3, 4]</sup>, Stochastic Petri nets (SPN)<sup>[5, 6]</sup>, and Colored Petri Nets (CPN)<sup>[7, 8]</sup>. TPN is one of the most widely used Petri net in engineering projects, especially for discrete time programming problem (e.g., production scheduling problem). On the one hand, TPN is used to analyze the properties of objects, such as reachability, safety, aliveness and persistence<sup>[9-14]</sup>. For another aspect, TPN is used as optimization model to obtain the best solution together with optimization algorithm, such as L1 algorithm<sup>[15]</sup>, Particle Swarm Optimization (PSO)<sup>[16]</sup>, Ant Colony Optimization (ACO)<sup>[17]</sup>, Simulated Annealing (SA)<sup>[18]</sup> and Genetic Algorithm (GA)<sup>[19-21]</sup>. For Flexible Manufacturing System (FMS) Scheduling Problem and Job-Shop Scheduling Problem (JSP), GA is a frequently used optimization algorithm.

## Definitions and concepts related to TPN

The basic Petri net is Place/Transition-net (P/T-net) and usually is defined as a five tuple<sup>[20]</sup>:  $PN = (P, T, F, W, M_0)$ , where  $P$  is a finite set of places;  $T$  is a finite set of transitions;  $F$  is a set of directed arcs;  $W$  defines the weight of each arc. For those arcs with no weight,  $W$  is equal to one.  $M_0$  is the initial marking. In addition,  $P \cap T = \emptyset$  and  $P \cup T \neq \emptyset$ <sup>[9]</sup>. In P/T-net modeling, using the concept of conditions and events, places represent conditions and transitions represent events<sup>[1]</sup>.

Timed Petri Net (TPN) is the extension of the basic Petri nets by adding time parameter to places ( $P$ ) (Timed Place Petri nets, TPPN), transitions ( $T$ ) (Timed Transition Petri nets, TTPN) or directed arcs ( $F$ ) (Timed Arcs Petri nets, TAPN). Generally, TPPN is the most widely used TPN to model the process of operation, such as the process of machining work pieces. In this paper, we use TPN to refer to TPPN if not specified particularly.

When modeling FMS scheduling process, every operation is modeled by a transition pair and a place, while the transition pair represents the beginning and ending of an operation, and the place represents the operation itself. Moreover, the operation stations of work pieces and the resource used in the operation, such as equipment and workers, are also represented by places. An example of TPN model of operation is shown as Figure 1.

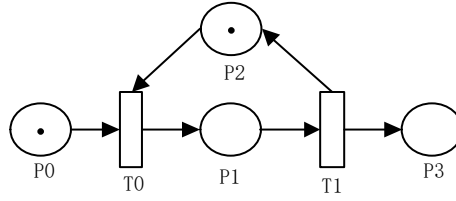


Fig. 1 An example of TPN model of operation

The meanings and parameters of places and transitions in figure 1 TPN model are shown in Table 1.

Table 1 The meanings and parameters of the TPN model

NO.	Meanings	Parameters
T0	Beginning of operation	—
T1	Ending of operation	—
P0	Ready to be machined	Number of work pieces
P1	The process of Operation	Operation time
P2	Resource(equipment/worker)	Number of resources
P3	Finish machining	Number of work pieces

Where, *T0* and *T1* represent the beginning and ending of operation, respectively; State places *P0* and *P3* represent the state of ready to be machined and the state of finish having been machined, respectively; the token of these two places represent the number of work pieces. Operation places *P1* represents the process of operation and the time parameter is defined as the time cost in this operation process. Resource place *P2* represents the equipment in use, such as machines and workers.

### Optimization approach

Genetic algorithms (GA) are heuristic stochastic search methods based on biological evolution mechanisms. GA uses several genetic operators and strategies to find better solutions for a certain problem by selectively exploring new regions of the solution space with a set of individuals. Crossover and mutation are the most common genetic operators, which emulate the reproduction and spontaneous alteration of individuals in living species. An individual, also known as a chromosome, represents a solution to the problem and is usually coded by binary or integer<sup>[9, 22]</sup>. The set of individuals used by GA is called a population. Implicit parallelism and global searching ability are the main characteristics of GA. Therefore, GA is suitable for optimizing complex NP, such as FMS scheduling problem. In this paper, an improved genetic algorithm based on the Simplified TPN is proposed via improving the genetic operators and simplifying the reachability checking with the contract conditions of the mathematical model.

Based on the simplified TPN and the improved Genetic Algorithm, an optimization approach is proposed in this paper and the detail steps shown as following.

**Step 1**, Simplify the model with the approach introduced in section 1.2.

**Step 2**, Code each transition of the Simplified TPN with integer coding mechanism introduced in 2.1.

**Step 3**, Calculate the order contracts and mutual exclusion contracts of the simplified TPN with the steps of 1.3.1. Based on the contracts, the mathematical model can be obtained with the target function defined in 1.3.2.

**Step 4**, Initialize the generation of chromosomes randomly.

**Step 5**, Check the reachability of the initial population. As introduced in 2.2, the reachability checked chromosome has fewer transitions and represents a reachable schedule plan.

**Step 6**, Use the evaluation operator to generate the next generation chromosomes.

**Step 7**, Calculate the fitness of each chromosome.

**Step 8**, If the optimization approach satisfies the stop conditions, then stop, otherwise go to **step 6**.

## Application

In order to evaluate the proposed optimization approach, we apply it on a classical schedule problem.

**Problem description.** A job shop schedule system includes five work pieces and six machines. The work pieces are marked with  $J_1, J_2, J_3, J_4, J_5$ , and the machines are marked with  $M_1, M_2, M_3, M_4, M_5$  and  $M_6$ . The six machines are divided into three types denoted as I, II, III, as shown in Table 2.

Table 2 The types of machines						
Types	I		II		III	
Machine	$M_1$	$M_2$	$M_3$	$M_4$	$M_5$	$M_6$

The jobs information, i.e., the machine types and work time of each steps, are shown in Table 3.

Table 3 The Jobs information of work pieces							
Object	Jobs	Machine	Time(h)	Object	Jobs	Machine	Time(h)
$J_1$	①	III	4	$J_4$	①	I	4
	②	I	5		②	III	4
	③	II	3		③	II	7
$J_2$	①	II	3	$J_5$	①	III	3
	②	III	4		②	II	5
	③	I	2		③	I	4
	④	II	5				
$J_3$	①	I	4				
	②	II	6				
	③	III	4				

When optimizing the best schedule plan, the following assumptions should be assumed.

- 1) Only one work piece can be machined by one machine at any time.
- 2) Each job cannot be stopped before being finished.
- 3) The next job can be started only after the previous job is finished.

The target of this schedule problem is to find out the best schedule plan which needs the least time.

**The simplified TPN of the schedule system.** The TPN is the base to optimize the schedule problem, thus we build the TPN of the problem and simplify it with the approach described in 1.2. The simplified TPN is shown in Figure 2. The places of the Simplified TPN represent the machining states of work pieces, e.g.,  $P_0, P_1, P_2, P_3$  and  $P_4$  represent the states of work pieces which are ready to be machined.  $M_1, M_2, M_3, M_4, M_5$  and  $M_6$  are resource places that represent the six machines.

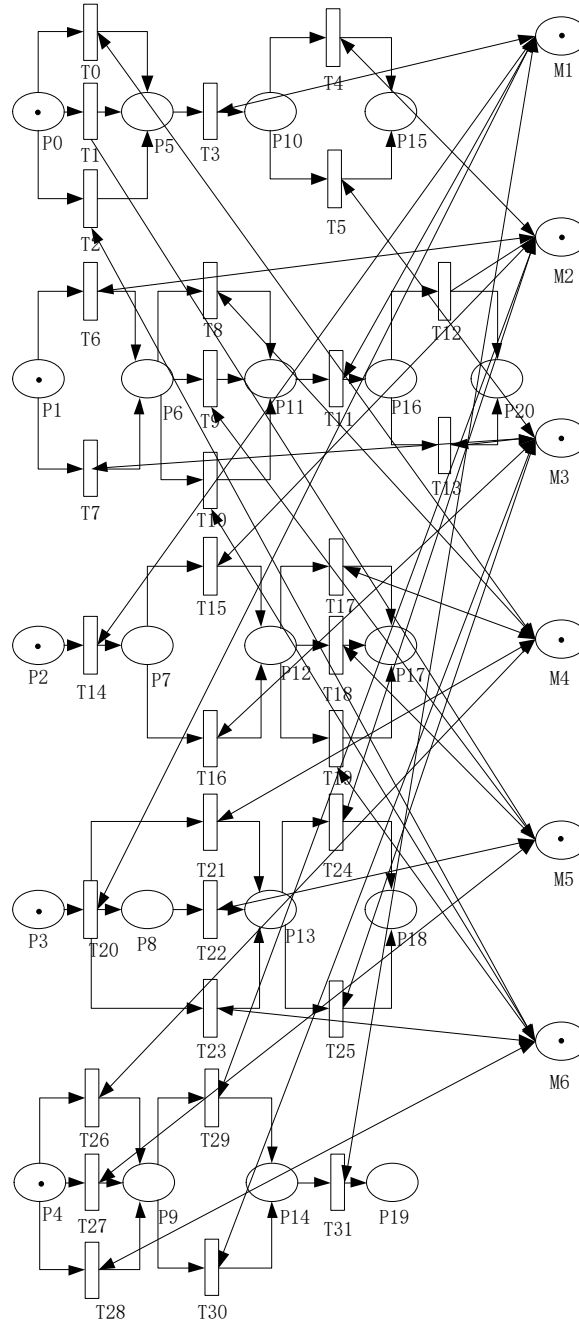


Fig. 2 The Simplified TPN of the schedule system

The transitions in the simplified TPN represent the operation processes of each job. The meaning and time parameters are shown in Table 4.

Table 4 The meanings and time parameter of the TPN

Transition	Time	Meaning	Transition	Time	Meaning	Transition	Time	Meaning
T <sub>0</sub>	4	J <sub>114</sub>	T <sub>1</sub>	4	J <sub>115</sub>	T <sub>2</sub>	4	J <sub>116</sub>
T <sub>3</sub>	5	J <sub>121</sub>	T <sub>4</sub>	3	J <sub>132</sub>	T <sub>5</sub>	3	J <sub>133</sub>
T <sub>6</sub>	3	J <sub>212</sub>	T <sub>7</sub>	3	J <sub>213</sub>	T <sub>8</sub>	4	J <sub>224</sub>
T <sub>9</sub>	4	J <sub>225</sub>	T <sub>10</sub>	4	J <sub>226</sub>	T <sub>11</sub>	2	J <sub>231</sub>
T <sub>12</sub>	5	J <sub>242</sub>	T <sub>13</sub>	5	J <sub>243</sub>	T <sub>14</sub>	4	J <sub>311</sub>
T <sub>15</sub>	6	J <sub>322</sub>	T <sub>16</sub>	6	J <sub>323</sub>	T <sub>17</sub>	4	J <sub>334</sub>
T <sub>18</sub>	4	J <sub>335</sub>	T <sub>19</sub>	4	J <sub>336</sub>	T <sub>20</sub>	5	J <sub>411</sub>
T <sub>21</sub>	4	J <sub>424</sub>	T <sub>22</sub>	4	J <sub>425</sub>	T <sub>23</sub>	4	J <sub>426</sub>
T <sub>24</sub>	7	J <sub>432</sub>	T <sub>25</sub>	7	J <sub>433</sub>	T <sub>26</sub>	3	J <sub>514</sub>
T <sub>27</sub>	3	J <sub>515</sub>	T <sub>28</sub>	3	J <sub>516</sub>	T <sub>29</sub>	5	J <sub>522</sub>
T <sub>30</sub>	5	J <sub>523</sub>	T <sub>31</sub>	4	J <sub>531</sub>			

In Table 4,  $J_{ikm}$  represents the  $k$ th job of work piece  $i$  with machine  $M_m$  ( $m=1, 2, 3, 4, 5, 6$ ).

**The coding of the simplified TPN.** Analyzing the simplified TPN, there are total 32 transitions. Therefore, according to the coding mechanism introduced in 2.1, the initial chromosomes are composed of number 0 ~ 31, such as:

**Chro2** (0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31).

**Mutual exclusion contracts.** The mutual exclusion contracts are not only one of the conditions of reachability of chromosome, but also can be used to reduce the number of genetics of chromosome. The mutual exclusion contracts of the simplified TPN can be obtained and shown in Table 5.

Table 5 The mutual exclusion contracts of the Simplified TPN

No.	Contracts	No.	Contracts	No.	Contracts	No.	Contracts
1	(0,1,2)	2	(4,5)	3	(6,7)	4	(8,9,10)
5	(12,13)	6	(15,16)	7	(17,18,19)	8	(21,22,23)
9	(24,25)	10	(26,27,28)	11	(29,30)		

**Order contracts.** According to the definition of the order contracts and the method of obtaining the contracts, the order contracts of the simplified TPN are shown as Table 6.

Table 6 Order contracts of the Simplified TPN

No.	Contracts	No.	Contracts	No.	Contracts	No.	Contracts	No.	Contracts
1	(0,3)	2	(1,3)	3	(2,3)	4	(3,4)	5	(3,5)
6	(6,8)	7	(6,9)	8	(6,10)	9	(7,8)	10	(7,9)
11	(7,10)	12	(8,11)	13	(9,11)	14	(10,11)	15	(11,12)
16	(11,13)	17	(14,15)	18	(14,16)	19	(15,17)	20	(15,18)
21	(15,19)	22	(16,17)	23	(16,18)	24	(16,19)	25	(20,21)
26	(20,22)	27	(20,23)	28	(21,24)	29	(22,24)	30	(23,24)
31	(21,25)	32	(22,25)	33	(23,25)	34	(26,29)	35	(27,29)
36	(28,29)	37	(26,30)	38	(27,30)	39	(28,30)	40	(29,31)
41	(30,31)								

The reachable chromosome only includes 16 transitions, e.g., **Chro2\***(the reachable chromosome of **Chro2**). **Chro2\*** (7, 14, 2, 15, 20, 28, 10, 29, 22, 11, 3, 12, 25, 17, 31, 5)

**The results.** According to the optimization approach proposed in 2.5, the results are obtained through implementing it in C++. The total time of the best schedule plan is 20 hours and the fired time (start time and end time) of the transitions are shown in Table 7. The number of individual of each population is 100 and the total number of the populations is 2000.

Table 7 The start time and end time of transitions

Transition	Start time	End time	Transition	Start time	End time
T6	0	3	T14	0	4
T0	0	4	T15	4	10
T20	4	9	T28	4	7
T9	4	8	T30	7	12
T22	9	13	T11	9	11
T3	11	16	T12	11	16
T25	13	20	T17	13	17
T31	16	20	T4	16	19

The Gantt graph of the best schedule plan is shown in Figure 3 .

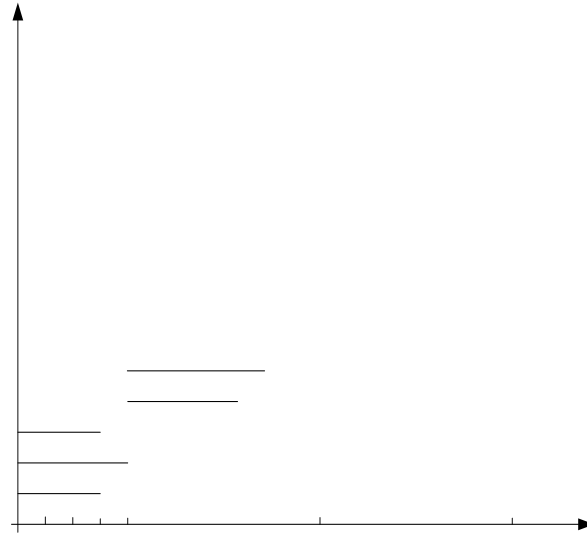


Fig. 3 The Gantt graph of optimization results

## Summary

In order to evaluate the proposed optimization approach, we use the L1 algorithm<sup>[15]</sup> to optimize the same schedule problem shown in 3.1. The total time of the results of L1 algorithm is 25 hours and the fired time of the schedule plan is shown in Table 8.

Table 8 The optimization results of L1 Algorithm

Transition	Start time	Finish time	Transition	Start time	Finish time
T0	0	4	T6	0	3
T14	0	4	T27	0	3
T8	4	8	T15	4	10
T20	4	9	T30	4	9
T3	10	15	T17	10	14
T22	10	14	T4	15	18
T11	15	17	T12	18	23
T25	18	25	T31	18	22

By comparing the results of the two optimization approaches, we can see that the proposed approach can get better results.

What's more, in order to evaluate the efficiency of this approach, the comparison with traditional Genetic Algorithm<sup>[23]</sup> is also conducted and the results are shown in Table 11. The number of chromosomes of initial population is 100 and max number of populations is 500. Since it is a stochastic process, it requires multiple replications. We run the experiments 10 times and take the mean of 10 individuals as the final results.

Table 9 The comparison with traditional GA

	Result	CUP time
Our Algorithm	22 hours	26.847seconds
Traditional Algorithm	25 hours	32.137 seconds

In Table 11, the cup time contains not only algorithm time, but also the time to display the best results of each population. However, for the same number of populations and the same platform, the display time of the two algorithms can be supposed as the same. From the results, we can conclude that the proposed algorithm not only can find better results, but also less time.

## Reference

- [1] Tadao Murata. Petri Nets: Properties, Analysis and Applications[J]. PROCEEDINGS OF THE IEEE, 1989, 77(4): 541-580.

- [2] Bijan Sarkar, Bimal Samanta. Application of Petri nets for systems modelling and analysis [J]. OPSEARCH, 2012, 49(4): 334-347.
- [3] Xiao Yi-zhong. The research and application of workflow system based on timed Petri nets [D]. Shanghai Jiaotong University, 2006
- [4] Yu Jia. Study on workflow model based on timed petri net [D]. Zhejiang Gongshang University, 2010
- [5] Maen Atli, Zhouhang Wang, H. Kondo Adjallah. Coloured stochastic Petri nets modelling for the reliability and maintenance analysis of multi-state multi-unit systems[J]. Journal of Manufacturing Technology Management, 2014, 25(4): 476-490.
- [6] Ding Feng. Research on the Use of Colored Timed Petri Net and Stochastic Petri Net [D]. Hefei University of Technology, 2009.
- [7] Hoda Davarzani Seyed Hessameddin Zegordi. Developing a supply chain disruption analysis model: Application of colored Petri-nets[J]. Expert Systems with Applications, 2012, 2012(39): 2102-2111
- [8] Li Jingkui. The Reaserch of Parallel Testing and development of tool based on colored Petri nets, [D]. Nanjing University of Aeronautics and Astronautics, 2010.
- [9] Farooq Ahmad Ayesha Sadiq, Sher Afzal Khan, Jose C. ,Valverde, Tabbasum Naz,Muhammad Waqas Anwar. Modeling and analysis of departure routine in air traffic control based on Petri nets[J]. Neural Comput & Applic, 2014, 2014(25):1099-1109.
- [10] Hammadi Bennoui. Interacting behavioral Petri nets analysis for distributed causal model-based diagnosis[J]. Auton Agent Multi-Agent Syst, 2014, 2014(28): 55-181.
- [11] Cheng-Ying Yang Victor R.L. Shen, Jia-Hong Lee. Analysis of the problem description in the Algorithm for Inventive Problem Solving using Petri nets[J]. Knowledge-Based Systems, 2012, 2012(35):290-303
- [12] Liang Qi Yuyue Du, And Mengchu Zhou. Analysis and Application of Logical Petri Nets to E-Commerce Systems[J]. TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS: SYSTEMS, 2014, 44(4): 468-481.
- [13] Abbas Dideban Meysam Zareieen, Ali Asghar Orouji. Safety analysis of discrete event systems using a simplified Petri net controller[J]. ISA Transactions, 2014, 2014(53):44-49.
- [14] Alessandro Giua Isabel Demongodin. Dynamics and steady state analysis of controlled Generalized Batches Petri Nets[J]. Nonlinear Analysis: Hybrid Systems, 2014, 2014(2):33-49.
- [15] Wang Chao. Research on Job--shop Scheduling Problem Based on Petri Nets [D]. BeiJing Jiaotong University, 2011.
- [16] OIN Na, YUE Xiao-bo, LIU Wu. Particle swarm optimization algorithm of JSP based on Petri net [J]. Computer Applications , 2008, 28(9):2166-2169.
- [17] Wang Xiao-rong, WU Tie-jun. Flexible jobshop Scheduling Based on Petri-net model: ACO-GA hierarchiacal evolutionary optimization approach[J]. Jouranl of Zhejiang University (Engineering Science), 2004, 38(3): 286-291.
- [18] TAO Ze, SUI Tian-zhong, XIE Li-yang, LIU Xiao-xia. Dynamic Scheduling Optimization of Dual-Resource Based on Petri Net and GASA [J]. Journal of Northeastern University(Natural Science) , 2007, 28(3): 405-409.
- [19] Y. M. Pan Albert W. L. Yao. A Petri Nets and Genetic Algorithm Based Optimal Scheduling for Job Shop Manufacturing Systems[C]. IEEE International Conference on System Science and Engineering, 2013:99-104.

- [20] Kiyoshi Itoh Tad Gonsalves. GA optimization of Petri net-modeled concurrent service systems[J]. Applied Soft Computing, 2011, 2011(11):3929-3937.
- [21] Guoshan Zhang Zixiang Zhao, Zhigang Bing. Scheduling Optimization for FMS Based on Petri Net Modeling and GA[C]. Proceeding of the IEEE International Conference on Automation and Logistics, 2011:422-427.
- [22] HAO Dong, JIANG Chang-Jun, LIN Lin. Petri Net Based MOdeli ng and GA Based Scheduling for FMS [J]. CHINESE JOURNAL OF COMPUTERS , 2005, 28(2):201-208.