Proceedings of the 17th World Congress
The International Federation of Automatic Control
Seoul, Korea, July 6-11, 2008

Timed Petri nets model on bell-type batch annealing process and its simulation using SystemC platform

Zhang Xiaoping, Wang Zhigang, Liu Quanli, Wang Wei, Zhao Jun

Research Center of Information and Control, Dalian University of Technology, Dalian 116024,
Pepople's Republic of China (e-mail: wangwei@dlut.edu.cn)

Abstract: Based on the analysis of the technical flow of bell-type batch annealing in cold rolling plant, a Timed Petri net model for describing batch annealing process is established in this paper. In order to solve the concurrence and time sequence of annealing process in the course of simulation, the mapping rules between Petri nets and SystemC platform are constructed by using the SystemC simulation platform which is widely employed in the design of large-scale integrated circuit. Accordingly, the simulation model of bell-type batch annealing process is implemented. The simulation result shows the proposed model and simulating method is better than the traditional object oriented discrete event simulation on the simulating speed and the scheduling result.

Keywords: bell-type batch annealing process; Timed Petri nets; SystemC simulation platform; discrete event simulation

1. INTRODUCTION

The bell-type annealing is an important procedure in steel industry, by which the cold rolling products with high quality are produced. In general, the numbers of annealing bases, coolers, furnaces and other resources are economically limited, and the heating and cooling time of diversified type of cold coils are largely different. Even if using the hydrogen bell-type furnace, the working time of an annealing plan is more than 30 hours. Therefore, the annealing stack order in the shop greatly influences the production efficiency, and it is significant to ensure the reasonable annealing scheduling of batch cold coil stacks to improve the whole throughput of cold rolling plant.

The production scheduling process of bell-type batch annealing is a typical multi-machine, multi-stage, re-entrant shop scheduling problem with complex constraints on resources, energy and so on. An optimal coil stacks combination on each base was obtained, but without dealing with the scheduling of annealing process (Venkatesh *et al.*, 1983). A mixed-integer linear programming (MILP) on the batch annealing scheduling was established, in which all bases were assumed to be empty in the beginning, and each stack was assigned to a base prior to scheduling (Moon et al. 1999). In order to reduce the complexity of the model, the original model was divided into two MILP models (MILP-1 and MILP-2) that were solved by time windows decomposing method. But the assumption mentioned above was not reasonable in the practical environment because of the continuity of manufacturing, and it was very hard to build up a better algorithm to solve the large-scale model within so many complicated restrictions.

A no-wait two-stage shop scheduling problem was used to describe scheduling process which was solved by the parameter and value sequencing strategy to guarantee the shortest annealing time (Wu *et al.*, 2006). However, the related resources scheduling in the annealing process was neglected in that paper. A process control computer system of batch annealing was developed, which included process control, annealing state tracking and operating guidance (Wen, 1983). That system saved labour effectively and utilized the equipment efficiently. Nevertheless, it did not give an optimal annealing scheduling solution. An optimal scheduling method for bell-type batch annealing process was proposed that combined an improved genetic algorithm with discrete event simulation techniques, and had improved production efficiency after applied to an annealing shop (Liu *et al.*, 2005). The construction of that simulation system using object-oriented event division method leaded to the high complexity of simulation program, and the running mode based on single process made the simulation time very long. A simulated prediction system using neural network modelling technique was developed (Deepankar *et al.*, 2006). Yet, the model was difficultly achieved along with the increase of the complexity of the practical problem. Therefore, it is necessary to find an easy modelling method and simulation technique to carry out the simulation scheduling of the annealing shop.

The bell-type batch annealing process is a typical DEDS (Discrete Event Dynamic System), whose model is very difficult to be described normally (Jerry Banks *et al.*, 2005).Petri nets characterized by describing system's asynchronism, concurrence and conflict adapt to model the DEDS (Moody *et al.*, 1998). Nowadays, many complicated production scheduling problems have been modelled with it (Desrochers *et al.*, 1995; Dimitris Kiritsis *et al.*, 1996; Armando *et al.*, 1997; Doo *et al.*, 1994; David *et al.*, 1994;Yang *et al.*, 1993). In this paper, the technical flow of the bell-type annealing shop is analyzed and a Timed Petri net model is developed, in which the place and the transition correspond to the working state and the event of procedure respectively. Referring to the advantages of SystemC platform

978-1-1234-7890-2/08/$20.00 © 2008 IFAC

10.3182/20080706-5-KR-1001.2067

that solves the concurrence and time sequence in large-scale integrated circuit design, and combining the Petri nets model, this paper establishes the mapping principles between them. Then, the Timed Petri net scheduling model of annealing shop is simulated. The simulation results with practical production data shows that the simulation runs more rapidly than the traditional object-oriented discrete event simulation, and it provides an excellent annealing operation guidance which guarantees a shorter working time and higher resource utilization.

2. TIMED PETRI NETS MODEL OF BELL-TYPE BATCH ANNEALING SCHEDULING

2.1 Timed Petri nets

With the development of theory and application in the past four decades, different categories of advanced Petri nets (coloured Petri net, controled Petri net, Timed Petri net, stochastic Petri net and so on) had been brought forward, and they become a graphical and mathematical modelling tool to represent many types of dynamic systems, especially DEDS (Y, 1998). A Petri net may be defined as a particular kind of bipartite directed graph, in which three types of objects are involved (Places, Transitions, and Directed arcs). If time variables are added, the Timed Petri nets are formed, which enable to conveniently describe the dynamic system along with the time activities. The places with time variables in a Timed Petri net generally have two kinds, instantaneous places and time-lapse place (Zuberek, 1991). The difference between them focuses on the firing of transitions, and the former place is to act on the transition immediately right after the state changes, while the latter needs a period of time before firing the corresponding transitions. Thus, the constitutional elements and mechanism in Timed Petri net are well corresponding to those of DEDS. In this paper, we represent the states of object as places, and the events occurring in dynamic system as transitions.

2.2 Timed Petri nets model of bell-type furnace scheduling

In the practical environment, a bell-type batch annealing shop consists of many annealing areas, and each of them includes a lot of parallel annealing bases, on which the coils stack (a annealing plan) that contains three to five coils is loaded by workers. Each annealing plan needs to be handled with 11 working procedures (Liu *et al.*, 2005), and the working time of each procedure is relatively fixed except for the heating and the cooling procedures. Owing to the limitation of the number of furnaces, coolers and cranes, the resource competition between equipment exists in each base area. Furthermore, the share resource competition between the base areas such as crane, nitrogen gas also exists. For example, there are four base areas that have the same collocation in an annealing shop of cold rolling plant. In each area, four bases, two furnaces, two coolers and two cranes are placed in all. If the furnaces in one base area are all occupied, then the loaded base has to wait for an available furnace until the occupied furnaces are released. On the other hand, if the crane is busy at this time, the idle furnace also needs to wait. Therefore, the reasonable annealing scheduling can avoid the conflicts in annealing process, and can ensure the effective usage of equipments and resources.

We combines the annealing coil stacks with the bases and regards them as a manufacturing workpiece; meanwhile, regards cranes, furnaces and coolers as the fixed facilities. Firstly, the annealing shop is devided into many modules through different base area, which avoids the incorrect match between the furnaces, the coolers and the bases. Then each module is described by a Petri subnet; in other words, the Timed Petri nets model of the whole shop can be constituted by a series of Petri subnets. To a certain Timed Petri subnet model of a base area, the events created by various working procedures such as *beginning to load coils*, *finishing loading coils* and so on (Liu *et al.*, 2005) are expressed by transition T_{ijb} and T_{ije}, in which *i* denotes the annealing plan number; *j* denotes the sequence number of annealing procedures; *b* and *e* denote the beginning event and the ending event, respectively. The states involved working procedures such as *waiting for loading coils*, *loading coils*, *resources idling* and so forth are represented by the timed places, p_{ij_wait}, p_{ij_busy} and p_{m_idle}, in which the meanings of *i* and *j* are the same as the mentioned above; *wait* indicates that *i*th annealing plan's *j*th working procedure is waiting for being processed; *busy* indicated that it is being processed; *idle* indicates that the equipment is vacant; *m* represents the equipments and the resources in annealing process (1 is base, 2 is furnace, 3 is cooler, 4 is crane and so forth), and its amount is expressed by the number of tokens in the places p_{m_idle}. To above example, the numbers of token in the places p_{m_idle} such as bases, furnaces, coolers and cranes are 4, 2, 2 and 2 respectively. The amount of tokens varies according to the occupancy or release of the equipment. For example, when one furnace is occupied at a certain time in the annealing process, its token number will be decreased by one; if released, add one in other way.

In this paper, a Timed Petri subnet model of one base area is illustrated by Figure 1. Based on the practical production situation, some equipments and resources are idle and others are busy, which can be represented by variation of the tokens. When an annealing plan is ready for processing, the transition T_s is fired, which means an annealing process begins. Each transition is fired by one or more places, and then the instantaneous place or time-lapse places are created. The duration of instantaneous places is determined by every working time (Liu *et al.*, 2005). Thus, all annealing plans go through 22 places and 22 transitions altogether such as *waiting for loading coils*, *beginning to load coils*, *loading coils*, *finishing loading coils* and so on. For instance, the beginning to load coils transition T_{i4b} is fired by the places, p_{m_idle}, p_{2_idle} and p_{4_idle}, then entering into the place p_{i4_busy}. At that time, the token numbers of the places, p_{2_idle} and p_{4_idle}, subtract one respectively. After eight minutes, the transition T_{i4e} is fired, the crane is released, and the token number of p_{4_idle} adds one. When entering into the p_{i5_wait}, if the heating resources are sufficient, then enter into the p_{i5_busy}; otherwise, wait for the sufficient heating resources. With such simulation, all annealing plans can go through the whole annealing process based on the running principle of the Timed Petri nets, and the proposed model directly describes the annealing working process of a base area.

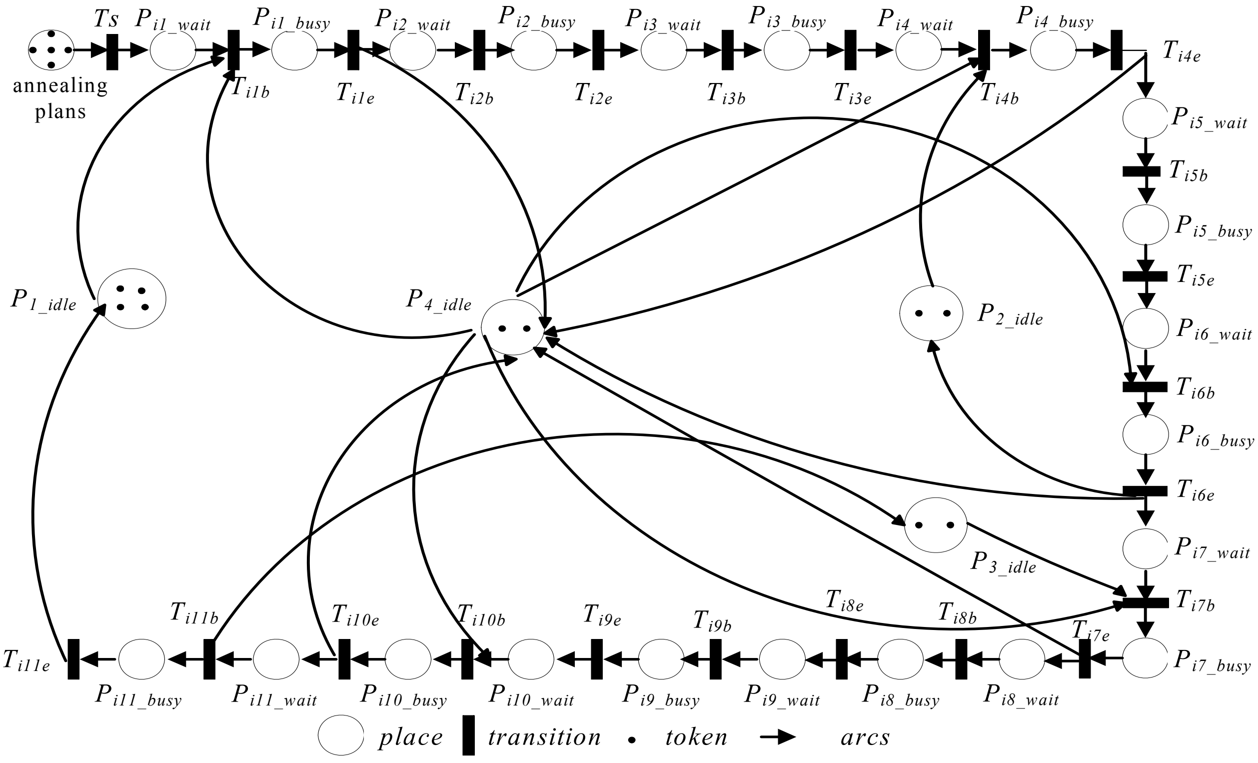

Figure 1 Timed Petri subnet model of a base area in the annealing shop

3. SIMULATION OF TIMED PETRI NETS MODEL BASED ON SYSTEMC PLATFORM

3.1 SystemC simulation platform

SystemC is originally brought forward to solve the design of VLSI (Very Large Scale Integrated-circuit) (Bhasker, 2002). Its essential units are some classes of modules (*SC_MODUE*) with process member function (*SC_THREAD*), in which other child modules and channels (ports and signals) for connecting modules are involved. Events can be defined as changes of signals at input ports of module, and the process in module is the basic functional unit, which is triggered by the corresponding event. Processes are organized as coroutines, and provide the mechanism for concurrent simulation. The sequence of events and the concurrence of processes are driven by a continuously running clock; meanwhile, an event-driven and multi-process interaction simulation kernel accelerates the simulation speed of the model. Holger Vogt et al. (2003) developed a model using SystemC platform and simulated the production process in the semiconductor fabrication factories. However, there is little research outcome in the literatures that uses SystemC platform to simulate bell-type batch annealing process. Since SystemC can supplement the disadvantage of information deficiency, data description and hardware depiction shortage in Petri nets, and the Petri nets modelling can assist SystemC to directly realize graphical expression, this paper bands both of them together to model and simulate the annealing scheduling process.

3.2 Mapping between Timed Petri nets model and SystemC platform

Since the state transformation of the model presents different characteristics due to the varied structure and parameter, it is necessary to check the speciality of practical system through analysis or simulation method (Zhou, 1998). The simulation technique can analyse and estimate the performance of dynamic system through simulating the model. SystemC platform models and simulates the practical complex system in terms of microcosmic pattern, and provides a system-level and high-level abstraction. Based on the characteristics of the composing contents of Petri net and SystemC, the paper maps the modelling elements of Petri net such as place, transition and time variable onto the programming elements of SystemC such as process, event, time under the following mapping principle. ($\rightarrow$ represents a mapping)

1) To a Petri subnet system S= $\{CP_i\}, CP_i \rightarrow$ modules in SystemC (*SC_MODULE*)

A dynamic system is divided into several components by the concepts of hierarchical, concurrent and serialization. These components are represented as modules or the Petri subnet.

2) To place P, P→process in SystemC (*SC_THREAD* or *SC_METHOD*)

A place represents a state of dynamic system in Petri net model and can be described by a *SC_THREAD* process or a *SC_METHOD* process in SystemC. A *SC_THERAD* process is suspended during the simulation until an event occurs, while a *SC_METHOD* process can merely run forward to the end once it has been triggered by an event. Here, we adopt the process *SC_THREAD*.

3) To transition T, T→event in SystemC (*SC_EVENT*)

A transition represents the event that changes the state of system in Petri net, which can be described by an event in SystemC, named *SC_EVENT*. Only if the conditions that trigger the event are satisfied, the event occurs by the method *notify()*, which is encapsulated in SystemC platform.

4) To the time delay D, the time variable in Timed Petri net, D→ time in SystemC (*SC_TIME*)

The time delay D represents the duration of system state in Timed Petri net, and can be described by the time *SC_TIME* in SystemC. The time *SC_TIME* can define different time period that represents the state duration with the method *wait (t, SC_NS)* in SystemC platform.

In the modelling and simulating of this paper, we firstly divide the whole system into several different subsystems by the module concept of SystemC platform, then we establish the Timed Petri net model of each subsystem, and realize the simulated conceptual model according to the mapping principle mentioned above.

3.3 Model implementation using SystemC

Since SystemC has the object-oriented features, we only need to establish a based module if the system contains a few of similar modules. Thus, other modules similar to the based one can accept or reject its characteristics and behaviors. That makes the model expand easily, shares the programme code effectively and improves the implemented efficiency of the model. Each module can be implemented in SystemC based on the constitution of Petri net model and the mapping principles (see 2.2 section). Here, we take an example of implementation. The definition of the Timed Petri net model mentioned in Figure 1 in SystemC is as follows:

SC_MODULE (*area1*) //the base area 1
{*sc_signal*<bool>*base [4]*; //the state of each base (0 is idle, 1 is busy)
………
sc_event b_load; //loading coils event begins(transition T_s)
………
sc_time time_list[1][10]; // the duration of each state
int worktime; // total annealing working time
int resource_efficiency; // resource utilization ratio
SC_CTOR (*area1*) // base area 1
{*SC_THREAD* (*wait_load*); // waiting for loading coils process (the place p_{i1_wait})
sensitive<<*start*; //triggered by the event of beginning to annealing(the transition T_s)
SC_THREAD(*load*), //loading coils process (the place p_{i1_busy})
sensitive<<*b_load*; // triggered by the event of beginning to loading coils (transition T_{i1b})
………}
void *wait_load*() //the implementation function of the waiting for loading coils process
{bool *flag*=*juge_load*(); // whether annealing resources are sufficent
if (*flag*= =0)
b_load.notify(); //trigger the event of beginning to load coils
else
b_load.notify(time_list[0][0],SC_NS); //trigger the event of beginning to load coils after *time_list[0][0] SC_NS*
}
void *load*() //the realization function of loading coils process
{*wait (time_list[1][0]*, *SC_NS*); //load coils for *time_list[1][0] SC_NS*
e_load.notify(); //trigger the event of finishing loading coils, (the transition T_{i1e}) }};

From the model implementation, the encapsulated class, module, event and process simplify the procedures, in which a base area, namely a Timed Petri subnet model, is represented by module *SC_MODULE*, and the 22 production states described by the places in Petri subnet model is represented by the process *SC_THREAD* that are triggered by the corresponding 22 transitions. These transitions are represented by the events *SC_EVENT* in SystemC, and the event-driven and multi-process interaction simulation kernel makes the 22 processes run in parallel. In the running of the Petri net model, a lot of transitions can be triggered at one time. In this paper, we refer to the priority grades setting in Liu *et al.* (2005), which is that each beginning transition T_{ijb} is fired successively based on the priority under some conditions. In other words, the firing level of the transition *beginning to unload coils* is lowest, while the transition *beginning to unload furnace* is highest, and the same types of transitions are fired under the FIFO principle. When a series of annealing plans complete the arrangement and get ready to be annealed, the SystemC program begins. Firstly, the event *start*, namely transition T_s, is fired, and then the process *wait_load* begins to run. If the annealing resources signal *flag* is equal to zero, the event *b_load* (beginning to load coils) is fired by method *notify()*. Subsequently, the process *load* begins to run until this annealing process finishes. During this process, the simulation clock is driven by the occurring event at a time, and the system states are recorded.

4. SIMULATION AND ANALYSIS OF A BELL-TYPE BATCH ANNEALING SCHEDULING

In this paper, the simulation data comes from the practical production (Liu *et al.*, 2005), in which 36 annealing plans were being annealed at an annealing recipe, and other fifteen annealing plans were waiting for being annealed. In other words, at that moment, the production state in the annealing shop was that not all bases, furnaces, coolers and cranes were idle. When building the Time Petri net model of this annealing shop, we partition the whole shop into four base areas depending on the different type of 46 bases, and the four areas are modelled by the Timed Petri subnet respectively. The numbers of base in each area are 11, 11, 12 and 12 respectively, and the bases can be represented as *A-i*, B-*j,* C-*k* and D-*h* (i=1,2,…,11, j=1,2,…,11, k=1,2,…,12, h=1,2,…,12). The total number of furnaces or coolers is 23, and they can be represented by CH-*pq* or *HH-pq*, where *HH* represents the furnace, *CH* represents the cooler, p=1,2,…,23, q indicates the base area (*A, B, C, D*). The numbers of crane and push car are respectively 2, and they are represented by *CR-m* and *PC-n*, where m=1,2 and n=1,2.

Due to the large number of the practical bases, this paper only presents some of the annealing scheduling results on a few of bases during the Morning Shift of one day (see Table 1), in which the annealing operation guidance of each base is given. For example, to an annealing plan (No. 1048233) on the No. 2 base of A area, based on the operation guidance in the table, the worker can unload the *HH-3A* furnace of this base with *C-1A* crane at 6:12 in the morning, and load the cooler on it with *C-2A* at once. The two processes take 8 minutes, and at 6:20 the annealing plan on this base begins to be cooled. Therefore, under such operation guidance, each annealing recipe on all bases can be processed in order, which avoids blocking a series of annealing procedures and makes full use of the resources. Table 2 shows the comparison of simulation results between the method presented in this paper and the scheduling method proposed by Liu *et al.* (2005). The comparison indicates that with the increase of the annealing plan amount the simulation speed by the presented approach presented is evidently higher than that of Liu *et al.* (2005) using object-oriented discrete event simulation. In virtue of the reduction of simulation time, we extend the searching scope of the annealing order; therefore, a better scheduling solution can be obtained. When 10 annealing plans are involved in a batch scheduling, the furnace utilization ratio obtained by the presented model is increased by 7.6%, and the total annealing time is reduced by 41.2 hours than Liu *et al.* (2005). So, all the annealing plans can be well annealed by the solution obtained by the simulation system in this paper with a higher resource utilization ratio and a shorter annealing time.

Table 1 The annealing operation guidance of some of annealing plans

Operation time	Base	Operation	Annealing plan	Furnace	Cooler	Crane	Push car
09-06 06:12	*A-02*	unloading furnace	1048233	*HH-3A*		*CR-1*	
09-06 06:12	*A-02*	loading cooler	1048233		*CH-6A*	*CR-2*	
09-06 06:20	*A-02*	cooling coils	1048233		*CH-6A*		
09-06 07:48	*A-03*	post-purge	1048201		*CH-7A*		
09-06 08:18	*A-03*	unloading cooler	1048201		*CH-7A*	*CR-2*	
09-06 08:26	*A-03*	unloading coils	1048201			*CR-1*	*PC-1*
09-06 12:17	*A-04*	unloading furnace	1048235	*HH-14A*		*CR-1*	
09-06 12:17	*A-04*	loading cooler	1048235		*CH-16A*	*CR-2*	
09-06 12:25	*A-04*	cooling coils	1048235		*CH-16A*		
09-06 13:56	*A-08*	unloading furnace	1048238	*HH-12A*		*CR-1*	
09-06 13:56	*A-08*	loading cooler	1048238		*CH-11A*	*CR-2*	
09-06 07:59	*A-10*	unloading furnace	1048240	*HH-19A*		*CR-1*	
09-06 07:59	*A-10*	loading cooler	1048240		*CH-19A*	*CR-2*	
09-06 08:07	*A-10*	cooling coils	1048240		*CH-19A*		
09-06 09:41	*A-11*	unloading furnace	1048246	*HH-21A*	*CH-23A*	*CR-1*	
09-06 09:41	*A-11*	loading cooler	1048246		*CH-23A*	*CR-2*	
09-06 09:49	*A-11*	cooling coils	1048246				
09-06 13:31	*B-02*	post-purge	1048225		*CH-5B*		
09-06 08:18	*B-04*	unloading furnace	1048220	*HH- 2B*		*CR-1*	
09-06 08:18	*B-04*	loading cooler	1048220		*CH-7B*	*CR-2*	
09-06 08:26	*B-04*	cooling coils	1048220		*CH-7B*		
09-06 14:00	*B-06*	post-purge	1048184		*CH-12B*		
09-06 12:58	*B-07*	unloading furnace	1048248	*HH- 18B*		*CR-1*	
09-06 12:58	*B-07*	loading cooler	1048248		*CH-9B*	*CR-2*	
09-06 13:06	*B-07*	cooling coils	1048248		*CH-9B*		
09-06 07:35	*B-08*	unloading furnace	1048224	*HH- 11B*		*CR-1*	
09-06 07:43	*B-08*	loading cooler	1048224		*CH-8B*	*CR-1*	

Table 2 The comparison of simulation result

Number of annealing plan	Simulation time(s)		Furnace utilization ratio		Total annealing time(h)	
	Method of this paper	Method of Liu *et al.*(2005)	Method of this paper	Method of Liu *et al.*(2005)	Method of this paper	Method of Liu *et al.*(2005)
5	8	20	94.5907%	92.2762%	51.33	57.52
10	10	30	87.7299%	80.1422%	54.00	95.20
15	15	45	85.4495%	81.8485%	73.87	86.80
20	20	60	85.2056%	79.1594%	75.23	102.08

4. CONCLUSIONS

The Timed Petri net can conveniently describe the dynamic characteristcs of DEDS. In this study, through analyzing the technical flow of bell-type batch annealing in cold rolling plant, the Timed Petri net model of annealing process scheduling in the shop is presented, which describes the whole annealing process practically. Moreover, this paper analyses the features of SystemC platform in simulating the DEDS, constructes the mapping relationship between the Timed Petri net and SystemC platform, and implements the model with SystemC according to the proposed mapping rules. The simulation results show that this simulation model can complete the whole annealing scheduling process at a high speed, and obtains the a better scheduling result with shorter total annealing time and higher resources utilization. In addition, the worker can work in order according to the detailed operation guidance provied by the simulation model. This method combined the Petri net modelling and SystemC simulation can be used to model and simulate other similar production system.

ACKNOWLEDGEMENTS

This research is supported by the National 863 Project (2007AA04Z156) and the National Natural Science Foundation of China (60604026). All of these supports are appreciated.

REFERENCES

Armando Walter Colombo, Ricardo Carelli and Benjamin Kuchen (1997). A temporised Petri net approach for design, modelling and analysis of flexible production systems. *The International Journal of Advanced Manufacturing Technology*, **vol.13**, **No.3**, pp.214-226.

Bhasker, J. (2002). A SystemC Primer. Star Galaxy Publisher.

Desrochers, A.A. and Al-Jaar, R.Y. (1995). Applications of Petri nets in manufacturing systems. IEEE Press, New York.

Deepankar Pal a, Amlan Datta a, Satyam S, *et al.*(2006). An Efficient Model for Batch Annealing Using a Neural Network.Materials and Manufacturing Processes, **vol. 21, No.5,** pp.567 – 572.

Doo Yong Lee and DiCesare, F.(1994). Scheduling flexible manufacturing systems using Petri nets and heuristic search. IEEE Transactions on Robotics and Automation, **vol.10**, **No.2**, pp.123-132.

David, R. and Alla, H. (1994). Petri Nets for Modeling of Dynamic-Systems: A Survey. *Automatica*,**vol.30**, **No.2**, pp.175-202.

Dimitris Kiritsis and Michel Porchet (1996). A generic Petri net model for dynamic process planning and sequence optimization. *Advances in Engineering Software*, **vol.25**, **No.1**, pp.61-71.

Holger Vogt (2003). Discrete-Event Simulation Using SystemC: Interactive Semiconductor Factory Modeling with FABSIM. Proceedings of the 2003 Winter Simulation Conference, **vol.2**, pp. 1383-1387.

Jerry Banks, *et al.* (2005). Discrete-event System Simulation. Machine Press, China.

Liu, Q.L., Wang, W., *et al.* (2005). Optmal Scheduling Method for a Bell-type Batch Annealing and Its Application. *Control Engineering Practice,* **vol.13**, pp.1315-1325.

Moody, J.O. and Antsaklis, P. J. (1998). Supervisory Control of Discrete Event Systems using Petri Nets.Kluwer Academic Publishers.

Moon, S. and Hrymakn, A.N. (1999). Scheduling of the Batch Annealing Process-deterministic case. *Computers and Chemical Engineering*, **vol.23**, **No.9**, pp.1193-1208.

Venkatesh, N., Moodie, C.L. and Ravindran, A. (1983). Production Scheduling of Annealing Furnaces. Purdue Lab.for Applied Industrial Control.

Wu, T. and Chen, R.Q. (2006). Heuristic Algorithm of Scheduling Model for Cold Coils Annealing. *Journal of Huazhong Unv. of Sci.& Tech,* **vol.34**, **No.9**, pp.58-60.

Wen, C.H. (1983). Computer Application in Batch Annealing Furnace Process Control. *Improved Productivity and Automation of Iron and Steel Industry*, Kaohsiung, Taiwan.

Y, C.Y. (1998). Theory of Petri nets. Electronic Industry Press, China.

Yang Kyu Lee and Sung Joo Park(1993). OPNets: an object-oriented high-level Petri net model for real-time system modeling. Journal of Systems and Software archive, **vol.20**, **No.1**, pp.69-86.

Zuberek, W.M. (1991). Timed Petri nets: definitions, properties and applications. *MicroElectronics and Reliability*, **vol.31**, **No.4**, pp. 627-644.

Zhou, M.C. and Venkatesh, K. (1998). Modeling, Simulation and Control of Flexible Manufacturing Systems: A Petri Net Approach. World Scientific, Singapore.