FLOW PROFILE PROPAGATION METHOD FOR MODELLING AND SIMULATION OF PRODUCTION WITH CONTINUOUS PROCESSES

Andrzej BOŻEK, Marian WYSOCKI, Tomasz ŻABIŃSKI

Summary: A new method of production modelling and simulation has been proposed. The method combines flexible job shop production environment with features of continuous processing. The solution takes into consideration changes of machines speed and continuous flow of material between consecutive operations. The special kind of a sequence constraint has been introduced that ensures processing continuity of every operation. The rules of construction of simulation models that use the flow profile propagation method have been presented. These models are represented by timed coloured Petri nets structures.

Keywords: production modelling, production scheduling, continuous production, flexible job shop, timed coloured Petri nets.

1. Introduction

In the chapter, a new method of production modelling and simulation has been proposed. The method can be also used for scheduling after integration with a suitable optimization procedure. The research has been motivated by observation of production in a screw factory for which a scheduling module is being developed. The production has job-operation organization. There is a set of production jobs and every job consists of arbitrary given sequence of operations. A set of alternative machines is defined for every operation and one of them has to be selected for processing the operation. This production environment can be classified as *flexible job shop* [1]. However, features of machines and operations are different from that typical for the standard flexible job shop. Machines have defined speed calendars that assign speed of processing to consecutive periods of time. Material flows continuously through processes of production routes [2, 3], thus time ranges of consecutive operations can overlap. The concept of a sequence constraint has to be redefined in comparison with the standard flexible job shop: a next operation can start at the moment when the inflow of material from a previous operation is sufficient for ensuring continuous processing of the next operation (because unplanned machine stops are unacceptable). The time function of material inflow from a previous operation as well as a speed calendar of the machine appointed for a next operation have to be taken into consideration for calculating of sequence constraints. It is done in the presented flow profile propagation method.

2. Flow profile

The concept of a *flow profile* is crucial to presented method. It can be thought as a formal object designed for representing flow of material in a some point of a production system in the function of time.

2.1. Flow profile definition

A *flow profile* is a linear-segment non-decreasing right-continuous function *fp*. This function will be represented by *flow profile sequence* (*FPS*) of pairs (*t*, *a*) that are consecutive end points of line segments

$$FSP = \left(\left(t_1, a_1\right), \left(t_2, a_2\right), \ldots, \left(t_n, a_n\right)\right) \tag{1}$$

where:

$$n \in \mathbf{N}, \quad n \geq 2, \quad a_1 = 0, \quad \neg \underset{3 \leq p \leq n}{\exists} t_{p-2} = t_{p-1} = t_p, \quad \neg \underset{3 \leq p \leq n}{\exists} a_{p-2} = a_{p-1} = a_p,$$

$$\underset{p,q \leq n}{\forall} (p \leq q) \Rightarrow \left[\left(t_p \leq t_q\right) \wedge \left(a_p \leq a_q\right)\right], \quad \underset{p,q \leq n}{\forall} (p \neq q) \Rightarrow \left[\left(t_p \neq t_q\right) \vee \left(a_p \neq a_q\right)\right].$$

The first value *t* in each pair represents time and the second value *a* represents related material amount. For a sequence *FPS* the additional parameters have been defined:

- ST_{FPS} – represents *start time* (that is t_1) of *FPS*,
- ET_{FPS} – represents *end time* (that is t_n) of *FPS*,
- TF_{FPS} – represents *total flow* (that is a_n) of *FPS*.

The time range [ST_{FPS}, ET_{FPS}] delimits the part of a flow profile that will be called an *active part*. Amount of material changes in the range of the active part, so the process that generates the flow has to be active. The part of flow profile delimited by the time range [0, ST_{FPS}) will be called *left passive part*. Value of a flow profile is 0 in the left passive part. The part of flow profile delimited by the time range (ET_{FPS}, +∞) will be called *right passive part*. Value of a flow profile remains TF_{FPS} in the right passive part.

The value of a flow profile function fp^{FPS}, represented by a given sequence *FPS*, can be calculated for any time $t \in [0, +\infty)$ with the use of the formula

$$fp^{FPS}(t) = \begin{cases} 0, & 0 \leq t < ST_{FPS} \\ \text{linear interpolation of } FPS, & ST_{FPS} \leq t \leq ET_{FPS} \\ TF_{FPS}, & ET_{FPS} < t \end{cases} \tag{2}$$

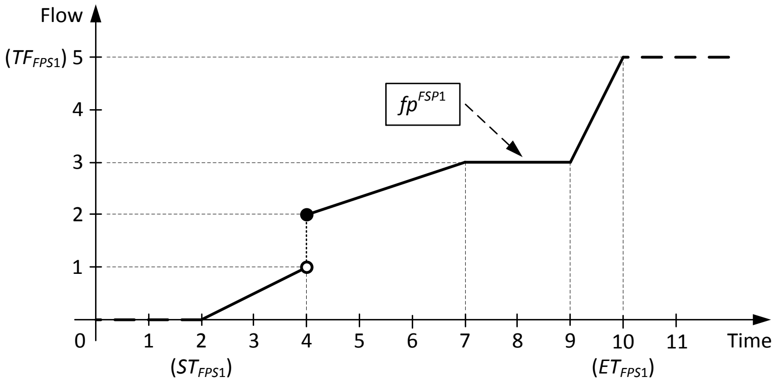

Fig. 1. An example flow profile

An example flow profile function, represented by sequence $FPS1$ = ((2, 0), (4, 1), (4, 2), (7, 3), (9, 3), (10, 5)) is shown in Figure 1. The passive parts of a flow profile are denoted by dashed line in Figure 1. The graphical representation of these parts will be omitted in next figures, only active parts will be presented.

2.2. Input and output flow profiles

The presented modelling method attributes two flow profiles to every continuous process in a production system: an input flow profile fp_{IN} and an output flow profile fp_{OUT} (Fig. 2). The input flow profile represents the material that inflows into the process. The output flow profile represents the material that outflows after processing. The difference $fp_{IN}(t) - fp_{OUT}(t)$ represents material amount that is being processing or waiting in a material buffer at time t.

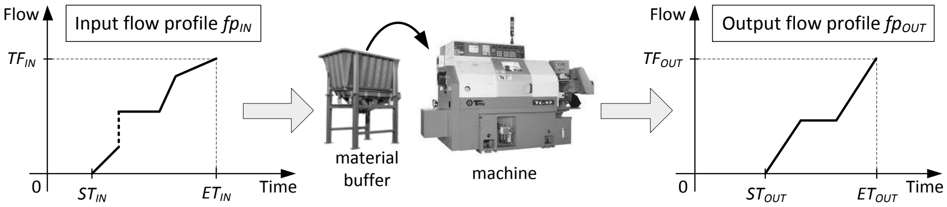

Fig. 2. Input and output flow profiles

The notation $fp_1 \rightarrow fp_2$ will be used if fp_1 is an input flow profile and fp_2 is an output flow profile of the same process. Two conditions have to be satisfied if $fp_1 \rightarrow fp_2$.

The first condition is that the total flow value of the both flow profiles should be the same ($TF_{fp1} = TF_{fp2}$). It is related to the assumption that the same material flows from the input to the output of the process. This may be not satisfied if the units of amount are different or/and if the material amount increases/decreases proportionally in the process. In such cases, normalization should be used to get input and output flow profiles with the same value of the total flow.

The second condition is that the flow profile fp_1 has to be *under* the flow profile fp_2. The notation $fp_1 \rhd fp_2$ (fp_1 is under fp_2) will be used for this relation that is defined as follows

$$fp_1 \rhd \mathrm{fp}_2 \Leftrightarrow \underset{t\in[0,+\infty)}{\forall} fp_1(t) \geq fp_2(t). \tag{3}$$

3. Flow profile propagation

Flow profile propagation is a method of modelling and simulation of production systems with continuous processes that makes use of the previously introduced concept of the flow profile. This method assumes that a machine environment and sequence relations between operations are defined similarly as for the well known *flexible job shop* problem [1]. However, the meaning of the term *operation* changes. An *operation* should be thought as an active part of an output flow profile of some process.

3.1. Method description

The flow profile propagation method can be described as follows:

1. There is a set of m machines $\{M_1, M_2, ..., M_m\}$.
2. A speed calendar SC_i is defined for each machine M_i. A speed calendar SC is defined as sequence of pairs $SC = ((t_1, v_1), (t_2, v_2), (t_n, v_n))$; in the range of time $[0, t_1]$ a machine has speed v_1, in the range of time $(t_1, t_2]$ the speed is v_2, and so on.
3. There is a set of j jobs$\{J_1, J_2, ..., J_j\}$.
4. A sequence of operations is defined $(O_{1,k}, O_{2,k}, ..., O_{o(k),k})$ for each job J_k.
5. A set of machines is defined $\{\mathrm{M}_{1,n}, \mathrm{M}_{2,n}, ..., \mathrm{M}_{m(n),n}\}$ for each operation O_n that can alternatively execute the operation.
6. An input flow profile $fp_{IN,k}$ is defined arbitrary for the first operation of each job J_k.
7. The goal of the simulation/scheduling is to select a machine and to find an output flow profile for each operation.
8. An output flow profile of every operation must satisfy the following restrictions:
   a) Every machine can execute only one operation at the same time, so only one active part of an output flow profile can be linked with machine at the same time.
   b) An input flow fp_{IN} and an output flow fp_{OUT} profiles of the same operation have to satisfy the condition $fp_{IN} \to fp_{OUT}$.
   c) Additional restrictions can be added for limiting the space of results.
9. An output flow profile of every operation is determined on the basis of the value of its start time (ST_{fpOUT}) and the speed calendar of the machine that processes the operation.
10. An output flow profile of an operation is also the input flow profile of the next operation in the production route.

The characteristic feature of the method is that consecutive flow profiles of operations are in a sense propagated one by one through production routes (points 8b and 10 above). This is the reason why the method has been named as *flow profile propagation*.

3.2. Output flow profile generation

An output flow profile fp_{OUT} can be generated unambiguously if three parameters are given: an input flow profile fp_{IN} related to fp_{OUT}, a speed calendar SC of the machine that processes the operation defined by fp_{OUT} and a arbitrary defined start time of fp_{OUT} (ST_{fpOUT}). Let $SC(t)$ denote the value of the machine speed in the moment t. The function fp_{OUT} is defined by the integrate

$$fp_{OUT}(t) = \int_{ST_{fpOUT}}^{t} SC(\tau)d\tau, \;\; ST_{fpOUT} \leq t \leq ET_{fpOUT}. \tag{4}$$

The end time of the generated flow profile (ET_{fpOUT}) has to be chosen to satisfy the condition $TF_{fpIN} = TF_{fpOUT}$. Formally, it can be defined as the upper integration limit that satisfies the relation

$$TF_{fpIN} = \int_{ST_{fpOUT}}^{ET_{fpOUT}} SC(\tau)d\tau. \tag{5}$$

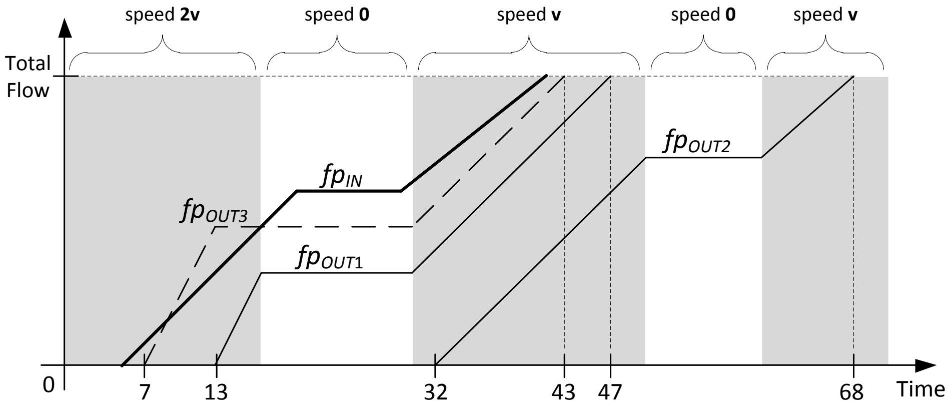

Fig. 3. Generation of an output flow profile

The algorithm that generates flow profile sequence FPS_{OUT} that represents the function fp_{OUT} defined by relation (4) is quite simple and it will not be presented here in details.

There are two important observations:

1. The length of the active part of the fp_{OUT} (so, the length of the operation) is not constant. It depends on ST_{fpOUT}. For example (Fig. 3), the active part of fp_{OUT1} has the length 34 (from 13 to 47) and the active part of fp_{OUT2} has the length 36 (from 32 to 68). It is different than for the standard flexible job shop where the length of every operation is constant and defined arbitrary.
2. The formulas (4) and (5) do not guarantee the relation $fp_{IN} \rhd fp_{OUT}$ between flow profiles. For example (Fig. 3), the profile fp_{OUT3} does not satisfy this relation. The time ST_{fpOUT} of an output flow profile has to be not less than *minimal start time* (see next point 3.3) to satisfy the condition $fp_{IN} \rhd fp_{OUT}$.

3.3. Minimal start time of output flow profile

A start time of an output flow profile of every operation is lower bounded. It cannot be smaller than a minimal value $MIN\_ST_{fpOUT}(fp_{IN}, SC)$ that is determined by the input flow profile fp_{IN} of the operation and the speed calendar *SC* of the machine that executes the operation. If the start time of the output flow profile were smaller, the condition $fp_{IN} \rhd fp_{OUT}$ (and effectively the condition $fp_{IN} \rightarrow fp_{OUT}$) would not be satisfied. Thus, the parameter $MIN\_ST_{fpOUT}(fp_{IN}, SC)$ represents sequence constraints in the flow profile propagation method. The algorithm has been designed that returns the value of this parameter on the basis of given fp_{IN} and *SC*. The algorithm has two parts (*the forward part* and *the backward part*). The operation of the algorithm is presented on the example (Fig. 4, 5).

The forward part of the algorithm (Fig. 4):

1. Divide the active part of the fp_{IN} into sections. A section is the range of the active part with a constant value of machine speed. There are three sections in the example: section I – speed *v*, section II – speed 0 (machine stopped) and section III – speed *v* again.
2. Select the first section (Fig. 4, section I).
3. Construct the set *SP* (*Set of Points*) of points that are bend points of fp_{IN} in the selected section (Fig. 4: section I – point A_1; section II – no points; section III – point C_4).

4. Add to the set *SP* the extreme left and right points of fp_{IN} in the selected section (Fig. 4: section I – points A_2, A_3; section II – points B_1, B_2; section III – points C_1, C_2).
5. If the selected section is not the first section add to the set *SP* the recently saved point *IP* (*intermediate point*).
6. For each point $p \in SP$ construct a line segment that includes point *p* and has slope equal of the machine speed (Fig. 4: section I – a_1, a_2, a_3; section II – b_1, b_2, b_3; section III – c_1, c_2, c_3, c_4). Construct the set *SLS* (*Set of Line Segments*) and add to this set the constructed line segments.
7. Select the line segment $ls^* \in SLS$ that is the lowest in the section (Fig. 4: section I – a_3; section II – b_3; section III – c_4).
8. If the current section is last go to the step 11 else go to the step 9.
9. Save the right end point of the line segment ls^* to the variable *IP* (*intermediate point*) (Fig. 4: section I – point B_3; section II – point C_3).
10. Select the next section (on the right) and go to the step 3.
11. Save the right end point of the line segment ls^* to the variable *FINAL POINT*.

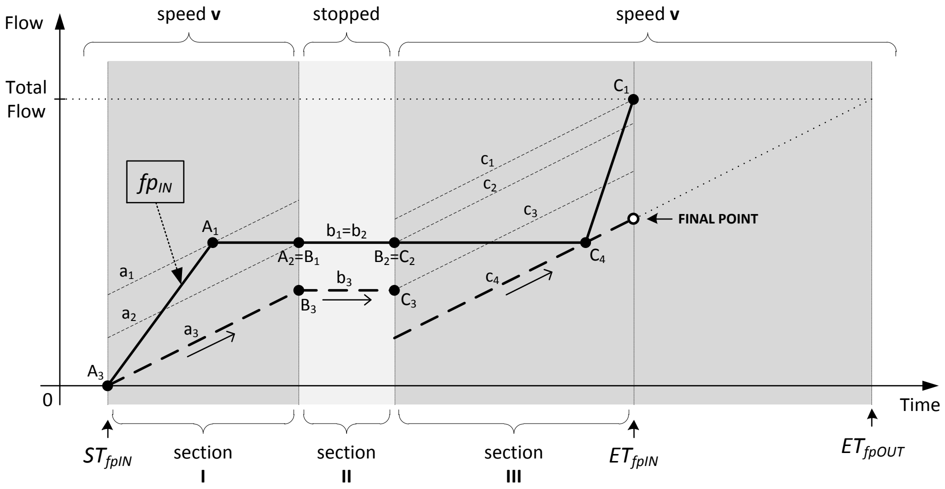

Fig. 4. Calculation the MIN_ST_{fpOUT} parameter, the forward part of the algorithm

The backward part of the algorithm (Fig. 5):

1. Select the last section (Fig. 5: section III).
2. Save *FINAL POINT* to the variable *RE* (*Right End*).
3. Construct a line segment ls^{**} that has slope equal of the machine speed and includes the recently saved point *RE* (Fig. 5: section III – f1; section II – f2; section I – f3).
4. If the line segment ls^{**} intersects the time axis in the selected section (Fig. 5: section I) go to the step 7, else go to the step 5.
5. Save the left end point of the line segment ls^{**} to the variable *RE* (Fig. 5: section III – point F_1; section II – point F_2).
6. Select the previous section (on the left) and go to the step 3.
7. The time coordinate of intersection of the line segment ls^{**} and the time axis is the value of $MIN\_ST_{fpOUT}(fp_{IN}, SC)$. Return the result and stop.

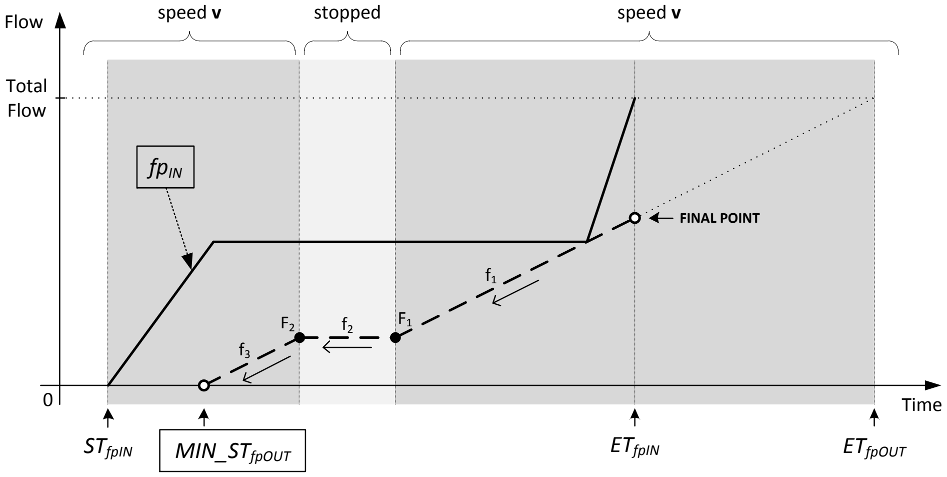

Fig. 5. Calculation of the MIN_ST_{fpOUT} parameter, the backward part of the algorithm

4. Modelling and simulation of production system

The formalism of timed coloured Petri nets and *CPN Tools* [4,5] software have been used for construction of simulation model of production system. Some aspects of modelling, simulation and scheduling of production systems with the use of Petri nets has been presented in the works [6,7] in more details.

There are a few advantages of using timed coloured Petri nets formalism for modelling of production systems:

1. Sequence relations and constraints that are present between operations can be easily modelled by the structure of a Petri net.
2. Time synchronisation of concurrent events is ensured by the formalism.
3. It is simple to combine many modelling method (for example, the flow profile propagation method and others) in one model.

The net substructure that represents one operation is the basic structural component of the model. It is shown in Figure 6. The shown substructure relates to the operation *OpX* that can be alternatively processed by the machines *M*1, *M*2, ..., *Mm*.

There are three types (colour sets) defined for places:

– *SP_CAL* (*SPeed CALendar*) – it represents a speed calendar of a machine in the form defined in the point 3.1.

– *FL_PROF* (*FLow PROFile*) – it represents a flow profile in the form of a flow profile sequence (*FSP*) defined in the point 2.1.

– *TIM_LIM* (*TIMe LIMit*) – it is timed type [4] that represents time restrictions in the model. A token of this type cannot be consumed by transition as long as its timestamp is greater than value of the *global clock* [4].

The transition *prep_OpX* (preparation for operation *X*) is fired if only a token with an input flow profile, bound to the variable *flProf*, is available. While this transition is fired the two actions are executed:

1. The token with the input flow definition is moved to the place *wait_OpX*.
2. For each machine the function *minimalDelay* is executed. This function calculates the difference between the value of $MIN\_ST_{fpOUT}(fp_{IN}, SC)$ and the value of the global clock.

The calculation is done on the basis of the algorithm described in the point 3.3. The parameter fp_{IN} is defined by the variable *flProf* and the parameter *SC* is defined by the variable *mCal*1 for the machine *M*1, the variable *mCal*2 for the machine *M*2, and so on. The tokens generated by the expressions 1`()@+ *minimalDelay*(*flProf*, *mCal*i) are inserted into the places *lim_M*i*_OpX* (i = 1, 2, ..., m). The timestamps of this tokens represent MIN_ST_{fpOUT} parameters.

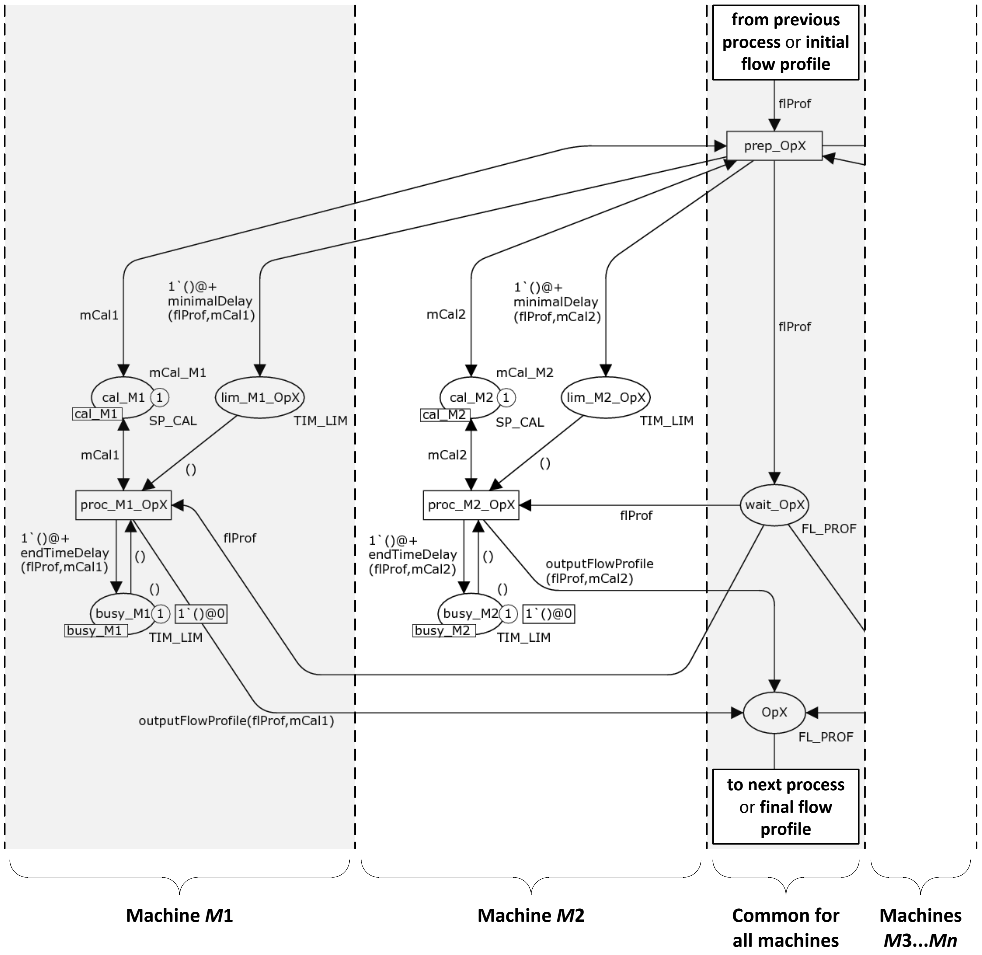

Fig. 6. The Petri net model substructure that represents one operation

The example state of the substructure with two machines (*M*1, *M*2) after execution of the transition *prep_OpX* is shown in Figure 7. The input flow profile has been moved to the place *wait_OpX*. This profile is represented by the sequence ((0, 0), (1150, 1533), (1650, 1700)). The MIN_ST_{fpOUT} parameter is 650 and 150 for the machines *M*1 and *M*2 respectively. The places *busy_M*1 and *busy_M*2 (one place for every machine, common for all operations that the machine can execute) represent reservation of machines by operations. So, the machine *M*1 has not be reserved so far and the machine *M*2 is reserved until the time 1927 by an operation other than *OpX*.

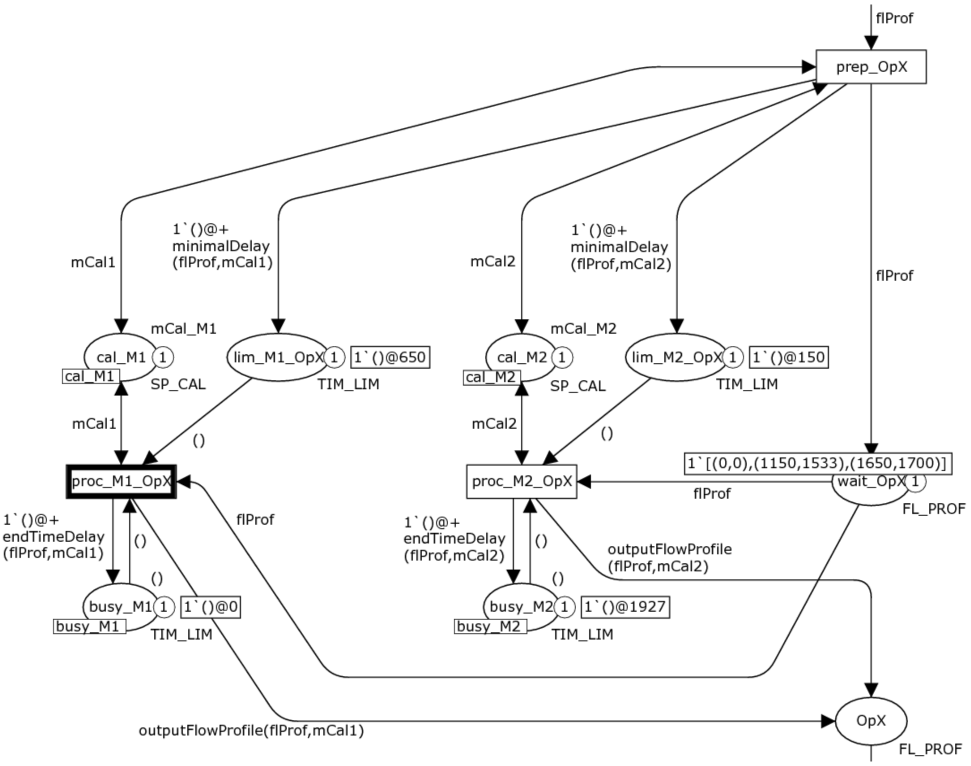

Fig. 7. Execution of the Petri net model – the state after execution of the transition *prep_OpX*

Marking of places *lim_M*i*_OpX* represents sequence constraints and marking of places *busy_M*i represents resource constraints in the modelled production system. A transition *proc_M*i*_OpX* becomes ready for execution if three conditions are satisfied simultaneously:

1. The value of global clock (that represents the current time) is not less than the timestamp of the token in the place *lim_M*i*_OpX* (the sequence constraint is not active).
2. The value of global is not less than the timestamp of the token in the place *busy_M*i (the resource constraint is not active).
3. A token is present in the place *wait_OpX* (an operation waits for processing).

The transition *proc_M*1*_OpX* will ready first, at time 650 (Fig. 7). While transition *proc_M*i*_OpX* is fired the three actions are executed:

1. Tokens are removed from the places *lim_M*i*_OpX*, *busy_M*i and *wait_OpX*. No other machine can start the same operation, because the place *wait_OpX* is common for all machines.
2. The function *outputFlowProfile* generates the output flow profile of the operation on the basis of relations (4) and (5). The current value of global clock is used as the parameter ST_{fpOUT}. The token that represents generated profile is inserted into the place *OpX*.
3. The token with the timestamp that equals ET_{fpOUT} is inserted to the place *busy_M*i. The needed calculation is done by the function *endTimeDelay*. This token blocks availability of the machine *M*i for other operations during the active part of the output flow profile of the operation *OpX*.

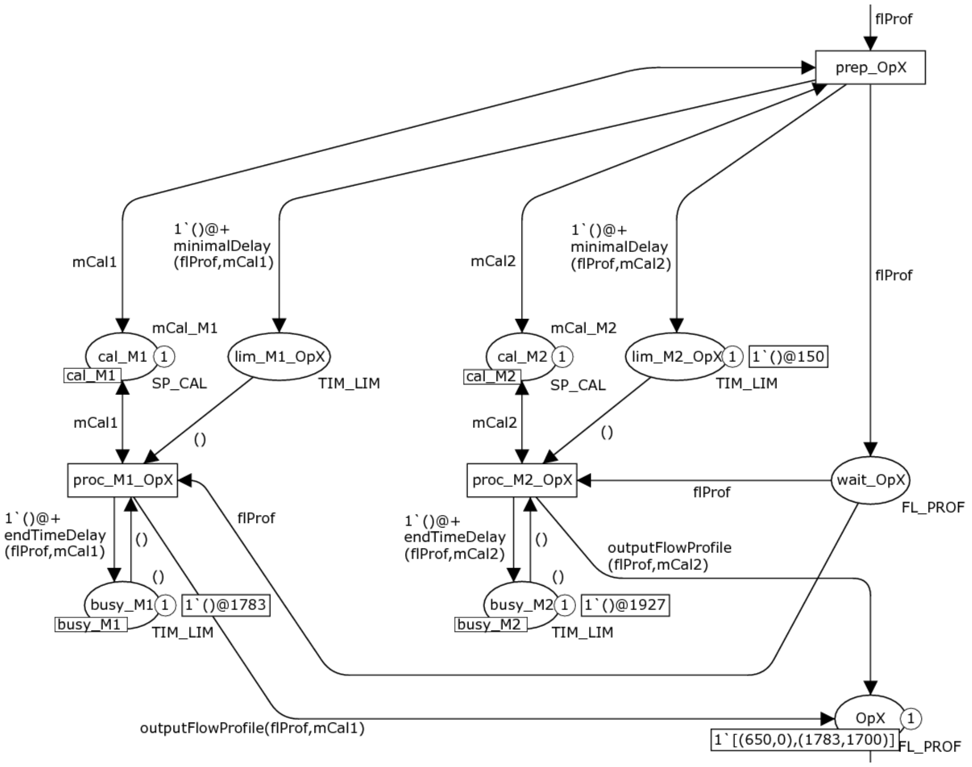

Fig. 8. Execution of the Petri net model – the state after execution of the transition *proc_M*1*_OpX*

The state of the substructure from Figure 7 after the execution of the transition *proc_M*1*_OpX* is shown in Figure 8. The generated output flow profile is represented by the sequence ((650, 0), (1783, 1700)). It is accessible for a next operation as the input flow profile or it finishes the job if it is the last operation of the job. The timestamp of the token in the place *busy_M*1 that equals 1783 blocks availability of the machine *M*1 during the active part of the generated output flow profile.

Every flexible job shop structure can be modelled by suitable connection of the basic substructures from Figure 6, one substructure for each operation. Of course, in a target application, the structure should be generated automatically and simulated by a dedicated algorithm.

5. An example

An example production problem with continuous processes has been defined for demonstration the method of flow profile propagation. There are 3 jobs *Job*1...*Job*3 and 6 machines *M*1...*M*6. Every job has 3 operations. Every operation can be processed by 2 machine alternatively. The sequence relations are presented in Table 1. It is mandatory to arbitrary define the input flow profile for the first operation of every job. In the example, a very simple pattern of that input flow profile has been used. These profiles have a form of a vertical line segment represented by the sequence FSP_{IN} = ((0, 0), (0, *totalAmount*)). It means that all amount of material (*totalAmount*) is accessible for processing in the first operations of jobs immediately when the simulation starts. The values of *totalAmount* that have been chosen: 2000 for *Job*1, 1800 for *Job*2 and 1700 for *Job*3.

Tab. 1. Sequence relations of the example production problem

	Operation 1	**Operation 2**	**Operation 3**
Job 1	M1 or M2	M3 or M4	M5 or M6
Job 2	M3 or M5	M1 or M6	M2 or M4
Job 3	M4 or M6	M2 or M5	M1 or M3

Speed calendars are defined as shown in Figure 9. Every machine has different speed. Every machine is available for working with the defined speed in some ranges of time and it is stopped (speed is 0) in other ranges of time.

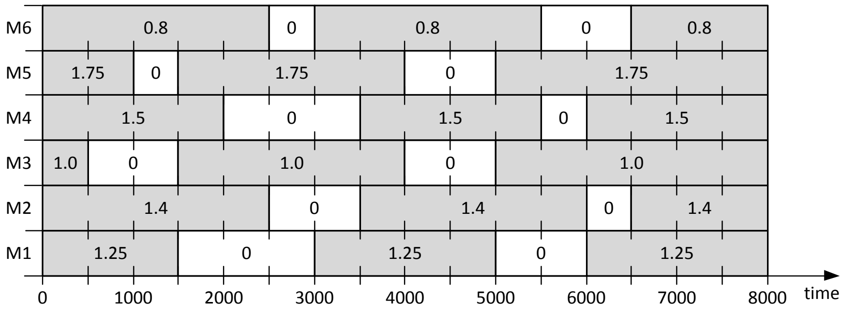

Fig. 9. The calendar of machines speed

A simulation model of the given flexible job shop problem has been prepared using *CPN Tools* software. The rules of modelling presented in the subchapter 4 have been used with one modification. The condition $fp_{IN} \vartriangleright fp_{OUT}$ in the function *minimalDelay* has been changed to the condition $fp_{IN}^{*} \vartriangleright fp_{OUT}$, where fp_{IN}^{*} is the input flow profile fp_{IN} shifted down by the value *minimalBuffer*. It causes that the difference $fp_{IN}(t) - fp_{OUT}(t)$ has at least the value *minimalBuffer*, so the material buffer is ensured. The value of *minimalBuffer* has been set to 200 for the simulation.

An example result of the simulation is presented in Figure 10 in the form of the output flow profiles plot. The first operations of every job have no active constraints and they start immediately at the time 0. It is easy to notice that sequence constraints are active for the operations (Job1, M5), (Job2, M6), (Job2, M2), (Job3, M2), and resource constraints are active for the operations (Job1, M4), (Job3, M3).

The detailed information about shapes of the output flow profiles can be dropped and the simulation result can be also presented in the form of the standard Gantt chart (Fig. 11).

6. Conclusion

The presented method combines flexible job shop production environment with features of continuous processing. It is dedicated as an element of the developed scheduling module for the screw factory "Gaweł Zakład Produkcji Śrub Sp. z o.o.". This module is a part of the project realized by Department of Computer and Control Engineering in the cooperation with the Green Forge Innovation Cluster [8].

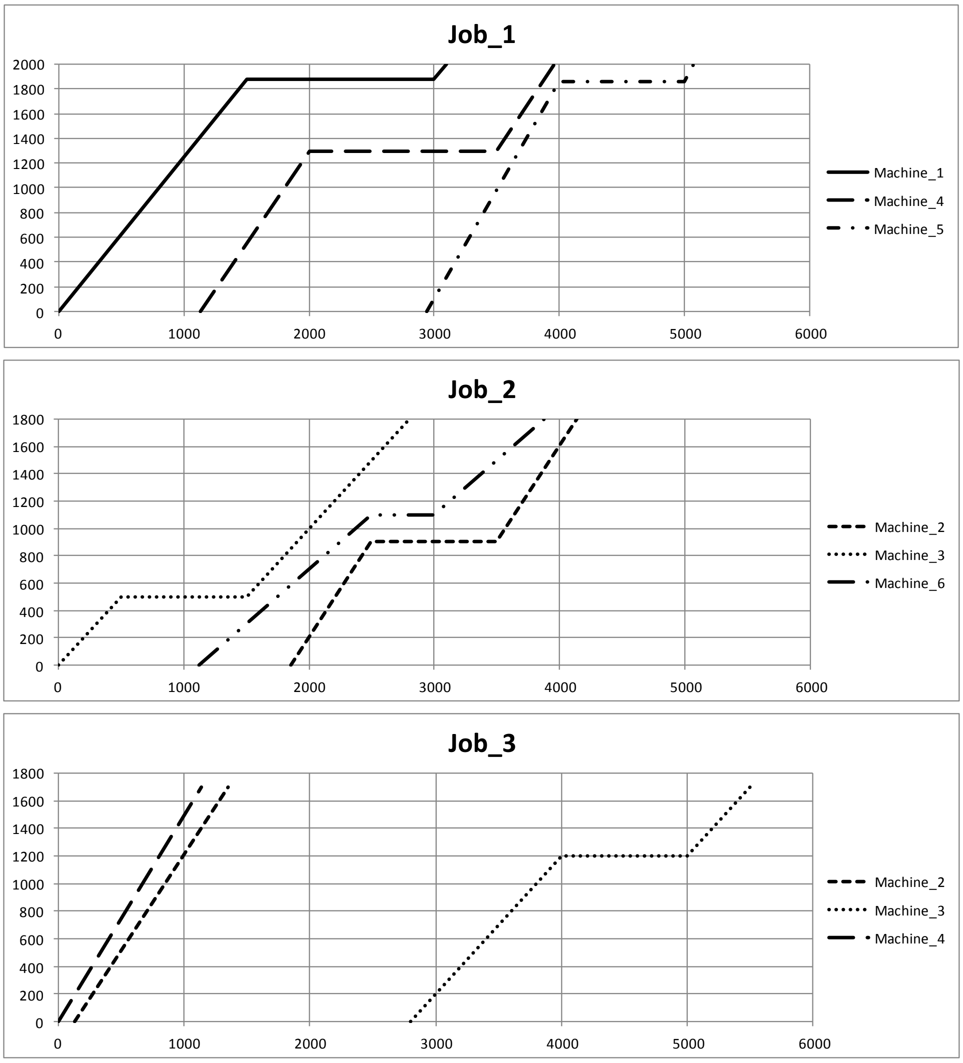

Fig. 10. The result of the production system simulation – the output flow profiles plot

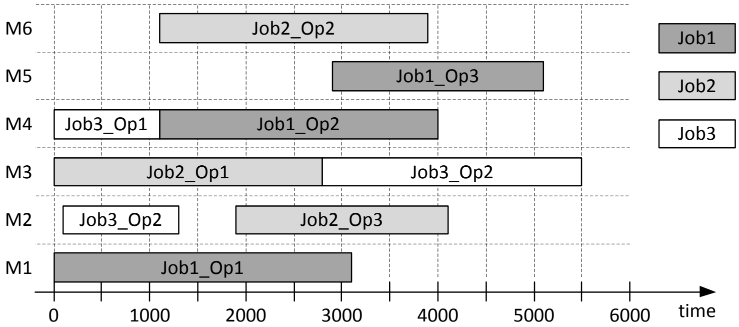

Fig. 11. The result of the production system simulation – the Gantt chart

The method will have to be integrated with other elements characteristic of production in the factory, for example: setup times, transport times, transfer batching, restriction of resources (tools, material, containers, etc.). It is planned to use the formalism of timed coloured Petri nets for the integration. The presented implementation of the method should be easy for integration with other discrete event systems, because the implementation has the discrete-like form – every continuous process is represented by a discrete event of its output flow profile generation.

The method can be adapted to other production systems. First of all, mass production of different fastener elements, steel elements and other small parts can have similar organization. It is also possible that the method will be usable for production systems with loose materials and liquid substances.

Acknowledgment

Computer hardware used for the research was bought as a part of the project No POPW.01.03.00-18-012/09 from the Structural Funds, The Development of Eastern Poland Operational Programme co-financed by the European Union, the European Regional Development Fund.

References

1. Błażewicz J., Ecker K.H., Pesch E., Schmidt G., Węglarz J.: Scheduling Computer and Manufacturing Processes, 2nd ed., Springer 2001.
2. Kallrath J.: Planning and scheduling in the process industry. OR Spectrum, 24, 2002, 219-250.
3. Neumann K., Schwindt C., Trautmann N.: Scheduling of continuous and discontinuous material flows with intermediate storage restrictions. European Journal of Operational Research, 165, 2, 2005, 495-509.
4. Jensen K., Kristensen L. M.: Coloured Petri Nets. Modeling and Validation of Concurrent Systems, Springer 2009.
5. CPN Tools Home Page, http://cpntools.org/
6. Tuncel G., Bayhan G. M.: Applications of Petri nets in production scheduling: a review. Int. J. Adv. Manuf. Technol. Springer-Verlag, 34, 7-8, 2007, 762-773.
7. Bożek A., Żabiński T.: Colored timed Petri Nets as a tool of off-line simulating for intelligent manufacturing systems. Electrical Review, 9/2010, 101-105.
8. Żabiński T., Mączka T., Jędrzejec B.: Control and Monitoring System for Intelligent Manufacturing – Hardware and Communication Software Structure, 7th Conference on Computer Methods and Systems, Cracow, Poland, 2009, 135-140.

MSc Eng. Andrzej BOŻEK
PhD DSC, Ass. Prof. Marian WYSOCKI
PhD Eng.Tomasz ŻABIŃSKI
Department of Computer and Control Engineering
Rzeszow University of Technology
35-959 Rzeszow, Powstancow Warszawy 12
tel.: (0-17) 865 12 25
fax: (0-17) 854 29 10
e-mail: abozek@kia.prz.edu.pl, mwysocki@kia.prz.edu.pl, tomz@kia.prz.edu.pl