SMART PRODUCTION SYSTEMS: ARCHITECTURE AND APPLICATION TO MULTI-JOB MANUFACTURING

Pooya ALAVIAN
University of Michigan, Ann Arbor, MI, USA, alavian@umich.edu

Peter DENNO
National Institute of Standards and Technology, Gaithersburg, MD, USA, peter.denno@nist.gov

Semyon M. MEERKOV
University of Michigan, Ann Arbor, MI, USA, smm@umich.edu

Smart Production Systems (SPS) are manufacturing systems capable of self-diagnosing and providing the operation management with an advice concerning performance improvements, along with rigorously quantified outcomes of their potential deployment. This paper presents a version of SPS architecture and illustrates its application to a class of widely used flexible manufacturing systems referred to as Multi-Job Production.

Key words: Smart production systems; Self-diagnostics; Multi-job production; Work-based model; Product-mix performance portrait;

1. Introduction

Smart Production Systems (SPS) are manufacturing systems capable of self-diagnosing and providing the operation management with an advice concerning performance improvements, along with rigorously quantified outcomes of their potential deployment. Numerous versions of SPS architecture are possible. The one developed here is based on a supposition that to be "smart", a production system should be equipped with an Artificial Intelligence Block (AIB) consisting of the following three units:

- information unit (IU)
- analytics unit (AU)
- optimization unit (OU).

The purpose of IU is to acquire information concerning the current status of the equipment and provide it to AU and OU. The required information typically entails machines' average up- and down-time, cycle time, probabilities of blockages and starvations, average buffer occupancy, etc., along with the system throughput. In many cases, modern manufacturing plants do have some kind of Production Monitoring Systems, which collect the data from a huge number of PLCs and other production monitoring sensors. Experience shows, however, that these data may or may not contain the information necessary for the two subsequent AIB units. Even if the required information is available, it practically always needs "cleaning-up" to be appropriate for utilization. In fact, in every project, which we have carried out on a factory floor during the last 30 years, the most time and efforts have been devoted to collecting the data and cleaning it up, so that it would be usable for system diagnostics and design of continuous improvement projects. Thus, developing and implementing an effective IU is an extremely important and time-consuming problem. Unfortunately, no rigorous methods for solving this problem are available, and IUs must be developed "manually" by the SPS developers in cooperation with plant personnel (preferably, involved in operating the Production Monitoring Systems). Without an effective IU, no SPS will be successful, no matter how powerful methods are used in AU and OU.

The purpose of AU is to investigate the system performance for any given set of machine and buffer parameters (i.e., calculate the average throughput, *TP*, average work-in-process, *WIP*, and probabilities of blockages, *BL*, and starvations, *ST*), diagnose its health (e.g., production losses), identify its bottlenecks (BNs), and quantify the effects of various "what if" scenarios of continuous improvement. To accomplish this, AU must be equipped with a mathematical model of the system at hand, which may be constructed and validated off-line, but continuously updated on-line, based on the information provided by IU. In addition, AU must contain methods for performance analysis and BN identification in the available system model. In most cases, these methods can be found in the rich theory of stochastic production systems analysis and design (see, for instance, Viswanadham and Narahari (1992), Askin and Standridge (1993), Buzacott and Shanthikumar (1993), Papadopoulos et al. (1993), Gershwin (1994), Altiok (1997), Papadopoulos et al. (2009), Curry and Feldman (2009), and Li and Meerkov (2009)). So, designing AUs is, mostly, a matter of selecting a right method for the system at hand or, in some cases, extending the existing methods to a new class of systems (see, e.g., Section 2).

The purpose of OU is to provide an optimal improvement plan. This is based on the results obtained in AU as well as the models of internal structure of bottleneck operations. For instance, while AU provides the identified bottleneck operation and its desired cycle time reduction, OU uses the internal model of the bottleneck in order to find the critical path improvement steps and, therefore, an optimal path for the improvement project implementation. Typically, OU is based on optimization techniques, which are readily available in the filed of optimization (see, for instance, Bazaraa et al. (2009), Papadimitriou and Steiglitz (1998), Boyd and Vandenberghe (2004), and Ruszczynski (2006)).

This paper is intended to describe the development of AIB for a class of production systems, referred to as Multi-Job Production (MJP). Accordingly, the outline is as follows: Section 2 introduces MJP serial lines and work-based model to be used for their analysis. Section 3 presents a method for performance and bottleneck analysis in MJP serial lines. Using this method, Section 4 investigates quantitative and qualitative features of MJP serial lines as a function of their main characteristic feature – the product-mix. Section 5 presents and illustrates the AIB for MJP serial lines, developed using the results of Sections 3 and 4. Finally, the conclusions and the directions of future work are given in Section 6. Due to space limitations many details and all proofs are omitted and can be found in Alavian et al. (2016).

2. MJP Systems: Description, Definition, and Work-based Model

2.1. Description and definition

Multi-job production is a class of flexible manufacturing systems, intended to produce different products (or job-types) within the same production system. MJP is widely used in product assembly, e.g., in automotive assembly plants, engine and battery plants, computer and appliance assembly, etc.

To illustrate MJP operation, consider an automotive assembly plant manufacturing two car models, A and B. In each area of the plant, i.e., body shop, paint shop, and final assembly, each job-type follows the same sequence of manufacturing operations. Let $\boldsymbol{r} = (r_A, r_B)$ be the product-mix, where r_A is the fraction of automobiles A to be manufactured and $r_B = 1 - r_A$ is that of B. The jobs are released one-by-one into the body shop in a sequence defined by the product-mix and a corresponding build-schedule and then proceed to the paint shop and final assembly. For instance, a segment of a release sequence may be $\cdots BAAABABAAB \cdots$, generated by the product-mix with $r_A = 2/3$ and $r_B = 1/3$. The jobs are transported from one operation to another (typically, by conveyors, which serve also as buffers) in the sequence of the release. Each job-type is processed by the machines (operations or stations) with zero (or practically zero) setup time, but requires different processing time at some or all machines.

Based on the above, the class of MJP systems is defined as follows:

- The required system performance is specified in terms of the product-mix, which may be changing frequently (e.g., on a daily basis).

- The jobs are released one-by-one (without batching) according to the product-mix and a corresponding build-schedule.
- All jobs undergo identical sequence of manufacturing operations, but require different amount of work at some or all operations.
- The setup times are zero.
- In-process buffers are non-dedicated (i.e., store different job-types in the sequence of arrival).
- The jobs are processed on a first-come, first-served basis.
- The machines are unreliable and experience random breakdowns.
- The processing time of each machine is deterministic, but job-dependent.

Unfortunately, current literature does not offer methods for MJP systems performance analysis as a function of the product-mix. To develop such methods, the next subsection introduces a work-based model of serial lines. Based on this model, Section 3 presents the sought methods for performance and bottleneck analysis in MJP systems with exponential machines.

2.2. Work-based model of MJP serial lines

As mentioned above, the approach developed here is based on a novel work-based model, instead of the traditional part-based model, of production systems. This implies that, unlike the traditional approach, where the analyses are carried out in terms of 'parts produced', in this paper the research is carried out in terms of the 'work produced', which is insensitive to whether a single- or multi-job manufacturing takes place. Given the work produced, the throughput of each job-type and other performance metrics can be calculated using the product-mix.

More specifically, the work-based model is defined as follows:

(i) Each machine, $m_i, i = 1, \cdots, M$, is characterized by its work-capacity, W_i (in units of work/min).

(ii) Each job-type, $J_j, j = 1, \cdots, S$, is characterized by its work-requirements, $w_{ij}, i = 1, \cdots, M; j = 1, \cdots, S$ (in units of work/job-type), i.e., by the vector $\boldsymbol{w}_j = [w_{1j}, \cdots, w_{Mj}]$. The set-up times for each job-type are zero.

(iii) The jobs are released one-by-one (without batching) according to a given product-mix, $\boldsymbol{r} = [r_1, \cdots, r_S]$, $\sum_{j=1}^{S} r_j = 1$, where r_j is the fraction of job-type j to be manufactured; the release sequence is formed by releasing each job-type j with probability $r_j, j = 1, \cdots, S$.

(iv) The buffers are not dedicated.

While these features of the model are novel, the remaining ones follow standard conventions used in serial lines modeling and analysis (as, for instance, in Li and Meerkov (2009)):

(v) Machines are characterized by the breakdown and repair rates, λ_i and μ_i (in units of 1/min), respectively; this implies that the machines are exponential with the average up- and downtime given by $T_{up,i} = \frac{1}{\lambda_i}$ and $T_{down,i} = \frac{1}{\mu_i}$, and with machine efficiency $e_i = \frac{\mu_i}{\lambda_i + \mu_i}$.

(vi) The first machine is not starved and the last machine is not blocked.

(vii) Specific technical conventions, under which the MJP lines are analyzed in this paper, are:
- the machines obey the blocked-before-service assumption;
- the breakdowns are time-dependent;
- the flow model description is used;
- the job release also follows the flow model convention.

Discussion: (a) The machine work-capacity, W_i, is defined by the technological operation it carries out. For instance, for a welding operation the work-capacity is the number of welds it can carry out per unit of time; for turning, milling or drilling operations, it is the feed rate of a cutting instrument; for robotic or manual assembly, it is the number of assembly steps carried out per unit of time; etc. The units of work in job work-requirements, w_{ij}, are the same as in the corresponding machines (but in terms of work per job-type, rather than work per unit of time).

(b) As it follows from (i) and (ii), the time necessary to process a job-type j on machine i (i.e., the cycle time of machine i for processing job j) is

$$\tau_{ij} = \frac{w_{ij}}{W_i}, \quad i = 1, \cdots, M; j = 1, \cdots, S. \tag{1}$$

While in the part-based model the cycle time is an independent variable, (1) indicates that in the work-based model it is not: w_{ij} and W_i are the independent variables. This allows for investigating the effect of the job work-requirements on the system's throughput and bottleneck.

(c) Model (i)-(vii) can be used for analysis of single-job production (SJP) as well. In this case, $S=1$ and $w_{ij}=w_i$.

The performance metrics of production systems in the framework of work-based model (i)-(vii) are as follows: $TP_j, j=1,\cdots,S$ – the average number of jobs of type j produced by the last machine per unit of time; $TP=\sum_j TP_j$ – the average total number of jobs produced by the last machine per unit of time; $WIP_i, i=1,\cdots,M-1$ – the average number of jobs in buffer b_i; $ST_i, i=2,\cdots,M$ – the probability that machine m_i is starved; $BL_i, i=1,\cdots,M-1$ – the probability that machine m_i is blocked. The methods for evaluating these performance metrics and identification of bottlenecks as functions of the product-mix are described next.

3. Performance Analysis and Bottleneck Identification

3.1. Performance analysis

Consider an MJP serial line defined by assumptions (i)-(vii) and denote its performance characteristics as $TP=\sum_{j=1}^{S} TP_j$, WIP_i, ST_i, and BL_i. To evaluate these characteristics, the following three-stage procedure is introduced:

Stage I. Given the work-requirements $w_{ij}, i=1,\cdots,M, j=1,\cdots,S$, define the work-requirements of the virtual job at machine m_i as the average work imposed on m_i under the product-mix $[r_1,\cdots,r_S]$, i.e.,

$$w_{i,v} := \sum_{j=1}^{S} r_j w_{ij}. \tag{2}$$

Stage II. Consider the virtual SJP line consisting of the machines and buffers of the original MJP line, but manufacturing the virtual job. Denote this line as SJP$_v$ and its performance metrics as TP_v, $WIP_{i,v}$, $ST_{i,v}$, and $BL_{i,v}$. Evaluate these performance metrics using the recursive aggregation technique described in Li and Meerkov (2009) by expressions (11.40)-(11.49) (with the machine capacity c_i replaced by $W_i/w_{i,v}$). As a result, SJP$_v$ is approximated by $\widehat{\text{SJP}}_v$ with the performance characteristics $\widehat{TP}_v$, $\widehat{WIP}_{i,v}$, $\widehat{ST}_{i,v}$, and $\widehat{BL}_{i,v}$.

Stage III. Calculate the estimates of the performance characteristics of the original MJP line according to

$$\begin{aligned} &\widehat{TP}_j = r_j\widehat{TP}_v, j=1,\cdots,S, && \widehat{WIP}_i = \widehat{WIP}_{i,v}, i=1,\cdots,M-1,\\ &\widehat{ST}_i = \widehat{ST}_{i,v}, i=2,\cdots,M, && \widehat{BL}_i = \widehat{BL}_{i,v}, i=1,\cdots,M-1. \end{aligned} \tag{3}$$

The accuracy of this analysis method has been analyzed in Alavian et al. (2016). The results are as follows:

- Stage I induces practically no errors in all four performance metrics for all M and S considered.
- Stage II does introduce errors in all performance metrics. The errors in *TP* are two-to-four times smaller than those in *WIP*. The errors in *BL* and *ST* are practically identical. All the errors are increasing functions of *M* and practically independent of *S*. We note that these errors are similar to those observed in evaluating asynchronous exponential SJP lines (Li and Meerkov 2009, Section 11.2).
- Similar to Stage I, Stage III introduces practically no errors in all performance metrics.

3.2. Bottleneck identification

Recall that in the framework of part-based model, BN is defined as machine m_i with the largest effect on the system throughput quantified as

$$\frac{\partial TP}{\partial c_i} > \frac{\partial TP}{\partial c_j}, \quad \forall j \neq i, \tag{4}$$

where $c_k = 1/\tau_k$ is the capacity of machine m_k, and τ_k is its cycle time. Since in the work-based model the average cycle time is $\frac{w_{k,v}}{W_k}$ and the only variable characterizing the machine is W_k, expression (4) becomes:

$$w_{i,v}\frac{\partial TP}{\partial W_i} > w_{j,v}\frac{\partial TP}{\partial W_j}, \quad \forall j \neq i. \tag{5}$$

We use this expression for BN identification in the framework of work-based model (see Alavian et al. (2016) for details).

4. Behavior of MJP Serial Lines as a Function of the Product-Mix

For simplicity, we analyze this behavior for the case of two job-types being manufactured. Note that in the case of $S = 2$, the product-mix is specified by a single variable $r = r_1$ (since $r_2 = 1 - r_1$). Similar results for $S > 2$ have been obtained as well.

It turns out that both qualitative and quantitative properties of $TP(r)$ and BN(r) in MJP systems depend on the relationship between the jobs work-requirements. To characterize this relationship, consider an MJP serial line producing two job-types, J_1 and J_2, with work-requirements, w_{i1} and w_{i2}, $i = 1, \cdots, M$, respectively, and with product-mix r. Let the bottleneck of this line be the machine denoted as BN$_{J1}$ when $r = 1$, and BN$_{J2}$ when $r = 0$.

DEFINITION 1. Given a serial MJP line defined by assumptions (i)-(vii), jobs J_1 and J_2 are called non-conflicting if BN$_{J1}$ = BN$_{J2}$. Otherwise the jobs are conflicting.

THEOREM 1. *Consider an MJP serial line defined by assumptions (i)-(vii) and producing two non-conflicting job-types, J_1 and J_2, with BN$_{J1}$ = BN$_{J2}$ = m_k. Then, if all buffers are of infinite or zero capacity,*

(a) $BN_v(r) = m_k$, for all $r \in [0,1]$;

(b) $TP_v(r)$ is strictly monotonically increasing if $TP_{J1} > TP_{J2}$; strictly monotonically decreasing if $TP_{J1} < TP_{J2}$; constant if $TP_{J1} = TP_{J2}$.

Proof: See Alavian et al. (2016).

For any finite sequence of N_i's, $TP_v(r)$ and BN(r) can be evaluated by the methods of Section 3 and the results obtained conform with those of Theorem 1 in over 95% of cases among the 24000 production lines analyzed.

THEOREM 2. *Consider an MJP serial line defined by assumptions (i)-(vii) producing two conflicting job-types, J_1 and J_2. Then, if all buffers are of infinite or zero capacity,*

(a) BN(r) has at most $M-1$ switches in the interval $[0,1]$; each machine can be a bottleneck only in a single interval of $[0,1]$;

(b) $TP_v(r)$ has the following properties:
- *if the number of switches of BN(r) is $1 \leq K \leq M$, then $TP_v(r)$, $r \in [0,1]$, has $K+1$ intervals of continuous differentiability; the BN(r) switches occur at the values of r, where $TP_v(r)$ is non-differentiable;*
- *if $w_{BN_{J1},1} > w_{BN_{J1},2}$ and $w_{BN_{J2},2} > w_{BN_{J2},1}$, then there exist r' and r'' such that $TP_v(r) > \max\{TP_{J1}, TP_{J2}\}, \forall r \in (r', r'')$.*

Proof: See Alavian et al. (2016).

The second bullet in part (b) of this theorem states that in the case of conflicting jobs, there exists a range of product-mixes, where the total throughput of MJP is larger than that of SJP of any constituent job-type. This phenomenon takes place because SJP overloads respective bottlenecks, while MJP with the "right" product-mix leads to more balanced work allocation. Using the method of Section 3, it has been shown that a similar behavior takes place for any sequence of finite buffer capacities N_i's (see Alavian et al. (2016) for details).

5. AIB for MJP Serial Lines

The methods developed in Sections 3 and 4 are the foundations of AU and OU discussed in this section. The input to AU is the mathematical model of the system at hand shown in screenshot (a) of Figure 1. Since IU is not currently available, the machine and buffer parameters shown in this screenshot have been entered manually. Screenshot (b) shows the output of AU, which represents the self-diagnosed state of the system at hand. We refer to this type of output as the Product-Mix Performance Portrait or just Performance Portrait (PP). It shows the total throughput and bottleneck for all values of the product-mix (note that, as follows from Theorem 2, this PP indicates that the job-types involved are conflicting). Selecting a particular product-mix on the PP, returns screenshot (c) (corresponding to product-mix $r_1 = 0.25$ and $r_2 = 0.75$), which indicates the effect of various "what if" scenarios of BN improvement. If a specific scenario is selected, the information is transferred to OU, which returns the results shown in Figure 2, indicating the optimal way of cycle time reduction for BN operation. In addition, OU summarizes the results and produces the overall AIB output referred to as Advice for Operation Manager, shown in screenshot (a) of Figure 3. If the product-mix were different (say, $r_1 = 0.8$ and $r_2 = 0.2$), the advice would be as shown in screenshot (b).

This approach has been used for developing a continuous improvement project at an automotive assembly plant, and the results have found favorable acceptance by the management.

6. Conclusions and Future Work

This paper showed that analytical theory for stochastic production systems analysis and improvement is an enabler of the development and application of SPS, in particular for multi-job manufacturing. While the results reported here refer to serial MJP lines, derivation of similar results for MJP assembly systems is an important theoretical and industrial problem, especially in the framework of SPS design and implementation.

References

Alavian, P., P. Denno, S. M. Meerkov. 2016. Multi-job production systems: definitions, problems, analysis, and product-mix performance portrait of serial lines. Available online at web.eecs.umich.edu/%7esmm/publications/MJP/MJPfull.pdf.

Altiok, T. 1997. *Performance Analysis of Manufacturing Systems*. Springer-Verlag, New York, NY.

Askin, R. G., C. R. Standridge. 1993. *Modeling and Analysis of Manufacturing Systems*. Wiley, New York, NY.

Bazaraa, M., J. J. Jarvis, H. D. Sherali. 2009. *Linear Programming and Network Flows*. 4th ed. John Wiley.

Boyd, S., L. Vandenberghe. 2004. *Convex Optimization*. Cambridge University Press.

Buzacott, J. A., J. G. Shanthikumar. 1993. *Stochastic Models of Manufacturing Systems*. Prentice Hall, Englewood Cliff, NJ.

Curry, G. L., R. M. Feldman. 2009. *Manufacturing Systems Modeling and Analysis*. Springer.

Gershwin, S. B. 1994. *Manufacturing Systems Engineering*. Prentice Hall, Englewood Cliff, NJ.

Li, J., S. M. Meerkov. 2009. *Production Systems Engineering*. Springer. Chinese translation, 2012.

Papadimitriou, C. H., K. Steiglitz. 1998. *Combinatorial Optimization: Algorithms and Complexity*. Dover publications.

Papadopoulos, H. T., C. Heavy, J. Browne. 1993. *Queueing Theory in Manufacturing Systems Analysis and Design*. Chapman and Hill, London, UK.

Papadopoulos, T. C., M. E. J. O'Kelly, M. I. Vidalis, D. Spinellis. 2009. *Analysis and Design of Discrete Part Production Lines*. Springer.

Ruszczynski, Andrzej. 2006. *Nonlinear Optimization*. Princeton University Press.

Viswanadham, N., Y. Narahari. 1992. *Performance Modeling of Automated Manufacturing Systems*. Prentice Hall, Englewood Cliff, NJ.

(a)

(b)

(c)

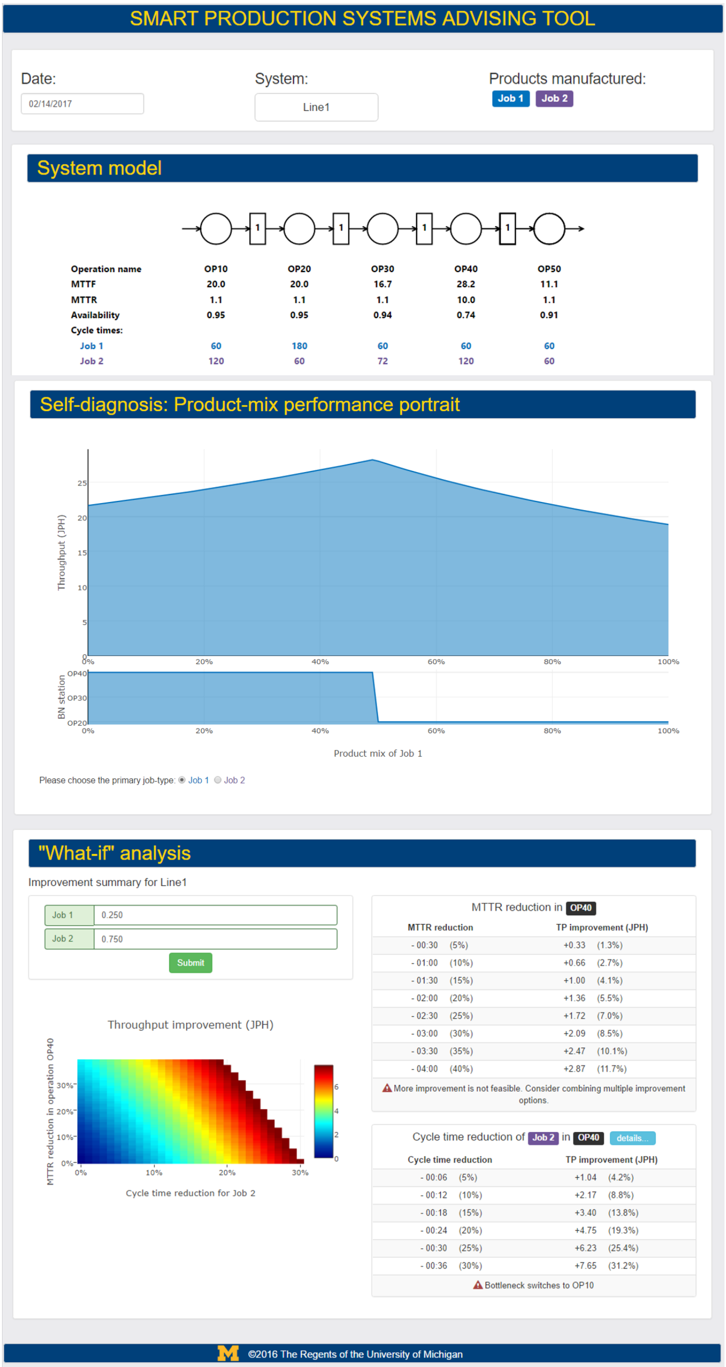


Figure 1. AU operation

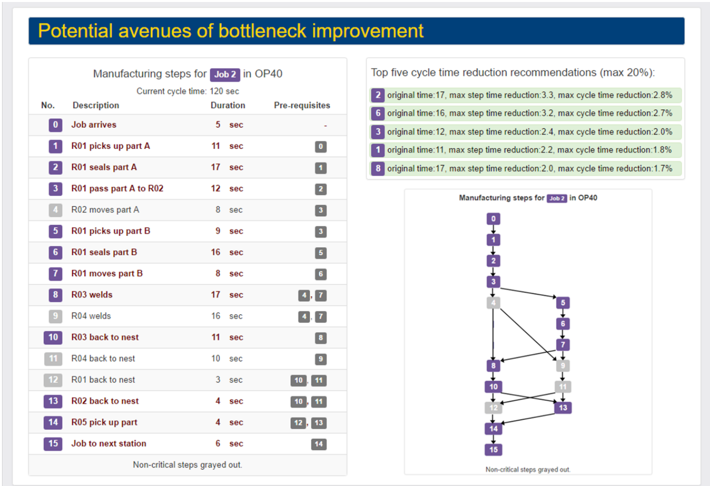
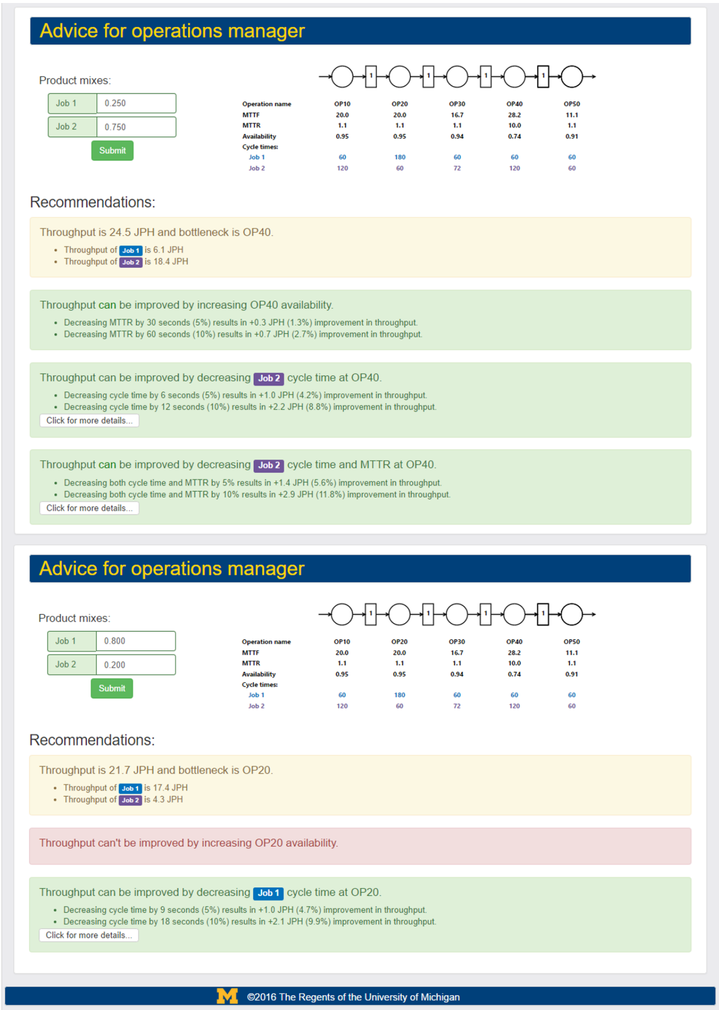


Figure 2. OU operation



(a)

(b)

Figure 3. Output of AIB