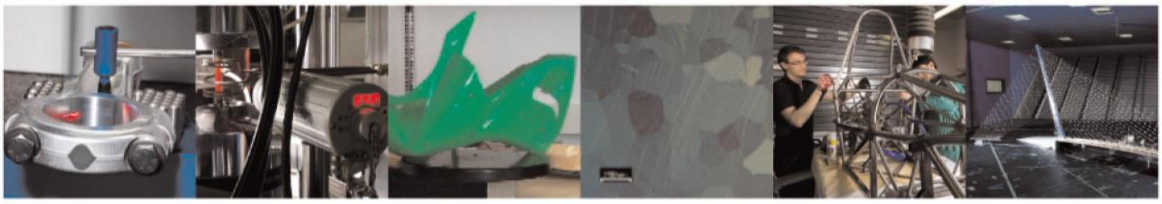

A statistical framework of data-driven bottleneck identification in manufacturing systems

Yu, Chunlong; Matta, Andrea

To appear in the *International Journal of Production Research*
Vol. 00, No. 00, 00 Month 20XX, 1–17

A statistical framework of data-driven bottleneck identification in manufacturing systems

Chunlong Yu [a] and Andrea Matta [b] *

[a] *Department of Mechanical Engineering, Politecnico di Milano, Milan, Italy*;
[b]*Department of Industrial Engineering & Management, School of Mechanical Engineering, Shanghai Jiao Tong University (SJTU), Shanghai, P.R. China*

(*v4.1 released December 2014*)

Data-driven bottleneck identification has received an increasing interest during the recent years. This approach locates the throughput bottleneck of manufacturing systems based on indicators derived from measured machine performance metrics. However, the variability in manufacturing systems may affect the quality of bottleneck indicators, leading to possible inaccurate detection results. This paper presents a statistical framework (SF) to decrease the data-driven detection inaccuracy caused by system variability. Using several statistical tools as building blocks, the proposed SF is able to analyze the logical conditions under which a machine is detected as the bottleneck, and rejects the proposal of bottleneck when no sufficient statistical evidence is collected. A full factorial design experiment is used to study the parameter effects of the SF, and to calibrate the SF. The proposed SF was numerically verified to be effective in decreasing the wrong bottleneck detection rate in serial production lines.

Keywords: Data driven; Manufacturing systems; Bottleneck

1. Introduction

Throughput is the most relevant metric to evaluate the efficiency of a production system. High throughput can be a key factor of the company competitiveness and usually brings large profit to the companies. Yet, the system throughput is significantly constrained by few machines in the system, commonly called bottlenecks (Wang et al. 2005). Technically, the bottleneck is defined as the machine to which the overall system throughput has the largest sensitivity (Li et al. 2009). Hence, system throughput improvement is quite related to improving the bottleneck. Unfortunately, locating the bottleneck in complex manufacturing systems is not easy, because a direct measure of the throughput sensitivity does not exist in factory floors (Kuo et al. 1996).

Bottleneck detection is a crucial and important issue in the production system related research. During the last two decades, several methods for detecting bottleneck in factory floors have been proposed in the literature. These make use of the on-line measurable metrics, either from buffers or machines, for obtaining indirect measures of the throughput sensitivity. Table 1 gives a summary of the detection methods and their main characteristics, relevant details are available in Betterton et al. (2012).

In general, bottleneck detection methods are based on one or more bottleneck measures derived from the collected machine performance metrics (e.g., buffer level, machine blockage and starvation time). These metrics can also be estimated coupled with analytical models or simulation models. However, the lack of exact analytical results for complex production systems and the long development time and possible misinterpretation of simulation results greatly limit the wide application of

* Corresponding author. Email: matta@sjtu.edu.cn

Table 1. Reviewed bottleneck detection methods

Method	Authors	Year	Adopted metrics	Applicable system	Num. of bottleneck conditions
Queue length	Lawrence and Buss	1994	Buffer level	General	M-1 *
Utilization	Hopp and Spearman	2000	Machine utilization	General	M-1
Active period	Roser et al.	2002	Machine blockage and starvation probability	General	M-1
Queue time	Faget and Herrmann.	2005	Buffer time	General	M-1
OTE	Muthiah adn Huang	2007	Several	General	M-1
Inactive period	Sengupta et al.	2008	Machine blockage and starvation probability	Serial line	M-1
ITV	Betterton et al	2012	Machine interdeparture time	Serial line	M-1
Arrow	Kuo et al.	1996	Machine blockage and starvation probability	Serial line	1 or more
	Biller et al.	2008	Machine blockage and starvation probability	Serial line with reworks	Several
Turning point	Li et al.	2009	Machine blockage and starvation probability	Serial line	3 or more
	Li	2009	Machine blockage and starvation probability	General	3 or more

* M: number of machines in the system.

model-based methodology in complex production systems. Actually, with the rapid development of information and communication technology (ICT), how to collect, analyze and make use of real data has become a key for the future manufacturing systems. Under this background, the tendency of detecting the bottleneck without building an analytical or simulation model, but merely with real-time data collected from the manufacturing systems, is increasing (Li et al. 2009, Betterton and Silver 2012). Such detection approach is known as Data-driven Bottleneck Detection.

The data-driven approach has obvious advantages, it avoids all the limitations in model building and potential errors from model assumption violation. Moreover, it allows dynamic bottleneck control and improvement based on short-term data analysis (Li and Ni. 2009). However, the accuracy to locate the bottleneck is closely related to the variability of the data coming from the field because of manufacturing system randomness. Indeed, variability can be introduced by unscheduled downtime of machines, process time variation, machine setups, recycle, etc. (Hopp and Spearman 2000). These uncertainties render machine performance metrics behaving as random variables, with an underlying joint distribution that put all of them in correlation. Therefore, estimation errors are inevitably related to the data-driven bottleneck detection approach, because the machine performance metrics are evaluated by using a finite stream of online records. This may result in possible unreliable bottleneck measures and, finally, inaccurate detections for the manufacturing companies. In the literature there are still not many concerns about this problem. Kuo et al. (1996) and Roser et al. (2001) calculated the confidence interval of bottleneck measures to provide statistical evidence of the detection results. However, detailed studies about how these approaches improve the detection accuracy were not given. Moreover, their treatments are not general enough to be adopted in other bottleneck detection methods with more bottleneck measures or more complex rules.

This paper proposes a Statistical Framework for data-driven detection approach, it is constructed with versatility allowing to be coupled with almost all bottleneck detection methods. The SF assesses the reliability of bottleneck detection results, rejecting the proposal of bottleneck when there is no statistical evidence. The main result is the decrease of wrong bottleneck identifications. This allows the manufacturing companies avoiding unnecessary machine improving activities, and increasing the production throughput more cost-effectively.

The rest of this paper is organized as follows. Section 2 offers a problem description. Section 3 reviews two bottleneck detection methods which will be coupled with the SF. Section 4 describes the proposed SF. Section 5 illustrates the application of the proposed SF to some numerical cases. Conclusion, discussion and possible future developments are presented in section 6.

2. Problem description

2.1 *Bottleneck definition*

In this paper, we define the bottleneck as follows:

Definition 1. A machine is the bottleneck if the sensitivity of the system throughput to its throughput in isolation is higher than all other machines in the system:

$$\frac{\triangle TH_{sys,i}}{\triangle TH_i} > \frac{\triangle TH_{sys,j}}{\triangle TH_j}, \quad \forall j \neq i$$

where $\triangle TH_{sys,i}$ is the system throughput increment due to $\triangle TH_i$, which is the increment of isolated throughput of single machine i.

In general, there are three factors affecting the isolated throughput: machine cycle time, machine up time and machine down time. Regarding to each factor and considering it independently, the bottleneck is further categorized as cycle time bottleneck (Chiang et al. 2001), uptime bottleneck and downtime bottleneck (Chiang et al. 2000). These bottlenecks are equivalent in some cases but this is not always true (Li et al. 2009). To be specific, in this paper we will focus on the cycle time bottleneck identification.

2.2 *Problem statement*

In a production system consisting of M machines and some finite input buffers, there exists a machine m_{bn}, which is the bottleneck according to Definition 1. Yet its location is unknown. To identify it, the bottleneck detection method $\mathscr{M}$ is adopted. Assume the production system has been in its steady state. The detection starts with measuring all the required machine performance metrics $\mathbb{D} = \{D_1, D_2, ..., D_q\}$ from the factory floor. Let t be the time length of the measurement. Then, the collected dataset $\hat{\mathbb{D}} = \{\hat{D}_1, \hat{D}_2, \ldots, \hat{D}_q\}$ is used as inputs for $\mathscr{M}$, which outputs the detected bottleneck $\hat{m}_{bn} \in \{m_0, m_1, \ldots, m_M\}$. Specifically, $\hat{m}_{bn} = m_0$ if no bottleneck is detected.

Due to the estimation errors of $\mathbb{D}$ derived from system variability, there exists the probability that $\mathscr{M}$ cannot locate the true bottleneck m_{bn}. We name this probability the wrong detection rate P_w, which is calculated with the following expression:

$$P_w = \lim_{N \to +\infty} \frac{\sum_{l=1}^{N} \mathbf{1}(\hat{m}_{bn,l} \neq m_{bn}) \cdot \mathbf{1}(\hat{m}_{bn,l} \neq m_0)}{N} \tag{1}$$

where $\mathbf{1}$ is the indicator function equal to 1 if condition in the argument is true and 0 otherwise, N is the number of independent and identical detection replications, $\hat{m}_{bn,l}$ represents the detected bottleneck in the l-th replication.

Actually, in data-driven bottleneck detection, system variability is one of the main factors causing detection inaccuracy. The problem considered in this paper is to propose a statistical framework able to recognize and avoid the wrong results caused by system variability, so as to reduce the P_w in data-driven bottleneck identifications.

3. Examples of bottleneck detection methods

The proposed SF is general enough to be coupled with all the bottleneck detection methods listed in Table 1. Due to limited space, in this paper we will validate the SF by adopting two powerful detection methods, the Arrow method (AM) and the Turning point method (TPM), as examples. The details of these two methods are reviewed as follows.

3.1 *Arrow method*

The bottleneck is detected with the following rules (Kuo et al., 1996):

Rule 1. Let mb_i and ms_i be the blockage and starvation probabilities of machine m_i, M be the number of machines in the serial production line. If the following condition holds:

$$mb_i > ms_{i+1} : i = 1, \ldots, M-1,$$

the bottleneck is downstream of m_i, and an arrow is directed from m_i to m_{i+1}. If the following inequality holds:

$$ms_i > mb_{i-1} : i = 2, \ldots, M,$$

the bottleneck is upstream of m_i, and an arrow is directed from m_i to m_{i-1}. Then, the machine with no departing arrows is detected as the bottleneck.

Rule 2. If multiple bottlenecks are detected using Rule 1, the one with the highest bottleneck severity is the system bottleneck. The *bottleneck severity* is defined as:

$$S_1 = ms_2 - mb_1, \quad S_M = mb_{M-1} - ms_M,$$
$$S_i = (mb_{i-1} + ms_{i+1}) - (mb_i + ms_i) : i = 2, \ldots, M-1$$

3.2 *Turning point method*

The bottleneck is detected with the following rules (Li et al., 2009):

Rule 3. With the same notation of mb_i, ms_i and M described before, machine m_j is the turning point of a production segment $\{m_{st}, \ldots, m_{ed}\}$,where $st, ed \in [1, M]$, $st \leq j \leq ed$ and $st \neq ed$, if the following inequalities are satisfied:

$$mb_i - ms_i > 0 : i = st, st+1 \ldots, j-1, j \neq st, j \neq ed,$$
$$mb_i - ms_i < 0 : i = j+1, \ldots, ed, j \neq st, j \neq, ed,$$
$$mb_j + ms_j < mb_{j-1} + ms_{j-1} : j \neq st, j \neq ed,$$
$$mb_j + ms_j < mb_{j+1} + ms_{j+1} : j \neq st, j \neq ed.$$

If $j = st$:

$$mb_{st} - ms_{st} > 0 \quad \text{and} \quad mb_{st+1} - ms_{st+1} < 0$$
$$\text{and} \quad mb_{st} + ms_{st} < mb_{st+1} + ms_{st+1}$$

If $j = ed$:

$$mb_{ed-1} - ms_{ed-1} > 0 \quad \text{and} \quad mb_{ed} - ms_{ed} < 0$$
$$\text{and} \quad mb_{ed} + ms_{ed} < mb_{ed-1} + ms_{ed-1}$$

If there is only one turning point in the production system, the turning point is the bottleneck.

Rule 4. If there are multiple turning points in the production system, the turning point with the maximum bottleneck index is the bottleneck. The *bottleneck index* is defined as:

$$S_{st} = \frac{ms_{st+1}}{mb_{st} + ms_{st}}, \quad S_{ed} = \frac{mb_{ed-1}}{mb_{ed} + ms_{ed}},$$
$$S_i = \frac{(mb_{i-1} + ms_{i+1})}{mb_i + ms_i} : i = st+1, \ldots, ed-1$$

4. Statistical framework

Bottleneck detection is a logical procedure to judge whether a machine is or not the bottleneck. This judgment can be decomposed into a series of more basic logical operations. These operations, in the case of data-driven detection, are to verify a set of bottleneck conditions using the on-filed data. Examples are as in Rule 1. The detection accuracy depends on two factors: first, whether the bottleneck conditions developed in a detection method can reveal the essence of the bottleneck; second, whether the data collected are representative to characterize the machine. Previous researches were mainly devoted to the first factor but seldom considered the second. Actually, estimation error of the performance metrics derived from system variability can greatly distort the facts of bottleneck conditions, leading to reduction of the detection accuracy. To avoid this problem, several statistical analysis tools are incorporated into one scheme, which renders the possibility to assess the reliability of the detection result.

4.1 *Proposition of detection*

4.1.1 Bottleneck condition verification

For a bottleneck detection method $\mathscr{M}$, let $\mathbb{C}_i(\mathscr{M}) = \{c_{i1}, c_{i2}, \ldots, c_{in}\}$ be the set of n bottleneck conditions under which m_i is detected as the bottleneck. To our knowledge, for any $\mathscr{M}$ presented in Table 1, any bottleneck condition c_{ik} can be generally formalized into a comparison of two variables deriving from machine performance metrics, as:

$$X_{ik}(\mathbb{D}) < Y_{ik}(\mathbb{D}) \tag{2}$$

Here, X_{ik} and Y_{ik} are named *bottleneck indicators*, which can be easily derived from the description of bottleneck detection method. For instance when AM is used, $\mathbb{C}_i(\text{AM}) = \{ms_i < mb_{i-1}, mb_i < ms_{i+1}\}$; in the case multiple local bottlenecks are detected simultaneously, for any of them, some global conditions (as Rule 2) are required to identify the global bottleneck.

Judging whether c_{ik} is satisfied in long-term perspective is actually to verify $\mu_{X_{ik}} < \mu_{Y_{ik}}$, where μ stands for the true mean. Actual methods verify this inequality in a deterministic sense neglecting the randomness of the bottleneck indicators. Instead of simply comparing the sample means, we propose to verify c_{ik} with the following hypothesis test:

$$\begin{aligned} H_0 &: \mu_{X_{ik}} = \mu_{Y_{ik}} \\ H_1 &: \mu_{X_{ik}} < \mu_{Y_{ik}} \end{aligned} \tag{3}$$

Standard statistical methods such as *t-test* can be adopted to test (3) in the condition that the observations of (X_{ik}, Y_{ik}) are independently and normally distributed. But in manufacturing context, the collected data are generally correlated and might be non-normal. Data independency and normality can be obtained by applying the *batch means technique* (Law, 2007), which will be introduced in detail in section 4.2. However, as noted by Chen and Kelton (2007), batch means that appear to be independent are not necessarily normally distributed and vice versa. Indeed, data normality is sometimes difficult to obtain especially when the metric distribution is restricted by some physical constraints. In order to release the normality requirement, we propose to use the *Mann-Whitney U test* for (3). This test has greater efficiency than the *t-test* on non-normal distributions and, in most of the cases, it is nearly as efficient as the *t-test* on normal distributions.

4.1.2 Multiple test problem

In order to conclude m_i is the bottleneck, it is necessary to accept all the bottleneck conditions in $\mathbb{C}_i(\mathscr{M})$ simultaneously. This is actually a multiple test problem. Indeed, when many hypotheses are tested simultaneously, and each test has a specified Type I error probability, the probability that at least some Type I errors are committed increases.

To solve multiple test problems, the simplest and most general procedure is the Bonferroni approach that controls the *familywise error* (FWER), i.e., the probability of making one or more false rejection, below α by adjusting the significance level of individual test to α/n. However, the Bonferroni method is an example of a single-stage testing procedure. In such procedure, control of the FWER has the consequence that the larger the number of hypotheses in the family, the smaller the average power for testing the individual hypothesis. To solve this problem, Holm (1979) presented a multistage test procedure that sequentially rejects the null hypothesis. The Holm's method is less conservative and widely applicable.

Besides the Holm's method, Hochberg (1988), Hommel (1988) and Rom (1990) proposed different multistage test procedures based on the Simes equality (1986). These procedures are more powerful, but strictly speaking, they are based on the assumption that the test statistics of the multiple hypothesis are independent from each other. However, this is not always guaranteed in the bottleneck detection procedure. Take $\mathbb{C}_i(\mathrm{AM})$ as an example, the test statistics of the two bottleneck conditions can be generally formulated as $T_{i1} = f(mb_{i-1}, ms_i)$ and $T_{i2} = f(mb_i, ms_{i+1})$, respectively. Due to the blockage and starvation propagation phenomenon, mb_i and mb_{i-1} are not considered as independent, as well as for ms_i and ms_{i+1}. Such correlation properties result in the dependence between T_{i1} and T_{i2} of certain extent. Consequently, the application of these methods in the context of bottleneck detection may be improper.

Benjamini and Hochberg (1995) introduced an approach, denoted as BH, to control the *false discovery rate* (FDR) instead of the FWER. FDR is the proportion of the rejected null hypotheses which are erroneously rejected. This method adapts for both independent and dependent test statistics. It has much more statistical power than the methods controlling the FWER but also higher Type I error as compromise.

In summary, Holm's method and BH method do not require independent assumption between bottleneck conditions. The former one is less powerful but more accurate, the latter one is just the opposite. These two methods are adopted to solve the multiple test problem in bottleneck detection. Detailed detection propositions are given in the next section.

4.1.3 Bottleneck detection proposition

Applying Holm's method and BH method respectively, two bottleneck detection propositions are developed.

Proposition 1. Let H^0_{ik} be the null hypothesis of the test to verify bottleneck condition c_{ik}, and let α be the significance level. First, arrange all the null hypothesis $H^0_{i1}, H^0_{i2}, \ldots, H^0_{in}$ in a non-descending order of their p-values: $H^0_{(1)}, H^0_{(2)}, \ldots, H^0_{(n)}$, with the corresponding p-value $P_{(1)} \leq P_{(2)} \leq \cdots \leq P_{(n)}$. Define

$$\xi = \min\{\max_{k\in[1,n]}\{(n-k+1)P_{(k)}\}, 1\} \tag{4}$$

If $\xi < \alpha$, the null hypothesis $H^0_{i1}, H^0_{i2}, \ldots, H^0_{in}$ are rejected and machine m_i is detected as the bottleneck. For further technical details, see Holm (1979).

Proposition 2. With the same notation of $P_{(k)}, \forall k$ and α, Define

$$\xi = \min\{\max_{k\in[1,n]}\{\frac{n}{k}P_{(k)}\}, 1\} \tag{5}$$

If $\xi < \alpha$, the null hypothesis $H^0_{i1}, H^0_{i2}, \ldots, H^0_{in}$ are rejected and machine m_i is detected as the bottleneck. For further technical details, see Benjamini and Hochberg (1995).

Proposition 1 controls the familywise (Type I) error in bottleneck detection, i.e., probability of accepting any wrong bottleneck conditions of m_i, under α. Theoretically, if the bottleneck conditions are correctly developed, this procedure guarantees the accuracy of any detected bottleneck above $1-\alpha$. On the other hand, the familywise Type II error, i.e., the probability to deny any bottleneck condition when it is actually satisfied, is closely related to the power of Holm's method. Unfortunately, the power of Holm's method decreases dramatically with the increasing of test number n. As an alternative, Proposition 2 can be used when n becomes large. In this way, we compromise between Type I error and Type II error.

4.2 *Batch means*

An important assumption made by many of the statistical tests is that the observations are an independent sample from some underlying distribution. Yet, data independence in manufacturing system is not always assured. In our paper, independent observations for the bottleneck indicators $(X_{ik}, Y_{ik}), \forall i, k$ are generated by the batch means technique (Law 2007), which is applied as follows.

Step 1 Initialize the batch number $g \leftarrow g_0$ and set the minimal allowed batch number g^*.
Step 2 Let $\hat{x}_1, \hat{x}_2, \ldots, \hat{x}_t$ and $\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_t$ be the t observations of X_{ik} and Y_{ik}. Divide them into g batches with batch size $w = t/g$, respectively.
Step 3 Calculate the sample mean of each batch, and the batch means can be obtained: $\bar{X}(1), \bar{X}(2), \ldots, \bar{X}(g)$ for X_{ik}, and $\bar{Y}(1), \bar{Y}(2), \ldots, \bar{Y}(g)$ for Y_{ik}.
Step 4 Test the independence of both batch means using the Von Neumann's ratio test (Bartels 1982). If both are independent, go to Step 6; otherwise go to Step 5.
Step 5 If $g > g^*$, reduce the batch number with $g \leftarrow g - 1$, and return to Step 2; otherwise go to Step 7.
Step 6 Output the obtained independent batch means for X_{ik} and Y_{ik}. Go to Step 8.
Step 7 Independent batch means cannot be obtained for X_{ik} and Y_{ik}. Go to Step 8.
Step 8 End the procedure.

Indeed, the batch number, or saying, number of observations after batching, is closely related to the statistical power of the bottleneck condition test (3). To guarantee the statistical power, the upper bound g_0 and the lower bound g^* should be chosen carefully. In our case, g_0 is set large enough to guarantee a 90% statistical power of the test in (3). g^* is a parameter to be tuned, which will be discussed in Section 5.2.

4.3 *Proposed detection scheme*

With the detection propositions and the discussed statistical techniques as building blocks, we propose a new detection scheme. In this scheme, Proposition 1 and 2 are applied to assess the bottleneck results obtained by conventional detection methods. The detection results are accepted or rejected according to their statistical evidence. This scheme has five method parameters as follows: $\mathscr{M}$, α, g^*, n^*, ε. $\mathscr{M}$ is the detection method, α is the significance level, g^* is the minimal allowed batch number, n^* is a threshold for the number of bottleneck conditions based on which we select the detection Proposition, $\varepsilon \in \{0, 1, 2\}$ is an integer value indicating what policy to take when independent batch means cannot be obtained. More specifically, when $\varepsilon = 0$, we reject the detected bottleneck without further considerations; $\varepsilon = 1$, we ignore the requirement of data independency for the Propositions; when $\varepsilon = 2$, we accept the detected bottleneck without further considerations. This scheme, i.e., the SF, is performed with the following steps.

Step 1 Set method parameters $\mathscr{M}, \alpha, g^*, n^*, \varepsilon$.
Step 2 Collect real-time data $\hat{\mathbb{D}}$ with length t. Input $\hat{\mathbb{D}}$ into $\mathscr{M}$ and obtain the detected bottleneck $\hat{m}_{bn}$. In the special case $\hat{m}_{bn} = m_0$, go to Step 7.
Step 3 List all the n bottleneck conditions used to detect the bottleneck $\hat{m}_{bn}$. For each of them,

Table 2. Case A, configurations of Line A1-A4

Line	PT (min)	CV	MTTF1 (min)	MTTR1(min)	MTTF2 (min)	MTTR2 (min)	Downstream buffer capacity
A1	[2.5 2.6 2.7 2.6 2.5]	0.1	300	50	-	-	50
A2	[2.5 2.6 2.7 2.6 2.5]	0.5	300	50	-	-	50
A3	[2.5 2.6 2.7 2.6 2.5]	0.1	300	50	20	1	50
A4	[2.5 2.6 3.0 2.6 2.5]	0.1	300	50	-	-	50

apply the batch means technique described in section 4.2 to obtain independent observations of the bottleneck indicators. If independent bottleneck indicators can be obtained for all bottleneck conditions, go to Step 5; otherwise go to Step 4.

Step 4 If $\varepsilon = 1$, go to step 5; If $\varepsilon = 2$, go to step 6; otherwise go to step 7.

Step 5 If $n < n^*$, apply Proposition 1 to verify whether $\hat{m}_{bn}$ is the bottleneck; otherwise apply Proposition 2. In the case $\hat{m}_{bn}$ is accepted as the bottleneck by the proposition, go to Step 6; otherwise go to Step 7.

Step 6 Output the bottleneck $\hat{m}_{bn}$. Go to Step 8.

Step 7 Output no bottleneck, suggest the user to increase data length t. Go to Step 8.

Step 8 End the procedure.

5. Numerical cases

To validate the proposed SF, we use simulation models constructed in the Arena ® software package. All simulation data are collected after a warm-up period of 120 shifts (1 shift = 480 minutes), which is used to remove the transient effects. The steady state of the each simulation model has been verified by using the Welch graphical procedure (Law 2007).

5.1 *Simulation models*

We use serial production lines with the following assumptions as test instances. The serial line consists of M machines arranged serially and M-1 buffers separating each consecutive pair of machines. The part processing time at each machine is variable, following a normal distribution with a coefficient of variation (CV). Machines are unreliable and can fail into one or more failure modes. Each failure mode is characterized by the time to failure and time to repair, both are exponentially distributed. Each buffer has finite capacity. Machines could be starved and blocked, except for the first machine that is never starved and the last machine that is never blocked. A single product is produced in the line and no machine setups are required. No scrap is produced.

We construct four groups of serial lines meeting the above described assumptions. The four groups of lines, referred to as Cases A, B, C and D, are described here. Case A is a group of five-machine serial lines with simple configurations. These lines will be used as standard cases to calibrate the SF. Case B consists of 3 serial lines, with which we aim to simulate scenarios in which the bottlenecks are in different locations, and they are due to different reasons, i.e., long processing time, low machine reliability and insufficient buffer capacity, respectively. Case C contains a long line. Besides the complex configuration, we introduce multiple failure modes for each machine aiming at simulate a real industrial case. Case D contains a troublesome line. This line was used as a counter example in Betterton et al.(2012); the proposed ITV method, as well as many other methods, failed to locate the bottleneck. Detailed parameters of these lines are reported in Table 2 to Table 5. Note that in Table 2 and Table 3, when a parameter is not identical for all machines, a set containing the parameter values of individual machine is given.

For all the constructed lines, the true bottleneck according to Definition 1 is located using the finite difference method as in (Betterton et al., 2012). The obtained bottlenecks are shown in Table 6. They will be used for judging the correctness of the detection results obtained in the following sections.

Table 3. Case B, configurations of Line B1-B3

Line	PT (min)	CV	MTTF (min)	MTTR*(min)	Downstream buffer capacity
B1	[2.5 3.0 2.5 2.8 2.5]	0.2	300	50	50
B2	2.5	0.2	[300 300 200 300 250]	[50 50 80 50 65]	50
B3	2.5	0.2	100	15	[30 30 10 15 -]

Table 4. Case C, configurations of Line C1

Machine	PT (min)	CV	MTTF1 (min)	MTTR1 (min)	MTTF2 (min)	MTTR2 (min)	Downstream buffer capacity
1	2.65	0.2	1500	200	90	2	60
2	2.85	0.2	450	50	60	5	12
3	2.50	0.2	500	80	50	3	41
4	3.00	0.2	800	20	80	2	93
5	2.40	0.2	650	60	48	5	22
6	2.75	0.2	690	85	30	2	90
7	2.80	0.2	820	40	60	4	80
8	2.65	0.2	520	120	55	6	99
9	2.30	0.2	1200	80	30	1	58
10	2.70	0.2	900	35	85	3	-

Table 5. Case D, configurations of Line D1

Machine	PT (min)	CV	MTTF (min)	MTTR(min)	Downstream buffer capacity
1	3.49	0.5	300	40	100
2	3.0	0.5	350	100	80
3	2.5	0.5	140	65	150
4	2.86	0.5	400	85	100
5	2.92	0.5	360	120	100
6	3.10	0.5	140	35	-

Table 6. True bottleneck locations in the simulation models

Line	A1	A2	A3	A4	B1	B2	B3	C1	D1
Bottleneck	m_3	m_3	m_3	m_3	m_2	m_3	m_4	m_2	m_2

5.2 *SF parameters effects and calibration*

In this section we aim at, first, understanding how the parameters of the SF will affect its detection performance; second, calibrating the SF, i.e., to determine a parameter combination which leads to high performance. The detection performance is evaluated by two indicators, P_w and P_c. Here P_c is named the correct detection rate, and it is defined as the probability that the detection outputs the correct bottleneck. Note that $P_w + P_c + P_0 = 1$, where P_0 is the probability that the detection outputs no bottleneck. A good performance is featured by low P_w and high P_c.

To study the effects of the SF parameters, a full factor design of experiments (DOE) is conducted with the following setups. The related factors and levels of the DOE are reported in Table 7. All cited factors results in a total of $2 \cdot 4 \cdot 3 \cdot 3 \cdot 3 = 216$ different parameter combinations. For each of them, 5 independent replications are performed. In each replication, two response variables are considered: P_w and P_c. Each of them is evaluated by applying the SF to solve a full set of bottleneck detection problems that can be described as follows. There are four different configurations for the production lines (L), which are the four lines in Case A, as $L = \{A1, A2, A3, A4\}$, and four configurations for the data length (t) available for the detection, as $t = \{1, 6, 18, 30\}$ (shifts). All the combinations of L and t result in 16 different detection problems. In each problem, the P_w and P_c are calculated using (1) with $N = 100$. Then, the two response variables are the average of all 16 problems.

As seen in the ANOVA analysis in Table 8 and Table 9, all the main factors have significant effect on P_w and P_c. According to Figure 1 and Figure 2, α and ε are the two most influential factors and have positive correlation with both P_w and P_c. Indeed, increasing α or ε means lowering standards to accept a detection result, which makes the SF accept more wrong results as well as correct results of insufficient statistical evidence. For the factor g^*, it seems that smaller g^* results in better performance of lower P_w and higher P_c, yet this is not true, the interaction effect between

Table 7. Factors and levels of the full factor design experiment

Factors	$\mathscr{M}$	α	g^*	n^*	ε
Levels	AM, TPM	0.05, 0.2, 0.5, 0.8	5, 10, 20	2, 4, 6	1, 2, 0

Table 8. ANOVA table of the SF with respect to Pw

Source	DF	Adj SS	Adj MS	F-Value	P
$\mathscr{M}$	1	2821	2821.5	2166.25	0.000
α	3	60059	200119.6	15370.45	0.000
g^*	2	264	131.9	101.26	0.000
n^*	2	34	16.8	12.88	0.000
ε	2	48967	24483.5	18797.70	0.000
$\mathscr{M} * \alpha$	3	193	64.4	49.44	0.000
$\mathscr{M} * g^*$	2	511	255.5	196.14	0.000
$\mathscr{M} * n^*$	2	30	15.1	11.58	0.000
$\mathscr{M} * \varepsilon$	2	60	30.2	23.19	0.000
$\alpha * g^*$	6	8718	1453.0	1115.54	0.000
$\alpha * n^*$	6	14	2.3	1.74	0.108
$\alpha * \varepsilon$	6	2192	365.4	280.54	0.000
$g^* * n^*$	4	10	2.6	2.01	0.091
$g^* * \varepsilon$	4	24978	6244.5	4794.35	0.000
$n^* * \varepsilon$	4	0	0	0	1.000
Error	1030	1342	1.3		
Total	1079	150193			

S=1.14126 R-sq = 99.11% R-sq(adjusted) = 99.06%

Table 9. ANOVA table of the SF with respect to Pc

Source	DF	Adj SS	Adj MS	F-Value	P
$\mathscr{M}$	1	3293	3292.9	2509.54	0.000
α	3	65739	21913.2	16700.35	0.000
g^*	2	2076	1038.1	791.12	0.000
n^*	2	29	14.3	10.93	0.000
ε	2	76967	38483.3	29328.74	0.000
$\mathscr{M} * \alpha$	3	80	26.8	20.44	0.000
$\mathscr{M} * g^*$	2	291	145.6	110.99	0.000
$\mathscr{M} * n^*$	2	27	13.6	10.40	0.000
$\mathscr{M} * \varepsilon$	2	44	22	16.73	0.000
$\alpha * g^*$	6	10628	1771.3	1349.96	0.000
$\alpha * n^*$	6	2	0.3	0.23	0.968
$\alpha * \varepsilon$	6	1611	268.5	204.62	0.000
$g^* * n^*$	4	10	2.6	1.99	0.093
$g^* * \varepsilon$	4	52106	13026.6	9927.75	0.000
$n^* * \varepsilon$	4	0	0	0	1.000
Error	1030	1352	1.3		
Total	1079	214256			

S=1.14549 R-sq = 99.37% R-sq(adjusted) = 99.34%

ε and g^* should be considered. As we find, when $\varepsilon = 1$ or 2 decreasing g^* leads to lower P_w and P_c, whereas when $\varepsilon = 0$ the trend is just opposite. As to the factor $\mathscr{M}$, TPM obtains lower P_w but also lower P_c than AM. Indeed, TPM has more bottleneck conditions, which makes the SF more rigorous in accepting a result. Finally, a slightly decreasing tendency of P_w and P_c is observed when n^* is increased.

As shown, trade-off between P_w and P_c is inevitable when setting the SF parameters. We choose the parameters as follows. On one hand, to assure a low P_w, we select low levels for α and ε, and adopt TPM as the detection method. On the other hand, to pursue a high P_c, low level is selected for g^*. At last, medium n^* is adopted. As a consequence, $\mathscr{M}$ =TPM, $\alpha = 0.05$, $\varepsilon = 0$, $g^* = 5$, $n^* = 4$ is set as the default parameter combination for the SF.

5.3 *Detection performance comparison*

In this section, we compare the performance of the three detection schemes: AM, TPM and SF with default settings, using the simulation models in Case B, C and D as test instances. For each

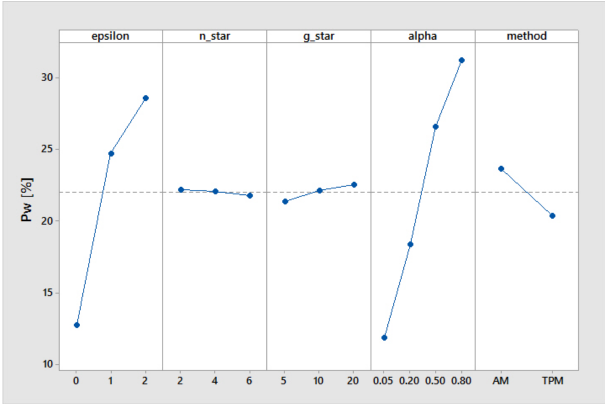

Figure 1. Main effect plot for P_w

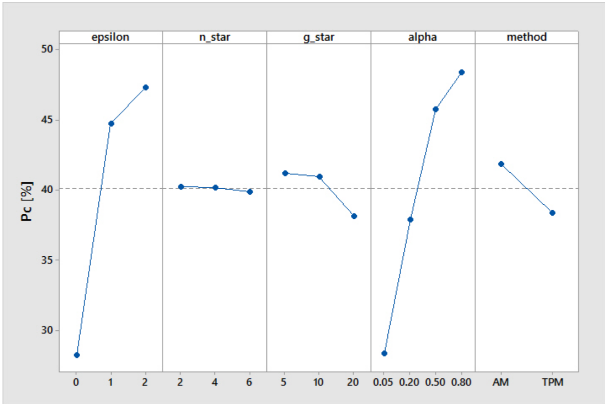

Figure 2. Main effect plot for P_c

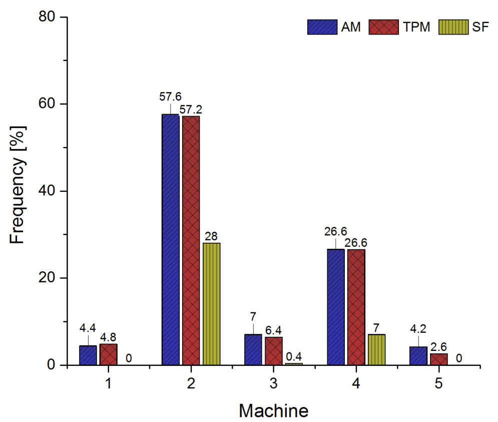

Figure 3. Detection results distribution over 500 experiments, where AM, TPM and SF are applied to identify the bottleneck in Line B1 with t=3 (shifts). The true bottleneck here is m_2

test instance, the AM, TPM and SF are adopted to identify the bottleneck, respectively. For each detection scheme, we record its detection results obtained by feeding with production data of different length, which varies from 3 to 90 shifts. For each combination of [Test instance, detection scheme, data length], 500 experiment replications are performed to evaluate the corresponding P_w and P_c.

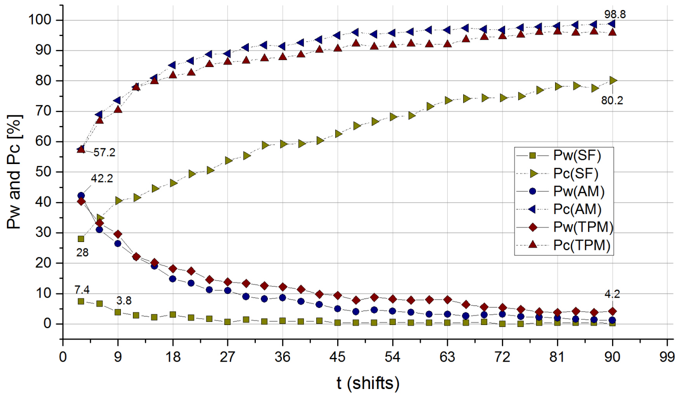

Figure 4. Detection performance comparison: Line B1

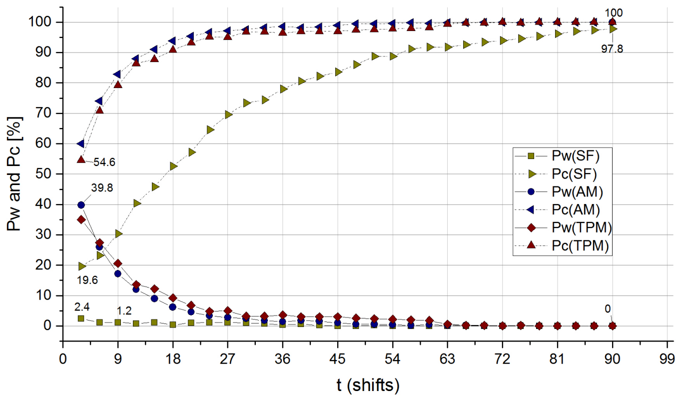

Figure 5. Detection performance comparison: Line B2

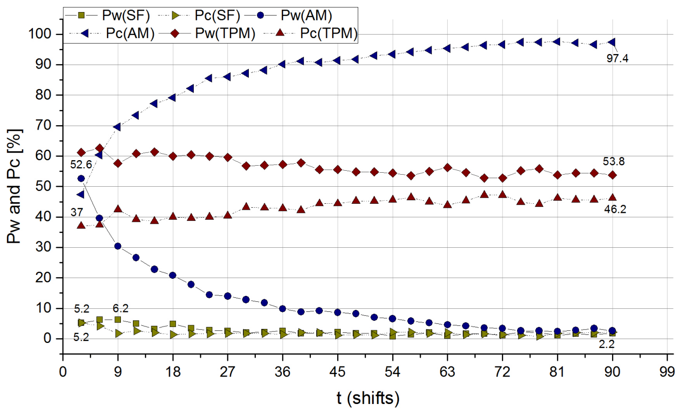

Figure 6. Detection performance comparison: Line B3

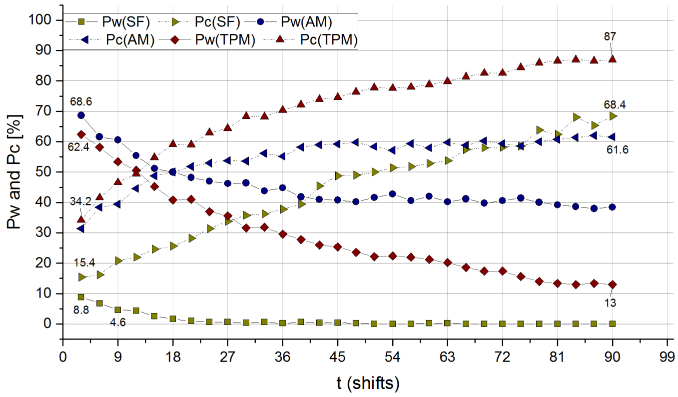

Figure 7. Detection performance comparison: Line C1

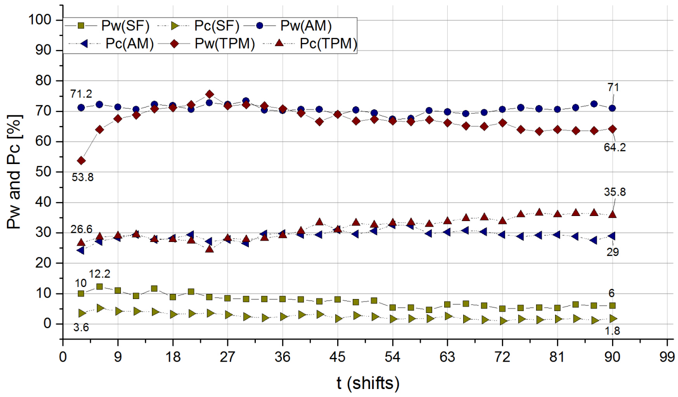

Figure 8. Detection performance comparison: Line D1

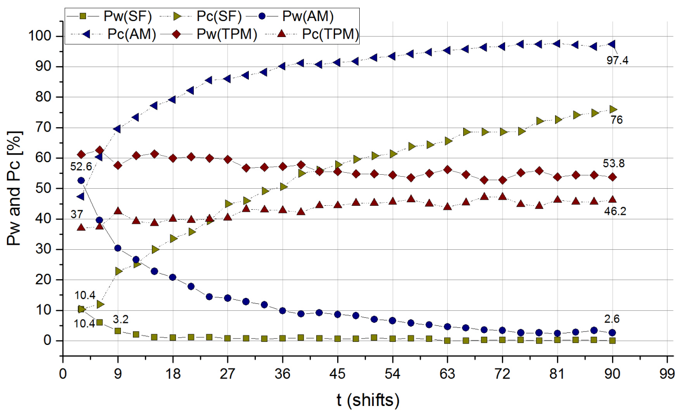

Figure 9. Detection performance comparison: Line B3. The SF adopts AM as the coupled $\mathscr{M}$, other parameter settings are as default

First, we make comparisons in Line B1. When $t = 3$ shifts, the AM and TPM cannot locate the true bottleneck m_2 in all replications, they give wrong detection results with a frequency around 40%, as in Figure 3. These wrong results distribute over all non-bottleneck machines but mainly on m_4, which is the second critical machine in the line. Whereas, the SF has a much lower P_w, because it rejects most of the wrong results proposed by the coupled $\mathscr{M}$ (TPM). Meanwhile, the SF rejects a portion of the proposed correct results as well, because these results have no sufficient statistical evidence. They are expected to be accepted when t is longer and more statistical evidences are collected. The detection results obtained with longer t are reported in Figure 4. As shown, P_w (AM) and P_w (TPM) decrease along t axis. Indeed, less estimation errors of $\mathbb{D}$ are incurred with the use of longer data and, therefore, the detection inaccuracy of AM and TPM decreases. Even so, the SF still shows great advantage over AM and TPM in avoiding wrong results, especially when t is short. On the other hand, P_c (SF) rises gradually along t axis and approaches to P_c (TPM). As expected, with the accumulation of statistical evidences, more correct results proposed by $\mathscr{M}$ are accepted by the SF.

Same comparisons are made in other production lines. Results are shown in Figure 5 to Figure 8. On one hand, in all cases, the SF suppresses the P_w at low level, the maximum P_w (SF) is merely 12.2% (Figure 8, t=2 shifts). Whereas the maximum P_w (AM) and P_w (TPM) both exceed 70%, as observed in Figure 8. It can be concluded that the SF is capable in avoiding wrong detection results in systems with different configurations. On the other hand, the P_c (SF) is lower than P_c (AM) and P_c (TPM) in all cases. Nevertheless, P_c (SF) increases with t and converges to the P_c of the coupled $\mathscr{M}$. Therefore, when the SF obtains no result, the users can input further records to obtain the correct bottleneck.

Two exceptions where P_c (SF) does not increase with t are observed, one is in Line B3 (Figure 6), and the other is in Line D1 (Figure 8). In both cases, the coupled $\mathscr{M}$ of the SF, i.e., TPM, fails to converge to the true bottleneck no matter how long the input record is. Indeed, the performance of the SF relies greatly on the coupled $\mathscr{M}$. To fix it, we have to select another $\mathscr{M}$ which is effective in these circumstances for the SF. For example in Line B3, we use AM instead of TPM as the coupled $\mathscr{M}$ and consequently, P_c (SF) increases gradually with t, as in Figure 9.

5.4 *Influence on bottleneck investment*

The purpose of bottleneck detection is to discover the most critical machine on which investment can bring the highest benefit. However the risk of detection error can have a great impact on the benefit. In this section, we study numerically how different detection schemes, i.e., AM, TPM and SF, will affect the benefit of bottleneck investment, with a simple reward model described below.

Assume a bottleneck investment project requires a cost I. Given the collected data of length t, if the correct bottleneck is detected and improved, a reward of $(1 + \tau)I$ is given, where τ is named *reward rate*; otherwise the reward is 0. The cost I is spent whenever a bottleneck is detected then improved, no matter it is correct or wrong. In the case no bottleneck is detected, no investment is made so no cost is incurred. The expected return on investment (ROI) can be calculated by:

$$ROI = \frac{I(1 + \tau)P_c - I(P_c + P_w)}{I} = \tau P_c - P_w \tag{6}$$

We calculate the ROI of different cases, results are shown in Figure 10. First, the benefit of applying the SF comes from its low P_w, which is the key for the users to avoid unnecessary investments. Therefore, when the available data is too short or the system configuration is too complex to assure a correct bottleneck, applying the SF is a wise decision which avoids loss. As seen in the case (Line B1, $\tau = 0.2$) and (Line B2, $\tau = 0.2$), when $t < 20$ shifts, the SF obtains a better ROI than AM and TPM. Also, as in the case (Line C1, $\tau = 0.2$), when the system has complex configurations, the SF outperforms AM and TPM along the t axis. Second, the disadvantage of the SF is the relatively low P_c, which makes the users become conservative in investing and leads

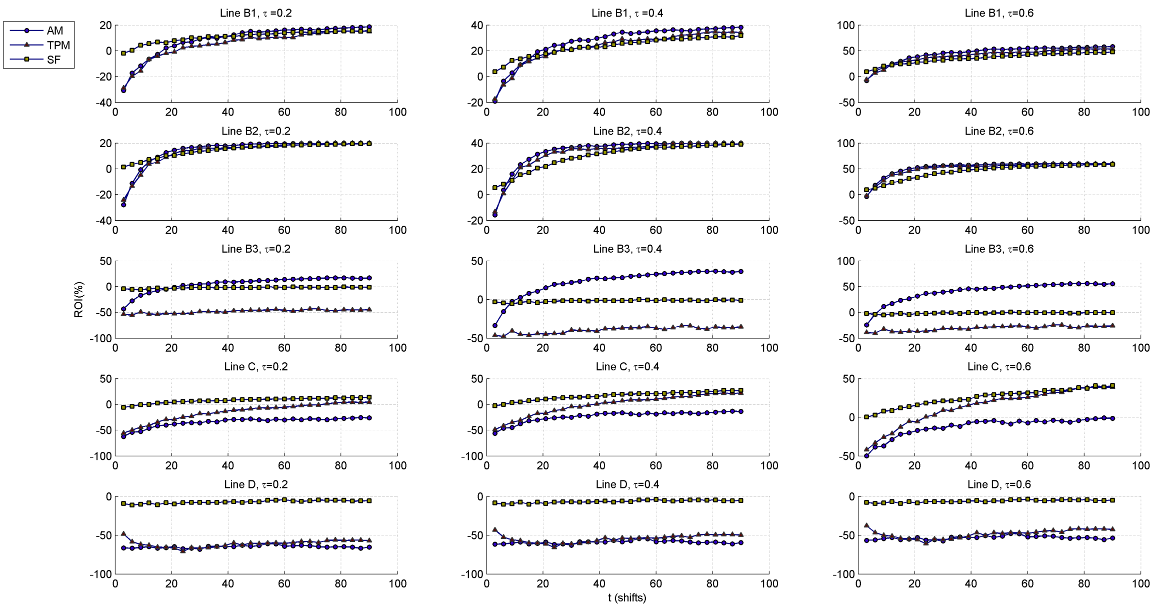

Figure 10. ROI of different bottleneck investment projects guided by AM, TPM and SF.

to opportunity costs. So when the reward rate is high, and the potential cost of not investing is large, the SF is less favored. For instance, in case (Line B1, $\tau = 0.6$), the ROI of SF becomes lower than that of AM and TPM for any $t > 9$ shifts. Third, when the characteristics of the production system are too tricky for a detection method to converge to the true bottleneck, applying the SF is able to pull down the P_w and prevent losses. Examples are found in the cases of Line D1 with different τ values.

6. Conclusion

By incorporating several statistical tools such as multiple testing technique and batch means technique, we proposed a statistical framework for the data-driven bottleneck detection procedure in manufacturing systems. To the best of our knowledge, this is the first research studying the influence of system variability on the accuracy of data-driven bottleneck detection and proposing solution to mitigate the inaccuracy. Simulation and comparisons demonstrate the effectiveness of the SF in avoiding wrong detection results. Bottleneck investments guided by the SF are more secure and result in higher ROI, especially when the available data stream is short and the system configuration is complex. In addition, the SF is easy to implement and needs no other information but only the on-field records used by the adopted bottleneck detection method.

The main disadvantage of the SF is that the P_c is not high enough. In some cases, the difference between P_c (SF) and P_c (coupled $\mathscr{M}$) can be larger than 25%. This is actually caused by the Type II error in the statistical tests. Indeed, the SF is constructed with some degree of freedom allowing the users to make trade-off between Type I and Type II error, or saying, between P_w and P_c. Although we have set the default parameters based on the analysis of 16 problems, the optimal settings is never static but rather "problem sensitive". It depends not only on the τ of the investing environment, but also on the system characteristics, and the available record length. Therefore, one future research is to develop an algorithm to adjust the SF parameters according to the problem structure.

Although the P_c (SF) can be increased by inputting more data, there are some practical situations where waiting for further data is not feasible. When no significant bottleneck is observed, the SF, instead of simply giving the "do not invest" instruction, could do more to make better use of the collected data. One idea is that the SF can try to locate a subset of critical machines which may contain the bottleneck. Thus the investment can be split among these machines. This could be an interesting feature to realize for future studies.

References

R. Bartels. 1982. "The Rank Version of Von Neumann's Ratio Test for Randomness." *Journal of the American Statistical Association* 77 (377): 40-46.

Y. Benjamini, Y. Hochberg. 1995. "Controlling the False Discovery Rate: A Practical and Powerful Approach to Multiple Testing." *Journal of the Royal Statistical Society. Series B (Methodological)* 57 (1): 289-300.

C. E. Betterton, S. J. Silver. 2012. "Detecting bottlenecks in serial production lines - a focus on interdeparture time variance." *International Journal of Production Research* 50 (15): 4158-4174.

S. Biller,J. Li, S. P. Marin, S. M. Meerkov, L. Zhang. "Bottlenecks in Production Lines with Rework: A Systems Approach." *Proceedings of the 17th World Congress The International Federation of Automatic Control*. Seoul, Korea, July 6-11, 2008.

E. Jack Chen, W. David Kelton. 2007. "A Procedure for Generating Batch-Means Confidence Intervals for Simulation: Checking Independence and Normality." *Simulation* 83 : 683-694.

S.-Y. Chiang , C.-T Kuo, S.M. Meerkov. 2000. "DT-Bottlenecks in Serial Production Lines: Theory and Application." *IEEE Transactions on Robotics and Automation* 16 (5) : 567–580

S.-Y. Chiang , C.-T Kuo, S.M. Meerkov. 2001. "c-Bottlenecks in Serial Production Lines: Identification and Application." *Mathematical Problems in Engineering* 7 : 543–578.

P. Faget, U. Eriksson, F. Herrmann. 2005. "Applying Discrete Event Simulation and an Automated Bottleneck Analysis as an Aid to Detect Running Production Constraints." *Proceedings of the 2005 Winter Simulation conference*, edited by M. E. Kuhl, N. M. Steiger, F. B. Armstrong, and J. A. Joines.

Y. Hochberg. 1988. "A Sharper Bonferroni Procedure for Multiple tests of significance." *Biometrika* 75 (4): 800–802.

S. Holm. 1979. "A Simple Sequentially Rejective Multiple Test Procedure." *Journal of Statistics* 6 (2): 65–70.

G. Hommel. 1988. "A Stagewise Rejective Multiple Test Procedure Based on a Modified Bonferroni Test." *Biometrika* 75 : 383–386.

W. Hopp, M. L Spearman. 2000. "Factory Physics: Foundations of Manufacturing Management." New York:McGraw Hill.

C. -T. Kuo, J. -T. Lim, S. M. Meerkov. 1996. "Bottlenecks in Serial Production Lines: A System-theoretic Approach." *Mathematical Problems in Engineering* 2 : 233–276.

A. M. Law. 2007 "Simulation Modeling & Analysis." 508–547. New York: McGraw Hill.

S. R. Lawrence, A. H. Buss. 1994. "Shifting Production Bottlenecks: Causes, Cures, and Conundrums." *Production and Operations Management* 3 (1): 21–37.

L. Li, Jun Ni. 2009. "Short-term Decision Support System for Maintenance Task Prioritization." *International Journal of Production Economics* 121 : 195–202.

L. Li, Q. Chang, J. Ni. 2007. "Data Driven Bottleneck Detection of Manufacturing Systems." *International Journal of Production Research* 47 (18): 5019–5036.

L. Li. 2009. "Bottleneck Detection of Complex Manufacturing Systems Using a Data-driven Method." *International Journal of Production Research* 47 (24): 6929–6940.

K. M. N. Muthiah, S. H. Huang. 2009. "Overall Throughput Effectiveness (OTE) metric for Factory-level Performance Monitoring and Bottleneck Detection." *International Journal of Production Research* 45 (20): 4753–4769.

D. M. Rom. 1990. "A Sequentially Rejective Test Procedure Based on a Modified Bonferroni Inequality." *Biometrika* 77 (3) : 663–665.

C. Roser, M. Nakano, M. Tanaka. 2001. "A Practical Bottleneck Detection Method." *Proceedings of the 2001 Winter Simulation Conference*, edited by B.A Peters, J.-S. Smith, D. J. Medeiros, and M. W. Rohrer.

C. Roser, M. Nakano, M. Tanaka. 2002. "Shifting Bottleneck Detection." *Proceedings of the 2002 Winter Simulation conference*, edited by E. Y?cesan, C.-H. Chen, J. L. Snowdon, and J. M. Charnes.

S. Sengupta, K. Das, R. P. Vantil. 2008. "A New Method for Bottleneck Detection." *Proceedings of the 2008 Winter Simulation conference*, edited by S. J. Mason, R. R. Hill, L.M?nch, O. Rose, T. Jefferson, J. W.

Fowler.
R. J. Simes. 1986. “An Improved Bonferroni Procedure for Multiple Tests of Significance.” *Biometrika* 73 : 751–754.
Y. Wang, Q. Zhao, D. Zheng. 2005. “Bottlenecks in Production Networks: an Overview.” *Journal of Systems Science and Systems Engineering* 14 (3): 347–363.