Bottleneck Detection in High-variety Make-to-Order Shops with Complex Routings: An Assessment by Simulation

Matthias Thürer* (corresponding author), Lin Ma, Mark Stevenson and Christoph Roser

Name: Prof. Matthias Thürer*
Institution: Jinan University
Address: School of Intelligent Systems Science and Engineering
Jinan University (Zhuhai Campus)
519070, Zhuhai, PR China
E-mail: matthiasthurer@workloadcontrol.com

Name: Lin Ma
Institution: Jinan University
Address: School of Management
Jinan University
510632, Guangzhou, PR China
E-mail: malin15102939217@163.com

Name: Prof. Mark Stevenson
Institution: Lancaster University
Address: Department of Management Science
Lancaster University Management School
Lancaster University
LA1 4YX - U.K.
E-mail: m.stevenson@lancaster.ac.uk

Name: Prof. Christoph Roser
Institution: Karlsruhe University of Applied Sciences
Address: Karlsruhe University of Applied Sciences
Moltkestr. 30,
76133 Karlsruhe, Germany
E-mail: Christoph.Roser@hs-karlsruhe.de

Keywords: *Theory of Constraints; Capacity Planning; Workload Control; Bottleneck Analysis; Job Shop.*

Bottleneck Detection in High-variety Make-to-Order Shops with Complex Routings: An Assessment by Simulation

Abstract

This study uses simulation to assess the performance of alternative methods for detecting *momentary* bottlenecks in high-variety contexts that produce on a to-order basis. The results suggest that using the utilization level of a station to detect bottlenecks leads to the best performance, but that this method suffers from high nervousness. Using the active period of a station appears to be a better overall choice for practice given its good performance and low nervousness. Meanwhile, methods that focus on the workload at a station are a viable alternative, but they may become dysfunctional in shops with directed routings and a limit on the queue. This negative effect is even stronger if the corrected workload measure is used, as recently suggested in the literature on short term capacity adjustments. Finally, using the inter-departure time detection method leads to the worst performance since: (i) it counterintuitively detects non-bottlenecks instead of bottlenecks; and, (ii) it is based on historical data, leading to a response delay.

Keywords: *Theory of Constraints; Capacity Planning; Workload Control; Bottleneck Analysis; Job Shop.*

1. Introduction

This study uses discrete event simulation to assess different methods for detecting momentary bottlenecks in the context of high-variety make-to-order production. It seeks to identify the best-performing methods and contingency factors affecting their application in practice. The importance of bottlenecks has been recognized since the emergence of the Theory of Constraints, which received broad research attention (Ikeziri *et al.*, 2019). As part of *The Goal*, Goldratt & Cox (1984) outlined a five-step process of continuous improvement: identify the bottleneck, exploit the bottleneck, subordinate everything else to this exploitation, elevate the bottleneck and, finally, go back to step 1 to determine if the bottleneck has changed. It is apparent that the first step, to identify the bottleneck, is of utmost importance (Pehrsson *et al.*, 2016; Kahraman *et al.*, 2020). Consequently, a large literature proposing different bottleneck detection methods has emerged (Roser & Nakano, 2015; Yu & Matta, 2016). This literature however typically focusses on production lines, i.e. contexts where every job visits every station in the same sequence.

In practice, many shops have much more complex routings. This includes make-to-order versatile manufacturing companies (Lizarralde-Aiastui *et al.*, 2020) which, in contrast to repeat business customizers, often produce a high variety of products (e.g. Muda & Hendry, 2003; Hines *et al.*, 2004; Stevenson *et al.*, 2005) in a job shop-like configuration (Hendry & Kingsman, 1989; Hendry *et al.*, 1998; Stevenson *et al.*, 2005). While there are other types of shops that produce a high variety of products, it is arguably these high-variety shops that are in the most urgent need of bottleneck detection methods since they are particularly prone to shifting bottlenecks (e.g. Lawrence & Buss, 1994). While there have been studies on bottleneck detection in job shops (e.g., Zhai *et al.*, 2011), prior studies have tended to focus on a deterministic scheduling context. In contrast, the high-variety make-to-order environment considered in this study is stochastic. This kind of environment has been considered, for example, in Roser *et al.* (2002a), but the authors only assessed the performance of a single bottleneck detection method and did not present a comparison of different methods.

Roser *et al.* (2002a, 2002b) distinguished between four types of bottleneck: a momentary (or sole) bottleneck, shifting bottlenecks (i.e. the preceding and momentary bottleneck during the shifting phase), an average bottleneck, and non-bottlenecks. In repetitive environments, average, shifting, and momentary bottlenecks often overlap, and information on the average bottleneck can be used to make system improvements since the production cycle will repeat. In high-variety environments, any overlap is only temporary, and information on the average bottleneck cannot be used since the same production situation will not repeat.

A practical example is the case of the Robert Bosch GmbH presented in Roser *et al.* (2014), where it was found that information on the momentary bottleneck is the most important in dynamic and unstable shop floor environments. Meanwhile, Subramaniyan *et al.* (2016) highlighted the importance of momentary bottlenecks for production and maintenance engineers when allocating resources on a real-time basis for bottleneck machines in the context of two different automotive manufacturing companies. Yet, despite the practical importance of momentary bottlenecks, it remains mostly unknown how bottleneck detection methods perform in identifying momentary bottlenecks, with the focus of many bottleneck detection methods presented in the literature or used in industry being on average bottlenecks (Roser *et al.* 2017). There is consequently a need to explore the performance of different bottleneck detection methods in a high-variety make-to-order shop with complex routings to provide guidance to managers on which method to apply to identify momentary bottlenecks in this context.

In response, this study uses discrete event simulation to addresses the following two Research Questions (RQs):

- RQ1: What is the best-performing bottleneck detection method in high-variety make-to-order job shops?
- RQ2: Are there contingency factors that guide the applicability of the different methods in this context?

The literature is first reviewed in Section 2 to identify the different bottleneck detection methods that need to be considered in our study, including contingency factors that may impact the applicability of existing bottleneck detection methods. The simulation model used to evaluate the performance of the considered methods is then outlined in Section 3. Section 4 presents the results before they are discussed in the context of previous literature in Section 5, where the managerial implications are also outlined. Conclusions are provided in Section 6 followed by the limitations and future research directions in Section 7.

2. Literature Review

Note that from here on the term “bottleneck detection method” refers to bottleneck detection methods that focus on *momentary* bottlenecks if it is not specified otherwise. The bottleneck definition adopted in this study is first outlined in Section 2.1. Section 2.2 then reviews the literature on bottleneck detection methods to identify the methods to be included in our study. Finally, Section 2.3 identifies potential contingency factors that need to be included in our study.

2.1 Bottleneck Definition

The term "bottleneck" is widely used in research and practice. However, when prompted for a definition, few academics appear to agree on what it means (Lawrence & Buss, 1994). In this study we follow the definition given by Roser & Nakano (2015) since: (i) it focusses on system performance and is not linked to a specific measure that would predetermine the bottleneck detection method, as in Lawrence & Buss (1994); and, (ii) it is sufficiently general compared to more mathematical expressions as proposed in, for example, Li (2018). However, unlike Roser & Nakano (2015), we focus on a make-to-order shop, meaning that the throughput is only one objective – on-time delivery is also an important objective. Thus, in this study we adopt the following revised definition:

> "*Bottlenecks are processes that influence both the throughput and the delivery time performance of the entire system. The larger the influence, the more significant the bottleneck.*"

2.2 Review of Bottleneck Detection Methods

This section introduces the bottleneck detection methods to be included in our study. We will focus on a shop with high variety routings. This consequently excludes bottleneck detection methods that require consistency in upstream and downstream stations, such as the *arrow method* (e.g. Kuo *et al*., 1996), the *inactive period method* (Li *et al*., 2007), the *turning point method* (Li *et al*., 2009, Li, 2018) and the *bottleneck walk method* (Roser *et al*., 2014). The existing bottleneck detection methods suitable for the production context considered in our study will be subdivided into methods that focus on the *queue* state and methods that focus on the *station* state. The former will be discussed in Section 2.2.1 and the latter in Section 2.2.2.

2.2.1 Bottleneck Detection Methods using the Queue State

A first set of bottleneck detection methods suitable for high-variety make-to-order contexts are methods that focus on the queue in front of a station. One of the biggest advantages of these methods is that inventory is directly observable and typically easily measurable. For example, the *maximum workload method* detects the bottleneck by measuring the workload W_s of each station s, with the bottleneck being the station with the maximum workload, that is $max\ (W_1, W_2, \dots, W_n)$, with *n* stations (Law & Kelton, 1991). This is equivalent to the *queue length method,* where the only difference is that it uses a different workload measure: the number of jobs. It detects the bottlenecks by measuring the queue lengths Q_s, with the bottleneck being the station with the largest queue length, that is $max\ (Q_1, Q_2, \dots, Q_n)$

(Lawrence & Buss, 1994). According to Little's Law (Little, 1966), the above is also similar to the *waiting time method*, which considers the station with the largest waiting time in front of a station to be the bottleneck (Roser *et al*., 2001).

The above bottleneck detection methods focus on the direct load at a station. This neglects the workload that is yet to arrive at a station. Oosterman *et al.* (2000) introduced different measures of the workload that include both the direct and upstream load. For example, the corrected aggregate workload, which gives the earliest possible indication that congestion is foreseen at a certain station (Land *et al.* 2015). To calculate the corrected aggregate load, a job contributes to the load of a station upon its entry to the shop and is excluded as soon as the operation at this station is complete. The corrected aggregate load contribution of a job to the i^{th} workstation in its routing is thereby determined by $\frac{p_{ij}}{i}$, where p_{ij} is the processing time of job j at station i. The corrected workload was argued to give the best representation of the future expected direct load of a station based on the mix of routings actually present on the shop floor (Oosterman *et al*., 2000).

2.2.2 Bottleneck Detection Methods using the Station State

A second set of bottleneck detection methods suitable for high-variety make-to-order contexts focus on the actual capacity resource, i.e. the station. For example, the *utilization method* (from Hopp & Spearman, 2000) detects the bottlenecks by measuring the utilization U_s, with the bottleneck being the station with the largest utilization, that is $max\ (U_1,\ U_2, \dots, U_n)$. The utilization method neglects differences in station state over time, only looking at the long term utilization (this is the active period divided by the sum of the active and inactive period). In contrast, the *active period method* only considers the duration that a station is working without interruption (Roser *et al*., 2002a). The station with the longest active period is considered to be the bottleneck (Roser & Nakano, 2015), as this station is the least likely to be interrupted by other stations and thus dictates the overall system output (Roser *et al*., 2001).

Meanwhile, the *inter-departure time variance method* identifies the station with the smallest work-in-process inter-departure time variance as the bottleneck (Betterton & Silver, 2012). The station's coefficient of variation for departures C_d is hereby a function of its own variation C_e, its utilization u, and the variation of arrivals from the upstream stations C_a, this is $C_d^2 = u^2 C_e^2 + (1 - u^2) C_a^2$. The inter-departure time variance is based on the supposed link between the active time and starvation/blockage. Since the bottleneck is argued to have a higher active time than other stations, it will cause upstream stations to be blocked and downstream stations to be starved. The increased blocking and starving at non-bottleneck stations will cause their inter-departure time variance to be larger, and the lower blocking and starving at the bottleneck will cause its inter-departure variance to be smaller (Betterton & Silver, 2012).

2.3 Contingency Factors Influencing Applicability

The existing literature suggests one important contingency factor: *the buffer limit*. The buffer limit results in two guidelines for the application of the above bottleneck detection methods. First, existing methods that use the queue state (Section 2.2.1) may become inaccurate in systems with finite queues. Second, existing methods that use the station state (Section 2.2.2) may become inaccurate in systems with infinite queues (since there is no blocking information to signal that the downstream queue is full). However, these findings are in the context of shops with constant, directed routings, i.e. production lines. In general, most of the studies comparing different bottleneck detection methods, such as Roser & Nakano (2015) and Yu & Matta (2016), focused on shops with directed routings.

This predominant focus on directed routings in the literature leads to a second important contingency factor: *the routing characteristics*. For example, existing methods that use information on upstream and downstream stations cannot be applied if there is no dominant flow since, in this case, upstream and downstream points simply do not exist. In general, it remains largely unknown how the bottleneck detection methods identified from the literature perform in shops with more complex routings, such as high-variety make-to-order shops, and whether the above guidelines still apply. This is considered a major shortcoming given that shifting bottlenecks are more likely in high-variety contexts and, consequently, the detection of momentary bottlenecks is of utmost importance.

3. Simulation Model

The modelled shop and job characteristics are first outlined in Section 3.1. The different methods considered to identify the momentary bottleneck are then described in Section 3.2 before Section 3.3 outlines how we respond to the identified bottleneck. Section 3.4 then outlines how shop floor control is exercised. Finally, the experimental design and the performance measures used are summarized in Section 3.5.

3.1 Job and Shop Characteristics

In order to implement our first contingency factor – routing characteristics – two shop types are considered. A Pure Job Shop (PJS; Melnyk & Ragatz, 1989), which is characterized by random and undirected routings, and a General Flow Shop (GFS; Oosterman *et al.* 2000), which is characterized by random direct routings. Both shops have been implemented in the Python© programming language using the SimPy© simulation module. Further, both shops contain seven stations, where each station is a single, constant capacity resource. The routing length of jobs varies uniformly from one to seven operations. All stations have an equal probability of being visited and a particular station is required at most once in the routing of a job. This routing vector for the PJS is then sorted for the GFS, so there are typical upstream and downstream stations.

In order to implement our second contingency factor, the queue space in front of each station is limited for the GFS. Three different limits are applied: 15 jobs, 20 jobs, and infinite (i.e. no limit). For the PJS, no limit is applied to avoid the mutual blocking of stations which may occur for undirected routings (Lödding *et al.*, 2003). So, in total, four different shop types are used: GFS 15 jobs limit, GFS 20 jobs limit, GFS no limit, and PJS no limit. As is typical for make-to-order shops, there is no finished goods inventory and jobs are delivered to the customer as soon as they have been completed.

Operation processing times follow a truncated 2-Erlang distribution with a mean of 1 time unit after truncation. The inter-arrival time of jobs to the shop follows an exponential distribution with a mean of 0.572 time units, which deliberately results in a utilization level of 100% at the bottleneck without adjustment. The high momentary utilization is possible given the bottleneck shifts that are described below.

We arbitrarily create an imbalance across stations. As in previous literature (e.g. Thürer *et al.*, 2017) Non-bottlenecks are created by reducing the corresponding processing times. We experimented with three different levels of bottleneck strength: low=10%; moderate=15%; and strong=20% processing time decrease. A bottleneck can be due to the station condition, i.e. the station working slower than normal, or due to changes in job properties, i.e. the workload of incoming jobs increases. In the first case, processing times are only adjusted once the job arrives at the station. In the second case, processing times are adjusted as soon as the job arrives at the system. There is one bottleneck station. In order to evaluate how fast and accurate a bottleneck detection method identifies a bottleneck, we shift the bottleneck during a simulation run. Two settings were considered: a shift every 50 jobs and a shift every 100 jobs. All stations have the same probability of being the next bottleneck.

Finally. due dates are set exogenously by adding a uniformly distributed random allowance factor to the job entry time. This factor was set arbitrarily between 26 and 36 time units.

3.2 Bottleneck Detection Methods

From the three bottleneck detection methods that use the queue state – maximum workload method (e.g., Law & Kelton, 1991), queue length method (e.g. Lawrence & Buss, 1994), and waiting time method (e.g. Roser *et al*., 2001) – we only consider the maximum workload method. Using the queue length instead of the workload leads to two problems. First, several stations may have the same queue length (*but* different workloads). And second, a station with many jobs in the queue does not necessarily constrain the system if these jobs are relatively small and there are fewer but much larger jobs at another station (Roser *et al*., 2003). Five different bottleneck detection methods – four from the literature and one newly developed in this study – will consequently be considered as follows:

- *Maximum Workload*: where the station with the maximum workload in the queue is the bottleneck.
- *Utilization*: where the station with the highest utilization is the bottleneck.
- *Active Period*: where the station with the longest uninterrupted active period is the momentary bottleneck.
- *Inter-Departure Time*: where the station with the lowest inter-departure time variance is the momentary bottleneck.
- *Corrected Workload*: where the station with the maximum corrected workload is the bottleneck.

Finally, the utilization method and the inter-departure time method require a time frame over which the utilization rates and inter-departure times are calculated. In this study, we considered the last 30 and the last 50 time units.

3.3 Response to Bottleneck: Capacity Adjustments

There is no possibility of accurately predicting what is the ‘real’ bottleneck since this itself would require a bottleneck detection method, which then per definition would be the best performing. In order to avoid this confirmation bias (Pohl, 2004), we measure the actual impact on system performance. This will be achieved by adjusting capacity at the station identified as the bottleneck by the five different bottleneck detection methods described above. There will be a 5% reduction in the processing time (after a possible adjustment to create the bottleneck).

3.4 Shop Floor Control – Priority Dispatching

As in previous studies on bottleneck detection, it is assumed that all jobs are accepted, materials are available, and all necessary information regarding shop floor routings, processing times, etc. is known. Jobs are released immediately to the shop floor on arrival. Jobs in the queues are prioritized according to operation due dates, which are calculated by backward scheduling from the due date. In this study, the allowance for the operation throughput times is given by the cumulative moving average, i.e. the average of all operation throughput times realized until the current simulation time.

3.5 Experimental Design and Performance Measures

The experimental factors are summarized in Table 1. A full factorial design was used with 336 scenarios (4x3x2x2x7x1), where each scenario was replicated 100 times. Results were collected over 10,000 time units following a warm-up period of 3,000 time units. These parameters allowed us to obtain stable results while keeping the simulation run time to a reasonable level.

Table 1: Summary of Experimental Factors

Experimental Factor	Level
Shop Type (4 level)	GFS (General Flow Shop) 15 jobs limit, GFS 20 jobs limit, GFS no limit and PJS (Pure Job Shop) no limit.
Bottleneck Strength (3 level)	low=10%; moderate=15%; and strong=20%
Cause of Bottleneck (2 level)	station and arriving workload
Timing of Bottleneck Shift (2 level)	Every 50 or every 100 jobs
Bottleneck Detection Methods (7 level)	Maximum Workload, Utilization (time frame 30 and 50 time units), Active Period, Inter-Departure Time (time frame 30 and 50 time units) and Corrected Workload
Capacity Adjustment Factor (1 level)	5%

Bottlenecks were defined in this study as processes that influence the throughput and the delivery performance of the entire system. To assess throughput performance, we measure the *mean lead time* – i.e. the mean of the completion date minus the arrival date across jobs. Please note that the fixed arrival rates in the simulation determine the total throughput that can be realized. Therefore, the throughput improvement capabilities of a bottleneck detection method will manifest in shorter lead times, rather than increased throughput. Delivery performance will be measured by: the *percentage tardy* – the percentage of jobs completed after the due date; and, the *mean tardiness* – that is, $T_j = max(0, L_j)$, with L_j being the lateness of job *j* (i.e. the actual delivery date minus the due date of job *j*).

4. Results

To obtain a first indication of the relative impact of the experimental factors, statistical analysis has been conducted by applying an Analysis of Variance (ANOVA). ANOVA is here based on a block design, which is typically used to account for known sources of variation in an experiment. In our ANOVA, we treat the shop type as the blocking factor. This allows the main effects of this factor and the main and interaction effects of our four factors related to bottlenecks and bottleneck detection – bottleneck strength, count between bottleneck shifts, location of bottleneck occurrence (upon arrival or at the station), and bottleneck detection method – to be captured. We do not present detailed results due to space limitations. All main effects were found to be statistically significant, as were about half of the two-way interactions. There were no significant three-way or four-way interactions.

The Scheffé multiple comparison procedure, which is arguably the most conservative of the commonly available *post-hoc* tests (Spurrier, 1999), was applied to obtain a first indication of the direction and size of the performance differences across bottleneck detection methods. Table 2 gives the 95% confidence interval. If this interval includes zero, performance differences are not considered to be statistically significant. We can observe significant performance differences for most pairs for at least one performance measure. Detailed performance results to further explore these differences will be presented next.

Table 2: Results for the Scheffé Multiple Comparison Procedure

Bottleneck Detection Method (x)	Method (y)	Lead Time lower[1)]	Lead Time upper	Percentage Tardy lower	Percentage Tardy upper	Mean Tardiness lower	Mean Tardiness upper
Utilization 30	Max Workload	-2.31	-0.59	-0.03	-0.01	-1.96	-0.32
Utilization 50	Max Workload	-2.00	-0.27	-0.02	-0.01	-1.86	-0.23
Active Period	Max Workload	-1.00	0.72	0.00*	0.02	-1.16*	0.48
Inter-Departure 30	Max Workload	3.00	4.72	0.05	0.07	1.86	3.50
Inter-Departure 50	Max Workload	3.04	4.76	0.06	0.07	1.89	3.52
Corrected Workload	Max Workload	1.14	2.86	0.00*	0.01	1.11	2.75
Utilization 50	Utilization 30	-0.55*	1.17	0.00*	0.01	-0.72*	0.92
Active Period	Utilization 30	0.44	2.17	0.02	0.03	-0.02*	1.62
Inter-Departure 30	Utilization 30	4.44	6.17	0.07	0.08	3.00	4.64
Inter-Departure 50	Utilization 30	4.49	6.21	0.07	0.09	3.03	4.66
Corrected Workload	Utilization 30	2.59	4.31	0.02	0.03	2.25	3.89
Active Period	Utilization 50	0.13	1.86	0.02	0.03	-0.11*	1.52
Inter-Departure 30	Utilization 50	4.13	5.86	0.07	0.08	2.91	4.54
Inter-Departure 50	Utilization 50	4.17	5.90	0.07	0.08	2.93	4.56
Corrected Workload	Utilization 50	2.27	4.00	0.01	0.03	2.16	3.79
Inter-Departure 30	Active Period	3.14	4.86	0.04	0.06	2.20	3.84
Inter-Departure 50	Active Period	3.18	4.90	0.05	0.06	2.23	3.86
Corrected Workload	Active Period	1.28	3.00	-0.01*	0.00	1.45	3.09
Inter-Departure 50	Inter-Departure 30	-0.82*	0.90	0.00*	0.01	-0.79*	0.84
Corrected Workload	Inter-Departure 30	-2.72	-1.00	-0.06	-0.05	-1.57*	0.07
Corrected Workload	Inter-Departure 50	-2.76	-1.04	-0.06	-0.05	-1.59*	0.04

[1)] 95% confidence interval; * not significant at α=0.05

4.1 Bottleneck Detection Method in PJS and GFS without Queue Limit (Blocking)

Results for the PJS and GFS without a queue limit are presented in Table 3. Table 3 gives the lead time, percentage tardy, and mean tardiness together with the number of changes in the detected bottleneck per 100 time units. The latter measure is used to assess the nervousness of the bottleneck detection methods.

Table 3: Results for "Strong" Bottleneck Strength in the Pure Job Shop and General Flow Shop without a Limit on the Queue

Bottleneck		Bottleneck Detection	Pure Job Shop (PJS)				General Flow Shop (GFS)			
When?	Where?		LT[1]	PT[2]	MT[3]	Change[4]	LT[1]	PT[2]	MT[3]	Change[4]
Every 50	arrival	Max Workload	17.17	5.8%	0.26	60.57	17.49	8.0%	0.46	57.43
		Utilization 30	17.02	5.7%	0.26	122.88	17.32	7.7%	0.44	127.22
		Utilization 50	17.24	6.1%	0.28	83.07	17.58	8.2%	0.47	87.80
		Active Period	17.47	6.9%	0.34	4.88	17.91	9.5%	0.59	4.46
		Inter-Departure 30	18.45	9.7%	0.54	34.96	18.90	12.2%	0.84	33.41
		Inter-Departure 50	18.45	9.8%	0.54	27.18	18.92	12.3%	0.85	25.54
		Corrected Load	17.10	5.6%	0.25	47.91	17.44	7.9%	0.46	44.22
	station	Max Workload	16.93	5.1%	0.21	62.48	17.19	7.1%	0.39	59.13
		Utilization 30	16.77	5.0%	0.21	121.45	17.02	6.8%	0.37	125.66
		Utilization 50	16.99	5.3%	0.23	81.29	17.27	7.3%	0.40	86.02
		Active Period	17.20	6.1%	0.28	5.05	17.56	8.4%	0.50	4.62
		Inter-Departure 30	18.08	8.5%	0.44	34.95	18.46	10.8%	0.70	33.48
		Inter-Departure 50	18.10	8.6%	0.45	27.02	18.49	10.9%	0.70	25.67
		Corrected Load	16.89	5.0%	0.21	49.65	17.15	7.1%	0.40	46.00
Every 100	arrival	Max Workload	17.73	7.5%	0.40	56.53	18.10	9.8%	0.63	52.91
		Utilization 30	17.59	7.4%	0.41	124.02	17.95	9.6%	0.62	128.78
		Utilization 50	17.81	7.9%	0.43	83.24	18.22	10.2%	0.66	89.10
		Active Period	18.11	8.9%	0.52	4.49	18.63	11.6%	0.81	3.99
		Inter-Departure 30	19.29	12.5%	0.83	34.72	19.89	15.5%	1.21	33.05
		Inter-Departure 50	19.30	12.6%	0.83	26.45	19.92	15.5%	1.21	24.97
		Corrected Load	17.68	7.3%	0.39	45.10	18.06	9.8%	0.63	41.54
	station	Max Workload	17.51	6.7%	0.34	58.76	17.84	9.0%	0.54	54.81
		Utilization 30	17.33	6.5%	0.34	122.95	17.65	8.7%	0.52	127.49
		Utilization 50	17.56	7.0%	0.36	81.67	17.90	9.3%	0.56	87.62
		Active Period	17.83	8.0%	0.44	4.62	18.28	10.6%	0.69	4.12
		Inter-Departure 30	18.89	11.1%	0.69	34.95	19.40	13.9%	1.01	33.21
		Inter-Departure 50	18.91	11.2%	0.69	26.66	19.44	14.0%	1.02	24.98
		Corrected Load	17.49	6.7%	0.34	47.35	17.83	9.1%	0.55	43.60

LT[1] – Lead Time; PT[2] – Percentage Tardy; MT[3] – Mean Tardiness; Change[4] – Changes in Detected Bottleneck per 100 time units

The results in Table 3 show that the *Max Workload*, *Utilization*, and *Corrected Workload* methods perform equivalent and the best across all of the bottleneck detection methods. However, the *Active Period* method excels in terms of nervousness. That is, it is by far the least nervous method. It is also the method with the least information requirements since it only needs to recall the last inactive period (i.e. one data point). The high nervousness of the *Utilization* method can be explained by the calculation of the utilization rate, which does not consider partly processed jobs at the beginning and the end of the period that is used for the calculation.

Meanwhile, the *Inter-Departure Time* method consistently performs the worst. If all stations are active, i.e. there is at least one job to be processed all of the time then the inter-departure time variance reflects the variability of the processing time. Since bottlenecks are created by reducing the processing times at non bottleneck stations, the variability at the set bottleneck is necessarily higher. Since the *Inter-Departure Time* method identifies the station with the lowest inter-departure time as the bottleneck, it counter-intuitively identifies the non-bottleneck. Note that Betterton & Silver (2012) also used the same coefficient of variation, this is the standard deviation divided by the mean; where higher means (set bottleneck) imply higher standard deviations (identified as non-bottlenecks). Hence, the *Inter-Departure Time* method appears to be dependent on the occurrence of blocking and starvation.

This dependence on the occurrence of blocking and starvation could also be argued to hold for the *Utilization* and *Active Period* methods. Having a small stable direct load over a time period will result in a 100% utilization and the maximum active period. Only disruptions provide discrimination across stations. However, disruptions do not necessarily result in longer operation throughput times. Yet it is the operation throughput times that determine the lead time and consequently delivery performance in our make-to-order context. It follows that the *Utilization* and *Active Period* methods perform well in our study because there is a direct link between a station that is active or utilized and the queue length.

Finally, the above conclusions are robust to all three environmental factors, i.e. routing direction (PJS *vs*. GFS), the time between bottleneck shifts, and whether a bottleneck occurs at arrival (e.g. due to an increase in the incoming workload) or at the station (e.g. due to reduced speed). The impact of our last environmental factor – the queue limit – will be provided in Section 4.2. But first, a more in-depth analysis of the results will be provided.

4.1.1 Analysis of the Results

To further explore the performance differences across bottleneck detection methods, we recorded when a station was detected as the bottleneck, when a station was set as the bottleneck (at arrival), and the direct load for an arbitrary station in the pure job shop. We used the PJS since here all stations show a similar pattern. Figure 1 gives the results for 5,000 time units (from 3,000 to 8,000) during an arbitrary simulation run. Note that we do not present results for all bottleneck detection methods to avoid redundancies.

From Figure 1 we can observe that there are three distinct time periods: Period 1, around 4,100 time units, is a period where the high direct load cannot be associated with the set bottleneck; Period 2, around 6,000 time units, is a period where there is a clear association between the set bottleneck and a high direct load; and, Period 3, around 6,500 to 7,000 time units, is a period in which being the set bottleneck results in a much lower direct load than Period 1. This highlights that there is no clear link between being the set bottleneck and the actual realized queue length in environments with high-variety routings and stochastic processing times. This discrepancy becomes even greater in the GFS, where upstream stations are more likely to constrain the system.

The *Max Workload*, *Utilization,* and *Active Period* methods react appropriately to all three periods of high direct load (by identifying the station as a bottleneck). In general, all three methods show a similar pattern in terms of bottleneck detection. However, the *Inter Departure Time* method appears to neglect the high direct load in Period 1 and Period 2. This is because it does not consider the queue length and, as argued above, detects non-bottlenecks instead of bottlenecks in the considered production context. Meanwhile, the *Active Period* method creates the most stable periods where the station is detected as a bottleneck, as was expected from the results in Table 3. This results in the lowest direct load during the three high load periods; however, from Table 3, it was apparent that it does not result in the best performance. Imagine another station that has a higher direct load but where the high load period only starts after the current station. This station will not be identified as the bottleneck by the *Active Period* method. It will be identified as the bottleneck by the *Utilization* method for some periods of time given the discreteness in the calculation of the utilization. It will also be identified as a bottleneck by the *Max Workload* method, which directly focusses on the direct load.

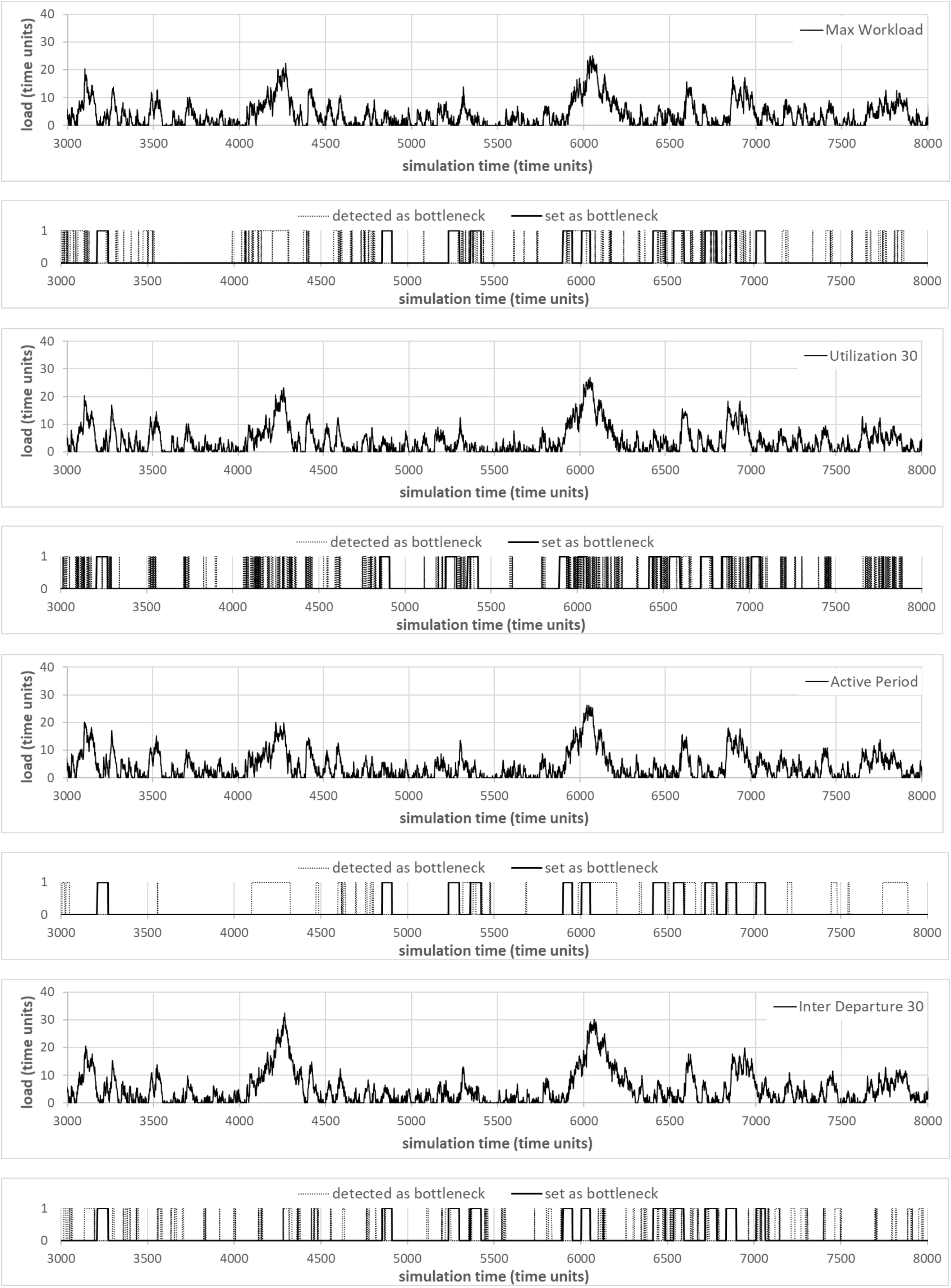

Figure 1: Performance Results Over Time for: Station Detected as the Bottleneck, Station Set as the Bottleneck, and Direct Load

4.2 Bottleneck Detection Method in the GFS with a Queue Limit (Blocking)

Table 4 provides the results for the general flow shop with a queue limit, i.e. scenarios where blocking may occur. Blocking is hereby defined as the situation where a job, having completed all of its processing requirements at a station, must remain at the station (and thus blocks station capacity) until space in the queue at the next station in its routing becomes available (Roser *et al.*, 2014).

While there is a general deterioration in performance if there is a queue limit, performance differences across bottleneck detection methods appear to be unaffected by the existence of a queue limit at moderate bottleneck strength. The same holds for strong bottlenecks. However there is a change if bottleneck strength is low, as can be seen from Table 5, which gives the results for low bottleneck strength and a bottleneck shift every 50 jobs. In Table 5, we observe a strong deterioration for *Max Workload* and *Corrected Workload*. This will be explored next.

Table 4: Results for "Strong" Bottleneck Strength in the General Flow Shop with the Limit on the Queue Set to 20 and 15 Jobs

Bottleneck		Bottleneck Detection	GFS Limit 20 jobs				GFS Limit 15 jobs			
When?	Where?		LT$^{1)}$	PT$^{2)}$	MT$^{3)}$	Change$^{4)}$	LT$^{1)}$	PT$^{2)}$	MT$^{3)}$	Change$^{4)}$
Every 50	arrival	Max Workload	17.80	8.9%	0.61	57.69	18.73	11.6%	1.01	58.92
		Utilization 30	17.59	8.5%	0.54	126.48	18.30	10.4%	0.81	124.62
		Utilization 50	17.88	9.2%	0.59	86.35	18.64	11.3%	0.87	83.18
		Active Period	18.22	10.4%	0.72	4.45	19.05	12.6%	1.05	4.41
		Inter-Departure 30	19.57	14.1%	1.16	33.62	21.07	18.1%	1.89	33.81
		Inter-Departure 50	19.58	14.2%	1.16	25.65	21.11	18.2%	1.90	25.95
		Corrected Load	17.76	8.9%	0.61	44.20	18.88	11.9%	1.16	43.32
	station	Max Workload	17.43	7.9%	0.50	59.51	18.28	10.5%	0.84	60.47
		Utilization 30	17.22	7.4%	0.44	124.94	17.84	9.2%	0.65	123.22
		Utilization 50	17.50	8.0%	0.49	84.74	18.26	10.2%	0.76	81.92
		Active Period	17.82	9.2%	0.58	4.62	18.53	11.2%	0.85	4.59
		Inter-Departure 30	19.00	12.4%	0.94	33.67	20.34	15.9%	1.56	33.72
		Inter-Departure 50	19.03	12.5%	0.94	25.81	20.37	16.1%	1.57	25.92
		Corrected Load	17.42	7.9%	0.52	45.99	18.39	10.9%	0.94	45.12
Every 100	arrival	Max Workload	18.65	11.4%	0.89	53.47	20.10	15.4%	1.59	54.75
		Utilization 30	18.44	11.1%	0.82	127.83	19.46	13.8%	1.24	125.78
		Utilization 50	18.72	11.7%	0.86	87.28	19.90	14.9%	1.37	83.93
		Active Period	19.19	13.3%	1.05	3.99	20.39	16.4%	1.59	3.93
		Inter-Departure 30	21.17	18.9%	1.87	38.65	23.40	24.3%	3.10	33.32
		Inter-Departure 50	21.18	19.0%	1.86	25.07	23.50	24.5%	3.17	25.36
		Corrected Load	18.61	11.4%	0.90	41.35	20.43	16.0%	1.88	39.63
	station	Max Workload	18.31	10.4%	0.77	55.35	19.64	14.2%	1.38	56.75
		Utilization 30	18.01	9.9%	0.66	126.52	18.94	12.3%	1.02	124.71
		Utilization 50	18.32	10.6%	0.72	85.79	19.36	13.3%	1.16	82.57
		Active Period	18.69	11.8%	0.85	4.10	19.81	14.6%	1.36	4.05
		Inter-Departure 30	20.32	16.5%	1.44	33.42	22.45	21.8%	2.60	33.28
		Inter-Departure 50	20.36	16.6%	1.47	25.23	22.48	21.8%	2.60	25.42
		Corrected Load	18.31	10.5%	0.78	43.54	19.88	14.6%	1.58	41.98

LT$^{1)}$ – Lead Time; PT$^{2)}$ – Percentage Tardy; MT$^{3)}$ – Mean Tardiness; Change$^{4)}$ – Changes in Detected Bottleneck per 100 time units

Table 5: Results for Low Bottleneck Strength in General Flow Shops with the Limit on the Queue Set to 15 and a Bottleneck Shift every 50 Jobs

	Arrival				Station			
	LT[1)]	PT[2)]	MT[3)]	Change[4)]	LT[1)]	PT[2)]	MT[3)]	Change[4)]
Max Workload	42.14	52.6%	17.11	48.57	40.33	50.7%	15.53	49.07
Utilization 30	31.74	40.8%	8.17	164.68	31.53	40.0%	8.07	163.98
Utilization 50	32.86	42.7%	8.99	125.91	32.02	41.1%	8.35	124.59
Active Period	37.54	49.2%	12.82	2.24	36.63	47.9%	12.09	2.28
Inter-Departure 30	62.41	64.7%	35.79	32.98	60.07	62.9%	33.69	33.15
Inter-Departure 50	62.19	65.0%	35.54	24.96	58.26	63.3%	31.81	24.83
Corrected Load	64.56	60.6%	38.56	18.36	58.48	58.8%	32.74	19.52

LT[1)] – Lead Time; PT[2)] – Percentage Tardy; MT[3)] – Mean Tardiness; Change[4)] – Changes in Detected Bottleneck per 100 time units

4.2.1 Analysis of Results

To better understand the aforementioned performance deterioration at low bottleneck strength, we recorded the blocking time, the number of occurrences of blocking, and the number of jobs currently queuing at the station that is blocked. The results for Station 1, Station 2, and Station 3 are provided in Table 6. We also recorded the percentage of time a station was identified as a bottleneck by a given method. These results are given in Table 7.

Table 6: Blocking Analysis for General Flow Shop, Queue Limit 15, Low Bottleneck Strength, Bottleneck Shift Every 50 Jobs, and Bottleneck Occurrence at Arrival – Release Blocking at Station 1, Station 2, and Station 3

	Release		Station 1			Station 2			Station 3		
	Dur.[1)]	Count[2)]	Dur.	Count	Load[3)]	Dur.	Count	Load	Dur.	Count	Load
Max Workload	0.9	6817	1.1	635	14.1	1.0	433	9.6	1.0	303	8.6
Utilization 30	0.9	5077	1.1	444	12.8	1.0	318	8.6	1.0	231	7.8
Utilization 50	0.9	5340	1.1	461	13.0	1.0	324	8.7	1.0	242	7.9
Active Period	0.9	6536	1.1	528	13.6	1.0	361	8.9	1.0	262	8.1
Inter-Departure 30	0.9	9685	1.2	747	14.7	1.1	512	9.6	1.1	365	8.6
Inter-Departure 50	0.9	9787	1.2	756	14.7	1.1	514	9.6	1.1	367	8.5
Corrected Workload	0.9	8460	1.1	885	14.7	1.1	524	10.3	1.1	359	8.8

Dur.[1)] - average blockage duration; Count[2)] - average occurrences per 10.000 time units; Load[3)] - average number of jobs queuing at blocked station when blocking occurred

Table 7: Analysis of Bottleneck Detection for General Flow Shop, Queue Limit 15, Low Bottleneck Strength, Bottleneck Shift Every 50 Jobs, and Bottleneck Occurrence at Arrival – Percentage of Time Identified as the Bottleneck

	Station 1	Station 2	Station 3	Station 4	Station 5	Station 6	Station 7
Max Workload	36.5%	13.7%	11.5%	10.3%	9.3%	7.8%	10.9%
Utilization 30	14.1%	13.9%	14.3%	14.5%	14.7%	14.2%	14.4%
Utilization 50	14.2%	13.9%	14.3%	14.7%	14.8%	14.1%	14.2%
Active Period	15.4%	13.8%	14.3%	14.7%	14.5%	13.7%	13.6%
Inter-Departure 30	17.0%	14.4%	13.7%	13.3%	13.4%	13.4%	14.7%
Inter-Departure 50	17.6%	14.4%	13.7%	13.1%	13.1%	13.1%	14.9%
Corrected Workload	66.7%	8.7%	6.6%	5.4%	4.5%	3.7%	4.4%

Two observations can be made from the blocking analysis in Table 6. First, most of the blocking is so-called ‘release blocking’, i.e. a job cannot enter the shop floor before the queue limit at the first station in its routing has been reached. Second, the *Max Workload* and the *Corrected Workload* methods have the highest occurrences of blocking. Meanwhile, Table 7 highlights an overemphasis on Station 1, which is much more often identified as the bottleneck than other stations by the *Max Workload* and *Corrected Workload* methods.

Station 1 is a gateway station in the GFS. This function as a gateway station is strengthened if a queue limit is applied. In this case, Station 1 acts as an order release function for most jobs, controlling when a job can enter the shop floor and consequently when it can arrive at a downstream station. As a result, queues are longer at Station 1, which leads to the overemphasis observed in Table 7. Since capacity adjustments are focused on Station 1, downstream stations have a higher relative average utilization rate. This in turn leads to higher congestion in the system and to more blocking (specifically at Station 1). However, the blocking does not trigger a shift in the detected bottleneck for *Max Workload* and *Corrected Workload*, rather the contrary. Thus, system congestion remains and performance deteriorates. Note that this effect only occurs if the average utilization of the system is already high, as is the case for low bottleneck strength. In contrast, both the *Active Period* and *Utilization* methods react to blocking and the associated disruption to the active period. Finally, the effect for the *Corrected Load* method is higher since it is calculated upon arrival at the shop and thus before release blocking occurs. Hence, the workload correction introduces an additional emphasis on the first station in the routing of a job compared to using the direct workload queuing at a station, as is the case for the *Max Workload* method.

Finally, the *Inter-Departure Time* method also becomes dysfunctional in a shop with blocking. However, in this case the main issue is the time delay that is necessarily introduced to calculate the inter-departure time. Imagine a scenario where Station 1 is identified as the bottleneck leading to an increased output until Station 1 is blocked since the queue at Station 2 is full. Station 2 should now become the bottleneck, but the inter-departure time is necessarily based on past data. As a result, Station 1 remains the identified bottleneck until the blocking is reflected in the dataset used for calculating the inter-departure time variance, and thus capacity is adjusted at Station 1, although it is blocked. This in turn leads to more blocking, as can be observed from Table 6.

5. Discussion

The previous section presented the results from our simulation experiments. This section discusses the main contributions of our work to research and practice.

5.1 Research Implications

This study started by asking: What is the best-performing bottleneck detection method in high-variety make-to-order job shops? Using discrete event simulation it was found that the *Utilization* method is arguably the best performing method in our study, but it is very nervous. From a practical perspective, the *Active Period* method appears to be a better choice given its good performance and very low nervousness. Methods focusing on the workload are viable best-of-both-world alternatives; however, they may become dysfunctional in shops with directed routings and a queue limit that leads to blocking given their overemphasis on the gateway station, which naturally has the largest average queue. This negative effect is even stronger for the *Corrected Workload* method, which was based on recent literature on short-term capacity adjustments (Land *et al*., 2015). Finally, the *Inter-Departure Time* method consistently leads to the worst performance. If there is a fairly stable load in front of a station, and consequently processing time and inter-departure time distributions overlap, the inter-departure time method identifies non-bottlenecks instead of bottlenecks if the coefficient of variation is the same for bottlenecks and non-bottlenecks. In this case, higher means (i.e. a set bottleneck) imply higher standard deviations (identified as non-bottlenecks). Meanwhile, if there is blocking then the *Inter-Departure Time* method may identify a blocked station as a bottleneck station given the time delay that is necessarily introduced because the inter-departure time needs to be calculated based on historical data.

The active period method is commonly argued to be the best performing bottleneck detection method. It is consequently the most applied in the context of data-driven implementations; for example, by: Zhai *et al*. (2011), who used the active periods in the optimal schedule, Subramaniyan *et al*. (2018), who used auto-regressive integrated moving average (ARIMA) models based on active periods, and Subramaniyan *et al*. (2020), who used hierarchical clustering based on active periods. The question remains: why does the utilization method perform better than the active period method in our study? Imagine a situation where Station 1 precedes Station 2, i.e. all jobs move from Station 1 to Station 2, the queues are infinite, and there is a continuous arrival rate. If the processing time of jobs at Station 2 is larger than at Station 1 then Station 2 should be the bottleneck. It should be the bottleneck since the queue at Station 2 increases much faster, and since adding capacity at Station 1 even increases the rate at which the queue at Station 2 increases. Yet Station 1 will be identified as the bottleneck since Station 1 precedes Station 2 and thus has an earlier start time for the active period. In theory, the utilization method suffers from the same weakness, but since we do not include the partial processing time of a job currently being processed at a station in the calculations, our utilization method shifts between Station 1 and Station 2. Note that this effect does not occur if the queue size is limited, since Station 1 will become blocked and thus Station 2 will have the longest active period. Meanwhile, another main advantage of the active period method over the utilization method is the identification of the strength of bottlenecks, and thus secondary bottlenecks (Roser *et al*., 2003; Roser & Nakano, 2015).

5.2 Managerial Implications

The above phenomenon leads directly to our second research question: Are there contingency factors that guide the applicability of the different methods in this context? The first contingency factor considered was the buffer limit. If we revisit the two general guidelines that emerged out of the literature review, then we confirm our first guideline that existing methods that use the queue state (Section 2.2.1) may become inaccurate in systems with finite queues. However, the reason behind this is not restricted discrimination across queues, but rather the overemphasis on the gateway station, which may lead to dysfunctional behavior in highly congested shops. There is typically no issue with discrimination given that the queue limit is just an upper bound. Meanwhile, our results do not confirm our second guideline that existing methods that use the station state (Section 2.2.2) may become inaccurate in systems with infinite queues (since there is no blocking information to signal that the downstream queue is full). The main reason for this is that there is a direct link between the active period and queue length in our study. Therefore, if this link exists, then bottleneck detection methods that focus on the station state can be considered a better choice than methods that focus on the queue state.

The second contingency factor considered was the routing direction. We found that the relative performance ranking of the different bottleneck detection methods was not affected by this factor. Moreover, results align with previous literature in the context of pure flow shops, i.e. constant directed routings (e.g. Roser & Nakano, 2015). Thus, the main impact of routing characteristics is that a random routing excludes bottleneck detection methods that require consistency in upstream and downstream stations, such as the arrow method (e.g. Kuo *et al.*, 1996), the inactive period method (Li *et al.*, 2007), the turning point method (Li *et al.*, 2009, Li, 2018) and the bottleneck walk method (Roser *et al.*, 2014).

6. Conclusions

Bottleneck detection is a first step in bottleneck management, leading to a large literature proposing different bottleneck detection methods. This literature however typically focusses on production lines, i.e. contexts where every job visits every station in the same sequence. This neglects shops with more complex routings. In response, this study has assessed the performance of five different bottleneck detection methods in a high-variety make-to-order shop considering two important contingency factors identified from the literature: the buffer limit and the routing characteristics. Results indicate that the active period or the utilization method is a better choice than a bottleneck detection method that focuses on the queue state. This relative ranking of bottleneck detection methods is also not affected by routing characteristics, which provides important guidelines for management on which bottleneck detection method to apply in which production context. For example, bottleneck detection methods that focus on the queue state should not be applied in shops with a finite buffer size, while the main constraint in terms of routing is that random routings exclude bottleneck detection methods that require consistency in terms of the upstream and downstream station in the routings.

7. Limitations and Future Research

A main limitation of our study is that we neglected the actual exploitation of the bottleneck (Step 2 and Step3) and directly jumped to its elevation (Step 4). Future research could explore the link between bottleneck detection and, for example, the Drum-Buffer-Rope approach (e.g., Darlington *et al.*, 2015) or Constant Load (Bagni *et al.*, 2020) that focusses on bottleneck exploitation. Meanwhile, we also did not consider limits on the finished goods inventory, while demand was the bottleneck for a significant amount of time. Specifically, the latter calls for more research, potentially linking bottleneck detection to the job entry or customer enquiry stage where the job acceptance decisions are made and consequently demand is realized. Our focus has been on the actual operational impact of bottleneck detection methods on the shop floor. Finally, future research could also seek to develop new bottleneck detection methods. We saw that bottleneck detection methods can be subdivided according to the measure used, and a bottleneck is necessarily defined in terms of the chosen measure. Thus, a first step is to define the objective of the system and how this is measured. For example, in our make-to-order system the main objective is delivery performance (primary measure) rather than throughput (secondary measure), whereas throughput is the primary measure in most of the previous literature on bottleneck detection. Developing new bottleneck detection methods for so-called lateness bottlenecks (Fang *et al.*, 2020) is a promising avenue for future research.

References

Bagni, G., Godinho Filho, M., Thürer, M., and Stevenson, M., 2020, Systematic Review and Discussion of Production Control Systems that emerged between 1999 and 2018, *Production Planning & Control*, (in print).

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

Darlington, J., Francis, M., Found, P. and Thomas, A., 2015, Design and implementation of a Drum-Buffer-Rope pull-system, *Production Planning & Control*, *26*, 6, 489-504.

Fang, W., Guo, Y., Liao, W., Huang, S., Yang, N. and Liu, J., 2020, A Parallel Gated Recurrent Units (P-GRUs) network for the shifting lateness bottleneck prediction in make-to-order production system, *Computers & Industrial Engineering*, 140, 106246.

Goldratt, E.M., and Cox, J., 1984, *The Goal: Excellence in Manufacturing*, North River Press: New York.

Hendry, L.C., and Kingsman B.G., 1989, Production planning systems and their applicability to make-to-order companies, *European Journal of Operational Research*, 40, 1–15.

Hendry, L.C., Kingsman, B.G., and Cheung, P., 1998, The effect of workload control (WLC) on performance in make-to-order companies, *Journal of Operations Management*, 16, 63–75.

Hines, P., Holweg, M., and Rich, N., 2004, Learning to evolve: A review of contemporary lean thinking, *International Journal of Operations and Production Management*, 24, 10, 994–1011.

Hopp, W.J., and Spearman, M.L., 2000, *Factory physics*, 2nd ed. New York, NY: McGraw-Hill.

Ikeziri, L.M., Souza, F.B.D., Gupta, M.C. and de Camargo Fiorini, P., 2019, Theory of constraints: review and bibliometric analysis, *International Journal of Production Research*, *57*, 15-16, 5068-5102.

Kahraman, M.M., Rogers, W.P., and Dessureault, S., 2021, Bottleneck identification and ranking model for mine operations, *Production Planning & Control*, (in print)

Kuo, C.-T., Lim, J.-T., and Meerkov, S. M., 1996, Bottlenecks in serial production lines: a system-theoretic approach, *Mathematical Problems in Engineering*, 2, 3, 233–276.

Land, M.J., Stevenson, M., Thürer, M., and Gaalman, G.J.C., 2015, Job Shop Control: In Search of the Key to Delivery Improvements, *International Journal of Production Economics*, 168, 257–266.

Law, A.M., and Kelton, W.D., 1991, *Simulation modeling and analysis*, New York: McGraw-Hill.

Lawrence, S. R., and Buss, A. H., 1994, Shifting production bottlenecks: causes, cures, and conundrums, *Production & Operations Management*, 3, 1, 21–37.

Li, L., Chang, Q., Ni, J., Xiao, G., and Biller, S., 2007, Bottleneck detection of manufacturing systems using data driven method, In *2007 IEEE international symposium on assembly and manufacturing* (pp. 76–81). IEEE.

Li, L., Chang, Q., and Ni, J., 2009, Data driven bottleneck detection of manufacturing systems, *International Journal of Production Research*, 47, 18, 5019–5036.

Li, L., 2018, A systematic-theoretic analysis of data-driven throughput bottleneck detection of production systems, *Journal of Manufacturing Systems*, 47, 43–52.

Little, J., 1961. A proof of the theorem L = λW. *Operations Research* 8, 383-387.

Lizarralde Aiastui, A., Apaolaza Pérez de Eulate, U. and Mediavilla Guisasola, M., 2020, A strategic approach for bottleneck identification in make-to-order environments: A drum-buffer-rope action research based case study, *Journal of Industrial Engineering and Management*, 13, 1, 18-37.

Lödding, H., Yu, K.-W., and Wiendahl, H.-P., 2003, Decentralized WIP-oriented manufacturing control (DEWIP), *Production Planning & Control*, 14, 1, 42–54.

Melnyk, S.A., and Ragatz, G.L., 1989, Order review/release: research issues and perspectives, *International Journal of Production Research*, 27, 7, 108–096.

Muda, M., and Hendry, L., 2003, The SHEN model for MTO SMEs: A performance improvement tool, *International Journal of Operations & Production Management*, 235, 470–486.

Oosterman, B., Land, M.J., and Gaalman, G., 2000, The influence of shop characteristics on workload control, *International Journal of Production Economics*, 68, 1, 107–119.

Pehrsson, L., Ng, A.H., and Bernedixen, J., 2016, Automatic identification of constraints and improvement actions in production systems using multi-objective optimization and post-optimality analysis, *Journal of Manufacturing Systems*, 39, 24-37.

Pohl, R., 2004, *Cognitive illusions: A handbook on fallacies and biases in thinking, judgement and memory*. Psychology Press.

Roser, C., Nakano, M., and Tanaka, M., 2001, A practical bottleneck detection method, In *Proceedings of the 33nd conference on Winter simulation* (pp. 949-953), IEEE Computer Society.

Roser, C., Nakano, M., and Tanaka, M., 2002a, Shifting bottleneck detection, In *Proceedings of the 34th conference on Winter simulation: exploring new frontiers* (pp. 1079–1086), Winter Simulation Conference.

Roser, C., Nakano, M., and Tanaka, M., 2002b, Throughput sensitivity analysis using a single simulation, *Simulation Conference Proceedings of the Winter*, Vol. 2, pp1087–1094.

Roser, C., Nakano, M., and Tanaka, M., 2003, Comparison of bottleneck detection methods for AGV systems, *Proceedings of the 2003 Winter Simulation Conference*, pp1192–1198.

Roser, C., Lorentzen, K., and Deuse, J., 2014, Reliable shop floor bottleneck detection for flow lines through process and inventory observations, *Procedia CIRP*, 19, 63–68.

Roser, C., and Nakano, M., 2015, A quantitative comparison of bottleneck detection methods in manufacturing systems with particular consideration for shifting bottlenecks, In *IFIP*

International Conference on Advances in Production Management Systems (273–281). Springer, Cham.

Roser, C., Lorentzen, K., Lenze, D., Deuse, J., Klenner, F., Richter, R., Schmitt, J. and Willats, P., 2017, Bottleneck Prediction Using the Active Period Method in Combination with Buffer Inventories, In *IFIP International Conference on Advances in Production Management Systems* (374-381), Springer, Cham.

Spurrier, J.D., 1999, Exact confidence bounds for all contrasts of three or more regression lines, *Journal of the American Statistical Association*, 94, 446, 483-488.

Stevenson, M., Hendry, L.C., and Kingsman, B.G., 2005, A review of production planning and control: The applicability of key concepts to the make to order industry, *International Journal of Production Research*, 43, 5, 869–898.

Subramaniyan, M., Skoogh, A., Gopalakrishnan, M., Salomonsson, H., Hanna, A. and Lämkull, D., 2016, An algorithm for data-driven shifting bottleneck detection, *Cogent Engineering*, 3, 1, p.1239516.

Subramaniyan, M., Skoogh, A., Salomonsson, H., Bangalore, P. and Bokrantz, J., 2018, A data-driven algorithm to predict throughput bottlenecks in a production system based on active periods of the machines, *Computers & Industrial Engineering*, 125, 533-544.

Subramaniyan, M., Skoogh, A., Muhammad, A.S., Bokrantz, J., Johansson, B. and Roser, C., 2020, A generic hierarchical clustering approach for detecting bottlenecks in manufacturing, *Journal of Manufacturing Systems*, 55, 143-158.

Thürer, M., Qu, T., Stevenson, M., Li, C.D., and Huang, G.Q., 2017, Deconstructing Bottleneck Shiftiness: The Impact of the Bottleneck Position in an Order Release controlled Pure Flow Shop, *Production Planning & Control*, 28, 15, 1223-1235.

Yu, C., and Matta, A., 2016, A statistical framework of data-driven bottleneck identification in manufacturing systems, *International Journal of Production Research*, 54, 21, 6317–6332.

Zhai, Y., Sun, S., Wang, J. and Niu, G., 2011, Job shop bottleneck detection based on orthogonal experiment, *Computers & Industrial Engineering*, 61, 3, 872-880.