International Journal of Production Research
Publication details, including instructions for authors and subscription information:
http://www.tandfonline.com/loi/tprs20

Workload control release mechanisms: from practice back to theory building

Matthias Thürer [a], Cristovao Silva [b] & Mark Stevenson [c]

[a] Institut für Werkzeugmaschinen und Fabrikbetrieb, Technical University Berlin, Pascalstr. 8-9, 10587 Berlin, Germany

[b] Mechanical Engineering Department, University of Coimbra, Pólo II Pinhal de Marrocos, 3030 Coimbra, Portugal

[c] Department of Management Science, Lancaster University, Lancaster University Management School, Lancaster, LA1 4YX, UK

Version of record first published: 03 Jun 2009.

International Journal of Production Research
Vol. 48, No. 12, 15 June 2010, 3593–3617

Workload control release mechanisms: from practice back to theory building

Matthias Thürer[a], Cristovao Silva[b] and Mark Stevenson[c]*

[a]*Institut für Werkzeugmaschinen und Fabrikbetrieb, Technical University Berlin, Pascalstr. 8–9, 10587 Berlin, Germany;* [b]*Mechanical Engineering Department, University of Coimbra, Pólo II Pinhal de Marrocos, 3030 Coimbra, Portugal;* [c]*Department of Management Science, Lancaster University, Lancaster University Management School, Lancaster, LA1 4YX, UK*

(*Received 11 July 2008; final version received 24 March 2009*)

Much Workload Control research has focussed on the order release stage but failed to address practical considerations that impact practical application. Order release mechanisms have been developed through simulations that neglect job size variation effects while empirical evidence suggests groups of small/large jobs are often found in practice. When job sizes vary, it is difficult to release all jobs effectively—small jobs favour a short period between releases and a tight workload bounding while large jobs require a longer period between releases and a slacker workload bounding. This paper represents a return from a case study setting to theory building. Through simulation, the impact of job sizes on overall performance is explored using all three aggregate load approaches. Options tested include: using distinct load capacities for small/large jobs and prioritising based on job size or routing length. Results suggest the best solution is assigning priority based on routing length; this improved performance, especially for large jobs, and allowed a short release period to be applied, as favoured by small jobs. These ideas have also been applied to a second practical problem: how to handle rush orders. Again, prioritisation, given to rush orders, leads to the best overall shop performance.

Keywords: production planning; shop floor control; supply chain management; decision support systems; production control

1. Introduction

Workload Control (WLC) is a method of planning and controlling production that has received much attention in recent years. While the customer enquiry and order acceptance stages are important, a large proportion of the literature focuses on the order release stage through which the level of Work-In-Process (WIP) on the shop floor is regulated (e.g., Hendry and Wong 1994, Missbauer 1997, Land and Gaalman 1998, Bertrand and Van Ooijen 2002, Breithaupt *et al.* 2002, Cigolini and Portioli-Staudacher 2002). The unifying theme in this research is the use of a pre-shop pool in which all jobs 'compete' against each other for release. Land and Gaalman (1998) explain that a pool can absorb fluctuations in the flow of incoming orders, reduce WIP costs, increase shop floor transparency, reduce waste caused by order cancellations, allow later ordering of raw materials and reduce the need to expedite jobs on the shop floor.

*Corresponding author. Email: m.stevenson@lancaster.ac.uk

ISSN 0020–7543 print/ISSN 1366–588X online

DOI: 10.1080/00207540902922810
http://www.informaworld.com

A pre-shop pool can be particularly important where there is instability, such as in the manufacture of bespoke or highly customised products where job sizes (e.g., unit processing times or quantities) vary. However, when job sizes do vary, it can be difficult to plan and control the release of all jobs effectively—jobs with a small workload favour a short period between releases and a tight bounding of the released workload while jobs with a large workload require more time between releases and a slacker bounding of the released workload. This is supported by Land (2006), who explains that a long release period delays certain jobs and can increase gross throughput times while a short release period can hinder the progress of large jobs. Despite the above, simulation studies have tended to ignore this problem at the release stage. Meanwhile, recent case study research identified accommodating job size variations within WLC theory as an important problem for researchers to address in order to improve the effective implementation of WLC in practice (Stevenson and Silva 2008).

In response, this paper explores means of balancing the needs of small and large jobs by attempting to improve the performance of large jobs whilst maintaining a short period between releases (also known as the check or release period), as favoured by small jobs. Oosterman *et al.* (2000) suggest that a 2-Erlang distribution may be a better approach (than the exponential distribution) to modelling the processing times found in real-life job shops and most studies since Oosterman *et al.* (2000) have adopted this distribution. The simulations described herein use both exponentially distributed and 2-Erlang distributed processing times in order to analyse the implications of the choice of distribution.

In an extension, this paper also seeks to build on recent research by Hendry *et al.* (2008) who investigated issues arising from implementing WLC through comparative case study analysis. The authors examined two implementation projects, one at a capital goods manufacturer in The Netherlands and one at a subcontract engineering firm in the UK. The authors investigated how implementation issues that arise in the context of WLC should be addressed to enable improved implementation in practice. The study identified 17 implementation issues and raised a series of research questions. These include: "how can future, replacement part, rush orders be considered most effectively within the WLC concept?" One solution the authors suggest is reserving a percentage of capacity for rush orders; however, while suggestions are made, the performance of means of handling rush orders within a WLC system are not tested. After investigating the issue of job size variation in this paper, the findings are used to explore this second important practical problem. This paper represents a return from recent field research to a theory building and testing environment and continues the recent trend in WLC research to more accurately reflect practical considerations in job shop simulations and in the development of theory in order to improve the practical applicability of the methodology (e.g., Perona and Miragliotta 2000, Bertrand and Van Ooijen 2002, Henrich *et al.* 2004).

The remainder of this paper is organised as follows. Section 2 reviews the literature on order release mechanisms before the research method is outlined in Section 3. Section 4 describes the simulation model and the different approaches we investigate to address job size variation. Simulation results are summarised and discussed in Section 5 before Section 6 extends the results to the problem of how best to handle rush orders within the WLC concept. Final conclusions are presented in Section 7.

2. Literature review

This review considers two core elements of this paper: (1) the influence of the size of a job on performance; and (2) order release mechanisms. Section 2.1 provides a short review of how job size has been modelled in the literature before Section 2.2 explores order release mechanisms. It is not our intention here to provide a comprehensive review of the literature on order release mechanisms—many exhaustive reviews of the literature have previously been presented (e.g., Philipoom *et al*. 1993, Wisner 1995, Bergamaschi *et al*. 1997, Sabuncuoglu and Karapinar 1999). However, two of the most important methodological aspects at the order release stage, included in the classification of order review/release mechanisms by Bergamaschi *et al*. (1997), are: the way in which the methodology accounts for the workload of a job over time; and the way in which the workloads of shop floor resources are bounded. The impact of processing times, a major contributing factor to overall job size variation, on these two elements is considered before the literature is assessed in Section 2.3.

For a broader review of production planning and control, see Zäpfel and Missbauer (1993) and Stevenson *et al*. (2005). For a review of WLC, see Land and Gaalman (1996).

2.1 *Modelling job size variation*

A selection of previous WLC simulation studies is summarised in Table 1 based on the summary of order review/release mechanisms by Wisner (1995). The table includes various approaches to modelling processing times. Job size variation is evident in many of the models but the problem that results from this variation is not addressed. It is also evident from Table 1 that recent studies favour a 2-Erlang distribution, as previously described.

To the best of our knowledge, the contribution and influence of different job sizes on overall shop performance, and ways of accommodating job size variation, has not been explicitly considered. Papers typically seek to avoid the impact of job size variation, especially the presence of large jobs, rather than to address the issue within the WLC methodology. Therefore, the processing times generated are typically much smaller than the release intervals used in the studies, avoiding problems in the relationship between the check period and the size of jobs, as noted by Land (2006).

Other contributions disregard processing time variation even further. For example, alternative approaches to WLC, including card-based methods like CONWIP, often do not consider the size of jobs at all in the release decision. Instead, they control the number of cards (or jobs) in circulation and treat each job in the same way. It is acknowledged that these simplifications may reflect the characteristics of the environment for which the methodologies are designed. For example, Fowler *et al*. (2002) explain that in the semiconductor industry, where CONWIP has been implemented, it is not unreasonable to assume that processing times are constant. This is not a reasonable assumption in many other contexts.

2.2 *The impact of processing times on two aspects of order release mechanisms*

There are three notable approaches to accounting for the workload of a job over time when it is being considered for release.

(1) Aggregate load approaches attribute the workload of a job to relevant work centres at the moment of release irrespective of the routing of a job prior to arrival

Table 1. Sample of previous approaches to modelling processing times in WLC simulation studies.

Authors	Routing	Work centres	Operations	System	Processing times
Bertrand (1983)	R	5	U $\sim$ [1, 10]	H	Exp. distribution, $(1/\lambda)) = 1$ hour AND < 5
Shimoyashiro *et al*. (1984)	SR	33 (80)	[1, 15] mean 6	R	Mean 1; min. 0.1; max. 14 hours
Onur and Fabricky (1987)	R	6	U $\sim$ [4, 10]	H	Uniform distribution [3, 9] hours
Ragatz and Mabert (1988)	R	5	U $\sim$ [1, 8]	H	Exp. distribution, $(1/\lambda)) = 1$ hour AND < 4
Melnyk and Ragatz (1989)	R	6	U $\sim$ [1, 6]	H	Exp. distribution, $(1/\lambda)) = 1$ time unit
Park and Bobrowski (1989)	R	5 (10)	U $\sim$ [1, 5]	H	Exp. distribution, $(1/\lambda)) = 2.5$ hours AND < 4 hours
Melnyk *et al*. (1991)	R	6	U $\sim$ [2, 6]	H	Normal distribution, $\mu = 1.5$, $\sigma = 0.1$ hours
Ahmed and Fisher (1992)	R	5	U $\sim$ [1, 8]	H	Exp. distribution, $(1/\lambda) = 1$ hour AND < 4
Philipoom and Fry (1992)	F	5 (12)	5	H	Exp. distribution selected such that 90% capacity utilisation rate is obtained
Philipoom *et al*. (1993)	R	15	U $\sim$ [3, 7]	H	Exp. distribution selected such that 87% or 92% capacity utilisation rate is obtained
Hendry and Wong (1994)	R	6	U $\sim$ [1, 6]	H	Exp. distribution $(1/\lambda) = 1.5$ time units

Melnyk *et al*. (1994)	R	6	U ~ [2, 6]	H	Exp. distribution $(1/\lambda) = 1.5$ and $(1/\lambda) = 1.5$ bounded between 1.4 and 1.6 hours
Fredendall and Melnyk (1995)	R	6 (12)	U ~ [2, 6]	H	Exp. distribution, $(1/\lambda) = 1$ time unit
Park and Salegna (1995)	SR	6	U ~ [1, 6]	H	Exp. distribution selected such that the first WC has a utilisation level of 92% and the others 86%
Cigolini *et al*. (1998)	R	11	U ~ [1, 12]	H	30 job types
Hendry *et al*. (1998)	SR	15	[3, 13] mean 9	R	$\mu = 41$ hours
Land and Gaalman (1998)	R	6	U ~ [1, 6]	R	2-Erlang distribution, $\mu = 0.75$ days
Oosterman *et al*. (2000)	R, SR, F	6	U ~ [1, 6]	R	2-Erlang distribution, $\mu = 1$ day
Kingsman and Hendry (2002)	SR	15 (4)	[3, 13]	R	$\mu = 41$ hours
Bertrand and van de Wakker (2002)	R	15	Geometric function, $\mu = 5$, max $= 39$	H	Negative exponential distribution, $\mu = 1$ time unit
Bertrand and Van Ooijen (2002)	R	10	U ~ [1, 10]	H	Exp. distribution, $(1/\lambda) = 1$ time unit
Henrich *et al*. (2004)	R, F	12	U ~ [1, 12]	H	Gamma distribution, alpha = 2, beta = 0.5, $\mu = 1$, and variance = 0.5
Land (2006)	R	6	U ~ [1, 6]	H	2-Erlang distribution, $\mu = 1$ time unit

Routing: random routing (R); semi-random/dominant flow routing (SR); flow shop routing (F).
System: real system characteristics (R); hypothetical system characteristics (H).

at a work centre (e.g., Bertrand and Wortmann 1981, Hendry and Kingsman 1991, Kingsman 2000, Kingsman and Hendry 2002, Stevenson 2006a, Stevenson and Hendry 2006). The workload hence includes direct and indirect load without distinguishing between the two. The traditional aggregate load method pays particular attention to the set-up and processing times of jobs in the determination of the workload but has been criticised for having difficulty in providing sufficient control in job shop simulations (e.g., Perona and Portioli 1998, Oosterman *et al.* 2000). Adaptations of the traditional aggregate load approach include the corrected and extended aggregate load approaches.

(2) Probabilistic approaches (e.g., Bechte 1988, 1994, Wiendahl 1995) assign a percentage of the workload of a job to relevant work centres at release, based on the probability of the job reaching the work centre in the planning period. Breithaupt *et al.* (2002) criticise probabilistic approaches for neglecting the influence of processing times on order progress.

(3) Time bucketing approaches (e.g., Bobrowski 1989) divide the planning horizon into load periods/time buckets; forward or backward scheduling is then used to assign a job to a load period and it is only included in the period for which it will be the direct load. In recent years, the time bucketing approach has received little attention in the literature.

Of the above approaches, job size variation has a particularly detrimental effect on the aggregate load release method. For example, in relation to the traditional aggregate load approach to WLC:

- When a large job is released, it will have a big impact on the current workloads of all work centres in its routing, even when it is queuing or being processed elsewhere. This can distort the 'true state' of the shop floor and affect the release of other jobs from the pool. It could result in some work centres being left idle and others overloaded.
- Grouping machines can improve the timeliness of feedback information from the shop floor. This can be particularly important for the aggregate load method; however, when processing times are large, the workload requirements of a job can be misrepresented if machine capacities are grouped (Stevenson and Silva 2008).

Workload bounding refers to the use of parameters to restrict the workload (e.g., on the shop floor). The bounding of the workload is related to the period between releases. Perona and Portioli (1998) demonstrate the need to adjust the interval between releases when considering small and large orders. Large workload limits and long periods between releases would allow large jobs to be released but would undermine overall control of workloads. Hence, a large release period may solve one problem but deteriorate the speed of release for small jobs. If customers expect a short delivery lead time for small orders, the increase in pool waiting time for these orders may affect due date adherence.

Traditionally, the workload is controlled using maximum and/or minimum bounds (or norms). A key research challenge is determining the level at which to set workload norms. This is a subject of much debate. Enns and Prongue Costa (2002) advise that a control level set too high is ineffective but that too low a level provides inadequate throughput. Land (2004) shows that although tightening workload norms hinders the timing of job release, queues on the shop floor fluctuate less and suggests that the difficulties experienced by jobs with long routings and/or large processing times when norms are tight can be compensated for by increasing job priority. It is rare that research in this area considers the impact of large jobs on the bounding of workloads; exceptions include Bechte (1988), Hendry (1989) and Cigolini and Portioli-Staudacher (2002). When the load limit is reached in Bechte's (1988) probabilistic approach, release is continued for one additional job that would visit the fully loaded work centre. Similarly, Hendry (1989) describes a 'Force Release' mechanism that allows the user to release a job which would exceed the upper bound of one or more shop floor resources. Cigolini and Portioli-Staudacher (2002) describe a workload balancing procedure based upon striking a balance between improving utilisation at an under-loaded work centre at risk of starvation at the expense of overload elsewhere. Individual work centres can be overloaded as long as the overall workload balance across all work centres is improved. These solutions provide flexibility which goes some way to allowing large jobs to be released.

2.3 *Assessment of the literature*

Job size variation is an important problem impacting the performance of existing WLC theory at the order release stage but one that has received insufficient attention to date. Existing theory has a tendency to treat all jobs equally. In contrast, it is argued here that where there are distinct differences in job size, disregarding the impact of this variation is inappropriate and such models are unlikely to result in an effective solution for all jobs. In what follows, we acknowledge that small jobs have different requirements to large jobs and experiment with adapting the release mechanism to reflect this. This includes allowing the workload norm to be exceeded (from Bechte 1988) and increasing job priority for large jobs (from Land 2004).

Job size variation has a particularly detrimental effect on the aggregate load method and hence it is the method in most need of development. Moreover, this is the simplest method and, given that it is argued that managers prefer simplicity, is considered the one most likely to be successfully implemented in practice. Therefore, the study will use aggregate load methods as the basis for workload accounting over time (the traditional, corrected and extended aggregate load methods). With regards to workload bounding: difficulties in setting effective workload norms may be caused by attempting to find a single bound that will meet the needs of all jobs. Therefore, we try to accommodate differences between groups of jobs more explicitly within the bounding of the WLC concept.

3. Methodology

3.1 *Empirical grounding for the study*

Recent case study research (Stevenson 2006a, b, Silva *et al.* 2006, Hendry *et al.* 2008, Stevenson and Silva 2008) identified practical considerations that affect how the WLC concept is used in practice. Among these is the importance of accommodating processing time variation within the WLC methodology, thus providing an empirical grounding for this study.

Company M (see Silva *et al.* 2006 and Stevenson and Silva 2008) produce one-off aluminium moulds for pre-series production and steel mould components for large series production (e.g., for the automotive and electronics industries). Each aluminium mould is engineered-to-order and typically comprises of a large number of components, some are very simple, others are more complex. Processing time variation across jobs is prominent, which results in high job size variation. Under the WLC concept that Silva *et al.* (2006) attempted to implement, all components had to 'compete' against each other for the same set of resources; this led to implementation problems and resulted in large jobs performing worse than small jobs. The poor performance of large jobs was particularly striking if one considered that the *relative* gross throughput time of large jobs should be smaller than the relative gross throughput of small jobs if delivery lead times are to be competitive. Even if small and large jobs performed equally well, based on gross throughput time as a percentage of a job's work content, the lateness of large jobs was not acceptable while, in contrast, a degree of deterioration in the performance of small jobs would be 'acceptable'. Thus, to differentiate according to job size, and to find an optimal balance between the requirements of job sizes, appeared to be vital in order to implement the system successfully in this context.

The authors have observed a similar phenomenon in a very different production setting—a plastic bag manufacturer. The majority of production orders are processed in less than 24 hours but, like in Company M, a significant proportion take more than one working day. Unlike in Company M, this is not due to differing product complexity but to differing order quantities. Again, job size variation caused significant problems for the application of existing WLC theory in this company.

3.2 *Research questions*

To overcome the detrimental effect of job size variation on performance, as noted from the literature and observed in practice, the research began with the following questions.

(1) How can the existence of groups of 'small' and 'large' job sizes be best incorporated within the order release mechanism of the Workload Control concept?
(2) How can a balance between the requirements of 'small' and 'large' processing times be best achieved in order to improve the release mechanism and overall shop performance?

The best way to explore this problem is considered to be through simulation; hence, this study represents model-based research driven by empirical findings. Bertrand and Fransoo (2002) explain that: "*in this class of research, the primary concern of the researcher is to ensure that there is a model fit between observations and actions in reality and the model made of that reality*". The authors also explain that: "*quantitative model-based research is a rational, objective, scientific approach*". Simulation thus provides us with a good means of testing and evaluating new ideas in a controlled environment that can be replicated by other researchers.

3.3 *Iterative approach to theory building*

This paper tests several release mechanisms that seek to avoid the problems outlined in the above sections and obtain a 'best-of-both-worlds' solution. The research follows an iterative approach to building, testing and refining theory, as illustrated in Figure 1. The concept of WLC is often cited as being developed to overcome the lead time syndrome (Mather and Plossl 1978). Throughout the 1980s and 1990s, WLC theory was developed, tested and refined through simulation. Refined theory has been incorporated within the

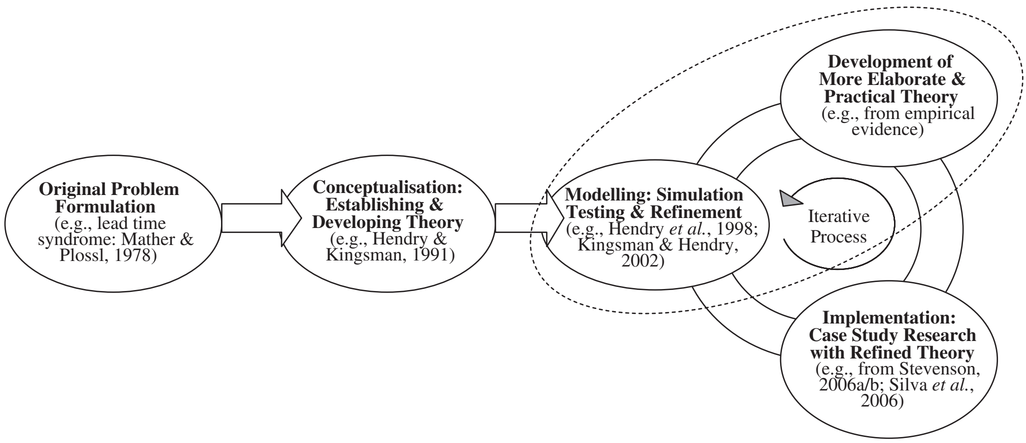

Figure 1. Theory–practice iterative research cycle.

design of decision support systems and applied during case study research. This study closes one iteration of the loop. It starts with the identification of a problem encountered in recent case study work, for which several possible solutions, representing practical extensions to WLC theory, are proposed. To test these solutions, the study returns to the simulation environment previously used by many authors to test the WLC theory. Replicating the traditional WLC simulation environment, which is a simplification of a real-life shop floor, allows research to identify the best solution to the problem encountered whilst maintaining consistency with the WLC simulation research methodology used in the past. The outcomes of these tests can be considered when implementing WLC systems in practice in the future, allowing the next iteration of research to confirm the effectiveness of the solutions proposed by this study. Hence, the paper demonstrates the complementary roles that case and simulation modelling research can play in the development of theory and improvement of practice.

4. Simulation model

4.1 *Shop characteristics*

A pure job shop simulation model, according to the characteristics outlined by Melnyk and Ragatz (1989), has been developed using SIMUL8© software. This model is used in many WLC simulation studies (e.g., Hendry and Wong 1994, Oosterman *et al*. 2000, Land 2006). The shop contains six work centres, where each is a single and unique source of capacity, which remains constant. The routing length varies from one to six operations. Each operation requires one specific work centre; routing and operation processing time characteristics are known upon job entry. A particular work centre is required at most once in the routing of a job; all stations have an equal probability of being visited. A First-Come-First-Served (FCFS) dispatching rule is used on the shop floor.

4.2 *Release mechanisms*

In this study, the assumption is that all orders are accepted, that materials are available, and that the process plan (including information regarding routing sequence, processing times, etc.) is known. Orders flow directly into the pre-shop pool; hence, like in most previous studies, a pool of confirmed orders is the starting point. At release time '*t*', jobs in the pool are considered according to shortest slack.

A job is attributed to the load of the work centres corresponding to its routing at the moment of release. If this aggregated load fits within the workload norm, the job is released to the shop floor. If one or more norms would be exceeded, the job must wait until at least the next release period. This procedure is repeated until all jobs in the pool at release time '*t*' have been considered for release once. Three aggregate load approaches are applied.

- The traditional (or classical) aggregate load approach (B), as described by Tatsiopoulos (1983), Hendry (1989) and Section 2.2 of this paper.
- The extended aggregate load approach (C), developed in response to problems caused by a lack of feedback information from the shop floor, as experienced by Tatsiopoulos (1983) while implementing the traditional aggregate load approach. Under the extended approach, a job contributes to the workloads of all stations in its routing until it leaves the shop floor. Hence, only feedback when the job leaves the shop floor is needed.
- The corrected aggregate load approach (B′), developed to account for the routing (and routing length) of jobs in the aggregation procedure (ignored by the traditional approach). Under the corrected approach (see Land and Gaalman 1996), the load contribution at the moment of release is depreciated according to the position of a work centre in the routing of a job. The further downstream a work centre is, the higher the depreciation factor.

In this study, the check period is set to 5 time units, i.e. jobs in the pool are considered for release every 5 time units. To avoid unnecessary complexity and enable a clear insight into the performance of the system, the planning horizon equals the check period.

4.3 *Job characteristics and due date setting procedure*

Due dates are set by adding a random allowance to the job entry time (see Equation (1)), as described by Oosterman *et al.* (2000) and Land (2006). Land (2006) states that the minimum value should cover a station throughput time of 5 time units (the maximum processing time plus one time unit) for a maximum of six operations plus a waiting time before release of 5 time units:

$$\text{Due date} = \text{Job entry time} + a, \quad \text{with } a \text{ uniformly distributed } [35, 60]. \tag{1}$$

Recent studies have modelled processing times using a 2-Erlang distribution. In this study, 2-Erlang and exponential distributions (both with a mean of 1 time unit) will be used in order to analyse the influence that the modelling approach has on performance. All relevant performance measures are arithmetically derivable from the two performance measures we collect. The chosen inter-arrival time of jobs (see Table 3) guarantees a machine utilisation rate of 90% for all the workload norms tested. Thus, for the workload norms tested, the output is not affected by the load limitation.

The characteristics of our job shop and jobs are summarised in Tables 2 and 3, respectively.

Table 2. Summary of simulated shop characteristics.

Shop characteristics	
Shop type	Pure job shop
Shop characteristics (real or hypothetical)	Hypothetical
Routing variability	Random routing, no re-entrant flows
No. of machines	Six
Interchange-ability of machines	No interchange-ability between machines
Machine capacities	All equal
Machine utilisation rate	90%
Shop floor dispatching policy	First-Come-First-Served

Table 3. Summary of simulated job characteristics.

Job characteristics	
No. of operations per job	Uniform [1, 6]
Operation processing times (exponential)	Exp. distribution, $(1/\lambda) = 1$
Operation processing times (2-Erlang)	2-Erlang, $\mu = 1$
Inter-arrival times	Exp. distribution, $(1/\lambda) = 0.633$
Set-up times	Not considered
Due date determination procedure	Job entry time $+ a$; a U $\sim$[35, 60]
Complexity of product structures	Simple independent product structures
Job characteristics (real or hypothetical)	Hypothetical

4.4 *Job size*

The main research objective is to analyse the influence of different job size on overall performance. Therefore, jobs are subdivided into 10 groups according to job size: nine groups are defined for jobs smaller than 9 time units (using an interval of 1 time unit); and one group is defined for jobs larger than 9 time units. To ease comparison, results for the different job sizes are summarised in two groups. Jobs larger than 3 time units are considered 'large jobs'; jobs less than or equal to 3 time units are considered 'small jobs'. All large jobs showed a similar performance pattern; the same is true of small jobs.

Figure 2 shows the distribution of job sizes using the exponential and 2-Erlang distributions. There is a notable difference between exponentially distributed processing times and 2-Erlang distributed processing times, particularly with regard to the number of large jobs. The exponential distribution shows a much higher number of very large jobs and a higher number of very small jobs. Job size for the 2-Erlang distribution is more settled around a mean of 3.5, showing less variance. Fifty percent of jobs on the shop floor are smaller than 3.5 time units (the expected value for job size given a mean routing length of 3.5 and processing time of 1 time unit) but represent only 30% of the total shop floor workload; 70% of the shop floor load is represented by the 50% of jobs larger than 3.5 time units.

4.5 *Experimental design*

In the first stage of experiments (the 'standard scenario'), the simulation model is run without any special conditions and the performance of the different job sizes is analysed.

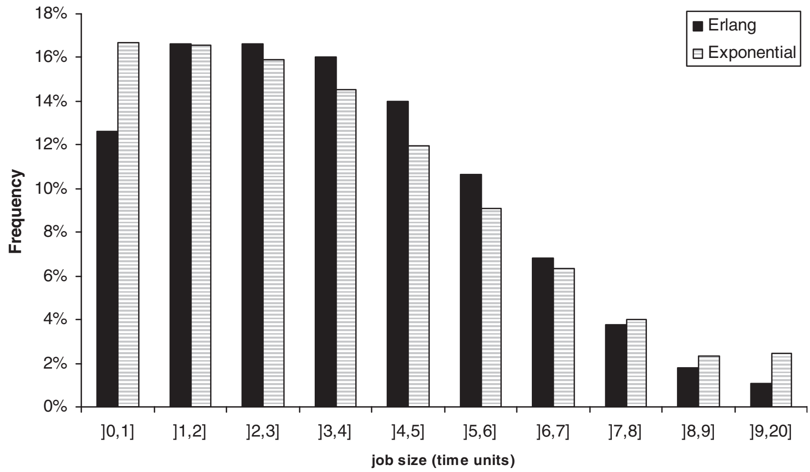

Figure 2. Job size distribution: exponential vs. Erlang.

Then, the following four approaches are implemented and will be compared with the standard scenario.

- *Distinct load capacities for small and large jobs*. The capacity of each work centre is divided into two parts and allocated proportionately to small and large jobs separately, according to the size of the jobs.
- *Prioritisation*. Jobs are prioritised in the pool according to job size or routing length. Either the largest job or longest routing is considered for release first.
- *Exceeding the workload norm*. The first job that exceeds the norm can be released. This should improve the performance of large jobs (more likely than small jobs to be the first to exceed the norm level).
- *Load correction*. Feedback from the shop floor, used in the traditional aggregate load approach, is corrected by the hypothetical downstream load. This represents the proportionate load of a job in-process at a work centre at job release but which is already complete. Under release method B, this proportion would continue to contribute until the whole job is complete at the work centre.

Each of the four approaches proposed above, plus the standard scenario, has been tested considering: two approaches for the generation of processing times, three aggregate load approaches and 13 load norm levels. This results in a full factorial design of experiments. The key results we focus upon are the gross (or total) throughput time and the (shop floor) throughput time. The (shop floor) throughput time describes the performance of the job after release and allows us to evaluate the performance of the shop floor. The gross throughput time, which incorporates the pool delay, provides an overview of the performance of the job across the whole system and indicates the percentage of late jobs to which it is directly related. Some preliminary tests were conducted in which mean job lateness was also analysed. These tests showed that the behaviour of the model was very similar in terms of mean job lateness and gross throughput time, i.e. good results in terms of gross throughput time meant good results in terms of mean lateness. Thus, the decision was made to focus on gross throughput time and to ignore mean job lateness during further testing.

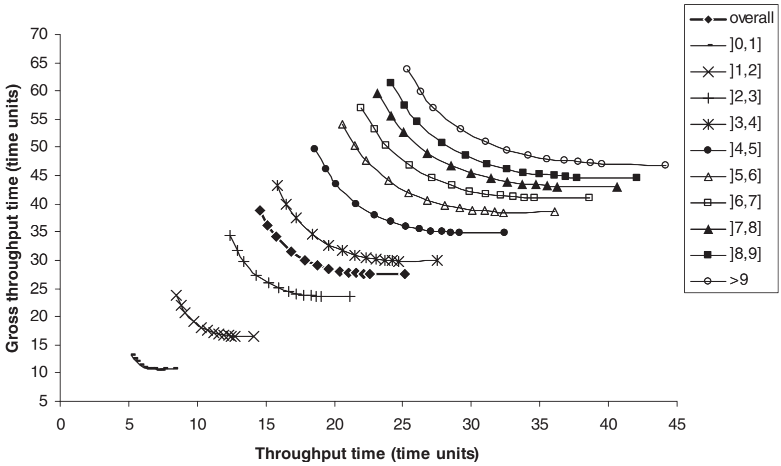

Figure 3. Performance of approach B under the standard scenario (2-Erlang).

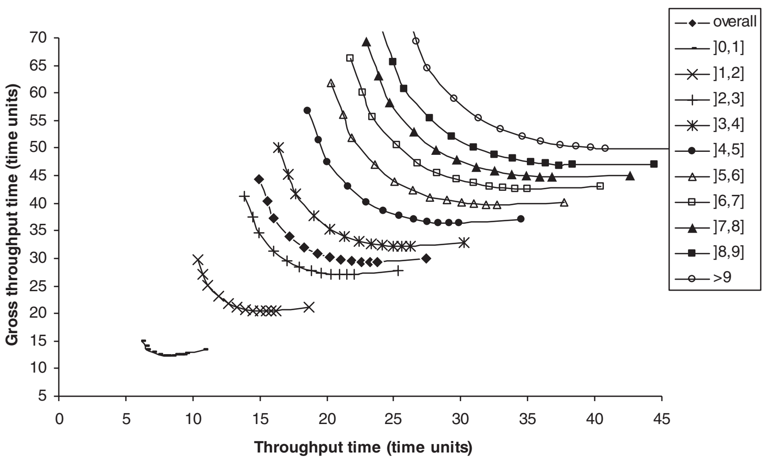

Figure 4. Performance of approach B under the standard scenario (exponential).

Results are obtained by tightening the norm level stepwise down from infinity, represented by the right-hand starting point of the curves that follow in Sections 5 and 6 (see Figures 3–11). A norm level of 100% is equivalent to the critical workload norm. The critical workload norm represents the point where the throughput time ceases to decrease, while the gross throughput time continues to rise; this will be determined empirically. Each experiment consists of 100 runs; results are collected over 10,000 time units; the warm-up period is set to 3000 time units to avoid start-up effects.

5. Results

5.1 *Results for the standard scenario*

Figures 3 and 4 show the results for release method B, the traditional aggregate load approach, under the standard scenario. As the norm is tightened, the shop floor

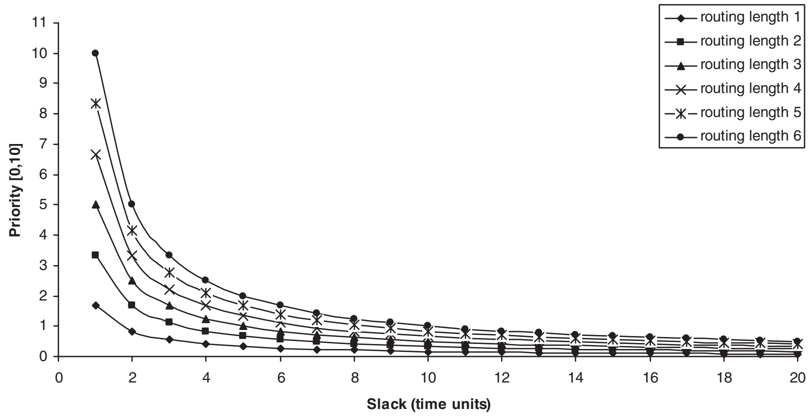

Figure 5. Conversion of priority according to routing length.

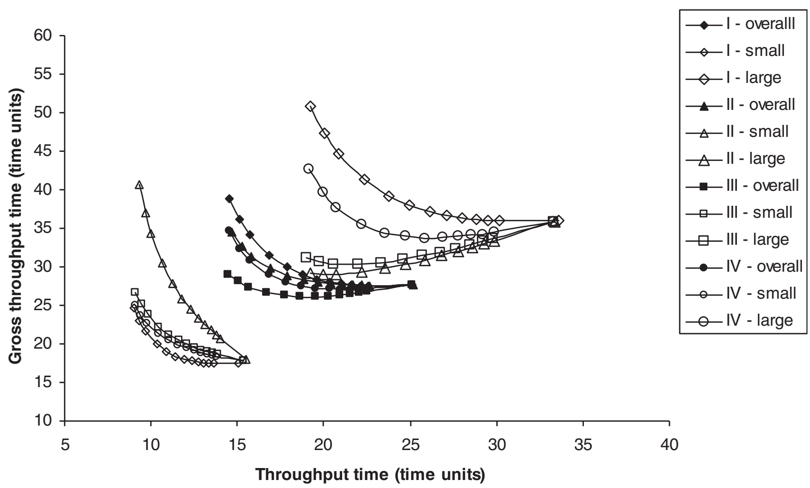

Figure 6. Performance of approach B with prioritisation (2-Erlang).

throughput time is reduced, caused by a reduction in the average waiting time in front of work centres. This, however, does not necessarily imply a reduction in the gross throughput time when the time in the pre-shop pool is also considered.

From the figures, it can also be concluded that, in the standard scenario, large jobs generally perform worse than small jobs (particularly noticeable if processing times are exponentially distributed due to the greater job size variance). For both distributions, the gross throughput time for large jobs is high relative to that for small jobs. To minimise the percentage of late jobs, the delivery lead time has to be large but this reduces the competitiveness of due date quotations a company can realistically make at the customer enquiry stage. Similar results, consistent with those obtained by Oosterman *et al*. (2000), have been obtained for the corrected and extended aggregate load approaches. The corrected approach performs the best out of the three and the extended approach performs the worst.

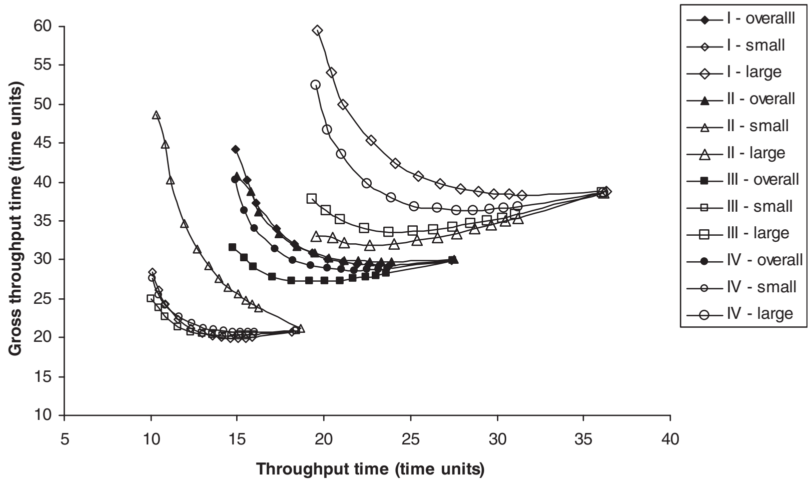

Figure 7. Performance of approach B with prioritisation (exponential).

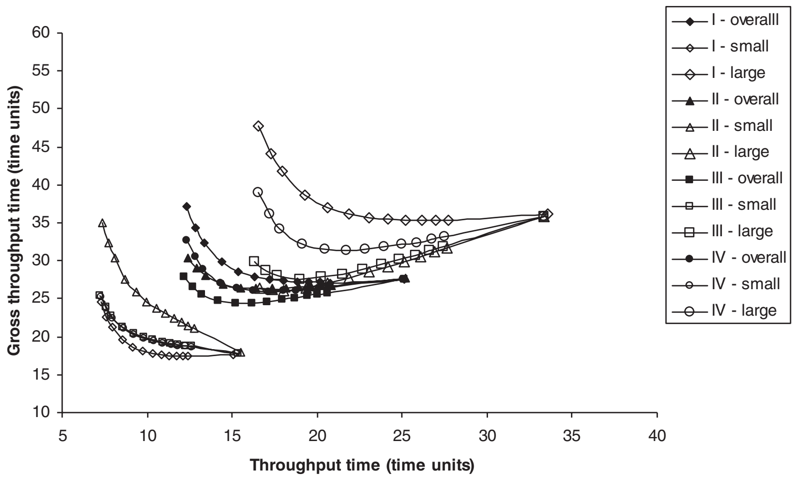

Figure 8. Performance of approach B′ with prioritisation (2-Erlang).

5.2 *Results based on different load capacities for small and large jobs*

One of the simplest potential solutions to our problem is to use different norm levels for small and large jobs and to distribute the load capacity of the shop floor proportionately according to the processing times of jobs. While this appears simple, using more than one norm increases the check period because capacity must be provided for both norms, leading to a greater gross throughput time. A longer period between releases implies a longer pool delay, which cannot be fully compensated for by any resulting gain in performance. To compensate, two solutions have been explored: (1) using different workload norms and check periods for small and large jobs; and (2) using two different

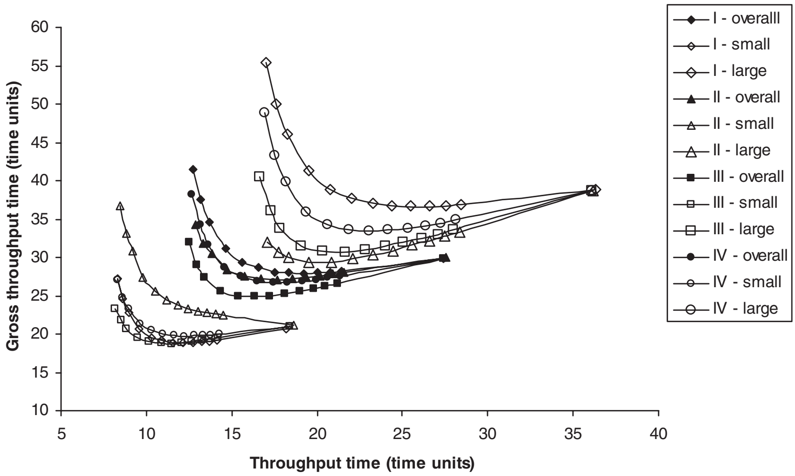

Figure 9. Performance of approach B′ with prioritisation (exponential).

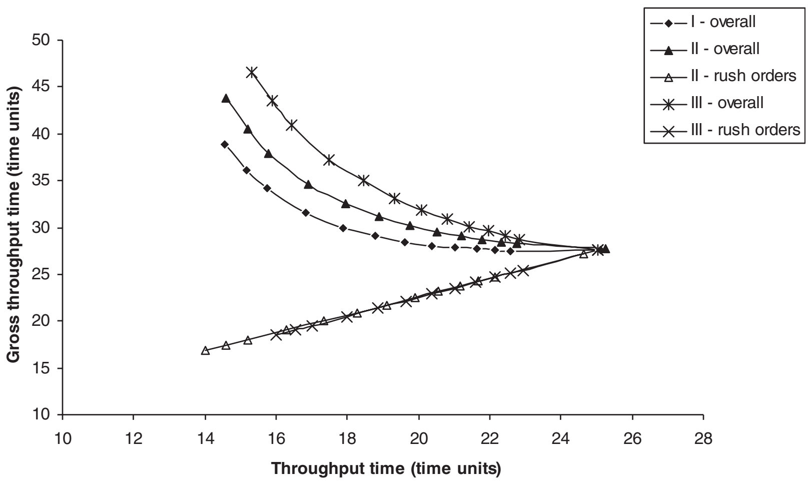

Figure 10. Performance of approach B for rush orders (2-Erlang).

check periods for small and large jobs but the same resources of load capacity. Consider the following.

(1) Using two different workload norms and check periods for small and large jobs leads to another challenge—how to set them, given that the load capacity and check period are inter-dependent? At each release point for small jobs, a percentage of capacity is kept free for large jobs. The minimum check period for large jobs is the period needed to provide enough free capacity for the release of large jobs (based on the maximum processing time). The more capacity reserved, the sooner large jobs can be released; this implies a shorter check period for large jobs and a larger check period for small jobs. Each improvement for one job size leads to

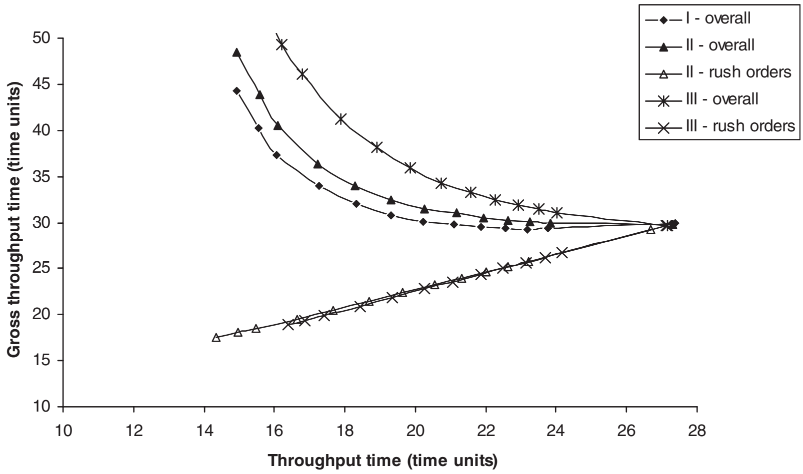

Figure 11. Performance of approach B for rush orders (exponential).

a deterioration for the other. Moreover, if only large jobs, and thus only large processing times are released, 'load gaps' begin to emerge which would otherwise be filled by jobs with a small workload contribution.

(2) Typical applications of using two different check periods, but where all jobs rely on the same resources of load capacity, favour small jobs; large jobs find it more difficult during the shorter of the two release periods to be released. Small jobs are released and contribute to the shop floor load thus reducing free capacity at the next (and longer) release period, thereby undermining the solution.

The results of applying different norms for small and large jobs did not improve performance. This might be an effect of the short planning horizon and rigid workload norms assumed in the simulations. Applying a long planning horizon, and allowing jobs to occasionally exceed the workload norm where appropriate, as is typical in real-life job shops, neutralises many of the restrictions that lead to poor performance. Another practical advantage is that this approach, using different resources of capacity for small and large jobs, lessens the detrimental effects tha job size variation has on aggregate methods (as described in Section 2.2). Therefore, it is concluded that the methods explored in this section are unlikely to lead to improvements in overall performance but may show more positive effects in practice.

5.3 *Results based on prioritisation methods*

Three different prioritisation methods have been tested, as outlined below.

- *Prioritisation according to job size*. Jobs are considered for release according to size and secondarily according to latest release date. Firstly, all jobs with a processing time greater than 9 time units are considered. Of these, the job with most immediate latest release date is considered first. This continues down through the other groups of job sizes, starting with jobs between 8 and 9 time units, until all jobs have been considered for release once.

- *Priority according to routing length*. Similar to above but according to routing length, starting with all jobs with a routing length of six operations.
- *Converted priority, according to routing length*. This aims to guard against the discrimination of small jobs which will occur in the above two prioritisation methods. Release precedence is determined by a combination of priority and slack, where slack is depreciated according to routing length. Thus, jobs with a larger routing length are given priority over jobs with a shorter routing length but with a similar slack. Figure 5 shows the new priority measure, standardised to a scale of [0, 10] for the different slack levels. Jobs are not further prioritised strictly according to routing length. Jobs with short routing lengths and a short slack receive priority over jobs with a larger routing length but longer slack.

To analyse the results, the above prioritisation rules have been compared with the standard scenario for the three aggregate load approaches. Scenario I represents the standard scenario; in scenario II, prioritisation is based on job size; in scenario III, prioritisation is based on routing length; and in scenario IV, prioritisation is according to 'converted priority'. Subsections 5.3.1 and 5.3.2 summarise the results of scenarios I–IV for the traditional, corrected and extended aggregate load approaches.

5.3.1 *Results for release method B: traditional aggregate load approach*

Figures 6 and 7 show the results obtained using the traditional aggregate load approach for scenarios I–IV for the jobs overall and for small and large jobs individually using the 2-Erlang and exponential distributions. It can be observed that if prioritisation is based on job size (scenario II), the performance for 2-Erlang distributed processing times is a slight improvement on the overall results obtained for the standard scenario (scenario I). However, if processing times are exponentially distributed, performance stays the same or deteriorates. Assigning priority according to job size improves performance for large jobs but significantly deteriorates performance for small jobs. This deterioration becomes even worse if processing times are exponentially distributed. There are two possible causes of these poor results, either: (1) the shop floor throughput time increases, caused by the influence of sequence changes at the release stage on the dispatching rule; or (2) the gross throughput time increases, from a longer pool delay as a result of the difficulties smaller jobs face in being released. As can be seen from the figures, the deterioration in performance of small jobs, and the improvement of large jobs, is mainly caused by the change in pool delay. Small jobs with a high routing length are difficult to release. A small job size does not necessarily imply a short routing length and *vice versa*. As a result, only considering job size in the release decision does not lead to an overall improvement. The improvement for large jobs does not fully compensate for the deterioration in small jobs.

If prioritisation is based on routing length (scenario III), results are very positive (compared with scenario I). The improvement for large jobs is almost the same as in scenario II, but the negative effect on the performance of small jobs is significantly less. The performance of small jobs is only slightly worse than in the standard scenario. Using the converted measure for prioritisation (scenario IV) improves the performance of small jobs compared with giving prioritisation strictly according to routing length; however, this improvement does not compensate for the deterioration in performance for large jobs. Hence, results for the traditional approach indicate that the best solution is scenario III, prioritisation based on routing length.

5.3.2 *Results for release methods B′ and C: corrected and extended aggregate loads*

Figures 8 and 9 summarise the results for release method B′ (the corrected load approach) for scenarios I–IV for the jobs overall and for small and large jobs individually. Results are very similar to those for the traditional approach. As previously, basing prioritisation on routing length (scenario III) yields the best results. Prioritisation according to job size (scenario II) yields slightly better results for large jobs than above but results in extremely poor performance for small jobs; the converted priority approach (scenario IV) leads to a slight improvement in the performance of small jobs but performance is much worse for large jobs. Results for release method C (the extended load approach) are not shown but the same conclusions as for release method B′ are also valid here. Through comparison, it can be concluded that the corrected aggregate load approach (B′) performs the best out of the three release methods and the best solution remains scenario III, prioritisation based on routing length.

5.4 *Results based on allowing the workload norm to be exceeded*

The Load Oriented Manufacturing Control (LOMC) concept presented by Bechte (1988), based on the probabilistic WLC approach, compensates for large jobs at the release stage. The norm level is relaxed; the first job that exceeds the load limit is still released to the shop floor, allowing very large jobs at the front of the queue in the pool to be released. In experimenting with using this idea in an aggregate load context, it has been difficult to control the emerging overload. Allowing workload restrictions to be exceeded can result in the shop spiralling out of control as, for example, the overload released at release time 't_x' has a negative influence on what can be released at release time: 't_{x+1}'. The extra (potentially very large) job that is released has to leave a given work centre before its workload is withdrawn and the capacity is made available for other jobs. The shop floor has to compensate for the overload and thus the capacity available for the release of other jobs is less. This hinders the release of especially large jobs in future periods; thus each time a job is released in this way, it stores up problems for the next release. No positive results have been obtained for release methods B, B′ and C.

5.5 *Results for the load correction approach*

Under the traditional aggregate load approach, jobs which are in-process at a given work centre at release time 't' contribute as a whole to the workload of the resource, adversely affecting the release of jobs from the pool, even though a proportion of the work has been completed and is thus hypothetically downstream. The workload of a work centre is only reduced when the whole of a job has left the work centre and this information has been fed-back from the shop floor. Under the load correction approach, the release procedure compensates for in-process jobs and corrects the load by deducting the hypothetical downstream load. Correcting the load should increase the capacity available for other jobs and make it easier for large jobs to be released. Despite this, no positive results have been obtained. Correcting the load showed no, or only a slight, improvement compared with the traditional approach.

5.6 *Discussion of results*

Results show that using different norms for small and large jobs and dividing the capacity of the shop floor according to job size or routing length is inadequate: it increases the check period and thus the pool delay. This effect could be improved by using a longer planning horizon, and a relaxed norm level, and is worthy of further exploration. Allowing jobs to exceed the workload norm once is also unsuitable for aggregate load methods: it causes an overload that is difficult to handle and to 'get under control'. Similarly, the load correction method has shown no positive effects.

The best approach is prioritisation; all scenarios based on prioritisation led to an improvement in overall performance compared with the standard scenario. Small jobs find it more difficult to be released but the increase in pool delay for small jobs is overshadowed by the pool delay reduction for large jobs. The question is: can deterioration in the performance of small jobs be accepted? In practice, perhaps the answer depends on the proportion of small and large jobs in the company's current job mix and the way in which the performance of the company is measured (i.e. is one on-time small job evaluated in the same way as one on-time large job or is the total work content of a job considered in determining performance?).

A small performance loss for small jobs may be acceptable if the performance of large jobs is clearly improved. It also seems practical to consider larger jobs for release first and then to fill the emerging gaps of free capacity with small jobs. Choosing which jobs are considered for release first has a significant influence on the pool delay and thus on the gross throughput time. In addition to the influence on the pool delay, prioritisation did not have a negative influence on shop floor throughput time performance. It was expected that the combination of changing the sequencing at the release and the FCFS dispatching rule would deteriorate the performance of small jobs at the direct load level. The jobs that are first released are also the first jobs to arrive in the queue in front of the work centre. It was expected that this would lead to deterioration in the performance of small jobs on the floor because there is always likely to be a large job being processed first. However, the negative influence is on the direct load, which is typically small and thus of less influence than the indirect load if the routing length is long. To summarise, consider the following.

- If jobs are prioritised according to size (scenario II), large jobs benefit the most. Jobs with a large routing length but small job size are unlikely to ever be released; this is a major contributing factor to the high average loss in performance for small jobs.
- If jobs are prioritised according to routing length (scenario III), a less significant improvement in performance for large jobs is observed but the deterioration of small jobs is much less, and the best overall performance is obtained.
- The performance of small jobs can be slightly improved using the converted priority method (scenario IV); however, much of the benefit for large jobs that results from prioritisation according to size or routing length is lost.

The way in which processing times are distributed is also important. If processing times follow a 2-Erlang distribution, overall performance is significantly better than if processing times follow an exponential distribution. Prioritisation according to routing length improved performance if processing times are exponentially distributed and thus if job size variation is high. Using this method, there is almost no difference in performance compared with a 2-Erlang distribution. For all approaches, release method B′

(the corrected approach) performed best and method C (the extended approach) performed the worst.

6. Handling rush orders

Despite the importance of rush orders in real-life job shops, where a company may receive an important urgent order at short notice, the topic has received little attention in the wider literature. A rare contribution is made by Wu and Chen (1997) who developed a model to estimate the cost of producing a rush order in an assemble-to-order context. Handling rush orders has not been adequately explored in the WLC literature. The question of how the emergence of rush orders can best be handled within the structure of the WLC concept is an important implementation issue highlighted by Hendry *et al.* (2008). While Hendry *et al.* (2008) suggest reserving a percentage of capacity for rush orders (based on their arrival rate) to cope with the problem, this idea has been rejected as it raises the check period—the same problem as identified in Section 5 when capacity was reserved for small and large jobs respectively. Therefore, following the results outlined in Section 5, this section briefly explores whether prioritisation, the best solution to handling job size variation, could play a similar role in handling rush orders or if allowing rush orders to exceed workload norms provides a better solution.

Figures 10 and 11 summarise the results obtained for release method B, the traditional approach, under three scenarios, for rush orders and the overall remaining orders. Scenario I represents the standard scenario without rush orders; in scenario II, priority is given to rush orders; and in scenario III, rush orders are allowed to exceed workload norms. The results for method B′, the corrected aggregate load, and method C, the extended aggregate load, are similar but not shown here. Method B′ performed best and method C the worst. From this brief extension to the analysis, it is concluded that prioritisation (scenario II) performs the best, especially if processing times are exponentially distributed. If rush orders are allowed to exceed the norm (scenario III), they cause the same uncontrollable overload as outlined in the previous section. The shop floor throughput time performance of rush orders deteriorates due to the uncontrolled load on the shop floor and the remaining jobs have a much longer pool delay caused by the disturbed feedback from the shop floor. Prioritisation has been tested up to a rush order proportion of 30%. This is considered very high—Hendry *et al.* (2008) suggested a rush order proportion of between 10 and 20%—however, the performance of rush orders remained relatively stable irrespective of changes in the rush order percentage. For the overall remaining orders, the lower the percentage of rush orders, the better the performance of non-rush orders.

In an additional approach (scenario IIIi), rush orders were allowed to exceed the norm without contributing to the load—on arrival, they were released directly to the shop floor and neglected by the WLC system. The occupation of the shop floor was maintained at the same level, meaning the WLC system parameters were adapted to the new lower load. Rush orders resulted in a significant loss in shop floor throughput time performance due to the uncontrolled overload on the shop floor. Hence, it is not possible to control only part of the shop floor using a WLC system; if WLC is to be effective, the whole shop floor must be controlled.

7. Conclusion

The order release stage of the Workload Control (WLC) concept has received much attention. Despite this, research has failed to address many of the practical considerations involved in the release of jobs that affect the ability to apply the concept in practice. This paper contributes to the available literature by representing a return from field work to a theory testing environment, demonstrating the complementary roles that case and modelling research can play in the development of theory. An original attempt to address the issue of variations in job size is presented. Several approaches have been tested to satisfy the special requirements of both small and large jobs and to improve the practical applicability of the WLC methodology.

Considering the research questions that were raised in Section 3.2: prioritisation appears to be the best solution to incorporating small and large job sizes within the release mechanism of the WLC concept, providing the best balance between the differing needs of the two job sizes. This improves the performance of large jobs while simultaneously allowing a short check period to be used, as favoured by small jobs. The results obtained for this solution also show greater stability and less deviation among the single results for each simulation run. Although this was not the intention of the work, we can conclude that the robustness of the system has also been improved.

In conclusion, giving priority to jobs with a large routing length is a more effective solution to the problem than reserving capacity for each job size or allowing jobs to exceed the norm. The same conclusion is also shown to be valid for rush orders, where prioritisation proved to be the best solution in order to handle the arrival of rush orders within the WLC concept. While the proposed solution for job size variation is consistent with the suggestion made by Land (2004), the solution for rush orders is in contrast to the suggestion made by Hendry *et al.* (2008). The results have implications for practice by showing that relatively simple methods can improve the performance of release mechanisms. Prioritisation is likely to be the solution that can be most realistically applied in practice—an important driver of theory. However, while prioritisation is considered a relatively simple method of improving the effectiveness of the release mechanism, whether the advantages prioritisation provides outweigh a slight increase in sophistication for the production planner can only be determined by returning to a case study setting—and so the cycle continues.

References

Ahmed, I. and Fisher, W.W., 1992. Due date assignment job order release and sequencing interaction in job-shop scheduling. *Decision Science*, 23, 633–647.

Bechte, W., 1988. Theory and practice of load oriented manufacturing control. *International Journal of Production Research*, 26 (3), 375–395.

Bechte, W., 1994. Load oriented manufacturing control just in time production for job shops. *Production Planning and Control*, 5, 292–307.

Bergamaschi, D., *et al.*, 1997. Order review and release strategies in a job shop environment: A review and a classification. *International Journal of Production Research*, 35 (2), 399–420.

Bertrand, J.W.M., 1983. The effect of workload dependent due-dates on job shop performance. *Management Science*, 29 (7), 799–816.

Bertrand, J.W.M. and Wortmann, J.C., 1981. *Production control and information systems for component-manufacturing shops*. Amsterdam: Elsevier.

Bertrand, J.W.M. and Fransoo, J.C., 2002. Operations management research methodologies using quantitative modelling. *International Journal of Operations and Production Management*, 22 (2), 241–264.

Bertrand, J.W.M. and Van de Wakker, A.M., 2002. An investigation of order release and flow time allowance policies for assembly shops. *Production Planning and Control*, 13 (7), 639–648.

Bertrand, J.W.M. and Van Ooijen, H.P.G., 2002. Workload based order release and productivity: A missing link. *Production Planning and Control*, 13 (7), 665–678.

Bobrowski, P.M., 1989. Implementing a loading heuristic in a discrete release job shop. *International Journal of Production Research*, 27, 1935–1948.

Breithaupt, J.W., Land, M.J., and Nyhuis, P., 2002. The workload control concept: Theory and practical extensions of load oriented order release. *Production Planning and Control*, 13 (7), 625–638.

Cigolini, R. and Portioli-Staudacher, A., 2002. An experimental investigation on workload limiting methods with ORR policies in a job shop environment. *Production Planning and Control*, 13 (7), 602–613.

Cigolini, R., Perona, M., and Portioli, A., 1998. Comparison of order review and release techniques in a dynamic and uncertain job shop environment. *International Journal of Production Research*, 36 (11), 2931–2951.

Enns, S.T. and Prongue Costa, M., 2002. The effectiveness of input control based on aggregate versus bottleneck workloads. *Production Planning and Control*, 13 (7), 614–624.

Fowler, J.W., Hogg, G.L., and Mason, S.J., 2002. Workload control in the semiconductor industry. *Production Planning and Control*, 13 (7), 568–578.

Fredendall, L.D. and Melnyk, S.A., 1995. Assessing the impact of reducing demand variance through improved planning on the performance of a dual resource constrained job shop. *International Journal of Production research*, 33 (6), 1521–1534.

Hendry, L.C., 1989. *A decision support system to manage delivery and manufacturing lead times in make to order companies*. Thesis (PhD). Lancaster University.

Hendry, L.C. and Kingsman, B.G., 1991. A decision support system for job release in make to order companies. *International Journal of Operations and Production Management*, 11, 6–16.

Hendry, L.C. and Wong, S.K., 1994. Alternative order release mechanisms: A comparison by simulation. *International Journal of Production Research*, 32 (12), 2827–2842.

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.

Hendry, L.C., *et al.*, 2008. Investigating implementation issues for workload control (WLC): A comparative case study analysis. *International Journal of Production Economics*, 112, 452–469.

Henrich, P., *et al.*, 2004. Reducing feedback requirements of workload control. *International Journal of Production Research*, 42 (24), 5235–5252.

Kingsman, B.G., 2000. Modelling input–output workload control for dynamic capacity planning in production planning systems. *International Journal of Production Economics*, 68 (1), 73–93.

Kingsman, B.G. and Hendry, L.C., 2002. The relative contributions of input and output controls on the performance of a workload control system in make to order companies. *Production Planning and Control*, 13 (7), 579–590.

Land, M.J., 2004. *Workload control in job shops: Grasping the tap*. Thesis (PhD). University of Groningen. Ridderkerk: Labyrint Publications.

Land, M.J., 2006. Parameters and sensitivity in workload control. *International Journal of Production Economics*, 104 (2), 625–638.

Land, M.J. and Gaalman, G., 1996. Workload control concepts in job shops—a critical assessment. *International Journal of Production Economics*, 46/47, 535–548.

Land, M.J. and Gaalman, G., 1998. The performance of workload control concepts in job shops: Improving the release method. *International Journal of Production Economics*, 56/57, 347–364.

Mather, H. and Plossl, G., 1978. Priority fixation versus throughput planning. *Production and Inventory Management*, 19, 27–51.

Melnyk, S.A. and Ragatz, G.L., 1989. Order review/release: Research issues and perspectives. *International Journal of Production Research*, 27 (7), 1081–1096.

Melnyk, S.A., Ragatz, G.L., and Fredendall, L., 1991. Load smoothing by the planning and order review/release systems: A simulation experiment. *Journal of Operations Management*, 10 (4), 512–523.

Melnyk, S.A., *et al.*, 1994. Evaluating variance control, order review/release and dispatching: A regression analysis. *International Journal of Production research*, 32 (5), 1045–1061.

Missbauer, H., 1997. Order release and sequence-dependent set-up times. *International Journal of Production Economics*, 49, 131–143.

Onur, L. and Fabricky, W.J., 1987. An input/output control system for the dynamic job shop. *IIE Transactions*, 19, 88–97.

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

Park, P.S. and Bobrowski, P.M., 1989. Job release and labour flexibility in a dual resource constrained job shop. *Journal of Operations Management*, 8 (3), 230–249.

Park, P.S. and Salegna, G.J., 1995. Load smoothing with feedback in a bottleneck job shop. *International Journal of Production Research*, 33 (6), 1549–1568.

Perona, M. and Portioli, A., 1998. The impact of parameters setting in load oriented manufacturing control. *International Journal of Production Economics*, 55, 133–142.

Perona, M. and Miragliotta, G., 2000. Workload control: A comparison of theoretical and practical issues through a survey in field, *Proceedings of the 11th international working seminar on production economics*. Innsbruck: Igls, 1–14.

Philipoom, P.R. and Fry, T.D., 1992. Capacity-based order review/release strategies to improve manufacturing performance. *International Journal of Production research*, 30 (11), 2559–2572.

Philipoom, P.R., Malhorta, M.K., and Jensen, J.B., 1993. An evaluation of capacity sensitive order review and release procedures in job shops. *Decision Sciences*, 24 (6), 1109–1133.

Ragatz, G.L. and Mabert, V.A., 1988. An evaluation of order release mechanisms in a job-shop environment. *Decision Sciences*, 19, 167–189.

Sabuncuoglu, I. and Karapinar, H.Y., 1999. Analysis of order review/release problems in production systems. *International Journal of Production Economics*, 62, 259–279.

Shimoyashiro, S., Isoda, K., and Awane, H., 1984. Input scheduling and load balance control for a job-shop. *International Journal of Production Research*, 22 (4), 597–605.

Silva, C., Roque, L., and Almeida, A., 2006. MAPP—A web-based decision support system for the mould industry. *Decision Support Systems*, 42 (2), 999–1014.

Stevenson, M., 2006a. Refining a workload control (WLC) concept: A case study. *International Journal of Production Research*, 44 (4), 767–790.

Stevenson, M., 2006b. *Bridging the gap: Refining a workload control concept for practical implementation*. Thesis (PhD). Lancaster University.

Stevenson, M. and Hendry, L.C., 2006. Aggregate load oriented workload control: A review and a reclassification of a key approach. *International Journal of Production Economics*, 104, 676–693.

Stevenson, M. and Silva, C., 2008. Theoretical development of a workload control methodology: Evidence from two case studies. *International Journal of Production Research*, 46 (11), 3107–3131.

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.

Tatsiopoulos, I.P., 1983. *A microcomputer-based interactive system for managing production and marketing in small component manufacturing firms using a hierarchical backlog control and lead time management technology*. Thesis (PhD). Lancaster University.

Wiendahl, H.P., 1995. *Load oriented manufacturing control*. Berlin: Springer.

Wisner, J.D., 1995. A review of the order release policy research. *International Journal of Operations & Production Management*, 15 (6), 25–40.

Wu, M.C. and Chen, S.Y., 1997. A multiple criteria decision-making model for justifying the acceptance of rush orders. *Production Planning and Control*, 8 (8), 753–761.

Zäpfel, G. and Missbauer, H., 1993. New concepts for production planning and control. *European Journal of Operational Research*, 67, 297–320.