Placement of Effective Work-In-Progress Limits in Route-Specific Unit-Based Pull Systems.

Nick Ziengs, Jan Riezebos, Remco Germs

Placement of Effective Work-In-Progress Limits in Route-Specific Unit-Based Pull Systems.

Journal:	*International Journal of Production Research*
Manuscript ID:	TPRS-2010-IJPR-1105
Manuscript Type:	Original Manuscript
Date Submitted by the Author:	04-Nov-2010
Complete List of Authors:	Ziengs, Nick; University of Groningen, Faculty of Economics and Business, Department of Operations Riezebos, Jan; University of Groningen, Faculty of Economics and Business, Department of Operations Germs, Remco; University of Groningen, Faculty of Economics and Business, Department of Operations
Keywords:	SHOP FLOOR CONTROL, DISCRETE EVENT SIMULATION, PULL SYSTEMS, DESIGN OF PRODUCTION SYSTEMS, LEAD-TIME REDUCTION
Keywords (user):	POLCA

Placement of Effective Work-In-Progress Limits in Route-Specific Unit-Based Pull Systems

Nick Ziengs, Jan Riezebos, and Remco Germs

Faculty of Economics and Business, Department of Operations, University of Groningen, Groningen, The Netherlands

Nick Ziengs (corresponding author)

University of Groningen, Faculty of Economics and Business, Department of Operations, PO BOX 800, 9700 AV Groningen, the Netherlands

Tel.: +31 (0) 50 363 8231; Fax: + 31 (0) 50 363 2032; E-mail: n.ziengs@rug.nl

Jan Riezebos

University of Groningen, Faculty of Economics and Business, Department of Operations, PO BOX 800, 9700 AV Groningen, the Netherlands

Tel.: +31 (0) 50 363 4853; Fax: + 31 (0) 50 363 2032; E-mail: j.riezebos@rug.nl

Remco Germs

University of Groningen, Faculty of Economics and Business, Department of Operations, PO BOX 800, 9700 AV Groningen, the Netherlands

Tel.: +31 (0) 50 363 8556; Fax: + 31 (0) 50 363 2032; E-mail: r.germs@rug.nl

Placement of Effective Work-In-Progress Limits in Route-Specific Unit-Based Pull Systems

Unit-based pull systems control the throughput time of orders in a production system by limiting the number of orders on the shop floor. In production systems where orders can follow different routings on the shop floor, route-specific pull systems that control the progress of orders on the shop floor by placing limits on the number of orders in (parts of) a routing, have shown to be effective in controlling throughput times. This is because route-specific pull systems are able to create a balanced distribution of the amount of work on the shop floor, which leads to shorter and more reliable throughput times. The placement of limits on work-in-progress in a route-specific pull system determines to a large extend the workload balancing capability of such a system. This paper shows how the placement of work-in-progress limits affects the workload balancing capability and thereby the throughput time performance of a route-specific unit-based pull system, namely POLCA.

Keywords: unit-based; pull systems; route-specific; workload balancing capability; POLCA

1 Introduction

Short and reliable throughput times are an important competitive advantage for Make-To-Order manufacturing firms who are confronted with routing variety. Throughput time performance can be improved by adequately controlling the release and dispatching of work to and on the shop floor. The release and dispatching of work is regulated by means of a material control system. These material control systems are referred to as pull systems when they limit the amount of work on the shop floor or on parts thereof (Hopp and Spearman 2004, Hopp and Spearman 2008). The placement of work-in-progress limits has shown to be effective in realizing shorter and more reliable throughput times.

In case of routing variety, the throughput time performance of a pull system largely depends on its capability to effectively balance the workload. A balanced workload refers to an even distribution of work among the workstations on the shop floor (Land and Gaalman 1998). Workload balancing is achieved by the appropriate placement of work-in-progress limits and decreases the time between the release and departure of work from the production system. The workload is said to be balanced effectively when this reduction in shop floor throughput time coincides with a reduction of total throughput time, i.e. the time between the moment of arrival and departure from the system.

A pull system is able to balance the workload effectively when route-specific information is used in the release and dispatching of work. Route-specific information refers to information about the availability of downstream capacity within a specific routing (Riezebos 2010). In card-based pull systems route-specific information is conveyed by means of production authorization cards. An available card signals that capacity is or will soon become available within a routing or part of a routing which is controlled by a work-in-progress limit. The placement of such a limit influences which information is used in the release and dispatching of work.

The placement of work-in-progress limits has been addressed before in literature on pull systems. The focus, however, has mostly been on repetitive manufacturing and the placement of work-in-progress limits within serial production lines (e.g. Conway *et al.* 1988, Dallery and Gershwin 1992, Gaury *et al.* 2001). Other manufacturing environments where routing variety is an important determinant of throughput time performance have often been overlooked. Nevertheless, routing variety is present in most manufacturing environments including Make-To-Order environments. We aim to show that routing variety brings about additional considerations for the effective placement of work-in-progress limits.

In this paper, we concentrate on the placement of work-in-progress limits in route-specific unit-based pull systems. More specifically, we are interested in how the placement of these limits affects the workload balancing capability of such a pull system. We use simulation to demonstrate the effects of the placement of work-in-progress limits throughout various parts of the shop floor. We restrict our focus to unit-based pull system, which limit the number of orders on the shop floor instead of the number of hours required to process an order, because they are the most straightforward pull systems which are still able to balance the workload effectively (Germs and Riezebos 2010).

The structure of this paper is as follows. In the next section we present a categorization of pull systems. We use this categorization to support our selection of an appropriate unit-based pull system, namely POLCA, which we use to determine the effects of the placement of work-in-progress limits on workload balancing capability. Based on our selection we further detail the research statement in Section 3. In Section 4 the simulation model and experimental design are discussed and in Section 5 the results of our simulations are presented. The final section concludes with a discussion of the results and directions for future research.

2 A card-based pull system categorization

Pull systems are material control systems which control the release and dispatching of work by liming the amount of work-in-progress which is allowed on the whole or parts of the shop floor (Hopp and Spearman 2004, Hopp and Spearman 2008). In card-based pull systems work-in-progress is limited throughout control loops on the shop floor. A control loop represents a part of the shop floor in which cards control work-in-progress. The number of cards within a control loop determines the size of the limit. An available card signals that downstream capacity is or soon will become available. Once the card is attached, work is allowed to progress to the next workstation in its routing. As such, work-in-progress limits enable downstream information related to the availability of capacity to be signaled upstream and used in releasing or dispatching work.

Although all card-based pull systems use this same basic mechanism to limit work-in-progress, there are a number of differences between pull systems. In this paper we distinguish these systems based on the following three characteristics, namely (1) unit- or load-based pull systems; (2) product-specific, product-anonymous, or route-specific control; and (3) connected or unconnected workstations.

Pull systems are either unit- or load-based. Unit-based systems control the number of orders on the shop floor. In a unit-based system a card represents a single order. The actual processing time requirements of that order are not taken into account and an available card is therefore only a rough approximation of the availability of capacity. KANBAN (Sugimori et al. 1977, Ōno 2003) and CONWIP (Spearman *et al.* 1990) are both examples of unit-based systems. Load-based systems, on the other hand, do take the processing time requirements of orders into account. Here, a card represents a predetermined amount of work rather than a single order. Load-Based POLCA (Vandaele et al. 2008) provides an example of a load-based pull system. Although load-based systems use a more accurate estimate of the processing time requirements of orders, unit-based systems are more straightforward and therefore more frequently used.

The second distinction is between product-specific, route-specific, and product-anonymous control (Riezebos 2010). In case of product-specific control, a card is dedicated to a specific product type. An available card signals whether capacity for a specific product will soon become available. Consequently, the use of product-specific control requires a separate buffer for each product type and is therefore not suited for manufacturing environments with a high degree of product or routing variety (e.g. Spearman *et al.* 1990, Krishnamurthy *et al.* 2004). KANBAN is the foremost example of product-specific control.

Route-specific control uses cards which are not dedicated to a specific type of product; instead cards are dedicated to a specific routing or part thereof. Hence, an available card signals whether capacity is or soon will become available for another order with the same (partial) routing, rather than for a specific product type. As such, information about the availability of capacity in the routing is considered when releasing or dispatching orders. POLCA (Suri 1998) and m-CONWIP (e.g. Germs and Riezebos 2010) are examples of route-specific control.

In case of product-anonymous control, orders progress regardless of product type or routing. That is, cards are not dedicated to a specific product type or (part of) a routing. CONWIP is a well known example of product-anonymous control. CONWIP authorizes the release of an order as soon as another order departs from the production system regardless of product type or routing.

Figure 1 illustrates the main differences between product-specific, product-anonymous, and route-specific pull systems in more detail. The dotted lines in the figure represent the control loops. The number within the cards indicates the control loop to which the card belongs. Product-specific cards are shaded according to the shading of their corresponding orders, whereas product-anonymous cards have a dark shade. Comparison of Figure 1a and 1c shows that the main difference between product-specific and product-anonymous control is the type of cards used, namely the use of either product-specific or anonymous cards. Alternatively, comparison of Figure 1c and 1b reveals that product-anonymous and route-specific control only differ in terms of the structure of control loops used. Here, the structure refers to the arrangement of control loops used to regulate the release and dispatching of orders in a pull system (Gaury, Kleijnen *et al.* 2001, Kleijnen and Gaury 2003).

<insert figure 1(a), 1(b), 1(c), and Legend here>

The last distinction is between connected and unconnected workstations. Workstations are connected if control loops span more than a single workstation. That is, workstations are connected if a single card signals information about the availability of capacity from one workstation to another workstation. Workstations can be either connected or unconnected in case of both product-specific and product-anonymous control. For instance, in case of product-specific control Kanban offers an example of unconnected workstations, whereas Generalized-Kanban (Buzacott 1989) or Extended-Kanban (Dallery and Liberopoulos 2000) are examples of connected workstations. In case of product-anonymous control Generic-Kanban (Chang and Yih 1994) provides an example of unconnected workstations and CONWIP is an example of workstations which are connected. Opposingly, in case of route-specific control workstations need to be connected. Workstations can either be connected by a single control loop (e.g. m-CONWIP) or by multiple overlapping control loops (POLCA). Both allow information related to the availability of capacity to be sent upstream within a routing (Gaury *et al.* 2001, Kleijnen and Gaury 2003, Gstettner and Kuhn 1996).

Table 1 provides an overview of a number of pull systems according to the categorization presented above. This overview is not meant to be exhaustive; rather we use it to identify a suitable pull system to study the effects of the placement of work-in-progress limits on effective workload balancing capability. Based on the categorization we selected a pull system which provides a large number of opportunities to study the placement of work-in-progress limits, namely POLCA. Although m-CONWIP also provides route-specific control we selected POLCA because it allows us to study the placement of work-in-progress limits without directly prioritizing specific routings. Alternatively, if no limit is placed for a routing in an m-CONWIP system that routing is prioritized over other routings. POLCA, therefore, allows us to evaluate whether the placement of a work-in-progress limit influences where the workload is balanced, whereas m-CONWIP does not.

POLCA is a route-specific unit-based pull system that uses multiple overlapping control loops to control the progress of work on the shop floor. In POLCA, each control loop connects two workstations. Similar to other card-based pull systems, POLCA requires a card to be attached before an order is allowed into the control loop. However, unlike most pull systems, before an order is allowed to move to the second workstation within the control loop an additional card needs to be attached because this workstation is also controlled by a second control loop. This card signals the availability of capacity for the third workstation within the routing. As such, POLCA is able to signal route-specific information to upstream workstations and balance the workload. Hence, POLCA allows us to evaluate the effects of the placement of multiple work-in-progress limits on its effective workload balancing capability. For additional details on POLCA we refer to Suri (1998), Suri and Krishnamurthy (2009), and Riezebos (2010).

<insert figure 1(a), 1(b), and 1(c) here, include caption 1(a), 1(b), 1(c), and 1>

3 Research Statement

In order to determine the effective workload balancing capability of a route-specific unit-based pull system, we distinguish between three commonly used measures of throughput time performance. These measures are based on the distinction between (1) the time an orders spends waiting before the shop floor and (2) the time an order spends on the shop floor. The average time between the arrival of orders and their release to the shop floor is referred to as the order pool time (OPT) and the average time between the release of orders to the shop floor and their departure from the production system is referred to as the shop floor throughput time (STT). The sum of the order pool time and shop floor throughput time is referred to as the total throughput time (TTT), i.e. the average time between the arrival of orders and their departure from the production system. For additional details on these measures we refer to Oosterman *et al.* (2000).

In general, placing a limit on work-in-progress results in a reduction of shop floor throughput time in the part of the shop floor that is limited. A pool of orders will accumulate before the limit. For instance, limiting work-in-progress on the whole shop floor results in a reduction in shop floor throughput time. The placement of such a limit prohibits the direct release of orders to the shop floor and therefore increases the order pool time of orders. Route-specific pull systems use this increase in orders before the shop floor to balance the workload at the moment of release and realize a reduction in shop floor throughput time.

Figure 2a shows the performance of a pull system which exhibits effective workload balancing capability by balancing the workload at the moment of release. The configuration at the rightmost end of the curve represents a non-limited system – identified by a square. Here, the configuration does not limit the workload and the shop floor throughput time will equal the total throughput time as orders are immediately released to the shop floor. The configuration – identified by small dots – becomes more constrained once we move to the left of the curve by gradually decreasing the number of cards. This results in a decrease in shop floor throughput time and an increase in order pool time. Since the pull system exhibits effective workload balancing capability, the increase in order pool time does not yet outweigh the reduction in shop floor throughput time. As such, the total throughput time decreases as well. The total throughput time is the lowest at the critical point of the curve. This is referred to as the optimal configuration since the critical point represents the maximum reduction in total throughput time which can be achieved by limiting work-in-progress – identified by a diamond shape. When moving even further to the left of the curve the increase in order pool time starts to offset the decrease in shop floor throughput time resulting in an increase in total throughput time. Hence, limiting work-in-progress on the shop floor in order to balance the workload at the release results in a trade-off between order pool time and shop floor throughput time.

The workload does not necessarily have to be balanced at the moment of release and can be balanced on the shop floor as well. Figure 2b shows the throughput time performance curve of a pull system which exhibits effective workload balancing capability by balancing the workload after the moment of release. Here, the release of orders is not delayed and orders are directly released to the shop floor upon the moment of arrival. Again, starting at the non-limited configuration and moving to the left of the curve, we find configurations where the shop floor and total throughput time have decreased due to limiting work-in-progress. By moving further to the left of the curve we reach the optimal configuration. Decreasing the number of cards even further will increases shop floor as well as total throughput time along the same 45 degree line.

The placement of work-in-progress limits needs to be carefully considered as the placement determines at which part of the shop floor the workload will be balanced. In general, previous literature on Workload Control and CONWIP (for a review see Stevenson *et al.* 2005) emphasizes the moment of release. That is, it is generally assumed that work-in-progress limits should cover the whole shop floor and the explicit placement of work-in-progress limits in some parts of the shop floor is not explicitly considered. By comparing the effects of placing a limit on work-in-progress in various parts of the shop floor we will be able to evaluate whether work-in-progress limits should be placed throughout the whole shop floor or only at some specific parts of the shop floor. We will address the following research question in the remainder:

"How does the placement of work-in-progress limits affect the effective workload balancing capability of a route-specific unit-based pull system?"

<insert figure 2(a), 2(b) here, include caption 2(a), 2(b), and 2>

4 Methodology

A discrete-event simulation model has been developed to study the effects of the placement of work-in-progress limits on the effective workload balancing capability of route-specific unit-based pull systems. The simulation model and experimental design are discussed in the following two subsections.

4.1 Model design

In our study we distinguish between two variants of a production system. In both cases a Make-To-Order policy is used and production is allowed to start only after an order has arrived. The shop floor topology and pull structure of both variants is shown in Figure

3. The arrangement of workstations on the shop floor is referred to as the shop floor topology. Both variants have a divergent topology where the number of workstations doubles each consecutive stage. Such a divergent topology allows the workload to be balanced at every stage. Hence, the divergent topology is ideally suited to examine the effect of placement on workload balancing capability and throughput time performance.

The two variants differ with respect to the number of production stages. The first variant has three consecutive stages consisting of 7 workstations (A-G) and the second has four consecutive stages consisting of 15 workstations (A-O). The capacity of a workstation is constant and a workstation is allowed to process one order at a time. Orders are processed by a single workstation at each stage. In addition, for both topologies the number of routings equals the number of workstations in the last stage of the topology, each routing is equally likely to occur, and the processing time of workstations doubles every stage. This ensures that all workstations have the same average utilization level which allows us to set a single card count for all control loops between two consecutive stages. As such, we set two card counts for the three-stage topology and three card counts for the four-stage topology. The addition of a fourth stage provides more opportunities for the placement of work-in-progress limits and enables us to closely examine the use of multiple overlapping loops per routing. Moreover, the comparison with the three-stage topology allows us to assess whether the number of stages influences the placement of such a limit.

The model has been constructed using a discrete-event simulation library of routines within Delphi named DESIMP. The model was verified by comparing the results of several replications to the model developed by Germs and Riezebos (2010). We found no significant differences using a paired *t*-test at a 95% confidence level.

<insert figure 3, include caption 3>

4.2 Experimental design

Table 2 provides an overview of the experimental factors and corresponding levels considered in our study, namely batch size, utilization level, and the distribution of inter-arrival and processing times. These experimental factors have been shown to influence the effective workload balancing capability of route-specific pull systems (see also Germs and Riezebos 2010). The processing times are either constant or Erlang-2 distributed. The average processing time of the first workstation is one time-unit and doubles each stage. The inter-arrival times are either constant or exponentially distributed. The average inter-arrival times are chosen so that the average utilization of the workstation is constant at 80%, 85%, or 90%. The batch size refers to the number of orders arriving simultaneously and is either 1 or 10.

For ease of comparison we divided the experiments into three series which are also listed in Table 2. In series 1 we consider the optimal configuration under the restriction that each control loop within a stage must have the same number of cards, i.e. [$n_1 = n_2 = \ldots = n_m$] where n_m represents the number of cards in the control loops starting at stage m. By placing a limit in the first stage we choose balance the workload at the moment of release. The critical point is found by gradually reducing the number of cards in all control loops. Series 1 allows us to determine the effective workload balancing capability in case work-in-progress is limited without considering the placement of the limits. This practice of uniformly limiting work-in-progress is very common (e.g. Spearman *et al.* 1990, Suri 1998, Krishnamurthy and Suri 2009). Therefore, this series serves as a benchmark against which we can compare the effective workload balancing capability when the placement of work-in-progress is explicitly considered.

In series 2 we relax the restriction on the number of cards and allow the number of cards to vary between stages, i.e. [n_1, n_2, …, n_m]. Relaxing the restriction on the number of cards enables us to identify the optimal configuration for each experimental design. The optimal configuration is found by identifying the critical point for this series of experiments. If we observe a reduction in total throughput time compared to series 1 this would mean that work-in-progress limits should not be uniformly placed throughout the shop floor, rather the placement of each work-in-progress limit should be carefully evaluated.

In series 3 we demonstrate the effects of the placement of work-in-progress limits in parts of the shop floor, i.e. [n_i, ∞], [n_i, n_j, ∞], etc. where n_i and n_j represent a finite number of cards in stage *i* and stage *j* respectively. In this series we determine the critical point given that one or more stages are not limited. This allows us to contrast the effects of the placement of work-in-progress limits throughout various parts of the shop floor. Thereby, we are able to determine at which stage the placement of a limit contributes most to the effective workload balancing capability of the pull system.

For all experiments we measure the average shop floor and total throughput time. The averages are used to construct the throughput time performance curves and determine the ratio of the achieved reduction in average shop floor and total throughput time relative to the non-limited configuration, [∞, ∞, ∞]. The averages are based on 100 replications with a run-length of 100.000 time-units. The averages collected within the first 25.000 time-units were disregarded in order to eliminate the initial transient. We used Welch's procedure, as detailed by Law and Kelton (2000), to confirm that the warm-up period was sufficient.

<insert table2, include caption 2>

5 Results

In this section we present the results of our simulation experiments. The results for each series of experiments is discussed in a separate subsection. In line with previous research (Germs and Riezebos, 2010), experiments with random processing times showed only modest or no effective workload balancing capability and have therefore been omitted from our numerical results.

5.1 *Series 1: same number of cards per stage [n_1=n_2=…=n_3]*

Figure 4 shows the absolute throughput time performance of the route-specific unit-based pull system for the three- and four-stage topology given the restriction that the number of cards in each control loop is the same. The effects of increased batch size, increased utilization level, and randomness of inter-arrival times are shown using four combinations of experimental factors. The first combination (a) shows the throughput time performance given a batch size of 1, a utilization level of 80%, and constant inter-arrival times. In the additional combinations (b, c, and d) we change the level of one of the experimental factors with respect to combination (a) in order to visualize the influence of the experimental factors on the effective workload balancing capability. In combination (b) we increase the batch size, in combination (c) the utilization level is increased, and in combination (d) we use randomly distributed instead of constant inter-arrival times. The throughput time performance of all combinations relative to the non-limited system is found in Table 3.

Figure 4 shows that the pull system is able to balance the workload effectively in both the three- and four-stage topologies. For all combinations (a, b, c, and d) it is shown that decreasing the number of cards results in a reduction in total throughput time compared to the non-limited system. Throughput time continues to decrease until the critical point is reached. The figure also shows that for all combinations a different effect on the effective workload balancing capability can be observed. The pull system achieves the largest reduction in total throughput time when having to cope with random (d) instead of constant inter-arrival times (a). The smallest reduction in total throughput time is observed when the utilization level increases from 80% (a) to 85% (c). In addition, the combinations show an increase in absolute total throughput time when changing the levels of the experimental factors. For the three-stage topology, increasing the batch size (b) results in the largest increase in total throughput time, whereas random instead of constant inter-arrival times (d) result in the smallest increase in total throughput time. Alternatively, for the four-stage topology an increase in batch size (c) results in the smallest increase in total throughput time and random instead of constant inter-arrival times (d) result in the largest increase in total throughput time.

Further comparison shows that the pull system is able to balance the workload more effectively in case of the four-stage topology. For example, the four-stage topology is able to achieve a reduction in total throughput time of 0.80%, whereas the three-stage topology is only able to achieve a reduction of 0.47% for combination (a). This difference can be attributed to the additional workstations the workload can be balanced among, i.e. workstations H, I, …, and O. However, the positive effects of a larger batch size, increased utilization, and random inter-arrival times diminish with the number of stages. For instance, given a batch size of 10, a utilization level 90%, and exponential inter-arrival times, the three-stage topology is able to achieve a 5.76% reduction of total throughput time, whilst the four-stage topology is only able to achieve a 3.95% reduction. For all experimental designs the average reduction in total throughput time is 2.52% (95% CI; 1.28 % - 3.77%) and 1.89% (95% CI; 1.14% - 2.64%) for the three- and four-stage topology respectively, which suggests that the three-stage topology is more robust in terms of effective workload balancing capability.

Table 3 also shows the number of cards used in each control loop. Overall, the four-stage topology requires a slightly larger number of cards in each control loop. For both topologies an increase in batch size, an increase in utilization level, and random instead of constant inter-arrival times result in an increase in the required number of cards. An increase in batch size in case of the three-stage topology proves to be an exception. Nevertheless, using the same number of cards in each control loops results in an improvement of throughput time performance over the non-limited system for all experimental designs.

<insert figure 4(a) and 4(b) here, include caption 4(a), 4(b), and 4>

<insert table 3, include caption 3>

5.2 Series 2: varying number of cards per stage [n_1, n_2, …, n_3]

Figure 5 shows the throughput time performance curves for combination (a) of series 1 and 2. The plots allow us to demonstrate the effects of the restriction that the same number of cards is used in each control loop. Comparing the curves of series 1 and 2 shows that the absolute reduction in total throughput time is considerably larger when we relax the restriction on the number of cards. This additional reduction is observable for both the three- and four-stage topologies, although it is larger for the four-stage topology. The figure also shows that the shop floor throughput time equals the total throughput time for the second series of experiments. Hence, for combination (a) of series 2 orders do not incur an order pool time and are released directly to the shop floor at the moment of arrival and a larger reduction in total throughput time is achieved without placing a work-in-progress limit in the first stage.

Table 4 shows that for all experimental designs a reduction in total throughput time can be observed when relaxing the restriction on the number of cards. The average reduction in total throughput time was 2.21% (95% CI; 1.53% - 2.87%) for series 1 and is 8.01% (95% CI; 6.89% - 9.12%) for series 2. Moreover, most experimental designs show a reduction in shop floor throughput time equal to the reduction in total throughput time. Only those experiments with both a batch size of 10 and random inter-arrival times are exceptions for the three-stage topology and experiments with random inter-arrival times are exceptions for the four stage topology. Still, in most cases work-in-progress is not limited in the first stage and the workload is more effectively balanced at the later stages of the topology. As such, the placement of a work-in-progress limit has a relatively large influence on the effective workload balancing capability of the pull system.

Table 4 also shows that for the optimal configuration of each experimental design one or more stages are not limited. For the three-stage topology either the first or the second stage is limited. In general, work-in-progress limits are placed in the second stage. However, limiting the first stage results in a larger reduction in total throughput time in case of increased batch size and random inter-arrival times. For the four-stage topology similar results are shown. Here, either one or two stages are limited. In case of constant inter-arrival times only the last stage is limited. However, in case of random inter-arrival times the first and the last stages are limited and no limit on work-in-progress is placed in the intermediate stage. Hence, for the optimal configuration of both the three- and four-stage topology the control loops within a routing do not overlap. As such, using a structure of overlapping loops results in a smaller reduction in total throughput time for these topologies than possible. In the next series of experiments we will examine the placement of work-in-progress limits in more detail.

<insert figure 5(a) and 5(b) here, include caption 5(a), 5(b), and 5>

<insert table 4, include caption 4>

5.3 Series 3: a finite number of cards in a limited number of stages [n_i, ∞], [n_i, n_j, ∞], etc

Table 5 provides an overview of the results of the third series of experiments. In series 3 we review the effects of the explicit placement of work-in-progress limits in some of the stages. The table shows that the placement of a work-in-progress level influences the effective workload balancing capability of the pull system. In general, the placement of work-in-progress limits in the last stage is more effective than the placement of work-in-progress limits in the first stage(s).This contradicts earlier research on Workload Control and CONWIP which advocates limiting work-in-progress in the first stages, thereby balancing the workload at the moment of release. For the three-stage topology limiting the first stage results in an average reduction of 4.00% (95% CI; 1.88% - 6.11%), whereas limiting the last stage results in an average reduction of 6.40% (95% CI; 5.22% - 7.58%). Four the four-stage topology, limiting the first stage results in an average reduction of 2.20% (95% CI; 0.89% - 3.51%), limiting the second stage results in an average reduction of 2.98% (95% CI; 2.36% - 3.60%), and limiting the last stage results in an average reduction of 6.89% (95% CI; 5.96% - 7.82%).

If we compare the optimal configuration of series 2 to the reduction achieved by limiting the last stage we see a relatively small increase in total throughput time. For the three-stage topology the average reduction in total throughput time was 6.82% (95% CI; 5.42% - 8.2%) and is 6.40% (95% CI; 5.22% - 7.58%) when limiting the last stage. For the four-stage topology the difference is slightly larger since the average reduction in total throughput time for series 2 was 9.19% (95% CI; 7.55% - 10.84%) and is 6.89%

(95% CI; 5.96% - 7.82%) when limiting the last stage. The increased reduction is due to the experiments with random inter-arrival times for the four-stage topology and increased batch size and random inter-arrival times for the three-stage topology.

Table 5 also lists the throughput time performance of the critical point given each restriction. The results show that given the change to limit two consecutive control loops it is always optimal to limit a single control loop. Consider, for instance, the optimal configuration for the four-stage topology when allowing a limit in the first two stages, i.e. $[\infty, n_i, n_j]$. For this restriction it is optimal to limit only the last stage and not the final two stages, i.e. $[\infty,\infty, n_j]$. The other configurations also show that limiting work-in-progress in two consecutive stages reduces the workload balancing capability and throughput time performance. Hence, using a structure of overlapping control loops results in a decrease in reduction in total throughput time and degrades the effective workload balancing capability of the pull system.

<insert table 5, include caption 5>

6 Conclusion

Short and reliable throughput times are an important competitive advantage for make-to-order companies. Route-specific, as opposed to product-specific or product-anonymous, pull systems are able to achieve shorter and more reliable throughput times by balancing the workload. A balanced workload is accomplished through the provision of route-specific information to upstream workstations in the form of available cards. This transfer of route-specific information is facilitated by a structure of route-specific control loops. The placement of a work-in-progress limit in these loops allows route-specific information to be used for the release and dispatching of orders. The information is then used to balance the workload at workstations where two or more route-specific control loops intersect.

In this paper we have shown that the placement of work-in-progress limits affects the workload balancing capability and throughput time performance of a route-specific unit-based pull system, namely POLCA. POLCA makes use of a structure of multiple overlapping control loops to pass on route-specific information and, thus, requires multiple work-in-progress limits for each part of a routing.

In general, limiting work-in-progress results in a decrease in both shop floor and total throughput time. Our results show that the reduction in total throughput time is largest when work-in-progress is not limited in the first stage(s) of control loops. Not limiting work-in-progress in the first stage(s) of control loops decreases the number of workstations at which the workload can be balanced. In addition, limiting work-in-progress in two consecutive stages always results in decreased workload balancing capability and throughput time performance. This suggests that the structure of overlapping control loops prohibits all required information to be transferred entirely upstream. That is, in a divergent topology control loops shared by multiple routings will hinder the flow of orders and information in both routings when there is a lack of available capacity in only one of those routings. Consequently, orders for which downstream capacity is available still incur an additional and unnecessary waiting time thereby diminishing throughput time performance.

These results contradict prior research on CONWIP or Workload Control which emphasizes delaying the release of orders to the shop floor over dispatching. We show that balancing the workload at a later stage outperforms balancing the workload at the moment of release for the chosen pull system. However, delaying the release of orders becomes increasingly important once the order pool gets larger due to, for instance, increased batch size or random inter-arrivals. Overall, we conclude that the placement of a work-in-progress limit is most effective at the last stage of a routing and the effects of an additional limit at the first stage is much smaller than the effects of limiting at the last stage.

Our results provide a number of opportunities for future research. First, similar to previous research (Germs and Riezebos 2010), we found only modest or no effective workload balancing capability in experiments with random processing times. As such, it would be interesting to see the degree improvement in terms of workload balancing capability when using a load-based variant. Second, control loops that connect more than two workstation might more accurately signal whether or not downstream capacity is available due to a larger part of the routing being included in the control loop. Hence, future research might look into the effects of extending the control loops to encompass more than two workstations.

References

Buzacott, J.A., 1989. Queueing models of Kanban and MRP controlled production systems. *Engineering Costs and Production Economics,* 17(1-4), 3-20.

Chang, T.M. and Yih, Y., 1994. Generic kanban systems for dynamic environments. *International Journal of Production Research,* 32(4), 889-902.

Conway, R., Maxwell, W., McClain, J.O. and Thomas, L.J., 1988. The Role of Work-In-Process Inventory in Serial Production Lines, *Operations research,* 36(2), 229-241.

Dallery, Y. and Liberopoulos, G., 2000. Extended kanban control system: combining kanban and base stock. *IIE Transactions,* 32(4), 369-386.

Dallery, Y. and Gershwin, S.B., 1992. Manufacturing flow line systems: a review of models and analytical results. *Queueing Systems,* 12(1), 3-94.

Fernandes, N.O. and do Carmo-Silva, S., 2006. Generic POLCA—A production and materials flow control mechanism for quick response manufacturing. *International Journal of Production Economics,* 104(1), 74-84.

Gaury, E.G.A., Kleijnen, J.P.C. and Pierreval, H., 2001. A Methodology to Customize Pull Control Systems. *The Journal of the Operational Research Society,* 52(7), 789-799.

Geraghty, J. and Heavey, C., 2005. A review and comparison of hybrid and pull-type production control strategies. *OR Spectrum,* 27(2), 435-457.

Germs, R. and Riezebos, J., 2010. Workload balancing capability of pull systems in MTO production. *International Journal of Production Research,* 48(8), 2345-2360.

Gstettner, S. and Kuhn, H., 1996. Analysis of production control systems kanban and CONWIP. *International Journal of Production Research,* 34(11), 3253-3273.

Hopp, W.J. and Spearman, M.L., 2008. *Factory physics*. New York: McGraw-Hill/Irwin.

Hopp, W.J. and Spearman, M.L., 2004. To Pull or Not to Pull: What Is the Question?. *Manufacturing Service Operations Management,* 6(2), 133-148.

Kleijnen, J.P.C. and Gaury, E.G.A., 2003. *Short-term robustness of production management systems: A case study*. European Journal of Operations Research. 148(2), 452-465.

Krishnamurthy, A. and Suri, R., 2009. Planning and implementing POLCA: a card-based control system for high variety or custom engineered products. *Production Planning & Control,* 20(7), 596-610.

Krishnamurthy, A., Suri, R. and Vernon, M., 2004. Re-Examining the Performance of MRP and Kanban Material Control Strategies for Multi-Product Flexible Manufacturing Systems. *International Journal of Flexible Manufacturing Systems,* 16(2), 123-150.

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

Law, A.M. and Kelton, W.D., 2000. *Simulation modeling and analysis*. Boston: McGraw-Hill.

Ōno, T., 2003. *Toyota production system : beyond large-scale production*. Portland: Productivity Press.

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.
Riezebos, J., 2010. Design of POLCA material control systems. *International Journal of Production Research*, 48(5), 1455-1477.

Spearman, M.L., Woodruff, D.L. and Hopp, W.J., 1990. CONWIP: A pull alternative to kanban, *International Journal of Production Research*, 28(5), 879-894.

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.

Sugimori, Y., Kusunoki, K., Cho, F. and Uchikawa, S., 1977. Toyota production system and Kanban system Materialization of just-in-time and respect-for-human system. *International Journal of Production Research,* 15(6), 553-564.

Suri, R., 1998. *Quick response manufacturing: a companywide approach to reducing lead times*. Portland: Productivity Press.

Vandaele, N., Van Nieuwenhuyse, I., Claerhout, D. and Cremmery, R., 2008. Load-Based POLCA: An Integrated Material Control System for Multiproduct, Multimachine Job Shops. *Manufacturing Service Operations Management,* 10(2), 181-197.

Figure 1(a). Product-specific control (KANBAN).

Figure 1(b). Route-specific control (m-CONWIP).

Figure 1(c). Product-anonymous control (generic-KANBAN).

Figure 1. Pull System categories.

Figure 2(a). Limited at the moment of release.

Figure 2(b). Limited after the moment of release.

Figure 2. Throughput time performance curves.

Figure 3. Three- and four-stage topology and pull structure

Figure 4(a). Three-stage topology.

Figure 4(b). Four-stage topology.

Figure 4. Shop floor and total throughput time for the three- and four-stage topology for series 1. Combination (a) batch size of 1, utilization level of 80%, constant inter-arrival times; (b) batch size of 10, utilization level of 80%, constant inter-arrival times; (c) batch size of 1, utilization level of 85%, constant inter-arrival times; (c) batch size of 1, utilization level of 80%, exponentially distributed inter-arrival times.

Figure 5(a). Three-stage topology.

Figure 5(b). Four-stage topology.

Figure 1. Shop floor throughput time performance for the three- and four-stage topology given the same number of cards (series 1) and a varying number of cards (series 2). Combination (a) batch size of 1, utilization level of 80%, constant inter-arrival times.

Table 1. A card-based pull system categorization.

Table 2. Experimental design.

Table 3. Optimal throughput time performance for a three- and four stage topology*.

Table 4. Optimal throughput time performance of a three-and four stage topology*.

Table 5. Optimal throughput time performance of a three- and four-stage topology*.

Figure 1(a). Product-specific control (KANBAN).

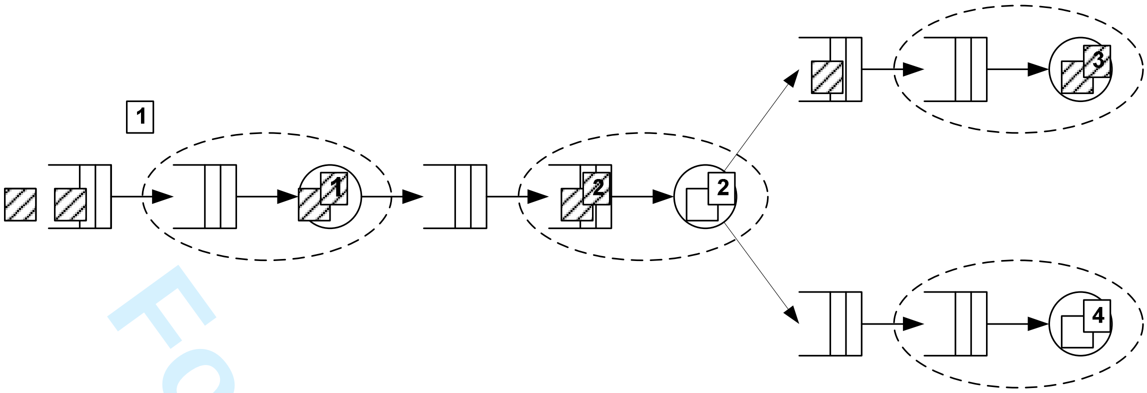


Figure 1(b). Route-specific control (m-CONWIP).

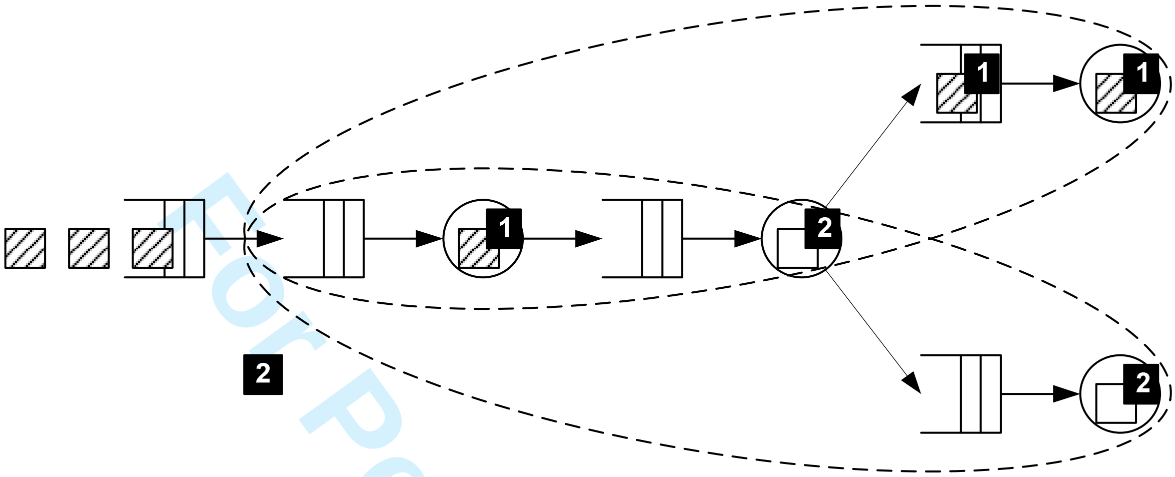
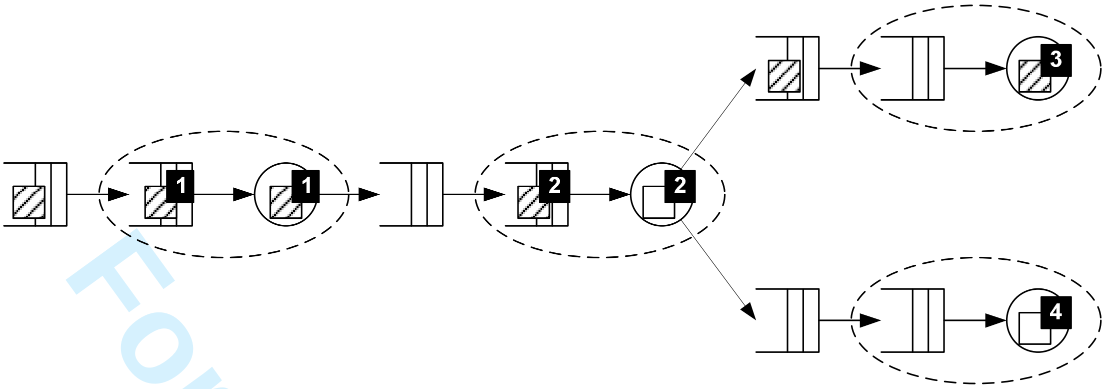


Figure 1(c). Product-anonymous control (generic-KANBAN).



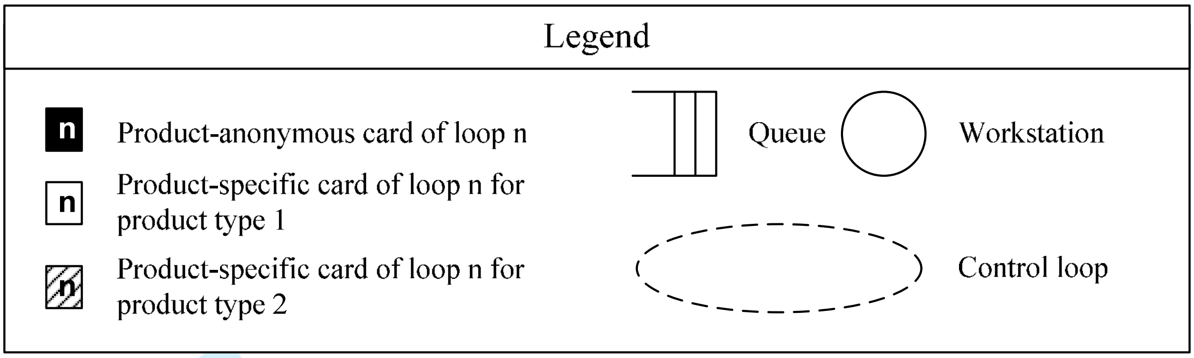

Legend
n
Product-anonymous card of loop n
n
Product-specific card of loop n for product type 1
n
Product-specific card of loop n for product type 2
Queue
Workstation
Control loop

Figure 1. Pull System categories.

Figure 2(a). Limited at the moment of release.

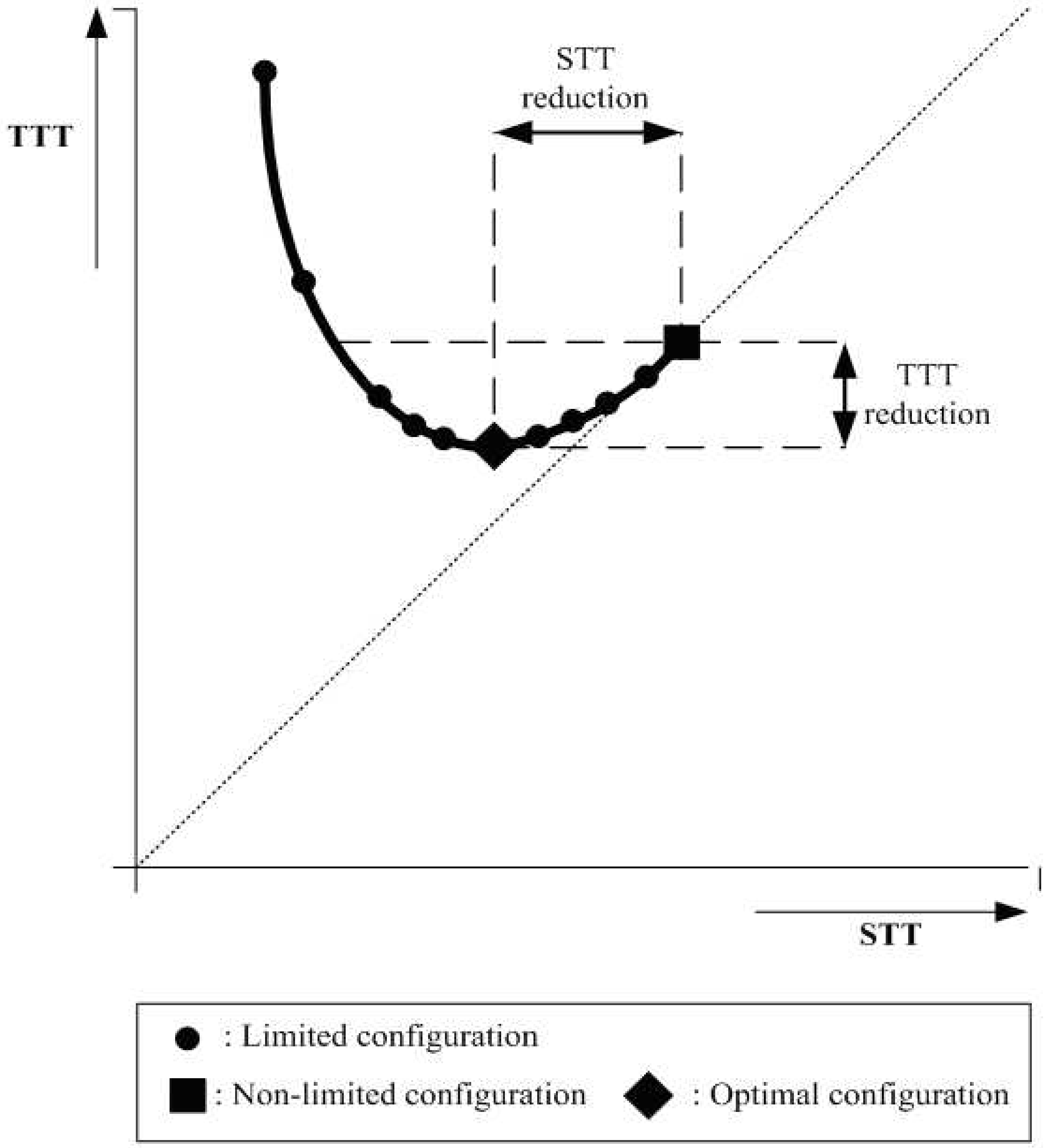

TTT
STT reduction
TTT reduction
STT
● : Limited configuration
■ : Non-limited configuration
◆ : Optimal configuration

Figure 2(b). Limited after the moment of release.

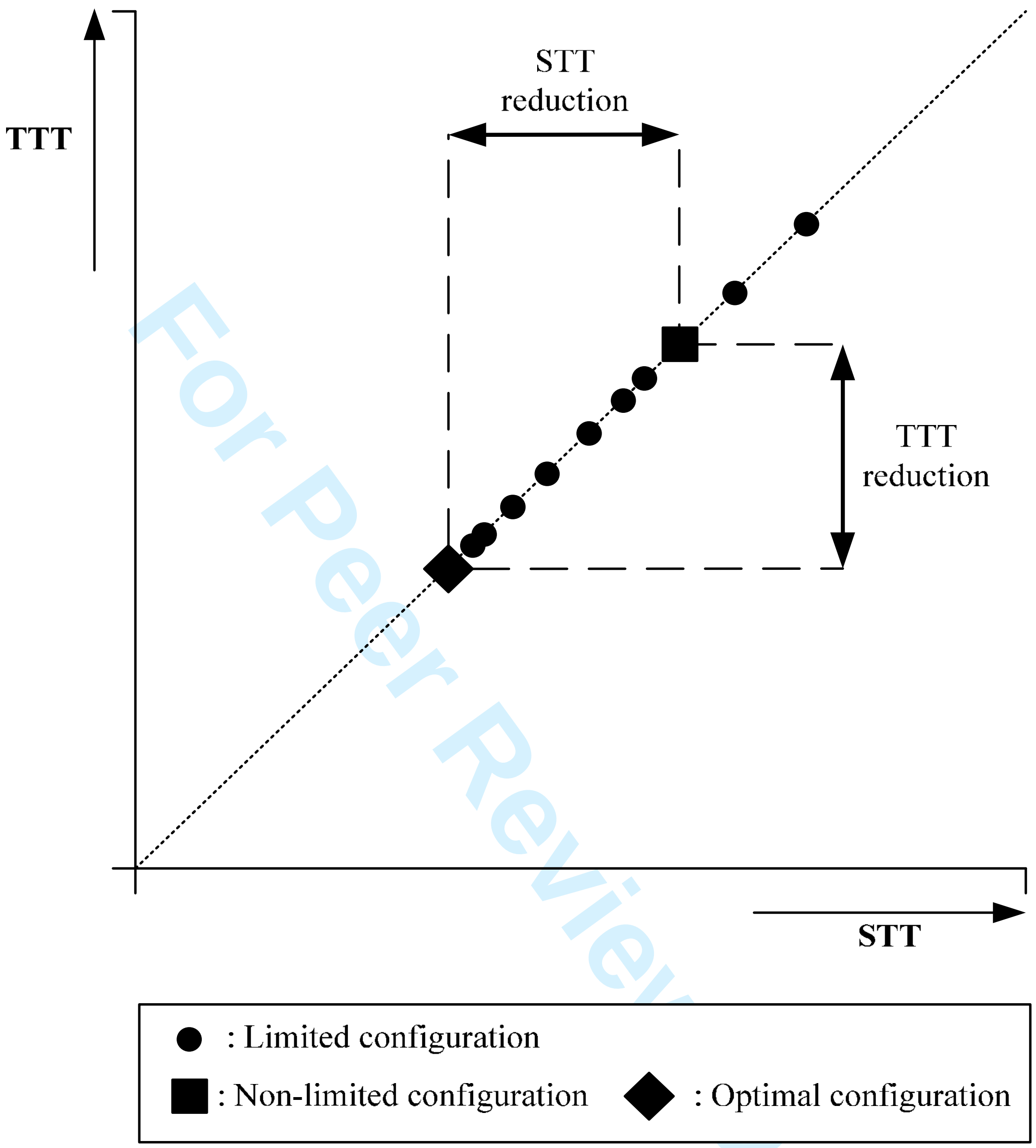


Figure 2. Throughput time performance curves.

Figure 3. Three- and four-stage topology and pull structure

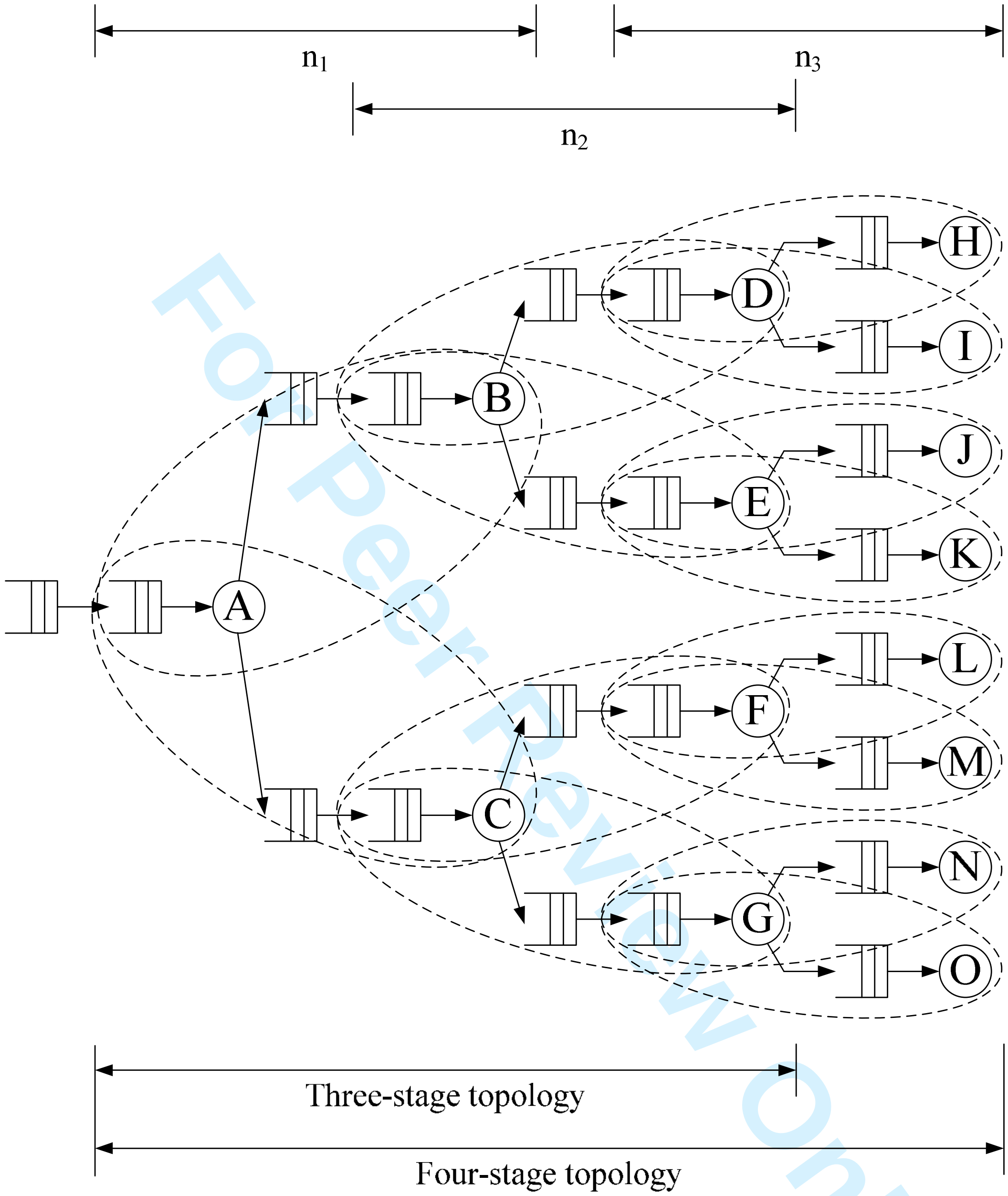


Figure 4(a). Three-stage topology.

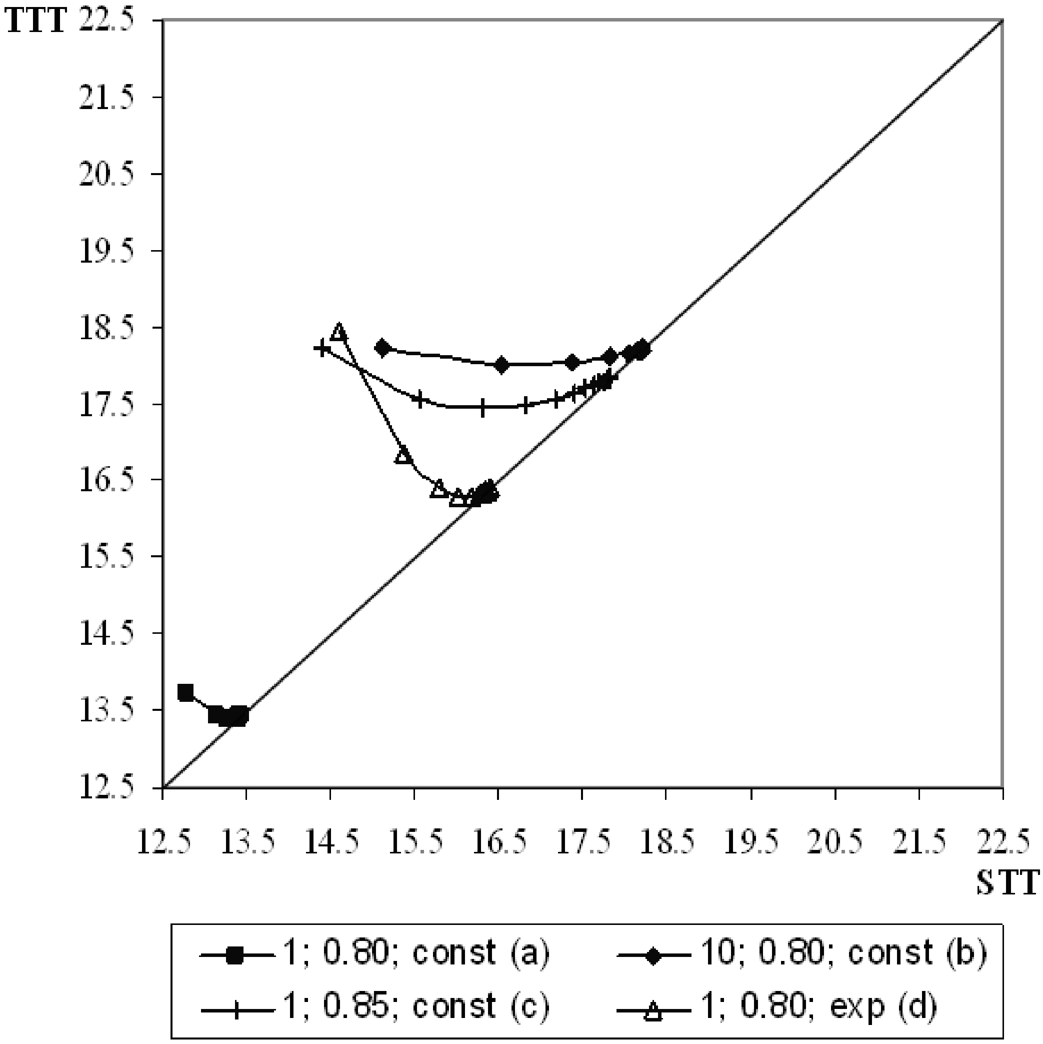


Figure 4(b). Four-stage topology.

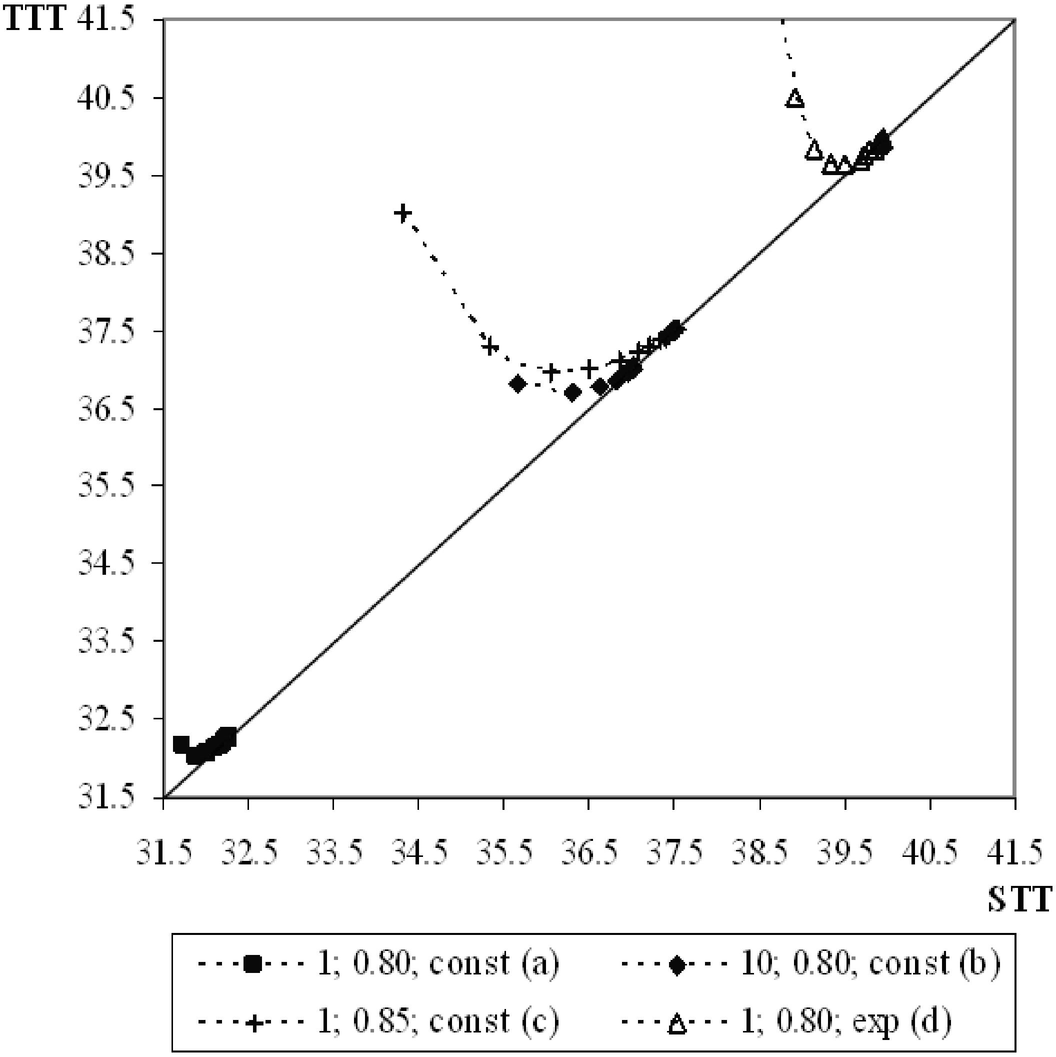


Figure 4. Shop floor and total throughput time for the three- and four-stage topology for series 1. Combination (a) batch size of 1, utilization level of 80%, constant inter-arrival times; (b) batch size of 10, utilization level of 80%, constant inter-arrival times; (c) batch size of 1, utilization level of 85%, constant inter-arrival times; (c) batch size of 1, utilization level of 80%, exponentially distributed inter-arrival times.

Figure 5(a). Three-stage topology.

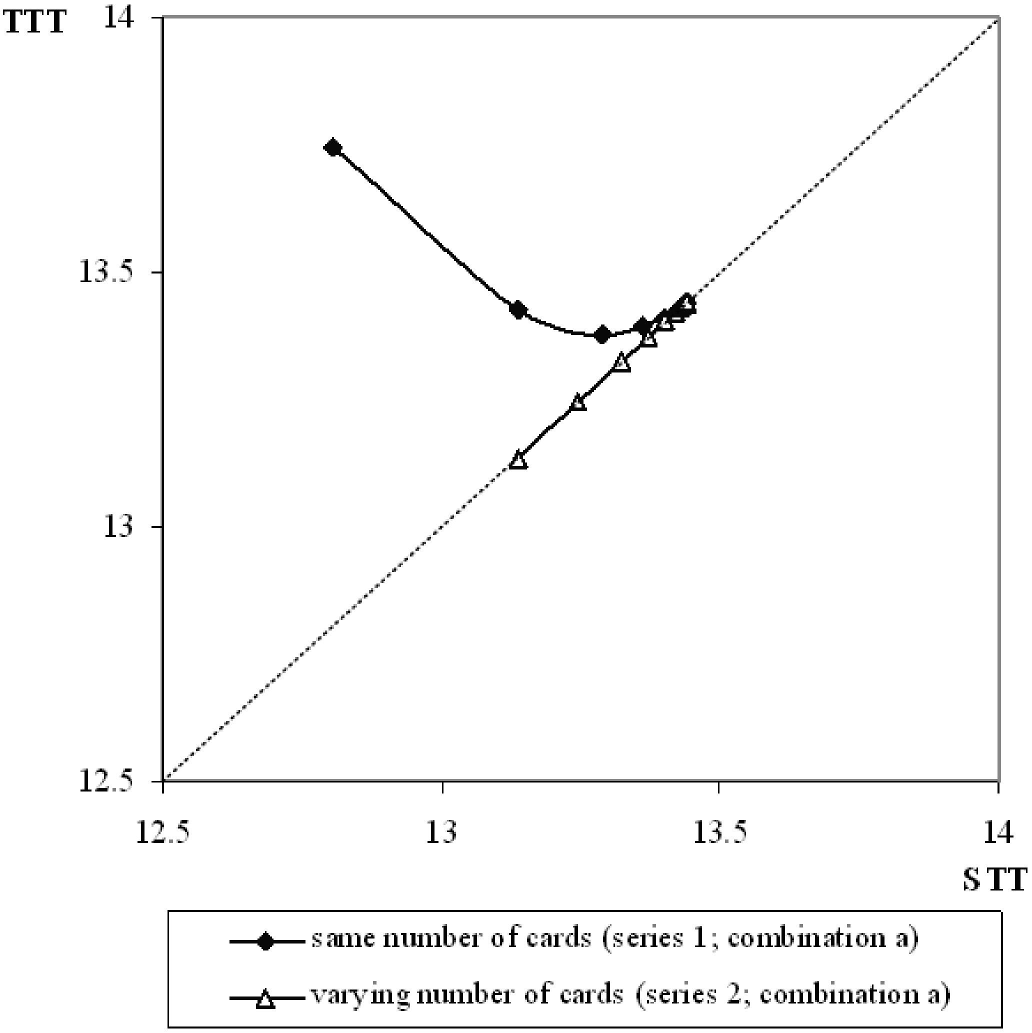


Figure 5(b). Four-stage topology.

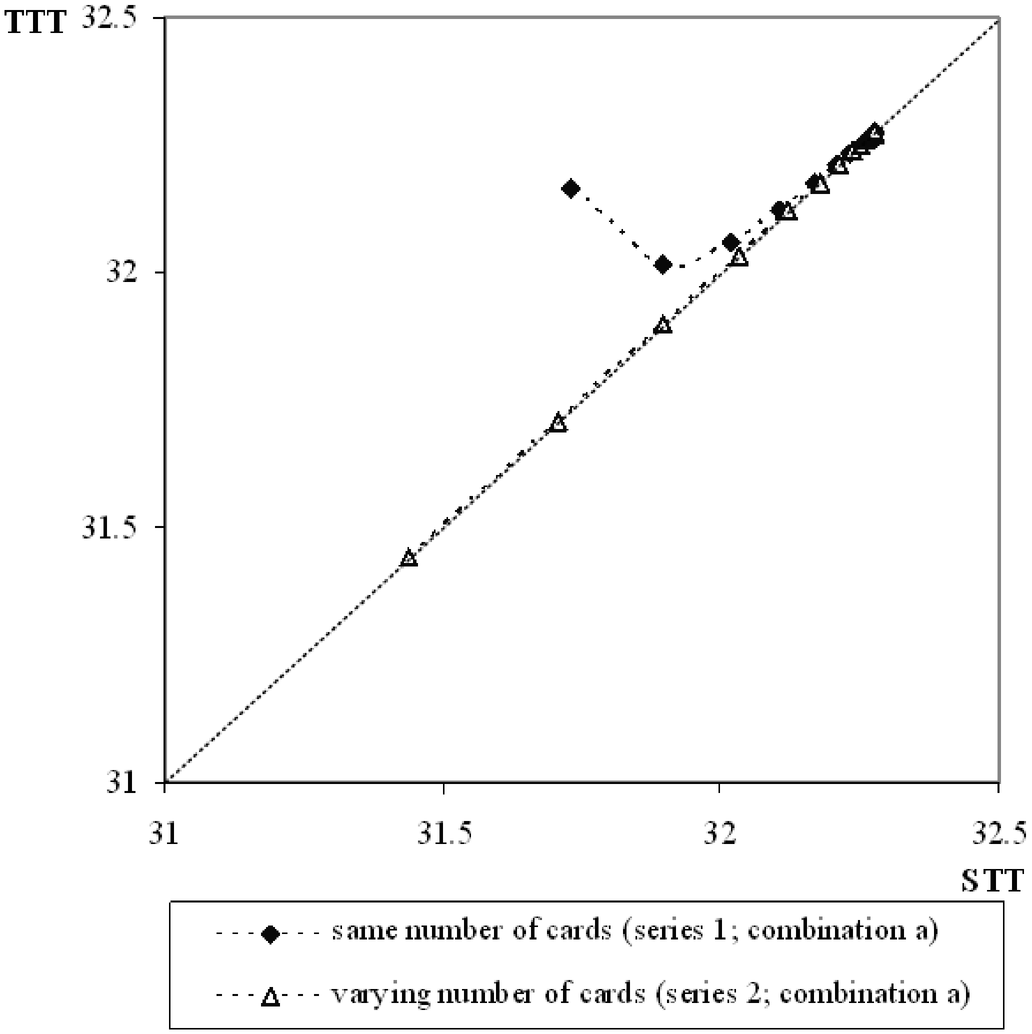


Figure 2. Shop floor throughput time performance for the three- and four-stage topology given the same number of cards (series 1) and a varying number of cards (series 2). Combination (a) batch size of 1, utilization level of 80%, constant inter-arrival times.

Table 1. A card-based pull system categorization.

Catagory	Name	Load-based/ Unit-based	Connected/ Unconnected	References
Product specific	Kanban	Unit-based	unconnected	Sugimori et al. (1977); Ōno (2003)
	Hybrid Kanban/Conwip	Unit-based	connected	Geraghy and Heavey (2005)
	Generalized Kanban	Unit-based	connected	Buzacott (1989)
	Extended Kanban	Unit-based	connected	Dallery and Liberopoulos (2000)
Product-anonymous	Generic Kanban	Unit-based	unconnected	Chang and Yih (1994)
	CONWIP[1]	Unit-based	connected	Spearman, Woodruff, and Hopp (1990)
Route-specific	POLCA	Unit-based	connected	Suri (1998); Riezebos (2010)
	Load-based POLCA	Load-based	connected	Vandaele et al (2008)
	G-POLCA	Load-based	connected	Fernandes and Carmo-Silva (2006)
	m-CONWIP[2]	Unit-based	connected	Spearman, Woodruff, and Hopp (1990); Germs and Riezebos (2010)

[1] CONWIP makes use of a single limit for the entire shop floor

[2] m-CONWIP makes use of a single limit for each routing

Table 2. Experimental design.

Series	Stage	Configuration	Order arrival pattern			Processing times
			Inter-arrival times	Utilization	Batch size	
1 [$n_1=n_2=\ldots=n_m$]	3; 4	1-20; ∞	const; exp	80; 85; 90	1; 10	const; Erlang-2
2 [$n_1,n_2,\ldots,n_m$]	3; 4	1-20; ∞	const; exp	80; 85; 90	1; 10	const; Erlang-2
3 [$n_1,\infty,\ldots,\infty$]	3; 4	1-20; ∞	const; exp	80; 85; 90	1; 10	const; Erlang-2

Table 3. Optimal throughput time performance for a three- and four stage topology[*].

		Constant inter-arrival time						Random inter-arrival time					
		Three-stage			Four-stage			Three-stage			Four-stage		
Batch size	Utilization	conf.	%TTT	%STT	conf.	%TTT	%STT	conf.	%TTT	%STT	conf.	%TTT	%STT
Constant processing time													
1	80%	[6,6]	0.47	1.14	[7,7,7]	0.80	1.18	[6,6]	2.23	8.56	[7,7,7]	1.52	2.66
	85%	[8,8]	0.71	1.43	[9,9,9]	0.86	1.59	[8,8]	2.73	9.13	[10,10,10]	1.74	3.63
	90%	[11,11]	1.09	2.49	[14,14,14]	1.35	1.79	[11,11]	3.31	11.61	[14,14,14]	2.28	4.85
10	80%	[5,5]	1.15	9.17	[7,7,7]	0.89	1.99	[10,10]	5.22	38.83	[11,11,11]	3.44	24.67
	85%	[7,7]	1.05	5.10	[9,9,9]	0.86	1.99	[13,13]	5.47	40.49	[15,15,15]	3.66	25.03
	90%	[10,10]	1.21	4.81	[14,14,14]	1.28	1.87	[18,18]	5.67	43.25	[21,21,21]	3.99	27.15

[*] Given the restriction that the same number of cards is used in each control loop

Table 4. Optimal throughput time performance of a three-and four stage topology[*].

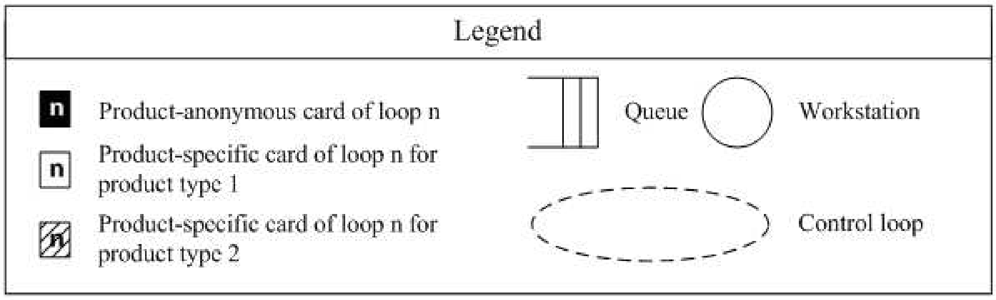
		Constant inter-arrival time						Random inter-arrival time					
		Three-stage			Four-stage			Three-stage			Four-stage		
Batch size	Utilization	conf.	%TTT	%STT	conf.	%TTT	%STT	conf.	%TTT	%STT	conf.	%TTT	%STT
Constant processing time													
1	80%	[∞,3]	3.27	3.27	[∞,∞,2]	5.34	5.34	[∞,3]	6.12	6.12	[2,∞,2]	7.46	15.21
	85%	[∞,3]	5.88	5.88	[∞,∞,2]	7.03	7.03	[∞,3]	7.76	7.76	[2,∞,2]	9.41	20.16
	90%	[∞,3]	8.32	8.32	[∞,∞,2]	9.24	9.24	[∞,3]	9.69	9.69	[3,∞,2]	11.70	24.37
10	80%	[∞,3]	3.36	3.36	[2,∞,2]	5.96	18.94	[2,∞]	8.17	60.61	[2,∞,2]	11.06	45.85
	85%	[∞,3]	4.56	4.56	[3,∞,2]	7.57	15.86	[2,∞]	8.71	63.23	[2,∞,2]	12.24	48.73
	90%	[∞,3]	6.83	6.83	[3,∞,2]	9.80	18.27	[2,∞]	9.17	65.66	[2,∞,2]	13.52	51.76

[*] Given no restriction on the number of cards used

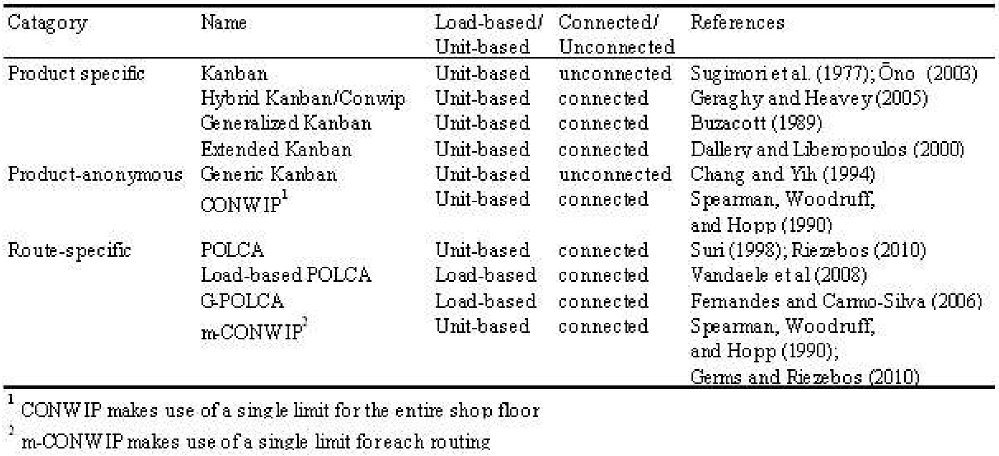
Table 5. Optimal throughput time performance of a three- and four-stage topology[*].

		Constant inter-arrival time						Random inter-arrival time					
Batch size	Utilization	conf.	%TTT	%STT	conf.	%TTT	%STT	conf.	%TTT	%STT	conf.	%TTT	%STT
Three-		[n, ∞]			[∞, n]			[n, ∞]			[∞, n]		
1	80%	[∞,∞]	0.00	0.00	[∞, 2]	3.72	3.72	[2,∞]	3.63	22.70	[∞, 2]	6.12	6.12
	85%	[∞,∞]	0.00	0.00	[∞, 3]	5.88	5.88	[2,∞]	4.53	27.15	[∞, 2]	7.76	7.76
	90%	[∞,∞]	0.00	0.00	[∞, 3]	8.32	8.32	[2,∞]	5.59	32.24	[∞, 2]	9.69	9.69
10	80%	[2,∞]	2.66	29.05	[∞, 2]	3.36	3.36	[2,∞]	8.17	60.61	[∞, 2]	6.31	6.31
	85%	[3,∞]	2.69	20.29	[∞, 2]	4.56	4.56	[2,∞]	8.71	63.23	[∞, 2]	6.84	6.84
	90%	[3,∞]	2.84	21.65	[∞, 3]	6.83	6.83	[2,∞]	9.17	65.66	[∞, 3]	7.39	7.39
Four-stage		[n, ∞, ∞]			[n, n, ∞]			[n, ∞, ∞]			[n, n, ∞]		
1	80%	[∞,∞,∞]	0.00	0.00	[∞, 2,∞]	1.49	1.49	[2,∞,∞]	1.68	10.76	[∞, 2,∞]	2.69	2.69
	85%	[∞,∞,∞]	0.00	0.00	[∞, 2,∞]	2.18	2.18	[2,∞,∞]	2.13	12.87	[∞, 2,∞]	3.43	3.43
	90%	[∞,∞,∞]	0.00	0.00	[∞, 2,∞]	3.26	3.26	[2,∞,∞]	2.61	15.28	[∞, 2,∞]	4.28	4.28
10	80%	[2,∞,∞]	1.40	14.38	[∞, 2,∞]	1.62	1.62	[2,∞,∞]	4.90	39.69	[2,∞,∞]	4.90	39.69
	85%	[3,∞,∞]	1.30	9.60	[∞, 2,∞]	2.08	2.08	[2,∞,∞]	5.33	41.82	[2,∞,∞]	5.33	41.82
	90%	[3,∞,∞]	1.28	9.75	[∞, 2,∞]	2.96	2.96	[2,∞,∞]	5.78	44.02	[2,∞,∞]	5.78	44.02
		[∞, ∞, n]			[∞, n, n]			[∞, ∞, n]			[∞, n, n]		
1	80%	[∞,∞, 2]	5.34	5.34	[∞,∞, 2]	5.34	5.34	[∞,∞, 2]	5.75	5.75	[∞,∞, 2]	5.75	5.75
	85%	[∞,∞, 2]	7.03	7.03	[∞,∞, 2]	7.03	7.03	[∞,∞, 2]	7.25	7.25	[∞,∞, 2]	7.25	7.25
	90%	[∞,∞, 2]	9.24	9.24	[∞,∞, 2]	9.24	9.24	[∞,∞, 2]	9.04	9.04	[∞,∞, 2]	9.04	9.04
10	80%	[∞,∞, 2]	4.72	4.72	[∞,∞, 2]	4.72	4.72	[∞,∞, 2]	5.75	5.75	[∞,∞, 2]	5.75	5.75
	85%	[∞,∞, 2]	6.29	6.29	[∞,∞, 2]	6.29	6.29	[∞,∞, 2]	6.47	6.47	[∞,∞, 2]	6.47	6.47
	90%	[∞,∞, 2]	8.50	8.50	[∞,∞, 2]	8.50	8.50	[∞,∞, 2]	7.29	7.29	[∞,∞, 3]	7.33	7.33
		[∞, n, ∞]			[n, ∞, n]			[∞, n, ∞]			[n, ∞, n]		
1	80%	[∞, 2,∞]	1.49	1.49	[∞,∞,2]	5.34	5.34	[∞, 2,∞]	2.69	2.69	[2,∞, 2]	7.46	15.21
	85%	[∞, 2,∞]	2.18	2.18	[∞,∞,2]	7.03	7.03	[∞, 2,∞]	3.43	3.43	[2,∞, 2]	9.41	20.16
	90%	[∞, 2,∞]	3.26	3.26	[∞,∞,2]	9.24	9.24	[∞, 2,∞]	4.28	4.28	[3,∞, 2]	11.72	22.01
10	80%	[∞, 2,∞]	1.62	1.62	[2,∞,2]	5.96	18.94	[∞, 2,∞]	3.58	3.58	[2,∞, 2]	11.06	45.85
	85%	[∞, 2,∞]	2.08	2.08	[3,∞,2]	7.57	15.86	[∞, 2,∞]	3.95	3.95	[2,∞, 2]	12.24	48.73
	90%	[∞, 2,∞]	2.96	2.96	[3,∞,2]	9.80	18.27	[∞, 2,∞]	4.27	4.27	[2,∞, 2]	13.52	51.76

[*] Given the restiction that one or more control loops are limited


Legend
n Product-anonymous card of loop n
n Product-specific card of loop n for product type 1
n Product-specific card of loop n for product type 2
Queue
Workstation
Control loop

211x63mm (72 x 72 DPI)

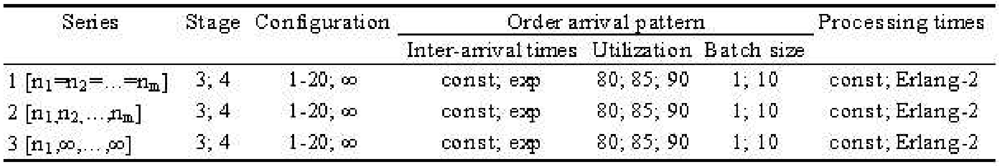


Catagory	Name	Load-based/ Unit-based	Connected/ Unconnected	References
Product specific	Kanban	Unit-based	unconnected	Sugimori et al. (1977); Ōno (2003)
	Hybrid Kanban/Conwip	Unit-based	connected	Geraghy and Heavey (2005)
	Generalized Kanban	Unit-based	connected	Buzacott (1989)
	Extended Kanban	Unit-based	connected	Dallery and Liberopoulos (2000)
Product-anonymous	Generic Kanban	Unit-based	unconnected	Chang and Yih (1994)
	CONWIP[1]	Unit-based	connected	Spearman, Woodruff, and Hopp (1990)
Route-specific	POLCA	Unit-based	connected	Suri (1998); Riezebos (2010)
	Load-based POLCA	Load-based	connected	Vandaele et al (2008)
	G-POLCA	Load-based	connected	Fernandes and Carmo-Silva (2006)
	m-CONWIP[2]	Unit-based	connected	Spearman, Woodruff, and Hopp (1990); Germs and Riezebos (2010)

[1] CONWIP makes use of a single limit for the entire shop floor

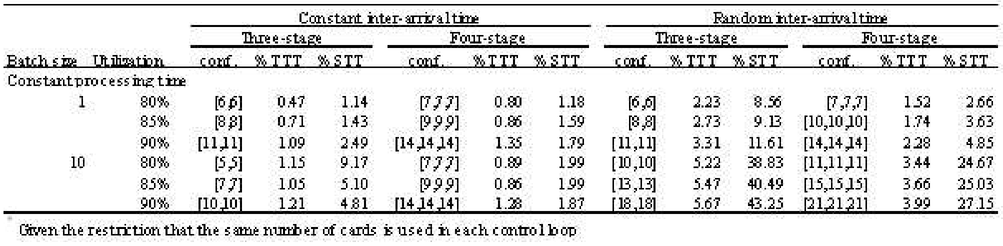
[2] m-CONWIP makes use of a single limit foreach routing

213x101mm (72 x 72 DPI)



Series	Stage	Configuration	Order arrival pattern			Processing times
			Inter-arrival times	Utilization	Batch size	
1 [n_1=n_2=…=n_m]	3; 4	1-20; ∞	const; exp	80; 85; 90	1; 10	const; Erlang-2
2 [n_1,n_2,…,n_m]	3; 4	1-20; ∞	const; exp	80; 85; 90	1; 10	const; Erlang-2
3 [n_1,∞,…,∞]	3; 4	1-20; ∞	const; exp	80; 85; 90	1; 10	const; Erlang-2

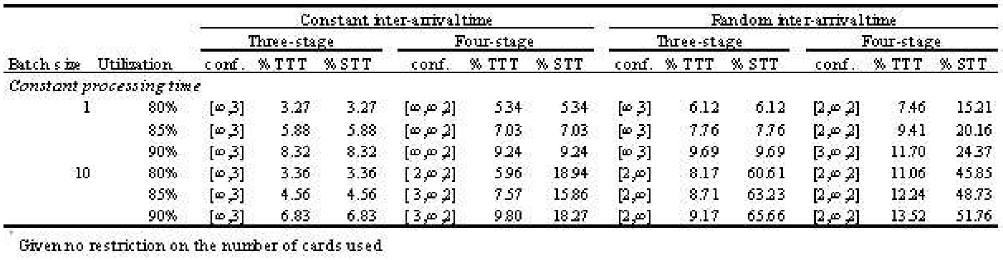
213x47mm (72 x 72 DPI)



		Constant inter-arrival time						Random inter-arrival time					
		Three-stage			Four-stage			Three-stage			Four-stage		
Batch size	Utilization	conf.	% TTT	% STT	conf.	% TTT	% STT	conf.	% TTT	% STT	conf.	% TTT	% STT
Constant processing time													
1	80%	[6,6]	0.47	1.14	[7,7,7]	0.80	1.18	[6,6]	2.23	8.56	[7,7,7]	1.52	2.66
	85%	[8,8]	0.71	1.43	[9,9,9]	0.86	1.59	[8,8]	2.73	9.13	[10,10,10]	1.74	3.63
	90%	[11,11]	1.09	2.49	[14,14,14]	1.35	1.79	[11,11]	3.31	11.61	[14,14,14]	2.28	4.85
10	80%	[5,5]	1.15	9.17	[7,7,7]	0.89	1.99	[10,10]	5.22	38.83	[11,11,11]	3.44	24.67
	85%	[7,7]	1.05	5.10	[9,9,9]	0.86	1.99	[13,13]	5.47	40.49	[15,15,15]	3.66	25.03
	90%	[10,10]	1.21	4.81	[14,14,14]	1.28	1.87	[18,18]	5.67	43.25	[21,21,21]	3.99	27.15

* Given the restriction that the same number of cards is used in each control loop

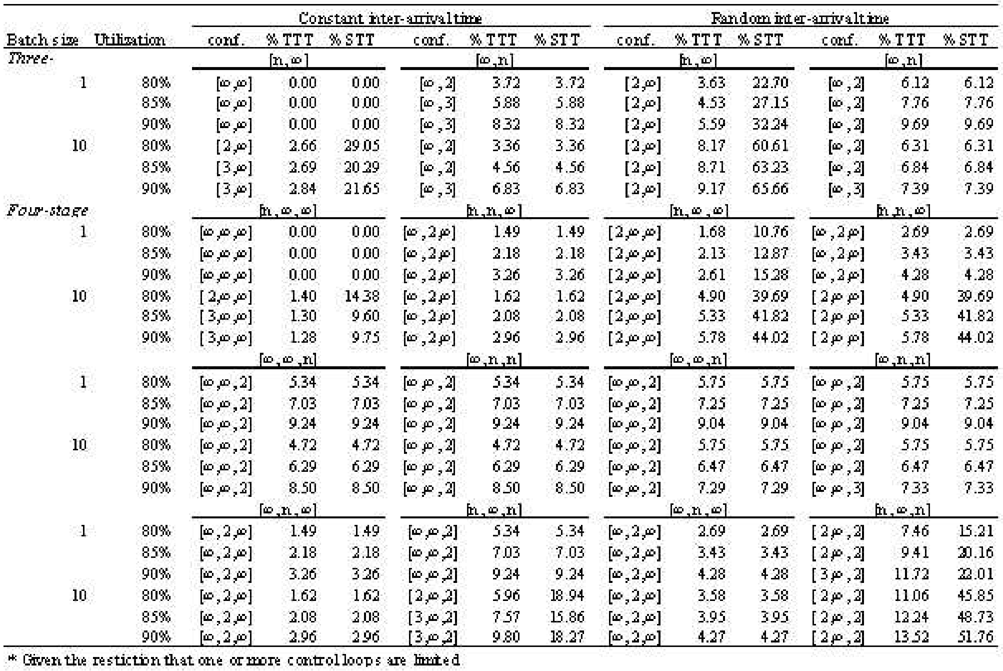
213x55mm (72 x 72 DPI)



		Constant inter-arrival time						Random inter-arrival time					
		Three-stage			Four-stage			Three-stage			Four-stage		
Batch size	Utilization	conf.	% TTT	% STT	conf.	% TTT	% STT	conf.	% TTT	% STT	conf.	% TTT	% STT
Constant processing time													
1	80%	[∞,3]	3.27	3.27	[∞,∞,2]	5.34	5.34	[∞,3]	6.12	6.12	[2,∞,2]	7.46	15.21
	85%	[∞,3]	5.88	5.88	[∞,∞,2]	7.03	7.03	[∞,3]	7.76	7.76	[2,∞,2]	9.41	20.16
	90%	[∞,3]	8.32	8.32	[∞,∞,2]	9.24	9.24	[∞,3]	9.69	9.69	[3,∞,2]	11.70	24.37
10	80%	[∞,3]	3.36	3.36	[2,∞,2]	5.96	18.94	[2,∞]	8.17	60.61	[2,∞,2]	11.06	45.85
	85%	[∞,3]	4.56	4.56	[3,∞,2]	7.57	15.86	[2,∞]	8.71	63.23	[2,∞,2]	12.24	48.73
	90%	[∞,3]	6.83	6.83	[3,∞,2]	9.80	18.27	[2,∞]	9.17	65.66	[2,∞,2]	13.52	51.76

* Given no restriction on the number of cards used

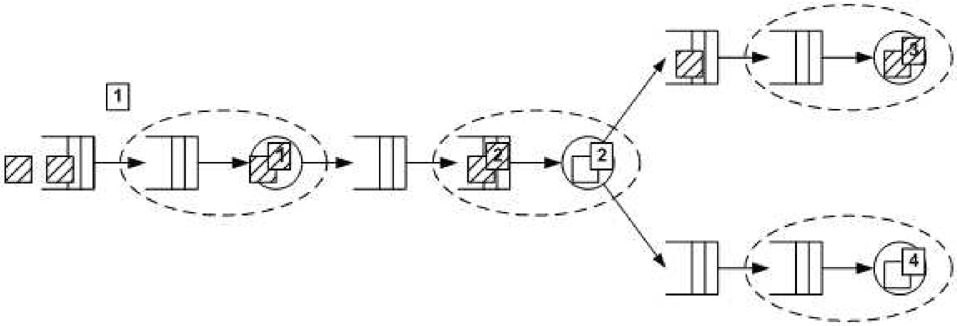
213x59mm (72 x 72 DPI)



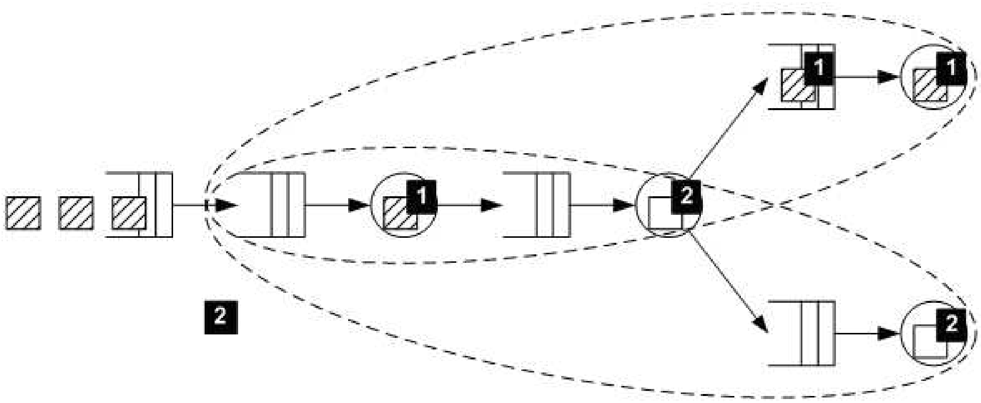
		Constant inter-arrival time						Random inter-arrival time					
Batch size	Utilization	conf.	% TTT	% STT	conf.	% TTT	% STT	conf.	% TTT	% STT	conf.	% TTT	% STT
Three-		[n,∞]			[∞,n]			[n,∞]			[∞,n]		
1	80%	[∞,∞]	0.00	0.00	[∞,2]	3.72	3.72	[2,∞]	3.63	22.70	[∞,2]	6.12	6.12
	85%	[∞,∞]	0.00	0.00	[∞,3]	5.88	5.88	[2,∞]	4.53	27.15	[∞,2]	7.76	7.76
	90%	[∞,∞]	0.00	0.00	[∞,3]	8.32	8.32	[2,∞]	5.59	32.24	[∞,2]	9.69	9.69
10	80%	[2,∞]	2.66	29.05	[∞,2]	3.36	3.36	[2,∞]	8.17	60.61	[∞,2]	6.31	6.31
	85%	[3,∞]	2.69	20.29	[∞,2]	4.56	4.56	[2,∞]	8.71	63.23	[∞,2]	6.84	6.84
	90%	[3,∞]	2.84	21.65	[∞,3]	6.83	6.83	[2,∞]	9.17	65.66	[∞,3]	7.39	7.39
Four-stage		[n,∞,∞]			[n,n,∞]			[n,∞,∞]			[n,n,∞]		
1	80%	[∞,∞,∞]	0.00	0.00	[∞,2,∞]	1.49	1.49	[2,∞,∞]	1.68	10.76	[∞,2,∞]	2.69	2.69
	85%	[∞,∞,∞]	0.00	0.00	[∞,2,∞]	2.18	2.18	[2,∞,∞]	2.13	12.87	[∞,2,∞]	3.43	3.43
	90%	[∞,∞,∞]	0.00	0.00	[∞,2,∞]	3.26	3.26	[2,∞,∞]	2.61	15.28	[∞,2,∞]	4.28	4.28
10	80%	[2,∞,∞]	1.40	14.38	[∞,2,∞]	1.62	1.62	[2,∞,∞]	4.90	39.69	[2,∞,∞]	4.90	39.69
	85%	[3,∞,∞]	1.30	9.60	[∞,2,∞]	2.08	2.08	[2,∞,∞]	5.33	41.82	[2,∞,∞]	5.33	41.82
	90%	[3,∞,∞]	1.28	9.75	[∞,2,∞]	2.96	2.96	[2,∞,∞]	5.78	44.02	[2,∞,∞]	5.78	44.02
		[∞,∞,n]			[∞,n,n]			[∞,∞,n]			[∞,n,n]		
1	80%	[∞,∞,2]	5.34	5.34	[∞,∞,2]	5.34	5.34	[∞,∞,2]	5.75	5.75	[∞,∞,2]	5.75	5.75
	85%	[∞,∞,2]	7.03	7.03	[∞,∞,2]	7.03	7.03	[∞,∞,2]	7.25	7.25	[∞,∞,2]	7.25	7.25
	90%	[∞,∞,2]	9.24	9.24	[∞,∞,2]	9.24	9.24	[∞,∞,2]	9.04	9.04	[∞,∞,2]	9.04	9.04
10	80%	[∞,∞,2]	4.72	4.72	[∞,∞,2]	4.72	4.72	[∞,∞,2]	5.75	5.75	[∞,∞,2]	5.75	5.75
	85%	[∞,∞,2]	6.29	6.29	[∞,∞,2]	6.29	6.29	[∞,∞,2]	6.47	6.47	[∞,∞,2]	6.47	6.47
	90%	[∞,∞,2]	8.50	8.50	[∞,∞,2]	8.50	8.50	[∞,∞,2]	7.29	7.29	[∞,∞,3]	7.33	7.33
		[∞,n,∞]			[n,∞,n]			[∞,n,∞]			[n,∞,n]		
1	80%	[∞,2,∞]	1.49	1.49	[∞,∞,2]	5.34	5.34	[∞,2,∞]	2.69	2.69	[2,∞,2]	7.46	15.21
	85%	[∞,2,∞]	2.18	2.18	[∞,∞,2]	7.03	7.03	[∞,2,∞]	3.43	3.43	[2,∞,2]	9.41	20.16
	90%	[∞,2,∞]	3.26	3.26	[∞,∞,2]	9.24	9.24	[∞,2,∞]	4.28	4.28	[3,∞,2]	11.72	22.01
10	80%	[∞,2,∞]	1.62	1.62	[2,∞,2]	5.96	18.94	[∞,2,∞]	3.58	3.58	[2,∞,2]	11.06	45.85
	85%	[∞,2,∞]	2.08	2.08	[3,∞,2]	7.57	15.86	[∞,2,∞]	3.95	3.95	[2,∞,2]	12.24	48.73
	90%	[∞,2,∞]	2.96	2.96	[3,∞,2]	9.80	18.27	[∞,2,∞]	4.27	4.27	[2,∞,2]	13.52	51.76

* Given the restiction that one or more control loops are limited

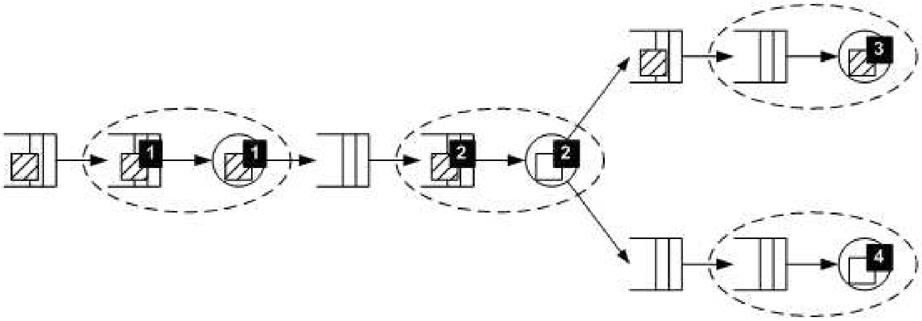
213x147mm (72 x 72 DPI)



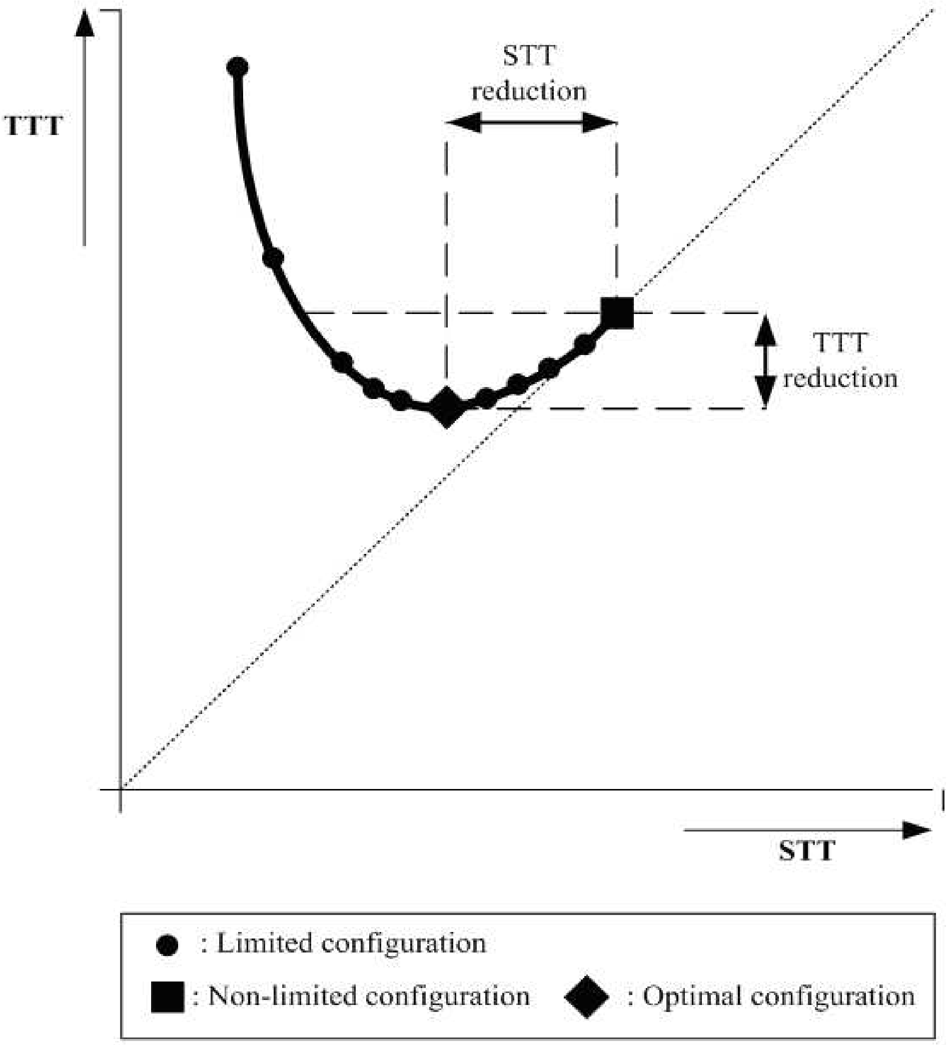
212x69mm (72 x 72 DPI)



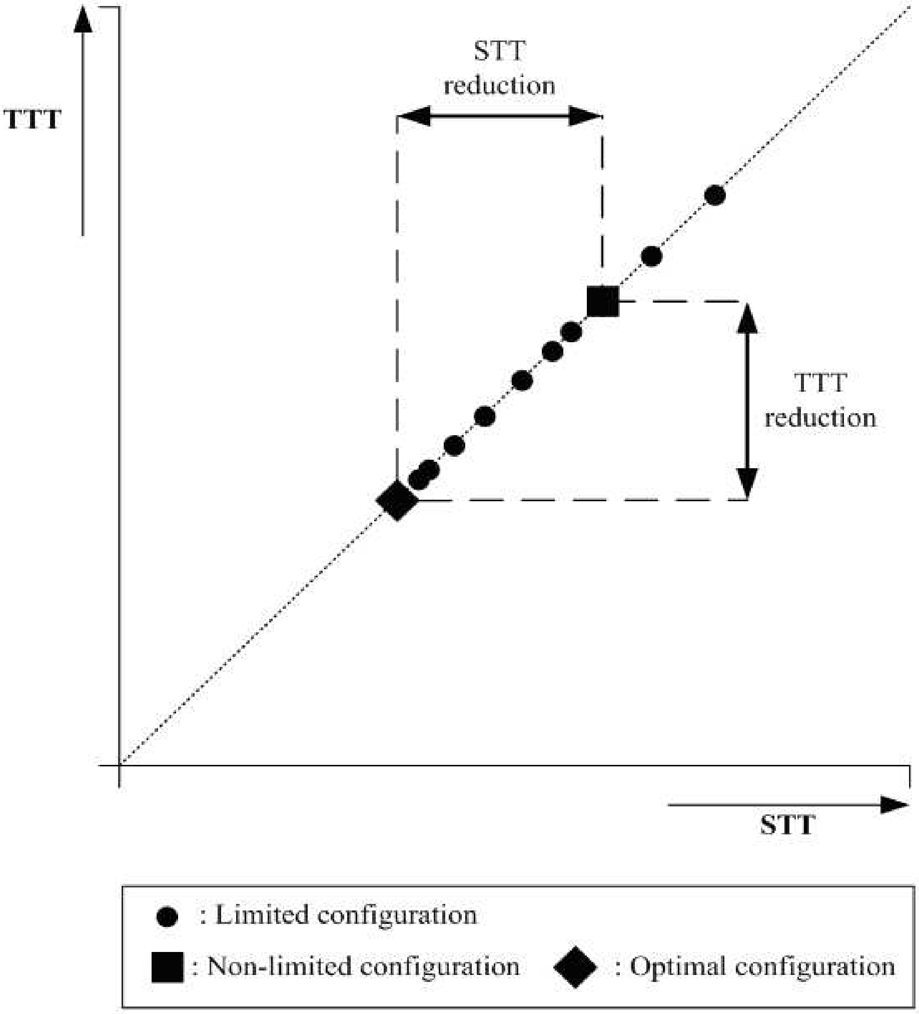
212x103mm (72 x 72 DPI)



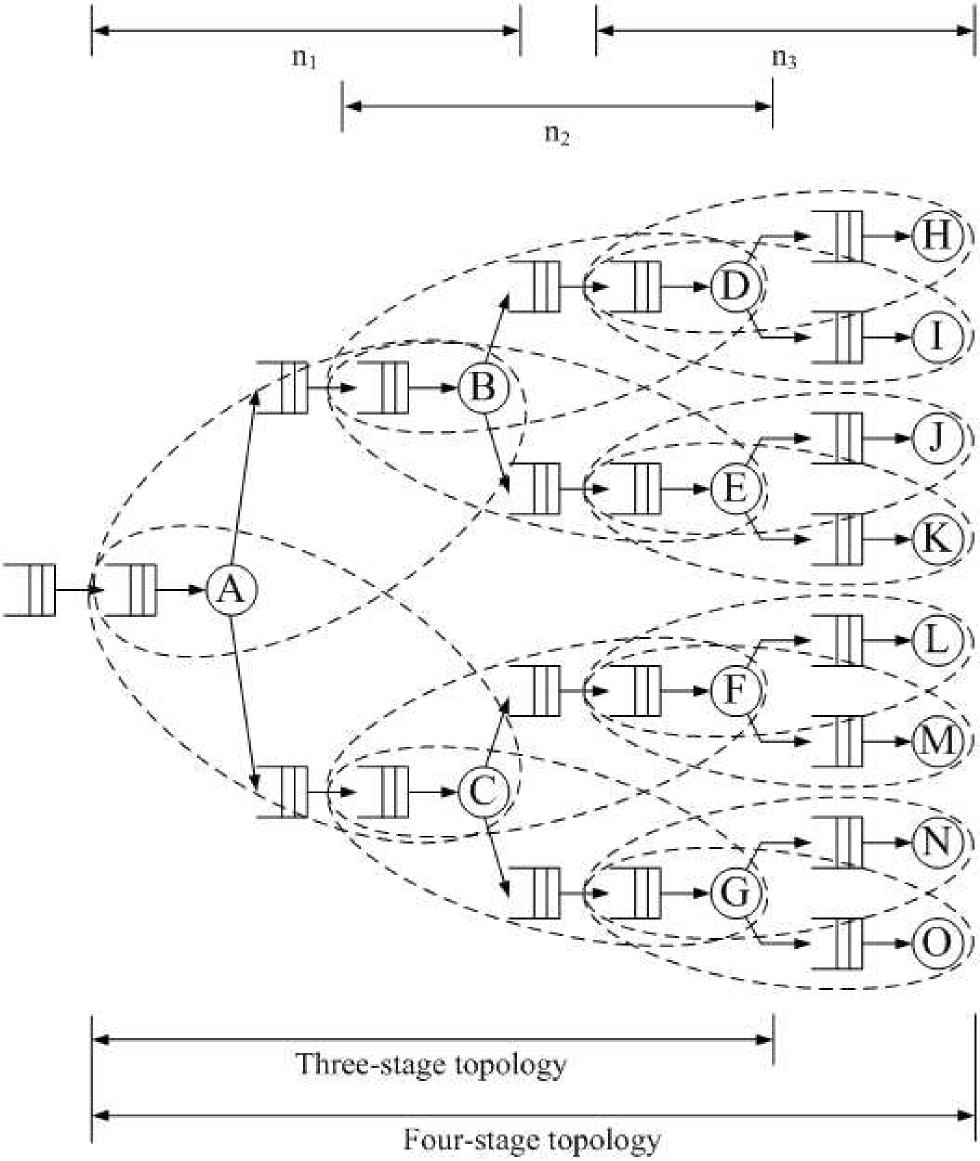
211x68mm (72 x 72 DPI)



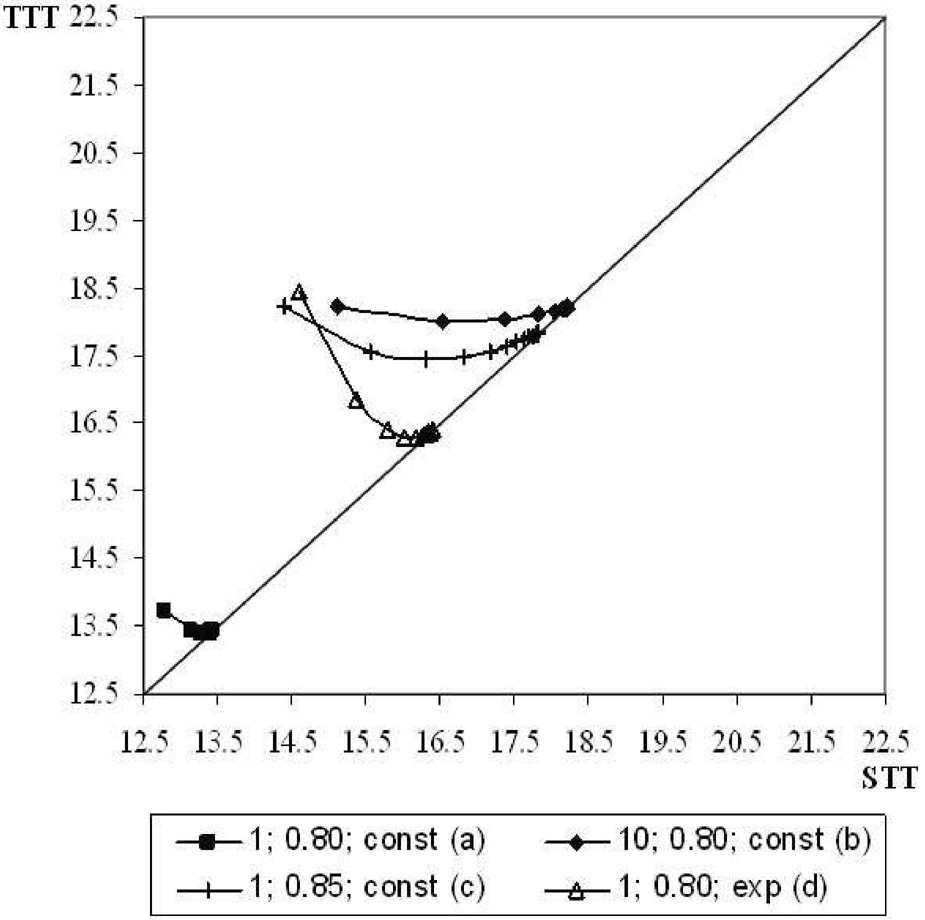
207x222mm (72 x 72 DPI)



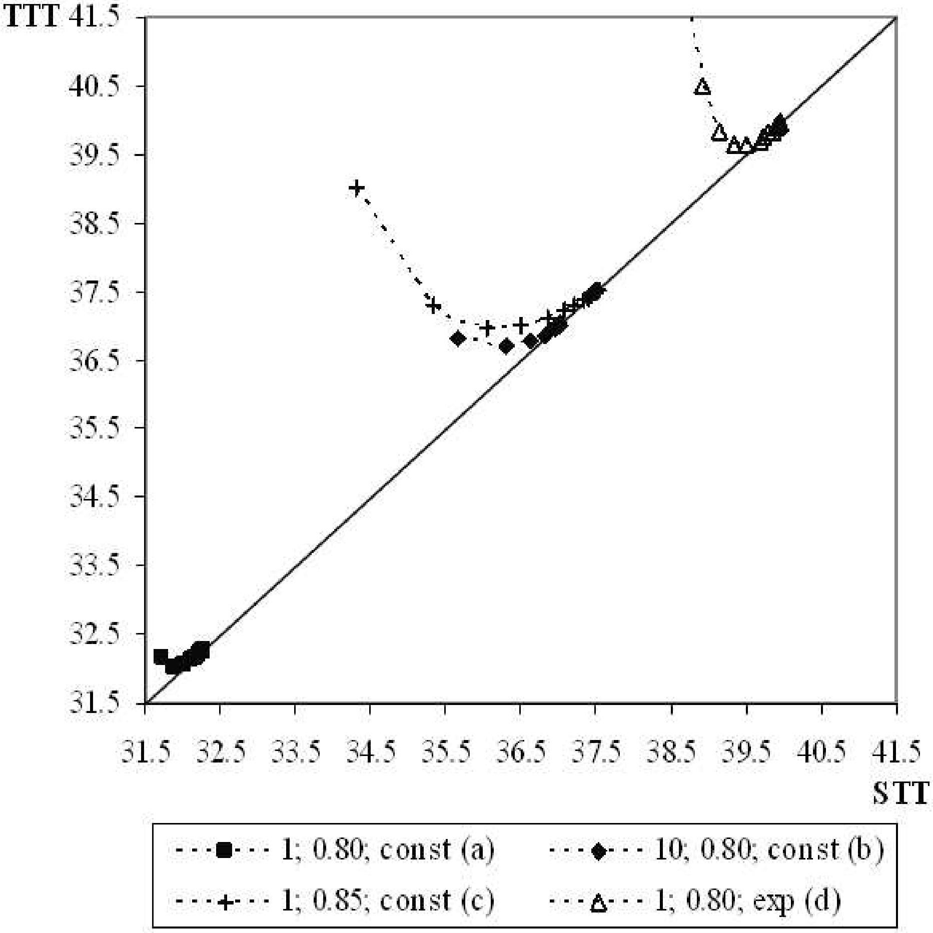
212x222mm (72 x 72 DPI)



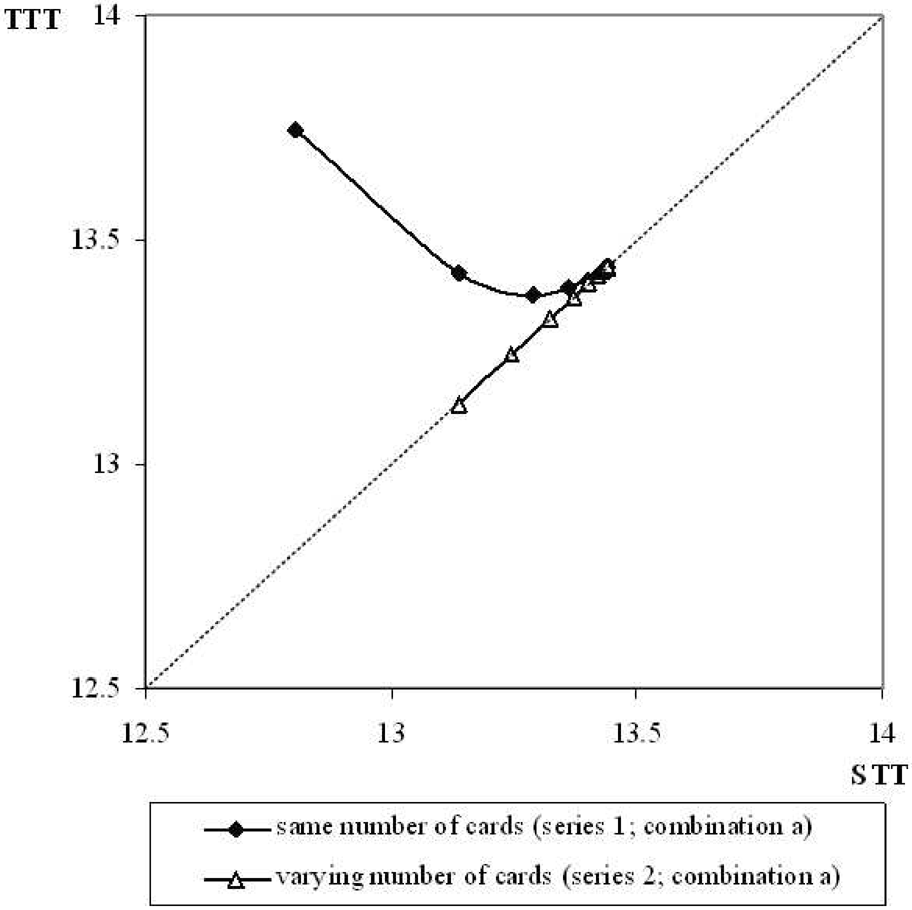
189x222mm (72 x 72 DPI)



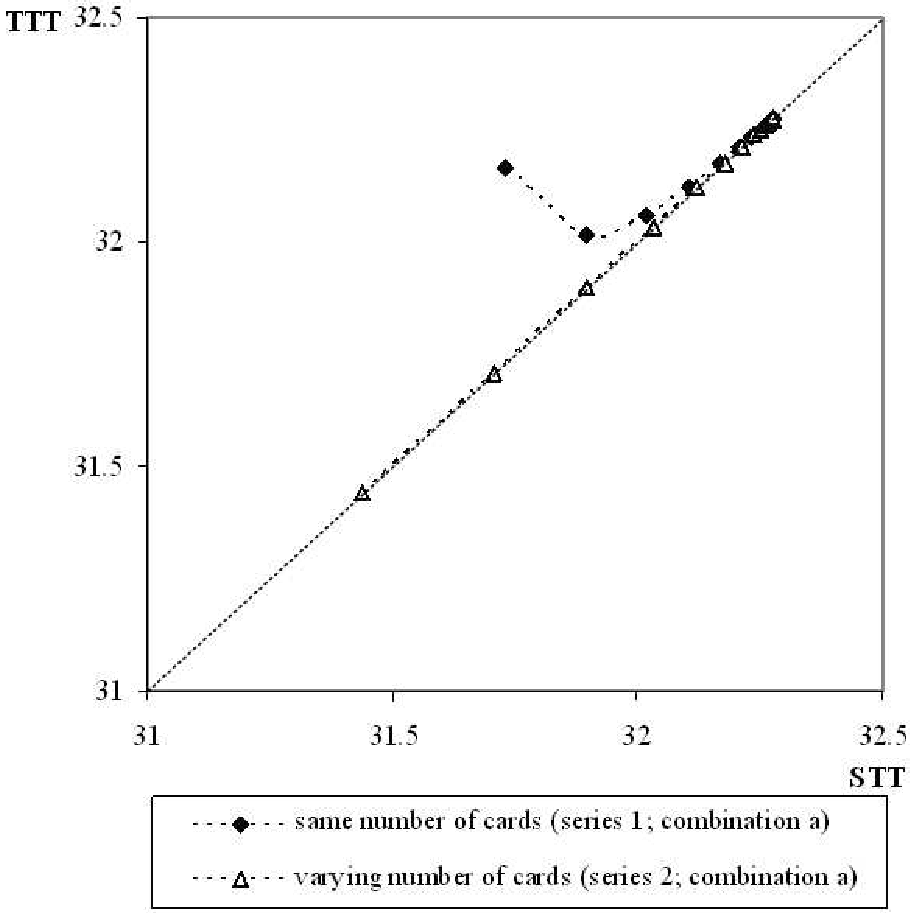
224x222mm (72 x 72 DPI)



221x222mm (72 x 72 DPI)



226x221mm (72 x 72 DPI)



226x222mm (72 x 72 DPI)