An approximate analytical method for evaluating the performance of closed loop flow systems with unreliable machines and finite buffers — Part I: Small Loops

N. Maggio[*], A. Matta[†], S. B. Gershwin[‡], T. Tolio[§]

February 14, 2003

Abstract

This paper describes an approximate analytical method for evaluating the average values of throughput and buffer levels of closed production systems with finite buffers. The method includes a new set of decomposition equations and a new building block model. We apply it to a three-machine, three-buffer loop in which machines can fail in more than one mode. The machines have deterministic processing times and geometrically distributed probabilities of failure and repair. The numerical results of the method are close to those from a simulation. The method performs well because it takes into account the correlation among the numbers of parts in the buffers.

1 Introduction

1.1 Closed Loop Systems

A *LOOP* is a material flow system that consists of work centers or *machines* separated by storage areas (*buffers*) in which material travels from machine to buffer to machine in a fixed sequence and returns to the first machine. An example is a manufacturing system, illustrated in Figure 1, in which raw parts enter the system from outside and are loaded onto pallets or fixtures at a loading station (machine M_1). The pallets and the associated parts then visit buffer B_1, machine M_2, ..., M_{K-1}, B_{K-1}.

[*]Politecnico di Milano; currently with Accenture, maggionicola@libero.it.

[†]Politecnico di Milano, Dipartimento di Meccanica, Sezione Tecnologie Meccaniche e Produzione, matta@tmhp1.mecc.polimi.it.

[‡]Massachusetts Institute of Technology, Department of Mechanical Engineering, Laboratory for Manufacturing and Productivity, gershwin@mit.edu.

[§]Politecnico di Milano, Dipartimento di Meccanica, Sezione Tecnologie Meccaniche e Produzione, tolio@tmhp1.mecc.polimi.it.

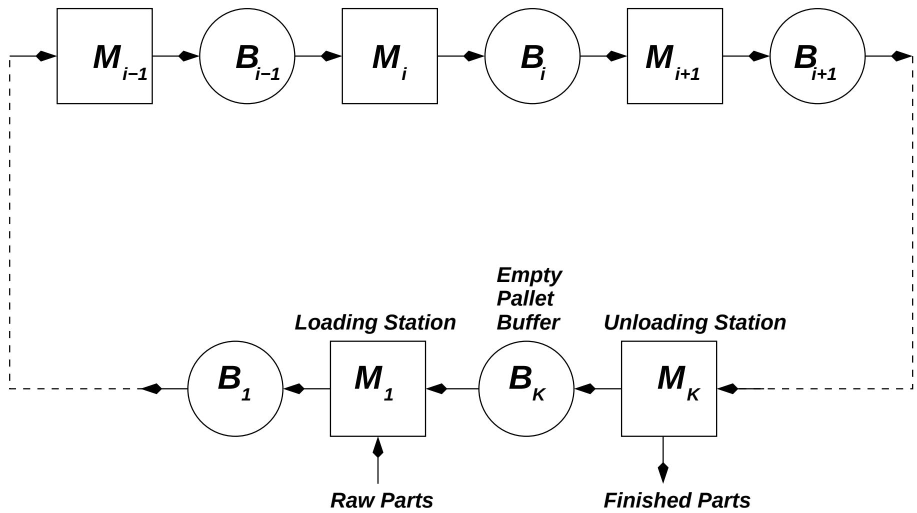


Figure 1: Illustration of a closed production line

Once all the operations have been performed, the part-pallet assembly goes to the unloading station (M_K) where the part is unloaded from its pallet. The finished part leaves the system, while the empty pallet goes to the empty pallet buffer (B_K) to wait for a new raw part.

The total number of pallets in the system — the *population* — is constant since pallets are not added or removed from the production line. For the pallets, the production line is closed. The production rate of parts is the same as the rate at which pallets travel through each workstation, and the distribution of parts in the system is the same as the distribution of pallets, except for the empty pallet buffer.

Examples of closed loop production lines with pallets or fixtures can be observed in automotive fabrication, electronic component assembly, food packaging, and consumer manufacturing industries. Such loops are common where work pieces are loaded onto a support in order to ensure accuracy and stability during operations.

In addition, loops occur in production systems controlled by CONWIP (constant work-in-process — Hopp and Roof 1998, Hopp and Spearman 1996, Spearman, Woodruff, and Hopp 1990); base-stock; PAC (production authorization cards — Buzacott and Shantikumar 1992, Buzacott and Shantikumar 1993); the control point policy (CPP) (Gershwin 2000); and other policies. There is a single loop like that of Figure 1 in a line controlled by CONWIP; in other cases, there are multiple loops. In systems controlled with such policies *tokens* or *production authorization cards* behave similarly to the pallets as described above, and the number of tokens (or another quantity) is constant either within the whole system or within a specific portion of the system.

1.2 Related Literature

Although several analytical methods have been developed for the analysis of open production lines (Dallery and Gershwin 1992), little work has been done on closed lines with unreliable machines and finite buffers. See the references in Tolio and Gershwin (1998). Frein, Commault, and Dallery (1996) proposed an analytical method for the performance evaluation of closed lines that is an extension of the decomposition method developed for open lines. However, this method does not fully capture the correlation that exists among the numbers of parts in each buffer of the system. The effects of this correlation increase when the number of machines decreases. As a result, the method is more accurate for larger systems than for smaller ones. By contrast, the method described here is better suited to small systems because its complexity increases as the system grows.

1.3 Contribution

This paper is a summary of the major results of Maggio (2000) and Matta and Tolio (2003). It presents an approximate analytical method for predicting the average throughput and average work in process in each buffer of a loop system. New decomposition equations are proposed for analyzing the behavior of a generalization of the Buzacott model, ie a discrete time, discrete state system which is synchronous, with unreliable machines and finite buffers. These new equations take into account the relationship between the maximum numbers of parts in the buffers and the propagation of interruptions of flow in the system. This relationship is due to the constant population. To do this we extend the multi-failure-mode approach of Tolio and Matta (1998) and Tolio, Gershwin, and Matta (2002). While the algorithm in this paper is only suited for three-machine systems, we describe approaches for extending it to systems of much greater size and complexity.

1.4 Outline

The class of systems that we model and analyze, and the important features of their behavior that require a new approach, are described in detail in Section 2. The decomposition technique is presented in Section 3. Unfortunately, the method grows in complexity as the size of the system increases, so we specialize it to three-machine, three-buffer systems in Section 4. Section 5 contains numerical results from the method and we conclude in Section 6.

2 Description of the system

2.1 Assumptions and Notation

2.1.1 Machines

We denote machine i by M_i and its downstream buffer by B_i ($i = 1, \dots, K$). We do not distinguish among parts, pallets, and tokens. We model the system as though the items neither enter nor leave, and we refer to the items as parts. There is therefore nothing special about M_1, M_K, or B_K. We use modulo K arithmetic in treating the indices that refer to machines and buffers, so M_1 is the machine downstream of M_K and M_{K+m} is the same as M_m.

The model we are considering is the same as the transfer line model of Tolio and Matta (1998) except for the loop structure. Processing times of each machine are equal and deterministic. A common fixed amount of time is required to perform each operation. Time is scaled so that each operation takes one time unit. We assume, also, that all the machines start their operations at the same instant. Transportation takes negligible time compared to the operation time.

Machines are unreliable and may have more than one failure mode; F_i represents the number of failure modes of machine M_i. At the beginning of each time step an operational machine M_i has a probability p_{ij} of failing in mode j ($j = 1, \ldots, F_i$). On the other hand, at the beginning of each time step, if machine M_i is down in mode j, it has a probability r_{ij} of being repaired. Therefore the quantity p_{ij} represents the probability that machine M_i enters failure mode j and r_{ij} is the probability of machine M_i being repaired in one time step while it is down due to a mode j failure.

As a consequence, the time to failure and the time to repair are geometrically distributed. By convention, repairs and failures occur at the beginnings of time steps and changes in the buffer levels (the amounts of material in the buffers) take place at the ends of time steps. Models with deterministic processing times and geometrically distributed times to failure and times to repair are referred as *synchronous models* (Dallery and Gershwin 1992) or Buzacott models.

We deal with unreliable machines that have operation-dependent failures (ODFs) (Buzacott and Hanifin 1978, Dallery and Gershwin 1992). That is, a machine can fail only while it is working.

2.1.2 Buffers

Buffers are finite; buffer B_i can hold a maximum of N_i parts. If B_i is full, M_i is not allowed to operate and is said to be *blocked*; if B_i is empty, M_{i+1} is *starved* and cannot operate. If M_i and M_{i+1} both operate during a time step, the level of B_i is unchanged. If neither M_i nor M_{i+1} operate, the level of B_i is also unchanged. If M_i operates and M_{i+1} does not operate during a time step, the level of the buffer increases by 1 at the *end* of the time step, ie, after the machine repair/failure states are changed. Similarly if M_i does not operate and M_{i+1} does during a time step, the level of the buffer decreases by 1 at the end of the time step.

2.1.3 Population

We assume that the number of parts in the system N^p — the population —is constant. We also assume that the population is greater than the size of the largest buffer and smaller than the sum of all the buffer sizes minus the size of the largest buffer. That is, assume B_m is the largest buffer. Then $N_m = \max_i N_i = N^{\mathbf{max}}$,

$$N^p > N^{\mathbf{max}} \tag{1}$$

$$N^p < \sum_{i=1}^{K} N_i - N^{\mathbf{max}} = \sum_{\substack{i=1 \\ i \neq z}}^{K} N_i \tag{2}$$

These assumptions are not restrictive since there is no benefit from increasing the size a buffer which can already hold all the parts; and there is no benefit from increasing the population if it can already fill up all the buffers except one. Simulation experience shows that when (1) is violated, a small increase in N^p gives a large increase in production rate; and when (2) is violated, a small decrease in N^p gives a large increase in production rate.

From (1) and (2),

$$N^{\max} < \sum_{\substack{i=1 \\ i \neq z}}^{K} N_i \tag{3}$$

That is, the size of the largest buffer is smaller than the sum of the sizes of the other buffers[1].

Let us analyze the implications of these assumptions. Suppose machine M_i is down. If the duration of the failure is long, buffer B_i becomes empty causing the starvation of M_{i+1}. If the duration is long enough, the starvation can propagate to all machines further downstream but can never reach machine M_{i-1} since we have assumed in equation (1) that not all parts can be contained in B_{i-1}. On the other hand if machine M_i is down, upstream buffer B_{i-1} continues to receive pieces from M_{i-1}. If the duration of the failure of M_i is long enough, B_{i-1} becomes full, causing the blocking of M_{i-1}. The blocking can propagate to all upstream machines but it can never reach machine M_{i+1} since we have assumed that all the parts cannot fill all the buffers except one, according to equation (2).

2.2 Disturbance propagation

When a machine in a open production line fails, starvation could propagate to all the machines downstream of it, and blocking could propagate to all the upstream machines. The propagation of blocking and starvation in a loop is different, and this difference requires a different kind of decomposition.

In a loop, the set of machines to which starvation and blockage can propagate depends on the sizes of the buffers and the population. Consider machines M_i, M_k and buffers B_i, B_{i+1}, ..., B_{k-1} between them. If the sum of the buffer sizes between M_i and M_k is greater than the loop population N^p, machine M_k can never block machine M_i because it cannot fill B_i. A failure of M_k, however, could cause the starvation of M_i, because all the parts can be contained in the buffers between M_i and M_k, and so B_{i-1} would be empty.

If the loop population N^p is greater than the sum of the buffer sizes between M_i and M_k, a very long failure of machine M_k will block M_i because all the buffers that are upstream of M_k and downstream of M_i will be full. Because this will leave some parts in buffer B_{i-1}, a long failure of M_k cannot starve M_i.

To summarize

[1] If (3) is violated but (1) is not, then the loop is equivalent to a tandem production system (a transfer line) comprised of all the machines and buffers except B_m appearing in the same sequence as in the the loop, in which M_{m+1} is the first machine and M_m is the last.

If $\sum_{j=i}^{k-1} N_j > N^p$	then a very long failure of M_k will not block M_i; but it will starve M_i.
If $\sum_{j=i}^{k-1} N_j < N^p$	then a very long failure of M_k will block M_i; but it will not starve M_i.

2.3 Thresholds

We can refine our understanding of the machines that could possibly be starved or blocked by a given machine. Suppose that machine M_k cannot cause the blockage of machine M_i. That is,

$$N^p < \sum_{j=i}^{k-1} N_j$$

Note that a long failure of M_k will cause M_i to be starved. Suppose also that M_k cannot cause the starvation of machine M_{i+1}. Then,

$$N^p > \sum_{j=i+1}^{k-1} N_j.$$

If M_k fails for a very long time, then buffers B_{i+1}, ..., B_{k-1} will be full, and buffers B_k, ..., B_{i-1} will be empty. Buffer B_i will have

$$L_{ik} = N^p - \sum_{j=i+1}^{k-1} N_j > 0$$

parts. This is feasible because

$$N_i - L_{ik} = N_i - \left(N^p - \sum_{j=i+1}^{k-1} N_j \right) = \sum_{j=i}^{k-1} N_j - N^p > 0$$

Now, suppose that at some time t, M_i is starved and there are $n_i(t) < L_{ik}$ parts in B_i. More specifically, M_i is starved due to the long failure of another machine and that machine is currently still down. Here, we argue that M_k cannot be that machine.

If the cause of the starvation was a failure of M_k, then buffers B_{k+1}, ..., B_{i-1} would be empty. All the parts would be in buffers B_i, ..., B_{k-1}. However, this is impossible. If all the buffers except B_i were full, there would be a total of

$$n_i(t) + \sum_{j=i+1}^{k-1} N_j$$

parts in buffers B_i, ..., B_{k-1}. But this is less than N^p and there are no parts in the other buffers. We can therefore conclude that a failure of M_k was *not* the cause of the starvation.

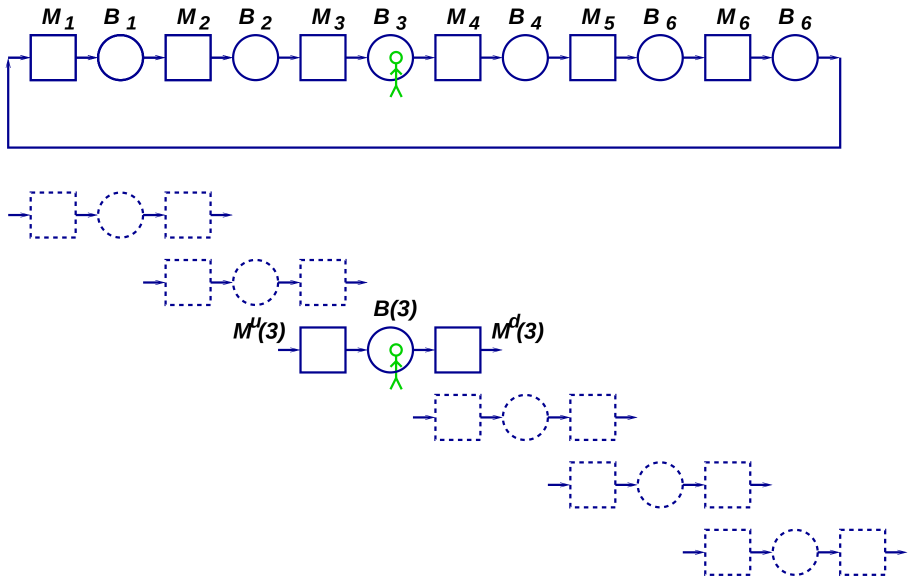


Figure 2: Decomposition method

This leads us to conclude that if M_i is starved and the level of the parts in buffer B_i is less than a certain *threshold*, then machine M_k cannot be the cause of the starvation. For the same reason, if there are too many parts (a number greater than a certain threshold) in buffer B_{i-1}, then M_i can not be blocked by machine M_k.

3 Solution methodology

3.1 Outline of the method

In decomposition methods like that of Gershwin (1987), Gershwin (1994), and Tolio and Matta (1998), the system to be analyzed (often called the *original system* or *real system*) is approximated by a set of two–machine lines (or *building blocks*) in which there is one two–machine line for each buffer in the system. Building block i, which corresponds to buffer B_i, is composed of an upstream machine $M^u(i)$, a downstream machine $M^d(i)$, and a buffer $B(i)$. (In a loop, the number of buffers is the same as the number of machines, so $i = 1, ..., K$.) Buffer $B(i)$ is the same size as B_i. $M^u(i)$ and $M^d(i)$ are often called *pseudo-machines*. A six-machine loop decomposed into six building blocks is shown in Figure 2.

The pseudo-machines approximate the aggregate behavior of portions of the line up- and downstream of buffer B_i. The basic idea of the decomposition method is to find the failure and repair parameters of the pseudo-machines, so that the material flow into and out of the buffer of each building block closely matches the flow of parts into and out of the corresponding buffer of the real system. In other words, consider an observer inside B_i who is able to see material entering and leaving the buffer but nothing else. We seek models of $M^u(i)$ and $M^d(i)$ such that he will believe us if we tell him that he is in $B(i)$ of building block i.

The flow of parts into buffer B_i (when the buffer is not full) will stop when M_i fails and when M_i becomes starved (ie, B_{i-1} becomes empty) due to a failure of one of the upstream machines. The observer who believes he is in $B(i)$ does not make such a distinction; he thinks that all interruptions of flow into $B(i)$ (when it is not full) are due to failures of $M^u(i)$.

Following Tolio and Matta (1998), we allow all the machines — those in the real system, and those in the building blocks — to have many failure modes. The modes of each of the building blocks will be the same as some of the modes of the real machines. Some of the failure modes of $M^u(i)$ are the same as those of M_i. We call them *local modes*. The other modes of $M^u(i)$ are the failure modes of M_{i-1} and all other machines that can starve M_i. We call them *remote modes*[2]. The observer sees the remote modes whenever B_{i-1} is empty[3].

Similarly, the flow out of the real buffer B_i stops (when B_i is not empty) if its downstream machine M_{i+1} is not able to work. This happens if M_{i+1} fails or if it is blocked (ie, buffer B_{i+1} is full) due to a failure of one of the further downstream machines. We assign $M^d(i)$ a set of local modes, which correspond to the failure modes of M_{i+1}, and a set of remote modes, which correspond to the modes of M_{i+2} and all other machines that can block M_{i+1}.

The problem then reduces to finding local and remote probabilities of failure and repair of the machines making up each building block, in order that the behavior of the flow into and out of the buffer $B(i)$ of two–machine line i closely matches that into and out of buffer B_i of the real system. If we can find all these parameters, we can determine the average throughput and buffer levels of the two–machine lines, and these will be close to the corresponding values of the real system (as indicated by the simulation comparisons in Section 5).

As indicated in Section 3.4.1, the probabilities of local failure and repair of two–machine line i are the same as those of the real machines adjacent to buffer B_i. Determining the probabilities of remote failures is more complex because the intervening buffers sometimes prevent failures from propagating.

3.2 Thresholds

Consider building block i, shown in Figure 3. If machine M_{i+1} can be blocked by machine M_k, $M^d(i)$ has a remote failure mode kj that corresponds to failure mode j of machine M_k, $j = 1, ..., F_k$. Among the unknown quantities that we must determine is the remote failure probability $p^d_{kj}(i)$. If M_{i+1} cannot be blocked by M_k, then $p^d_{kj}(i) = 0$.

In previous decompositions, the probabilities of failure $p^d_{kj}(i)$ were unknown quantities that were independent of the number of parts in $B(i)$. In this case, however, the threshold behavior discussed in Section 2.2 causes some of the probabilities to depend on the buffer level. As a result, a new two–machine model is required, which is described in Section 3.3.

For i and k such that M_{i+1} cannot be blocked by M_k for any level of the buffer, $p^d_{kj}(i) = 0$. For i and k such that M_{i+1} can be blocked by M_k for every level of the buffer, $p^d_{kj}(i)$ is a positive quantity — which we assume is independent of buffer level — that must be determined.

[2] In earlier papers by Tolio and his co-authors, the local and remote modes were called *real* and *virtual* modes.

[3] Omniscient modelers know which modes are local and which are remote, when B_{i-1} is empty and when B_i is full, but the observer does not.

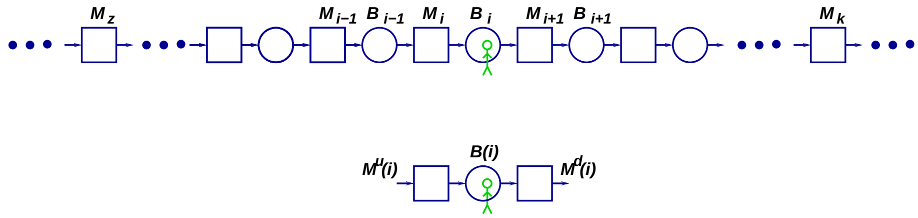


Figure 3: Illustration of a loop system and one of its building blocks

But for some i and k, the finite population N^p implies that if buffer B_i has too many parts, the remaining parts are not enough to fill all the buffers between M_{i+1} and M_k. Therefore, if the level of the buffer is greater than a threshold, $l_k^d(i)$, a failure of M_k cannot be the cause of blocking of machine M_{i+1}, so $p_{kj}^d(i)$ is 0. Machine M_k could produce blocking on machine M_{i+1} (and therefore it could contribute a failure mode to $M^d(i)$) only if the level of buffer $B(i)$ is less than or equal to $l_k^d(i)$.

A similar argument holds for the starvation of M_i. Suppose that machine M_i can be starved by machine M_z. This means that $M^u(i)$ has a remote failure mode that has probability $p_{zj}^u(i)$ of occurring (where j indicates one of the failure modes of the real machine M_z, $j = 1, \ldots, F_z$). If buffer B_i has too few parts, the remaining parts are too many to be contained in all the buffers space between M_{i+1} and M_z. Therefore M_i cannot be starved due to a failure of machine M_z, since the buffers between M_z and M_i cannot all be empty. More precisely, buffer B_{i-1} cannot be empty due to machine M_z being failed for a time long enough to make all the buffers between M_z and M_i empty. This observation leads us to say that $p_{zj}^u(i) = 0$ if the number of parts in $B(i)$ is less than a threshold $l_z^u(i)$. This symbol represents the threshold for the upstream machine $M^u(i)$ related to a failure of machine M_z. Machine M_z could produce starvation at machine M_i (and therefore it could affect $M^u(i)$) only if the level of buffer $B(i)$ is greater than or equal to $l_z^u(i)$.

The threshold $l_k^d(i)$ represents the largest number of parts in buffer B_i that allows all the buffers between M_{i+1} and M_k to be full at the same time. In that case all the buffers between M_k and M_i would be empty. Similarly, threshold $l_k^u(i)$ represents the smallest number of parts that allows all the buffers between M_k and M_i to be empty. Therefore, since machine M_{i+1} cannot be blocked by M_i and M_i cannot be starved by M_{i+1} (due to assumptions (1) and (2)), the two thresholds are the same, and we can simplify the notation:

$$l_k^u(i) = l_k^d(i) = l_k(i) \tag{4}$$

Let $n(t)$ be the number of parts in buffer $B(i)$ at time t. To summarize,

- if $n(t) < l_k(i)$ then $M^u(i)$ cannot be down in the remote mode that corresponds to failure mode j of machine M_k. Therefore $p_{kj}^u(i) = 0$.
- if $n(t) \geq l_k(i)$, the probability $p_{kj}^u(i)$ is independent of $n(t)$, and is an unknown to be determined.
- if $n(t) > l_k(i)$ then $M^d(i)$ cannot be down in the remote mode that corresponds to failure mode j of machine M_k. Therefore $p_{kj}^d(i) = 0$.

- if $n(t) \le l_k(i)$, the probability $p^d_{kj}(i)$ is independent of $n(t)$, and is an unknown to be determined.

Finally, if $n(t) = l_k(i)$ and all the buffers between M_{i+1} and M_k are full, then all the other buffers between M_k and M_i must be empty. Also, if $n(t) = l_k(i)$ and the downstream machine $M^d(i)$ is down in the remote mode corresponding to a mode j failure of machine M_k, then the upstream machine $M^u(i)$ must be down in its remote mode that corresponds to the same failure kj. Similarly if $M^u(i)$ is down in remote mode kj and $n(t) = l_k(i)$, then $M^d(i)$ must be down in its remote mode that corresponds to a mode j failure of M_k. That is, when $n(t) = l_k(i)$ either $M^u(i)$ and $M^d(i)$ are both down due to mode kj, or neither one is.

We define the quantity $\Psi(v, w)$ as the sum of the buffer sizes between M_v and M_w in the direction of flow, or

$$\Psi(v, w) = \sum_{z=v}^{w-1} N_z \tag{5}$$

The threshold is

$$l_k(i) = N^p - \Psi(i + 1, k) \tag{6}$$

The value of the threshold does not depend on the failure type but only on the buffer sizes between machines M_{i+1} and M_k and on the number of parts in the system, N^p.

3.3 Properties of the building block

3.3.1 Assumptions

In this section, we describe the two–machine line model that is used in the proposed solution method. A new model of the two–machine line is needed because of the thresholds described in Section 3.2. Thresholds — or any kind of buffer-level dependent failure probabilities — did not appear in earlier models of two–machine lines.

Each two–machine line is composed of two machines and one buffer. The first machine $M^u(i)$ is never starved and the second $M^d(i)$ is never blocked. Machines are unreliable and they have failure probabilities that may depend on the level of the buffer. We distinguish between two kinds of failures: those which can only occur for some values of the buffer level, or *BLD* (buffer level dependent[4]); and those which do not depend on the level of the buffer, or *BLI* (buffer level independent). All previous models in literature deal only with BLI failures.

When the level of the buffer is zero, the downstream machine cannot work since it is starved, and consequently it cannot fail. Therefore when the buffer is empty, the downstream machine cannot fail in any mode, BLI or BLD. Similarly, when the level of the buffer is equal to the buffer size (i.e. when the buffer is full), the upstream machine cannot work since it is blocked. Therefore when the buffer is full, the upstream machine cannot fail in any BLI or BLD mode.

[4] We use this general term although we are interested in only one specific kind of dependence: where the probability is determined by whether the level is above or below a threshold.

The numbers of BLD modes for the upstream and downstream machines are the same for the class of two-machine lines we study. To see this, assume that we are considering two-machine line i, which corresponds to buffer B_i in the original system. Each BLD mode of the upstream machine $M^u(i)$ is a remote mode that corresponds to a specific mode j of a specific machine M_k of the real system. When M_k fails in mode j, it causes machine M_i to be starved if the number of parts in buffer B_i is greater than a threshold, and that starvation appears to the observer in B_i as if $M^u(i)$ is down in the mode we are considering. When the level is below the threshold, M_i cannot be starved in that way, and so it appears that $M^u(i)$ cannot fail. At the same time, M_{i+1} can be blocked by M_k, so it appears that $M^d(i)$ can fail in the mode we are considering. If the level exceeds the threshold, $M^d(i)$ cannot fail in that mode. Therefore, for each BLD mode of $M^u(i)$, there is a corresponding BLD mode of $M^d(i)$, and the two modes have the same threshold.

Modes with the same threshold can be grouped. The size of each group from the upstream machine is the same as that for the corresponding group from the downstream machine with same threshold. The upstream machine cannot fail in a BLD mode when the number of work-pieces in the buffer is less than its threshold. In the same way the downstream machine cannot fail in a BLD mode when the number of work-pieces in the buffer is greater than its threshold.

The upstream machine cannot be down in any BLD mode kj when the level of the buffer is less than the threshold l_k. On the other hand, when the level of the buffer is greater than the threshold l_k, the downstream machine cannot be down in any BLD mode kj.

Finally, when the level of the buffer is equal to the threshold, either l_k and one of the two machines is down in BLD mode kj then the other machine is not able to process a part because it is also down in mode BLD kj. Taking into account these constraints, it is possible to derive a set of transition equations for the two–machine line.

In the following, we simplify notation by suppressing the index i, which elsewhere indicates which building block we are referring to.

3.3.2 States

The system is be modeled as a discrete time, discrete state Markov chain. The state of the system is indicated by $s(t) = (n(t), \alpha_u(t), \alpha_d(t))$ where:

- $n(t)$ represents the level of the buffer at time t ($n = 1, \ldots, N$; where N is the buffer size.)
- $\alpha_u(t)$ represents the state of the upstream machine M^u at time t and can assume the following values:

$$\alpha_u = \begin{cases} 1 \\ \rho_j \\ \lambda_{kj} \end{cases}$$

  where

  - ⋆ 1 means that the machine is up, i.e. it is able to work and process a part.
  - ⋆ ρ_j means that the machine is down in BLI mode j, $j = 1, ..., F^u_\rho$.

  - ⋆ λ_{kj} means that the machine is down in BLD mode kj. The index k indicates a group of failure modes with the same threshold l_k while j represents one of the $F_{\lambda k}$ different BLD modes of type k.

- $\alpha_d(t)$ represents the state of the downstream machine M^d at time t and can assume the following values:

$$\alpha_d = \begin{cases} 1 \\ \rho_j \\ \lambda_{kj} \end{cases}$$

  - ⋆ 1 means that the machine is up, i.e. it is able to work and process a part.
  - ⋆ ρ_j means that the machine is down in BLI mode j, $j = 1, ..., F_\rho^d$.
  - ⋆ λ_{kj} means that the machine is down in BLD mode kj. The index k indicates the group of failure modes with the same threshold l_k while j represents one of the $F_{\lambda k}$ (this number is equal to that of the upstream machine) different BLD modes of type k.

The transition equations are derived in Maggio (2000). As usual, they are used to determine the steady state probability $p(n, \alpha_u, \alpha_d)$ of each state. Then the performance measures — the average throughput and the average buffer levels — can be calculated.

3.3.3 Performance Measures

E^u is the probability that a part passes through the upstream machine in one time unit. It is given by the sum of the probabilities of the states in which the upstream machine is able to process a part. This happens when the machine is not down or not blocked (since the first machine of a two–machine line cannot be starved). Therefore E^u can be defined as

$$E^u = \sum_{n=0}^{N-1} p(n, 1, *) = \sum_{n=0}^{N-1} \sum_{\alpha_d} p(n, 1, \alpha_d) \tag{7}$$

where the symbol $*$ represents a summation over all the states of the machine (up or down either in a BLD or BLI mode).

E^d is the probability that a part passes through the second machine in a time unit. Therefore it represents the sum of the probabilities of the states in which the downstream machine is able to process a part. This happens when the machine is not down or not starved, since it cannot be blocked. Thus E^d can be defined as

$$E^d = \sum_{n=1}^{N} [p(n, *, 1)] \tag{8}$$

Material is conserved, since we do not consider any mechanism for creation or destruction of material. Therefore E^u and E^d must be equal in steady state. Conservation of flow in the system can be expressed as

$$E = E^u = E^d \tag{9}$$

where E is the average *throughput* of the two–machine line. The average buffer level of the two–machine line is

$$\overline{n} = \sum_{n=0}^{N} np(n,*,*) \tag{10}$$

Other important performance measures include the probabilities of blocking and starvation. The probability of finding the buffer empty (the probability of starvation of the second machine) is

$$P^s = p(0,*,1) \tag{11}$$

while the probability of finding the buffer full (the probability of blocking of the first machine) is

$$P^b = p(N,1,*) \tag{12}$$

3.3.4 Other Probability Measures

We define some quantitis that appear the decomposition equations. E^u_{bk} is defined to be the probability of the upstream machine being operational and the number of parts in the buffer being greater than or equal to the threshold l_k:

$$E^u_{bk} = \sum_{n=l_k}^{N-1} p(n,1,*) \tag{13}$$

Similarly, E^d_{ak} is the probability of the downstream machine being operational and the number of parts in the buffer being less than or equal to the threshold l_k.

$$E^d_{ak} = \sum_{n=1}^{l_k} p(n,*,1) \tag{14}$$

The probability of M^u being down in BLD mode kj is

$$X^u_{kj} = \sum_{n=0}^{N} p(n,\lambda_{kj},*) \tag{15}$$

Because the upstream machine cannot fail in a BLD mode if the number of parts in the buffer is less than the threshold, we must have

$$\sum_{n=0}^{l_k-1} p(n,\lambda_{kj},*) = 0 \tag{16}$$

Therefore the probability of M^u being down in mode BLD kj when the buffer level is greater than or equal to l_k is

$$X^u_{bkj} = \sum_{n=l_k}^{N} p(n,\lambda_{kj},*) = X^u_{kj} \tag{17}$$

The probability of M^d being down in mode BLD kj is

$$X^d_{kj} = \sum_{n=0}^{N} p(n, *, \lambda_{kj}) \tag{18}$$

Because the downstream machine cannot fail in a BLD mode if the number of parts in the buffer is greater than the threshold, we can write

$$\sum_{n=l_k+1}^{N} p(n, *, \lambda_{kj}) = 0 \tag{19}$$

Therefore the probability of M^d being down in mode BLD kj when the buffer level is less than or equal to l_k is

$$X^d_{akj} = \sum_{n=0}^{l_k} p(n, *, \lambda_{kj}) = X^d_{kj} \tag{20}$$

The probabilities that the buffer is empty due to the upstream machine being down respectively in BLI mode j and in BLD mode kj are

$$P^s_j = p(0, \rho_j, *) \tag{21}$$

$$P^s_{kj} = p(0, \lambda_{kj}, *) \tag{22}$$

Finally, the probabilities that the buffer is full due to the downstream machine being down respectively in BLI mode j and in BLD mode kj are:

$$P^b_j = p(N, *, \rho_j) \tag{23}$$

$$P^b_{kj} = p(N, *, \lambda_{kj}) \tag{24}$$

3.4 Decomposition Equations

The decomposition method works by constructing two–machine lines in which the behavior of the flow into and out of the buffer closely matches that of a corresponding buffer in the real line. To do this, we must carefully select the parameters of each two–machine line. In this section, we derive a set of equations that these parameters must satisfy to create the desired behavior.

3.4.1 Local failures

According to the definition of local failures (Section 3.1), the probabilities of local failures of machine $M^u(i)$ and their repairs are equal to those of machine M_i of the original system, The parameter $p^u_{ij}(i)$ represents the probability of machine $M^u(i)$ failing in mode ij. This mode reproduces the interruption of flow into buffer B_i caused by M_i being down in mode j. The parameter $r^u_{ij}(i)$ represents the probability of the resumption of flow due to M_i being repaired from failure j. Therefore we can write

$$p^u_{ij}(i) = p_{ij} \tag{25}$$

$$r^u_{ij}(i) = r_{ij} \tag{26}$$

Similarly for machine $M^d(i)$. The value $p^d_{i+1,j}(i)$ represents the local failure probability of machine $M^d(i)$ and reproduces the interruption of flow out of buffer B_i caused by M_{i+1} being down in mode j. The value $r^d_{i+1j}(i)$ represents the probability of the resumption of flow due to M_{i+1} being repaired from failure j. Thus,

$$p^d_{i+1,j}(i) = p_{i+1,j} \tag{27}$$

$$r^d_{i+1,j}(i) = r_{i+1,j} \tag{28}$$

3.4.2 Remote failures

Repair probabilities We must still determine the parameters of the remote failures of $M^u(i)$ and $M^d(i)$. To find the probabilities of repair, recall that failure propagation occurs through a set of full buffers or a set of empty buffers. When the remote machine is repaired, this set of buffers is no longer all full or all empty, and we assume that flow into or out of the buffer resumes instantaneously. That is, we ignore the time required for repairs to propagate[5]. Then we observe that the probability of recovering from a remote failure is equal to the probability of repairing the failed machine which caused the interruption of flow. This is because we are assuming that all repair time distributions are geometric (ie, memoryless), and because the unblocking of the downstream machine occurs at essentially the same time as the repair of the machine that caused the blockage (and similarly for the end of the starvation of the upstream machine). This repair probability does not depend on the condition of the system and is constant. The remote repair probabilities are therefore

$$r^u_{kj}(i) = r_{kj} \tag{29}$$

$$r^d_{k'j}(i) = r_{k'j} \tag{30}$$

Failure probabilities Failures, by contrast, do not propagate almost instantaneously. Time is required for the intervening buffers to fill or empty, and some failures will not propagate the whole distance because the repair occurs before the propagation reaches M_i or M_{i+1}. Consequently the remote failure probabilities ($p^u_{kj}(i)$ and $p^d_{kj}(i)$) are unequal to the failure probabilities (p_{kj}) of the real machines that cause the failures — in fact, they are smaller. We are not able to write simple equations for these quantities; we must find them indirectly by using properties of the original system.

For every failure there is a repair. Consequently, the steady-state probability of entering a failure state is equal to the probability of leaving that state. Suppose that $M^u(i)$ is down in remote mode

[5] This is equivalent to assuming that the number of machines in the network is much smaller than the mean time to repair of each machine. All decomposition methods with discrete material of which we are aware make an approximation like this, although not always explicitly. In continuous material systems, by contrast, the propagation of repairs is instantaneous. For example, consider a system in which all the machines have the same operation rate. If machine M_i is down and machines M_i, M_{i+1}, and M_{i+2}, are up and buffers B_i, B_{i+1}, and B_{i+2} are empty, B_{i+3} will gain material immediately after M_i is repaired.

kj. For this to happen, the number of parts in buffer $B(i)$ must be greater than or equal to $l_k(i)$. The probability of leaving this condition is therefore $X^u_{bkj}(i)$ times the remote repair probability $r^u_{kj}(i)$, or $X^u_{bkj}(i)\ r^u_{kj}(i)$.

Now let us consider the probability of $M^u(i)$ entering remote failure mode kj. Such a failure can happen only if buffer $B(i)$ contains at least $l_k(i)$ parts. The probability that $M^u(i)$ is up and there are at least $l_k(i)$ parts in buffer $B(i)$ is $E^u_{bk}(i)$. Therefore, the probability that $M^u(i)$ enters mode kj is $E^u_{bk}(i)\ p^u_{kj}(i)$, so

$$E^u_{bk}(i)\ p^u_{kj}(i) = X^u_{bkj}(i)\ r^u_{kj}(i) \tag{31}$$

Here $E^u_{bk}(i)$ is not the total throughput, but only the portion of it when the number of parts in the buffer is greater than or equal to $l_k(i)$. $E^u_{bk}(i)$ represents the probability that the upstream machine is able to process a part and $n \geq l_k(i)$.

Similarly for the downstream machine: $M^d(i)$ cannot fail in remote mode kj when $n > l_k(i)$, and when it is down, the number of parts cannot exceed this threshold. Therefore, for $M^d(i)$ to be down in mode kj, the number of parts in buffer $B(i)$ must be less than or equal to $l_k(i)$. The probability of leaving this condition is therefore $X^d_{akj}(i)r^d_{kj}(i)$.

The probability of $M^d(i)$ failing in this remote mode is the probability that it is up and the level of the buffer is less than or equal to $l_k(i)$, $E^d_{ak}(i)$, times the probability it fails in remote mode kj when a failure is possible, which is $p^d_{kj}(i)$. Therefore

$$E^d_{ak}(i)p^d_{kj}(i) = X^d_{akj}(i)r^d_{kj}(i) \tag{32}$$

The event that the upstream machine $M^u(i)$ is down due to a mode j failure of machine M_k is the event that that buffer B_{i-1} is empty due that failure. Thus the probability that $M^u(i)$ is down in the remote failure mode kj is the probability that B_{i-1} is empty because of the failure j of machine M_k, or, if $k \neq i-1$

$$X^u_{kj}(i) = P^s_{kj}(i-1) \tag{33}$$

If $k = i-1$ this reduces to

$$X^u_{i-1,j}(i) = P^s_j(i-1) \tag{34}$$

Similar considerations apply to the downstream machine $M^d(i)$. The event that $M^d(i)$ is down in the remote mode due to the failure j of machine M_k is the event that buffer B_{i+1} is full because of the failure j of machine M_k. The probability of this event is the same as the probability of blocking in building block $i+1$. Therefore, if $k \neq i+2$,

$$X^d_{kj}(i) = P^b_{kj}(i+1) \tag{35}$$

If $k = i+2$, this becomes

$$X^d_{i+2,j}(i) = P^b_j(i+1) \tag{36}$$

Note that at the same time that the observer in $B(i)$ sees a kj failure of $M^d(i)$, the observer in $B(i+1)$ sees a kj failure of $M^d(i+1)$ if $k \neq i+1$. This is because both failures are due to a mode j failure of machine M_k, and if buffer B_{i+1} is full due to that event, B_{i+2} must also be full.

3.5 Final Decomposition Equations

Using equations (17), (20), (29), (30), (33), (34), (35), and (36) in (31) and (32) we can determine the remaining unknown parameters. After some simplification, they become, for $i = 1, ..., K$:

$$p^u_{kj}(i) = \frac{P^s_j(i-1)}{E^u_{bk}(i)}\, r_{kj}, \qquad r^u_{kj}(i) = r_{kj}; \qquad \text{for } k = i-1; \tag{37}$$

$$p^u_{kj}(i) = \frac{P^s_{kj}(i-1)}{E^u_{bk}(i)}\, r_{kj}, \qquad r^u_{kj}(i) = r_{kj}; \qquad \text{for } k \neq i-1; \tag{38}$$

$$p^d_{kj}(i) = \frac{P^b_j(i+1)}{E^d_{ak}(i)}\, r_{kj}, \qquad r^d_{kj}(i) = r_{kj}; \qquad \text{for } k = i+2; \tag{39}$$

$$p^d_{kj}(i) = \frac{P^b_{kj}(i+1)}{E^d_{ak}(i)}\, r_{kj}, \qquad r^d_{kj}(i) = r_{kj}; \qquad \text{for } k \neq i+2; \tag{40}$$

for remote failures, and

$$p^u_{ij}(i) = p_{ij} \qquad r^u_{ij}(i) = r_{ij} \tag{41}$$

$$p^d_{i+1,j}(i) = p_{i+1,j} \qquad r^d_{i+1,j}(i) = r_{i+1,j} \tag{42}$$

for local failures. These equations are a set of independent nonlinear equations for calculating the unknown building block parameters. Conceptually, they may be solved iteratively: start with an initial guess for the unknown failure probabilities that appear on the right sides of these equations. Using this guess, evaluate the left sides, and this is the new guess of the unknowns for the unknowns. However, carrying this out involves some difficulties that are discussed in the next section.

3.6 Issues in the application of the decomposition equations

Just as in earlier decomposition methods, the solution of equations (37)-(42) involves the evaluation of certain quantities ($P^s_{kj}(i-1), E^u_{bk}(i)$, etc.) by using the steady-state probability distribution of the two-machine line. However, this is more difficult than in the past because of the thresholds discussed in Sections 3.2 and 3.3. Some failure probabilities depend on whether the buffer levels are above or below a threshold. This makes the two-machine line transition equations more complex than earlier versions. Worse, a further complication comes from the fact that there may be more than one threshold for some two-machine lines. To simplify the analysis we restrict our attention in the next section to three-machine loops with no more than one threshold in each building block. A much more general approach for dealing with this issue — and analyzing much more complex systems — is described in Werner (2001), Gershwin and Werner (2002), Levantesi (2001) and Gershwin and Levantesi (2002).

4 Application to Three-Machine Three-Buffer Loops

The general analysis of the previous section is specialized here to determine the performance measures of a loop composed of three machines and three buffers, with one failure mode for each machine. See Figure 4.

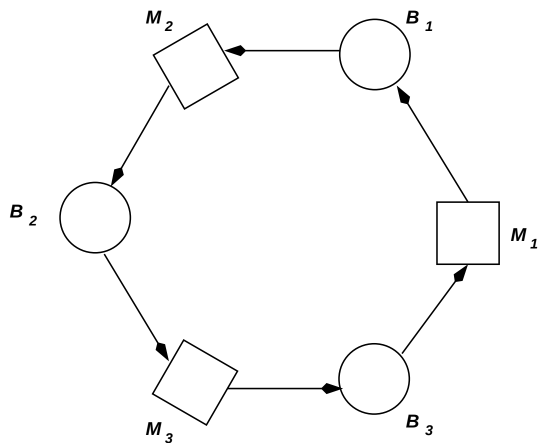


Figure 4: Three-machine loop.

The three-machine loop is decomposed, as shown in Figure 5, into three two-machine building blocks. The remote failure modes of the upstream machine of a building block represents the effect of starvation in the real line. However, since each real machine can be starved due to the failure of only one other real machine (because of the assumptions of Section 2.1.3), we have only one remote failure mode for each upstream machine. Each upstream machine $M^u(i)$ therefore has one local failure mode, the same as that of the real machine M_i immediately upstream of buffer B_i , and one remote failure mode (that of M_{i-1}). Similarly, the downstream machine $M^d(i)$ has one local failure mode, the failure mode of machine M_{i+1} and one remote failure mode (that of M_{i+2}), since each machine can be blocked due to the failure of only one other machine. (Recall that M_{i-1} and M_{i+2} are the same machine.)

From (6), the threshold $l(i)$ for building block i, due to the failure of machine M_{i+2} (ie M_{i-1}), is

$$l(i) = l_{i+2}(i) = N^p - N_{i+1} \tag{43}$$

Since the population of parts N^p is subject to equations (1) and (2) and since the size of the largest buffer observes equation (3), the threshold must satisfy

$$0 < l(i) < N_i \tag{44}$$

Note that there is exactly one threshold per buffer. This is because of the assumptions of Section 2.1.3) and because we are restricting our analysis to three-machine loops. It is not generally true.

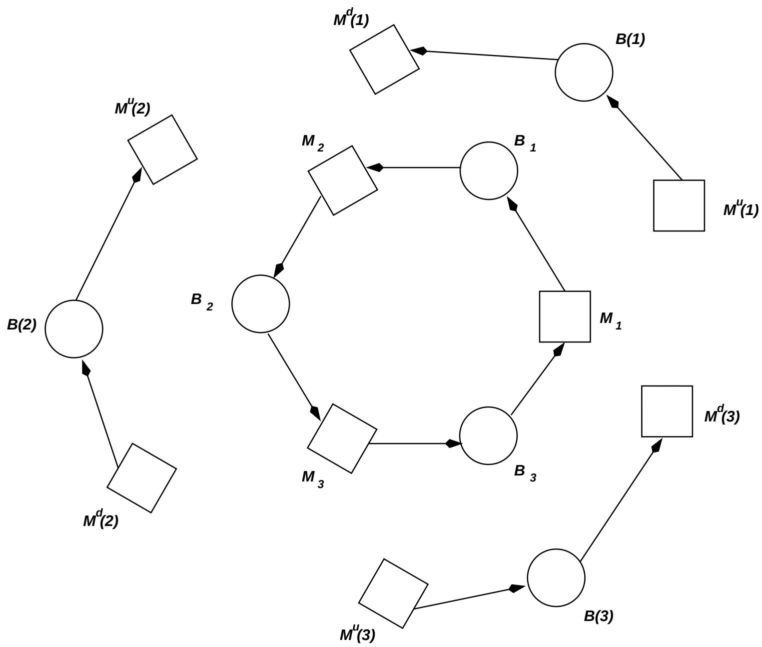


Figure 5: Decomposition method for three-machine three-buffer loops

4.1 Building Block

A complete description of the two-machine line is presented in Maggio (2000), in which the transition equations are described in detail.

In this version of the two machine line, there are $9(N+1)$ states (n, α_1, α_2), where

$$0 \leq n \leq N$$
$$\alpha_i = 1 \text{ or } \rho_i \text{ or } \lambda_i$$

in which, as in Section 3.3.2, 1 is the operational state, ρ_i is the BLI down state, and λ_i is the BLD down state. There is one BLI and one BLD state in each line, and only one threshold.

The state space is divided (in most cases) into five groups:

- the *lower boundary states*: $0 \leq n \leq 1$
- the *lower internal states*: $2 \leq n \leq l-2$
- the *threshold neighboring states*: $l-1 \leq n \leq l+1$
- the *upper internal states*: $l+2 \leq n \leq N-2$
- the *upper boundary states*: $N-1 \leq n \leq N$

In the lower boundary states (as in all earlier models) we have to take into account that when the level of the buffer n is equal to zero, the downstream machine cannot fail since it is starved.

In the upper boundary states (as in all earlier models), when n is equal to the buffer size N the upstream machine cannot fail since it is blocked. In the threshold neighboring states when n is equal to the threshold l, if one of the machines is down in the remote mode, then the other machine is also not able to process a part.

These three phenomena are separate from one to another only if the threshold is not equal to $0, 1, N-1$ or N. If the threshold assumes any of these values then two of the phenomena will occur at the same states and special versions of the equations are necessary.

The transition equations for all states other than the threshold neighboring states are the same as those in Tolio, Gershwin, and Matta (2002). The threshold state transition equations are provided in Maggio (2000).

4.2 Decomposition equations

The local failure parameters are given by (41) and (42). The following equations are related only to remote failures and follow from (37) and (39).

$$p^u_\lambda(1) = \frac{P^s(3)}{E^u_b(1)}\, r_3 \tag{45}$$

$$p^u_\lambda(2) = \frac{P^s(1)}{E^u_b(2)}\, r_1 \tag{46}$$

$$p^u_\lambda(3) = \frac{P^s(2)}{E^u_b(3)}\, r_2 \tag{47}$$

$$p^d_\lambda(1) = \frac{P^b(2)}{E^d_a(1)}\, r_3 \tag{48}$$

$$p^d_\lambda(2) = \frac{P^b(3)}{E^d_a(2)}\, r_1 \tag{49}$$

$$p^d_\lambda(3) = \frac{P^b(1)}{E^d_a(3)}\, r_2 \tag{50}$$

4.3 An Algorithm

The decomposition equations are nonlinear. Here, we describe an algorithm for solving them.

0. **Initialization:** for each of the three two–machine lines into which the loop is decomposed, the known parameters ($p^u_\rho(i)$, $p^d_\rho(i)$, $r^u_\rho(i)$, $r^d_\rho(i)$, $r^u_\lambda(i)$ and $r^d_\lambda(i)$) are specified and initial guesses for the unknowns ($p^u_\lambda(i)$, $p^d_\lambda(i)$) are chosen[6].

[6]For example, the initial guesses may be set to the failure probabilities of the machine (M_{i+2}) not adjacent to the buffer (B_i) whose building block is being initialized. The unknown $p^u_\lambda(i)$ represents the probability that machine M_{i+2} is down and its downstream buffer B_{i+2} is empty while $p^d_\lambda(i)$ represents the probability that M_{i+2} is down and

1. Evaluate the two–machine lines. Calculate $E_a^d(i)$, $E_b^u(i)$, $P^s(i)$ and $P^b(i)$.

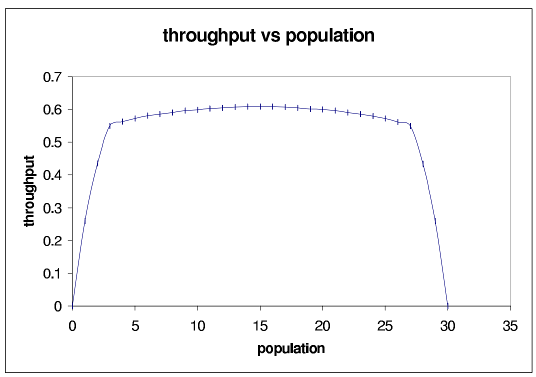
2. Evaluate $p_\lambda^u(i)$ and $p_\lambda^d(i)$ using (37)-(40).

3. Steps 1 and 2 are repeated until the termination condition is satisfied, that is until the remote probabilities converge. The termination condition is satisfied when the relative change in the six unknown parameters are all less than a specified quantity.

In our experience, which is described in Section 5, the algorithm is fast and accurate.

5 Numerical Results

In this section we examine the results of the analytical method and we compare them with those obtained from simulation. Of special interest is the effect of the number of parts on the production rate or throughput of the line. These results show that there exists an optimal number $\overline{N}$ of parts which maximizes the throughput of the line (as illustrated in Figure 6).



p1	0.06	r1	0.1	N1	7
p2	0.03	r2	0.7	N2	10
p3	0.04	r3	0.2	N3	13

Figure 6: Throughput from simulation as function of the number of parts

It has been shown (Perros 1990, Dallery and Towsley 1991) that for loops with identical machines, the curve is symmetric and the optimal population $\overline{N}$ is equal to half the sum of all the buffer sizes of the system. (See also Han and Park 2002.) Below the optimal number, the limited number of parts causes frequent starvation, and above it the system becomes congested and blockage occurs often. By solving the decomposition equations, we obtained the expected shape for the curve of the production rate as a function of the number of parts, as shown in Figure 7 for both a symmetrical and an asymmetrical loop.

Unlike other analytical methods, this algorithm does not check whether conservation of flow is satisfied. In Table 1 we display the the throughput of each building block and we compare them

its upstream buffer B_{i+1} is full. The remote probabilities of failure are actually smaller than the failure probabilities of M_{i+2}, but this is a reasonable initial guess.

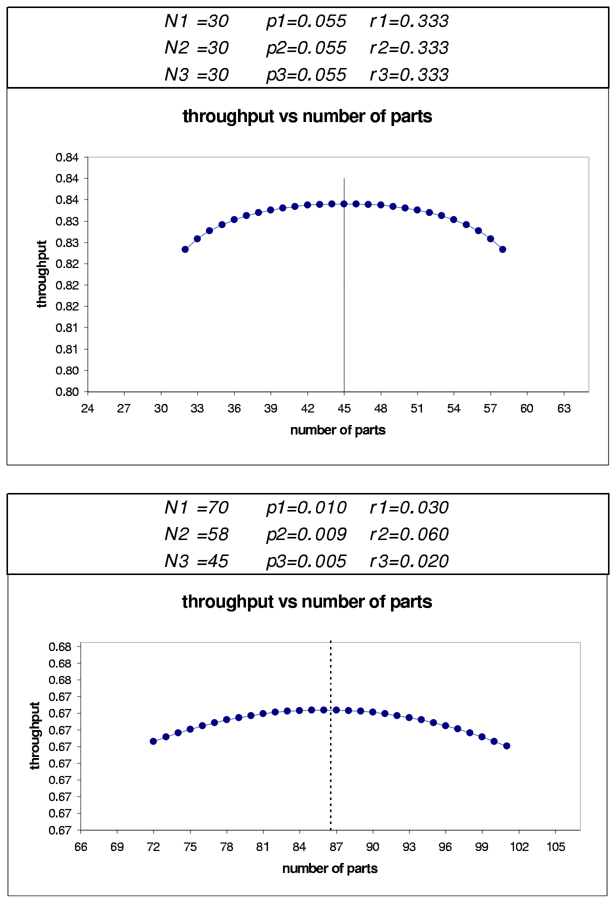


Figure 7: Analytical throughput as function of the number of parts

with that from simulation. Although the throughputs are different from one other, they are all very close to that from simulation.

Numerous experiments on different kinds of three-machine loop systems were carried out to test the accuracy, convergence reliability, and speed of the technique. We take an average of the throughputs of the building blocks and compare them with those obtained from a discrete event simulation. To ensure statistical significance, the length of simulation was chosen to be $10,000,000$ time units. The first set of experiments (Examples 1 – 4) considers symmetrical closed loop systems, that is, lines in which the machines are identical and the buffers have the same size. The second set (Examples 5 and 6) considers closed loop systems with identical machines but buffers with different sizes. The third set (Example 7) examines a loop with three different machines and identical buffers and the last set of runs (Example 8) deals with a completely asymmetrical closed line.

In Tables 3-15 numerical results of the analytical method are compared with those from simulation over a range of a different number of parts, while Table 2 provides the input parameters of the examples.

From a comparison of the analytical results with those from simulation, it can be seen that the method proposed seems to be fairly accurate. The error for the throughput is less then 0.8% for large buffers, but it increases to 1.24% for small buffers as illustrated in Table 7. The error in the average buffer level is almost always less than 3%, but it increases when the machines are not balanced or the buffers are different from one another. Furthermore we observed that the procedure converged in every case.

6 Conclusions and further research

6.1 Summary and Observations

This research was motivated by the need for a fast and accurate tool to evaluate the performance of small closed-loop material flow systems. We described in this paper an analytical model of the phenomena that occur in closed lines, and a decomposition method for predicting performance measures. This method provides quick results with generally small errors for three-machine, three-buffer loop systems. The error, compared with simulation, is small both for the average throughput and for the average buffer levels. We believe that similar results could be obtained by applying the method to loops with more than three machines. See Section 6.2.

The structure of the building block, however, suggests that the way in which the threshold affects the behavior of the two–machine line might cause difficulties for larger systems. As the number of machines increases, the number of thresholds that could appear in a given buffer also increases. Many of them will not affect the behavior of the two–machine line since they will be greater than the buffer size or negative. The feasible thresholds, those that are positive and less than the buffer size, however, lead to more complex behavior for states in which the number of parts is equal to a threshold. Since the number of thresholds in each buffer increases with the number of machines, boundary states and states adjacent to thresholds will more frequently overlap and special transition equations must be developed for these cases. As loops become larger, the building block equations become more complex since they must mode all the possible interactions between the threshold-neighboring states and the boundary states. These difficulties might be overcome by simplifying the phenomena due to the thresholds. Therefore, further research should extend the method to larger loop systems, using simplified two–machine line models.

6.2 Future Work

An approach to mitigating this difficulty is reported in Werner (2001), Gershwin and Werner (2002), Levantesi (2001) and Gershwin and Levantesi (2002). Gershwin and Werner (2002) eliminate the threshold issue in large closed loop systems by breaking up buffers at thresholds and introducing reliable machines between the new smaller buffers. All feasible thresholds are thereby moved to states in which buffers are empty or full. Simulation and other results suggest that the method works well.

Gershwin and Levantesi (2002) extend this method to systems with multiple closed loops. The method again works well, but substantial effort is required to analyze the ranges of blocking and starvation in the more complex network.

Acknowledgments

We are grateful for research support from the Singapore-MIT Alliance.

References

Buzacott, J. A. and L. E. Hanifin (1978). Models of automatic transfer lines with inventory banks — a review and comparison. *AIIE Transactions 10*(2), 197–207.

Buzacott, J. A. and J. G. Shantikumar (1992). A general approach for coordinating production in multiple-cell manufacturing systems. *Production and Operations Management 1*(1), 34–52.

Buzacott, J. A. and J. G. Shantikumar (1993). *Stochastic Models of Manufacturing Systems.* Englewood Cliffs, New Jersey: Prentice Hall.

Dallery, Y. and S. B. Gershwin (1992, December). Manufacturing flow line systems: A review of models and analytical results. *Queueing Systems Theory and Applications, Special Issue on Queueing Models of Manufacturing Systems 12*, 3–94.

Dallery, Y. and D. Towsley (1991). Symmetry property of the throughput in closed tandem queueing networks with finite capacity. *Operations Research 10*(9), 541–547.

Frein, Y., C. Commault, and Y. Dallery (1996). Modeling and analysis of closed-loop production lines with unreliable machines and finite buffers. *IIE Transactions 28*, 545–554.

Gershwin, S. B. (1987, March-April). An efficient decomposition method for the approximate evaluation of tandem queues with finite storage space and blocking. *Operations Research 35*(2), 291–305.

Gershwin, S. B. (1994). *Manufacturing Systems Engineering*. Prentice-Hall. See `http://web.mit.edu/manuf-sys/www/gershwin.errata.html` for corrections.

Gershwin, S. B. (2000, October). Design and operation of manufacturing systems: The control-point policy. *IIE Transactions 32*(10), 891–906.

Gershwin, S. B. and R. Levantesi (2002). An approximate analytical method for evaluating the performance of multiple-loop flow systems with unreliable machines and finite buffers. In preparation.

Gershwin, S. B. and L. Werner (2002). An approximate analytical method for evaluating the performance of closed loop flow systems with unreliable machines and finite buffers — Part II: General loops. In preparation.

Han, M.-S. and D.-J. Park (2002). Performance analysis and optimization of cyclic production lines. *IIE Transactions 34*, 411–422.

Hopp, W. J. and M. L. Roof (1998). Setting wip levels with statistical throughput control (stc) in conwip production lines. *International Journal of Production Research 36*(4), 867–882.

Hopp, W. J. and M. L. Spearman (1996). *Factory Physics: Foundations of Manufacturing Management*. Irwin/McGraw-Hill.

Levantesi, R. (2001). *Analysis of Multiple Loop Assembly/Disassembly Networks*. Ph. D. thesis, Politecnico di Milano.

Maggio, N. (2000). An analytical method for evaluating the performance of closed loop production lines with unreliable machines and finite buffer. Master's thesis, Politecnico di Milano.

Matta, A. and T. Tolio (2003). A new decomposition method for closed production lines with finite buffer capacity and unreliable machines. Technical Report 1-2003, Politicnico di Milano — Dipartimento di Meccanica.

Perros, H. G. (1990). Survey of closed queueing networks with blocking. *ACM Computing Surveys 22*(2), 83–121.

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

Tolio, T. and S. B. Gershwin (1998, March). Throughput estimation in cyclic queueing networks with blocking. *Annals of Operations Research 79*, 207–229.

Tolio, T., S. B. Gershwin, and A. Matta (2002, January). Analysis of two-machine lines with multiple failure modes. *IIE Transactions 34*(1), 51–62.

Tolio, T. and A. Matta (1998). A method for performance evaluation of automated flow lines. *Annals of the CIRP 47/1*, 373–376.

Werner, L. (2001). Analysis and design of closed loop manufacturing systems. Master's thesis, MIT OR Center.

N1	70	p1	0.01	r1	0.03
N2	58	p2	0.009	r2	0.06
N3	45	p3	0.005	r3	0.02

Number of Parts	throughput of first building block	throughput of second building block	throughput of third building block	Throughput from simulation	% error of the first building block throughput compared with simulation	% error of the second building block throughput compared with simulation	% error of the third building block throughput compared with simulation
71	0.657684	0.687390	0.674885	0.67037	-1.892%	2.539%	0.674%
72	0.658030	0.688082	0.670606	0.6716	-2.021%	2.454%	-0.148%
73	0.658410	0.688123	0.670839	0.67125	-1.913%	2.514%	-0.061%
74	0.658772	0.688146	0.671053	0.67287	-2.095%	2.270%	-0.270%
75	0.659115	0.688152	0.671249	0.67217	-1.942%	2.378%	-0.137%
76	0.659441	0.688140	0.671427	0.67141	-1.783%	2.492%	0.003%
77	0.659748	0.688113	0.671588	0.67252	-1.899%	2.319%	-0.139%
78	0.660038	0.688069	0.671733	0.67285	-1.904%	2.262%	-0.166%
79	0.660311	0.688010	0.671863	0.67176	-1.704%	2.419%	0.015%
80	0.660566	0.687936	0.671977	0.67334	-1.897%	2.168%	-0.202%
81	0.660803	0.687848	0.672075	0.67257	-1.750%	2.272%	-0.074%
82	0.661024	0.687745	0.672160	0.66983	-1.315%	2.675%	0.348%
83	0.661227	0.687629	0.672230	0.67241	-1.663%	2.263%	-0.027%
84	0.661412	0.687498	0.672285	0.67283	-1.697%	2.180%	-0.081%
85	0.661580	0.687355	0.672327	0.67058	-1.342%	2.502%	0.261%
86	0.661730	0.687198	0.672354	0.67328	-1.715%	2.067%	-0.138%
87	0.661863	0.687029	0.672368	0.67417	-1.826%	1.907%	-0.267%
88	0.661976	0.686848	0.672367	0.67145	-1.411%	2.293%	0.137%
89	0.662072	0.686654	0.672353	0.67063	-1.276%	2.389%	0.257%
90	0.662148	0.686449	0.672325	0.67364	-1.706%	1.901%	-0.195%
91	0.662205	0.686232	0.672282	0.67087	-1.292%	2.290%	0.210%
92	0.662242	0.686004	0.672225	0.67169	-1.407%	2.131%	0.080%
93	0.662259	0.685764	0.672153	0.67234	-1.499%	1.997%	-0.028%
94	0.662255	0.685515	0.672066	0.67167	-1.402%	2.061%	0.059%
95	0.662230	0.685255	0.671963	0.67206	-1.463%	1.963%	-0.014%
96	0.662182	0.684984	0.671844	0.67446	-1.820%	1.560%	-0.388%
97	0.662112	0.684705	0.671709	0.67354	-1.697%	1.658%	-0.272%
98	0.662019	0.684415	0.671558	0.67273	-1.592%	1.737%	-0.174%
99	0.661900	0.684117	0.671388	0.67189	-1.487%	1.820%	-0.075%
100	0.661757	0.683811	0.671200	0.67241	-1.584%	1.696%	-0.180%
101	0.661587	0.683496	0.670993	0.67074	-1.365%	1.902%	0.038%

Table 1: Conservation of flow

example 1					
p1	0.02	r1	0.3	N1	14
p2	0.02	r2	0.3	N2	14
p3	0.02	r3	0.3	N3	14

example 2					
p1	0.03	r1	0.4	N1	12
p2	0.03	r2	0.4	N2	12
p3	0.03	r3	0.4	N3	12

example 3					
p1	0.03	r1	0.2	N1	18
p2	0.03	r2	0.2	N2	18
p3	0.03	r3	0.2	N3	18

example 4					
p1	0.04	r1	0.2	N1	8
p2	0.04	r2	0.2	N2	8
p3	0.04	r3	0.2	N3	8

example 5					
p1	0.07	r1	0.3	N1	20
p2	0.07	r2	0.3	N2	25
p3	0.07	r3	0.3	N3	27

example 6					
p1	0.01	r1	0.1	N1	10
p2	0.01	r2	0.1	N2	12
p3	0.01	r3	0.1	N3	15

example 7					
p1	0.07	r1	0.5	N1	16
p2	0.01	r2	0.3	N2	16
p3	0.06	r3	0.7	N3	16

example 8					
p1	0.01	r1	0.03	N1	70
p2	0.009	r2	0.06	N2	58
p3	0.005	r3	0.02	N3	45

Table 2: Data for the examples considered

population	analytical throughput	throughput simulation	% error
16	0.905282	0.90446	0.091%
17	0.906746	0.90539	0.150%
18	0.907755	0.90591	0.204%
19	0.908417	0.90671	0.188%
20	0.908793	0.90686	0.213%
21	0.908915	0.90693	0.219%
22	0.908793	0.90683	0.216%
23	0.908417	0.90658	0.203%
24	0.907755	0.90602	0.191%
25	0.906746	0.90519	0.172%
26	0.905282	0.90426	0.113%
population	analytical Avg B1	AvgB1 simulation	% error
16	5.41860	5.34880	1.305%
17	5.74971	5.67220	1.366%
18	6.06967	5.99120	1.310%
19	6.38293	6.33850	0.701%
20	6.69234	6.64800	0.667%
21	7.00000	6.98900	0.157%
22	7.30766	7.32370	0.219%
23	7.61707	7.67270	0.725%
24	7.93033	7.98900	0.734%
25	8.25029	8.33280	0.990%
26	8.58140	8.65770	0.881%
population	analytical Avg B2	AvgB2 simulation	% error
16	5.41860	5.34080	1.457%
17	5.74971	5.65770	1.626%
18	6.06967	6.00570	1.065%
19	6.38293	6.30790	1.189%
20	6.69234	6.65530	0.557%
21	7.00000	7.00530	0.076%
22	7.30766	7.33630	0.390%
23	7.61707	7.66250	0.593%
24	7.93033	8.01750	1.087%
25	8.25029	8.34600	1.147%
26	8.58140	8.68610	1.205%
population	analytical Avg B3	AvgB3 simulation	% error
16	5.41860	5.31020	2.041%
17	5.74971	5.66990	1.408%
18	6.06967	6.00300	1.111%
19	6.38293	6.35340	0.465%
20	6.69234	6.69660	0.064%
21	7.00000	7.00550	0.079%
22	7.30766	7.33980	0.438%
23	7.61707	7.66470	0.621%
24	7.93033	7.99330	0.788%
25	8.25029	8.32110	0.851%
26	8.58140	8.65610	0.863%

Table 3: Example 1: Numerical results for a symmetrical loop

population	analytical throughput	throughput simulation	% error
14	0.900133	0.898460	0.186%
15	0.901805	0.899690	0.235%
16	0.902835	0.900520	0.257%
17	0.903399	0.900800	0.289%
18	0.903580	0.901000	0.286%
19	0.903399	0.900840	0.284%
20	0.902835	0.900330	0.278%
21	0.901805	0.899460	0.261%
22	0.900133	0.898470	0.185%

population	analytical Avg B1	AvgB1 simulation	% error
14	4.759880	4.674100	1.835%
15	5.082600	4.983500	1.989%
16	5.393360	5.345000	0.905%
17	5.697990	5.643700	0.962%
18	6.000000	6.012400	0.206%
19	6.302010	6.327400	0.401%
20	6.606640	6.663500	0.853%
21	6.917400	7.008400	1.298%
22	7.240120	7.336200	1.310%

population	analytical Avg B2	AvgB2 simulation	% error
14	4.759880	4.661700	2.106%
15	5.082600	5.019200	1.263%
16	5.393360	5.339800	1.003%
17	5.697990	5.684400	0.239%
18	6.000000	5.999800	0.003%
19	6.302010	6.346600	0.703%
20	6.606640	6.672600	0.989%
21	6.917400	7.004200	1.239%
22	7.240120	7.322000	1.118%

population	analytical Avg B3	AvgB3 simulation	% error
14	4.759880	4.664000	2.056%
15	5.082600	4.997200	1.709%
16	5.393360	5.315100	1.472%
17	5.697990	5.671800	0.462%
18	6.000000	5.987700	0.205%
19	6.302010	6.325900	0.378%
20	6.606640	6.663700	0.856%
21	6.917400	6.987300	1.000%
22	7.240120	7.341600	1.382%

Table 4: Example 2: Numerical results for a symmetrical loop

population	analytical throughput	throughput simulation	% error
20	0.804356	0.804310	0.006%
21	0.806614	0.805470	0.142%
22	0.808347	0.806770	0.195%
23	0.809660	0.807580	0.258%
24	0.810624	0.808010	0.324%
25	0.811283	0.808590	0.333%
26	0.811669	0.809040	0.325%
27	0.811796	0.809120	0.331%

population	analytical Avg B1	AvgB1 simulation	% error
20	6.708880	6.694600	0.213%
21	7.056700	6.993400	0.905%
22	7.393370	7.305400	1.204%
23	7.722280	7.659800	0.816%
24	8.045820	7.970100	0.950%
25	8.365660	8.375700	0.120%
26	8.683300	8.684200	0.010%
27	9.000000	9.006800	0.075%

population	analytical Avg B2	AvgB2 simulation	% error
20	6.708880	6.678900	0.449%
21	7.056700	7.062300	0.079%
22	7.393370	7.348300	0.613%
23	7.722280	7.673600	0.634%
24	8.045820	8.063900	0.224%
25	8.365660	8.338000	0.332%
26	8.683300	8.647200	0.417%
27	9.000000	9.038600	0.427%

population	analytical Avg B3	AvgB3 simulation	% error
20	6.708880	6.626400	1.245%
21	7.056700	6.944200	1.620%
22	7.393370	7.346200	0.642%
23	7.722280	7.666500	0.728%
24	8.045820	7.965900	1.003%
25	8.365660	8.286200	0.959%
26	8.683300	8.668500	0.171%
27	9.000000	8.954500	0.508%

Table 5: Example 3: Numerical results for a symmetrical loop

population	analytical throughput	throughput simulation	% error
28	0.811669	0.80887	0.346%
29	0.811283	0.80857	0.336%
30	0.810624	0.808	0.325%
31	0.80966	0.80737	0.284%
32	0.808347	0.80655	0.223%
33	0.806614	0.80545	0.145%
34	0.804356	0.80428	0.009%

population	analytical Avg B1	AvgB1 simulation	% error
28	9.3167	9.3373	0.221%
29	9.63434	9.6444	0.104%
30	9.95418	9.9907	0.366%
31	10.2777	10.31	0.313%
32	10.6066	10.649	0.398%
33	10.9433	10.985	0.380%
34	11.2911	11.355	0.563%

population	analytical Avg B2	AvgB2 simulation	% error
28	9.3167	9.3701	0.570%
29	9.63434	9.7015	0.692%
30	9.95418	9.9685	0.144%
31	10.2777	10.324	0.448%
32	10.6066	10.679	0.678%
33	10.9433	11.064	1.091%
34	11.2911	11.335	0.387%

population	analytical Avg B3	AvgB3 simulation	% error
28	9.3167	9.2925	0.260%
29	9.63434	9.6539	0.203%
30	9.95418	10.04	0.855%
31	10.2777	10.364	0.833%
32	10.6066	10.67	0.594%
33	10.9433	10.95	0.061%
34	11.2911	11.308	0.149%

Table 6: Example 3: Numerical results for a symmetrical loop

population	analytical throughput	throughput simulation	% error
10	0.7190	0.7107	1.168%
11	0.7208	0.7119	1.249%
12	0.7214	0.7128	1.211%
13	0.7208	0.7121	1.228%
14	0.7190	0.7108	1.154%
population	analytical Avg B1	AvgB1 simulation	% error
10	3.3286	3.3315	0.088%
11	3.6659	3.6691	0.088%
12	4.0000	4.0005	0.012%
13	4.3341	4.3324	0.040%
14	4.6714	4.6595	0.256%
population	analytical Avg B2	AvgB2 simulation	% error
10	3.3286	3.3283	0.008%
11	3.6659	3.6639	0.053%
12	4.0000	3.9992	0.020%
13	4.3341	4.3386	0.103%
14	4.6714	4.6700	0.031%
population	analytical Avg B3	AvgB3 simulation	% error
10	3.3286	3.3400	0.342%
11	3.6659	3.6668	0.026%
12	4.0000	4.0002	0.005%
13	4.3341	4.3288	0.123%
14	4.6714	4.6704	0.022%

Table 7: Example 4: Numerical results for a symmetrical loop

population	analytical throughput	throughput simulation	% error
29	0.76846	0.76731	0.15%
30	0.76976	0.76823	0.20%
31	0.77074	0.76862	0.28%
32	0.77148	0.76913	0.31%
33	0.77202	0.76935	0.35%
34	0.77238	0.76979	0.34%

population	analytical Avg B1	AvgB1 simulation	% error
29	8.76392	8.64930	1.33%
30	9.00128	8.91360	0.98%
31	9.22403	9.09020	1.47%
32	9.43501	9.35410	0.86%
33	9.63619	9.56500	0.74%
34	9.82921	9.79480	0.35%
35	10.01560	9.96430	0.51%

population	analytical Avg B2	AvgB2 simulation	% error
29	11.09070	10.97300	1.07%
30	11.65050	11.41400	2.07%
31	12.17340	12.05200	1.01%
32	12.66930	12.52200	1.18%
33	13.14350	13.00900	1.03%
34	13.59880	13.45900	1.04%
35	14.03770	13.90700	0.94%

population	analytical Avg B3	AvgB3 simulation	% error
29	9.46171	9.37720	0.90%
30	9.70548	9.67220	0.34%
31	9.95236	9.85690	0.97%
32	10.20670	10.12300	0.83%
33	10.47170	10.42500	0.45%
34	10.74960	10.74600	0.03%
35	11.04260	11.12700	-0.76%

Table 8: Example 5: Numerical results for a loop with balanced machines and different buffer sizes

population	analytical throughput	throughput simulation	% error
36	0.77267	0.76972	0.383%
37	0.77260	0.76980	0.364%
38	0.77239	0.76985	0.330%
39	0.77203	0.76947	0.332%
40	0.77149	0.76891	0.336%
41	0.77075	0.76861	0.279%
42	0.76976	0.76791	0.241%
43	0.76843	0.76724	0.156%

population	analytical Avg B1	AvgB1 simulation	% error
36	10.19760	10.18100	0.163%
37	10.37570	10.39300	-0.166%
38	10.55190	10.60900	-0.538%
39	10.72760	10.74300	-0.143%
40	10.90410	10.99800	-0.854%
41	11.08310	11.20400	-1.079%
42	11.26590	11.37300	-0.942%
43	11.45450	11.52100	-0.577%

population	analytical Avg B2	AvgB2 simulation	% error
36	11.35290	11.44600	-0.813%
37	11.68070	11.78300	-0.868%
38	12.02770	12.17400	-1.202%
39	12.39530	12.63900	-1.928%
40	12.78460	12.96900	-1.422%
41	13.19890	13.34100	-1.065%
42	13.64150	13.81600	-1.263%
43	14.11970	14.22900	-0.768%

population	analytical Avg B3	AvgB3 simulation	% error
36	14.46210	14.37100	0.634%
37	14.87150	14.82200	0.334%
38	15.26790	15.21600	0.341%
39	15.65300	15.61600	0.237%
40	16.02930	16.03100	-0.011%
41	16.40110	16.45400	-0.322%
42	16.77480	16.80900	-0.203%
43	17.16130	17.24900	-0.508%

Table 9: Example 5: Numerical results for a loop with balanced machines and different buffer sizes

population	analytical throughput	throughput simulation	% error
17	0.822956	0.818910	0.4941%
18	0.823172	0.818740	0.5413%
19	0.823172	0.819400	0.4603%
20	0.822956	0.818580	0.5346%
population	analytical Avg B1	AvgB1 simulation	% error
17	4.989790	4.959900	0.6026%
18	5.294450	5.206400	1.6912%
19	5.597060	5.465700	2.4034%
20	5.898380	5.770000	2.2250%
population	analytical Avg B2	AvgB2 simulation	% error
17	4.683890	4.831000	3.0451%
18	5.016340	5.178300	3.1277%
19	5.351500	5.479200	2.3306%
20	5.690140	5.758000	1.1785%
population	analytical Avg B3	AvgB3 simulation	% error
17	7.270310	7.209000	0.8505%
18	7.672930	7.615100	0.7594%
19	8.073750	8.055000	0.2328%
20	8.473760	8.471900	0.0220%

Table 10: Example 6: Numerical results for a loop with balanced machines and different buffer sizes

population	analytical throughput	throughput simulation	% error
18	0.881964	0.875030	0.7924%
19	0.882456	0.875300	0.8175%
20	0.882752	0.875670	0.8088%
21	0.882932	0.875910	0.8017%
22	0.883036	0.875810	0.8251%
23	0.883087	0.875880	0.8228%
population	analytical Avg B1	AvgB1 simulation	% error
18	1.914760	1.923800	-0.4699%
19	1.936500	1.960400	-1.2191%
20	1.955920	1.991300	-1.7767%
21	1.977110	2.032200	-2.7109%
22	2.003710	2.082800	-3.7973%
23	2.039760	2.179000	-6.3901%
population	analytical Avg B2	AvgB2 simulation	% error
18	4.904660	4.762900	2.9763%
19	5.687940	5.502600	3.3682%
20	6.498960	6.309400	3.0044%
21	7.326210	7.128300	2.7764%
22	8.159880	7.950000	2.6400%
23	8.991230	8.811900	2.0351%
population	analytical Avg B3	AvgB3 simulation	% error
18	11.273900	11.313000	-0.3456%
19	11.559400	11.536000	0.2028%
20	11.763100	11.699000	0.5479%
21	11.907400	11.839000	0.5778%
22	12.008700	11.967000	0.3485%
23	12.079400	12.008000	0.5946%

Table 11: Example 7: Numerical results for a loop with unbalanced machines and equal buffer sizes

population	analytical throughput	throughput simulation	% error
24	0.88309	0.87585	0.827%
25	0.88306	0.87579	0.830%
26	0.88298	0.87578	0.822%
27	0.88283	0.87563	0.822%
28	0.88257	0.87546	0.812%
29	0.88210	0.87553	0.751%
30	0.88122	0.87528	0.678%

population	analytical Avg B1	AvgB1 simulation	% error
24	2.09062	2.26850	-7.841%
25	2.16415	2.40270	-9.928%
26	2.27270	2.57630	-11.784%
27	2.43708	2.75910	-11.671%
28	2.69469	3.06380	-12.047%
29	3.11665	3.44700	-9.584%
30	3.83966	3.93740	-2.482%

population	analytical Avg B2	AvgB2 simulation	% error
24	9.81202	9.64680	1.713%
25	10.61400	10.47100	1.366%
26	11.38810	11.25800	1.156%
27	12.12360	12.01400	0.912%
28	12.80680	12.71300	0.738%
29	13.41710	13.30400	0.850%
30	13.92140	13.78400	0.997%

population	analytical Avg B3	AvgB3 simulation	% error
24	12.12830	12.08400	0.367%
25	12.16180	12.12600	0.295%
26	12.18510	12.16500	0.165%
27	12.20190	12.22600	-0.197%
28	12.21590	12.22300	-0.058%
29	12.23140	12.24800	-0.136%
30	12.25570	12.27700	-0.173%

Table 12: Example 7: Numerical results for a loop with unbalanced machines and equal buffer sizes

population	analytical throughput	throughput simulation	% error
72	0.672239	0.671600	0.0952%
73	0.672457	0.671250	0.1799%
74	0.672657	0.672870	-0.0317%
75	0.672839	0.672170	0.0995%
76	0.673003	0.671410	0.2372%
77	0.673150	0.672520	0.0936%
78	0.673280	0.672850	0.0639%
79	0.673395	0.671760	0.2433%
80	0.673493	0.673340	0.0227%
81	0.673575	0.672570	0.1495%
population	analytical Avg B1	AvgB1 simulation	% error
72	15.710500	16.284600	-3.5254%
73	16.058200	16.653400	-3.5740%
74	16.410900	17.138400	-4.2449%
75	16.768700	17.542400	-4.4105%
76	17.131800	17.686100	-3.1341%
77	17.500200	18.052600	-3.0599%
78	17.874000	18.490700	-3.3352%
79	18.253200	18.990000	-3.8799%
80	18.638000	19.241000	-3.1339%
81	19.028300	19.592000	-2.8772%
population	analytical Avg B2	AvgB2 simulation	% error
72	30.107800	30.450800	-1.1264%
73	30.642700	31.013100	-1.1943%
74	31.176900	31.330600	-0.4906%
75	31.710400	31.883200	-0.5420%
76	32.242900	32.571400	-1.0086%
77	32.774300	33.108400	-1.0091%
78	33.304600	33.414000	-0.3274%
79	33.833500	33.921600	-0.2597%
80	34.361000	34.543000	-0.5269%
81	34.886800	34.938700	-0.1485%
population	analytical Avg B3	AvgB3 simulation	% error
72	25.268900	25.264400	0.0178%
73	25.411000	25.333400	0.3063%
74	25.547400	25.530900	0.0646%
75	25.678500	25.574300	0.4074%
76	25.804700	25.742300	0.2424%
77	25.926400	25.838800	0.3390%
78	26.043900	26.095100	-0.1962%
79	26.157500	26.088300	0.2653%
80	26.267700	26.215900	0.1976%
81	26.374600	26.469100	-0.3570%

Table 13: Example 8: Numerical results for a loop with unbalanced machines and different buffer sizes

population	analytical throughput	throughput simulation	% error
82	0.673643	0.669830	0.5692%
83	0.673695	0.672410	0.1912%
84	0.673732	0.672830	0.1340%
85	0.673754	0.670580	0.4733%
86	0.673761	0.673280	0.0714%
87	0.673753	0.674170	-0.0618%
88	0.673730	0.671450	0.3396%
89	0.673693	0.670630	0.4567%
90	0.673641	0.673640	0.0001%
91	0.673573	0.670870	0.4029%
population	analytical Avg B1	AvgB1 simulation	% error
82	19.424300	19.718400	-1.4915%
83	19.825900	20.314800	-2.4066%
84	20.233200	20.721500	-2.3565%
85	20.646300	21.150800	-2.3853%
86	21.065100	21.564700	-2.3167%
87	21.489700	22.004000	-2.3373%
88	21.920100	22.313400	-1.7626%
89	22.356400	22.581200	-0.9955%
90	22.798500	23.278300	-2.0611%
91	23.246600	23.613000	-1.5517%
population	analytical Avg B2	AvgB2 simulation	% error
82	35.411000	35.747300	-0.9408%
83	35.933300	36.116900	-0.5083%
84	36.453600	36.524500	-0.1941%
85	36.971900	37.090600	-0.3200%
86	37.487900	37.531600	-0.1164%
87	38.001600	38.146700	-0.3804%
88	38.512800	38.543100	-0.0786%
89	39.021500	39.270400	-0.6338%
90	39.527500	39.509700	0.0451%
91	40.030600	40.055500	-0.0622%
population	analytical Avg B3	AvgB3 simulation	% error
82	26.478700	26.534200	-0.2092%
83	26.580100	26.568100	0.0452%
84	26.679300	26.753900	-0.2788%
85	26.776300	26.758400	0.0669%
86	26.871600	26.903600	-0.1189%
87	26.965400	26.849200	0.4328%
88	27.057900	27.143300	-0.3146%
89	27.149300	27.148200	0.0041%
90	27.240000	27.211800	0.1036%
91	27.330100	27.331300	-0.0044%

Table 14: Example 8: Numerical results for a loop with unbalanced machines and different buffer sizes

population	analytical throughput	throughput simulation	% error
92	0.673490	0.671690	0.2680%
93	0.673392	0.672340	0.1565%
94	0.673279	0.671670	0.2395%
95	0.673149	0.672060	0.1621%
96	0.673003	0.674460	-0.2160%
97	0.672842	0.673540	-0.1036%
98	0.672664	0.672730	-0.0098%
99	0.672468	0.671890	0.0861%
100	0.672256	0.672410	-0.0229%
101	0.672025	0.670740	0.1916%
population	analytical Avg B1	AvgB1 simulation	% error
92	23.700500	23.985100	-1.1866%
93	24.160400	24.302100	-0.5831%
94	24.626300	24.905200	-1.1198%
95	25.098200	25.188700	-0.3593%
96	25.576200	25.829100	-0.9791%
97	26.060200	26.468100	-1.5411%
98	26.550500	26.654800	-0.3913%
99	27.046900	27.036000	0.0403%
100	27.549700	27.576600	-0.0975%
101	28.058800	27.755600	1.0924%
population	analytical Avg B2	AvgB2 simulation	% error
92	40.530900	40.550700	-0.0488%
93	41.028200	41.082200	-0.1314%
94	41.522300	41.530400	-0.0195%
95	42.013300	41.972300	0.0977%
96	42.501000	42.385200	0.2732%
97	42.985300	42.525900	1.0803%
98	43.466200	43.280400	0.4293%
99	43.943500	43.752900	0.4356%
100	44.417300	44.111300	0.6937%
101	44.887400	44.720800	0.3725%
population	analytical Avg B3	AvgB3 simulation	% error
92	27.419800	27.464100	-0.1613%
93	27.509500	27.615500	-0.3838%
94	27.599400	27.564200	0.1277%
95	27.689600	27.838900	-0.5363%
96	27.780400	27.785600	-0.0187%
97	27.872100	28.005900	-0.4778%
98	27.964800	28.064600	-0.3556%
99	28.058900	28.211000	-0.5392%
100	28.154600	28.312000	-0.5559%
101	28.252000	28.523400	-0.9515%

Table 15: Example 8: Numerical results for a loop with unbalanced machines and different buffer sizes