A SEQUENCE OF SERVERS WITH ARBITRARY INPUT AND REGULAR SERVICE TIMES REVISITED

Benjamin Avi-Itzhak and Hanoch Levy

RUTCOR, Rutgers Center for Operations Research
Rutgers University, Busch campus, New Brunswick, New Jersey 08903

The March 1965 issue of *Management Science* saw a sequence of two papers by Avi-Itzhak and Yadin, the first presenting analysis of a two server tandem system, with manufacturing blocking and no intermediate queue, leading to the second paper, which analyzes an arbitrary number of tandem servers with manufacturing blocking.

In this work we present generalized results for the same systems under k-stage blocking. One-stage blocking is the so-called manufacturing blocking in which a job entering the j-th service position in the sequence requires attendance by the j-th server only. Upon completion of service the job will move to position $(j+1)$ if that position is not occupied. Otherwise it will continue to stay in position j, blocking it, until the next position is vacated. Two-stage blocking is the so-called communication blocking in which a job entering the j-th position needs simultaneous attendance by the j-th and the $(j+1)$-th servers. Thus start of service may be delayed until the $(j+1)$-th server is freed. In k-stage blocking the job requires at each position in the sequence the joint attendance of the server of that position together with the servers of the next $k-1$ positions.

One interesting result is that for $k > 1$ the waiting times are not order-insensitive while the $G/D/1$ equivalence is maintained.

(*Tandem Queues, Manufacturing Blocking, Communication Blocking, Optimal Ordering, Reversibility.*)

In Memory of Micha Yadin

1 INTRODUCTION

Many processes of interest, encounterable over a wide spectrum of human activities, can be modeled as a queueing system with an ordered sequence of N service stations serving incoming jobs (or customers) which are processed in order of arrival through the sequence of stations. When intermediate queueing spaces (buffers) are limited, as is frequently the case, the system may experience blocking, resulting in slow down of the flow of jobs through the tandem arrangement of servers. It was shown by Avi-Itzhak and Yadin (1965) that, in the general case, all intermediate queue sizes may be assumed to equal to zero. Intermediate queueing positions can be then viewed as servers with zero service times. Using this approach we will assume in what follows that queueing is allowed only before the first server in the sequence where the number of waiting positions is unlimited, and that each server has one service position capable of holding one job (being served or blocked).

The two most commonly studied blocking mechanisms are manufacturing blocking and communication blocking (for a bibliography see Perros (1989)). In the manufacturing blocking scheme, processing in service position j in the sequence starts immediately upon entrance of a job. At the completion of service the job moves to the next service position providing that it is not occupied. If the next service position is occupied the job waits in position j, blocking it, until position $j+1$ is vacated. Under this scheme blocking is possible only after service completion. In the communication blocking scheme, processing at service position j can start only when the next position is empty. Here blocking is possible before the start of processing only. Once processing is completed there is an instantaneous transfer of the job to the next service position. In both schemes blocking in service position N, the last one in the sequence, is not possible. The manufacturing scheme is descriptive of manufacturing, or assembly line, type environments while the communication scheme is descriptive of messages transition where each node in the sequence is capable of buffering one message only, and therefore transition from node j to node $j+1$ requires an empty buffer at node $j+1$.

Some later publications of interest dealing with manufacturing and communication blocking are Kelly (1982), Massey (1993) and Avi-Itzhak and Halfin (1993). In recent years a modified manufacturing blocking scheme, called Kanban blocking, has been proposed and studied by Mitra and Mitrani (1990) (1991), Buzacott (1989), Zipkin (1989) and several others. (For the origin of this scheme see Monden (1977)). In essence, in this scheme, the j-th server can continue to process jobs and place them in its own buffer when the $(j+1)$-th buffer is fully occupied. When all buffers are of size 1, which is the assumed case in our study, the Kanban blocking is identical to the manufacturing blocking.

In this study we introduce the *k-stage blocking* scheme under which a job requires at each position in the sequence the joint attendance of the server of that position together with the servers of the next $k-1$ positions. A server of a given position is allowed to attend to a job residing in another position only if the server's own position is empty. Toward the end of the sequence, when there are fewer than $k-1$ remaining positions, the job requires the joint attendance of the servers of all the remaining positions. Clearly, one-stage blocking is the manufacturing blocking scheme in which a job entering the j-th service position requires attendance by the j-th server only. Two-stage blocking is the communication blocking scheme in which a job entering the j-th position needs simultaneous attendance by the j-th and $(j+1)$-th servers and thus start of processing is delayed until the $(j+1)$-th server is freed. Three-stage blocking may be representative of a radio communications tandem network where transition from node j to node $j+1$ requires no noise emission from node $j+2$. In general the k-stage blocking provides a unifying framework for modeling discrete flow (customers, jobs, items) through a sequential provider of service (processing). For low values of k the service provider is also "discrete" and is made of individual servers in tandem, such that the flow is "jumpy", i.e., the customers move from one server to the next in abrupt jumps. When the values of k are large the service provider becomes "continuous", like a continuous production process or a strech of a highway lane, and the flow moves more smoothly. Vehicular traffic on a one lane road (which is the continuous service provider) may serve as an example. Suppose a moving car occupies a length of 120 ft. (its own length plus the minimal safety margin). Selecting $k=10$ means that each service position is of length 12 feet and the car cannot progress unless 10 adjacent positions (a stretch of 120 ft. of the lane) are available. The car will advance by jumping 12 ft. at a time. As k is increased while the length of the service positions is decreased, the progression of the car becomes smoother. For details of this type of application of the k-stage blocking framework see Avi-Itzhak (1994).

The March 1965 issue of *Management Science* saw a progression of two papers by Avi-Itzhak and Yadin (1965) and Avi-Itzhak (1965), the first presenting analysis of a two server tandem system with manufacturing blocking and no intermediate queue, leading to the second paper which analyzes a sequence of N tandem servers with arbitrary input an regular service times under manufacturing blocking. In the present paper we extend the results and theorems of these two papers to the general k-stage blocking scheme.

Section 2 treats the case of $k+1$ tandem servers with arbitrarily distributed service times which operate under k-stage blocking. The expressions for the maximum equilibrium throughput rate of the system and the queueing time transform, when assuming Poisson input, are presented. The case of Poisson input and constant service times is analyzed in greater detail where the equilibrium waiting time is shown to behave as in an $M/D/1$ system.

In section 3 we analyze N tandem stations with arbitrary input and regular processing times operating under k-stage blocking. We show that, unlike the manufacturing blocking case, when $k \geq 2$ the system's performance may be affected by the order of the stations. Still the waiting times of jobs are the same as in a $G/D/1$ queue, but the service time length in this $G/D/1$ queue is determined by the order of the stations.

In section 4 we identify an optimal order of the stations for the case $k=2$ and show that the problem of identifying such an order when $k=3$ is already NP-complete.

2 $N = k+1$ SERVERS UNDER k-STAGE BLOCKING

Here we assume $k+1$ servers in tandem with no intermediate queue. Processing times of job number n are denoted by $\overline{S}^{(n)} = (S_1^{(n)}, S_2^{(n)}, \cdots, S_{k+1}^{(n)})$, where $S_j^{(n)}$ is it's required processing duration in position j.

a. Just-In-Time (JIT) Input

Suppose first that the input process is on a JIT basis, i.e., at the instant that job n moves from service position 1 to service position 2 job $n+1$ enters into position 1. One way to visualize JIT input is to assume that the waiting line before service position 1 is never empty. Under JIT input the measure of performance of main interest is the throughput rate of the system which in essence represents the system's capacity. We define the occupation time of position 1 by job n, denoted by $T^{(n)}$, as the time elapsing from start of service to job n in position 1 until the system is ready to start service to job $n+1$ in position 1. Then

$$T^{(n)} = \max\left(S_1^{(n)}, S_{k+1}^{(n-1)}\right) + S_2^{(n)} + S_3^{(n)} + \cdots + S_k^{(n)}, \qquad n \geq 1 \tag{1}$$

where $S_j^{(0)} = 0$, $j = 1, 2, \cdots, k+1$.

The occupation time, $T^{(n)}$, will serve as a surrogate for service time in position 1. Job n renders position 1 to be inaccessible to other jobs for a duration of length $T^{(n)}$. Note that the residence time of job n in position 1, denoted by $R^{(n)}$, is not equal to $T^{(n)}$. Rather

$$R^{(n)} = \max\left(S_1^{(n-1)}, S_{k+1}^{(n-2)}\right) + \left(S_1^{(n)} - S_1^{(n-1)}\right) + S_2^{(n-1)} + S_3^{(n-1)} + \cdots + S_k^{(n-1)}.$$

The throughput rate for the first n jobs, denoted by TP_n, is now defined as

$$TP_n = \frac{n}{T^{(1)} + T^{(2)} + \cdots + T^{(n)}}. \tag{2}$$

For the case where the processing requirements of all jobs are identically distributed, namely, $S_j^{(n)}$ is distributed as S_j, $\forall n \geq 1$, $j = 1, 2, \cdots, k+1$, the occupation times of all jobs, except for job 1, are identically distributed as T. From relation (1) we obtain

$$\begin{aligned} T &= \max\left(S_1, S_{k+1}\right) + S_2 + S_3 + \cdots + S_k \\ &= \sum_{j=1}^{k+1} S_j - \min\left(S_1, S_{k+1}\right) \\ &= \max\left(S_1 + S_2 + \cdots + S_k, S_2 + S_3 + \cdots + S_{k+1}\right), \end{aligned} \tag{3}$$

and the long-run (equilibrium) throughput rate is given as

$$TH = \frac{1}{E(T)}. \tag{4}$$

From (3) it is clear that reversing the order of the service positions will not affect the throughput process (for reversibility of tandem queues see Muth (1979), Dattatreya (1978) and Yamazaki and Sakasegawa (1975), (1985)).

From (3) it is also evident that placing the two "slowest" service positions at the two ends of the sequence will maximize the throughput of the system. For this purpose positions m and n are the "slowest" if $E(\min(S_m, S_n)) = \max_{i \neq j} E(\min(S_i, S_j))$, $i, j = 1, 2, \cdots, k+1$. If processing times are constants it is best to place the longest processing time position at one end, the second longest at the other end and the remaining $k-1$ positions in-between. This optimal ordering seems to re-enforce the *bowl phenomenon* which was observed in numerous studies (Hillier and Boling (1967), (1979), EI-Rayah (1979), Pinedo (1982), Huang and Weiss (1990), Ding and Greenberg (1991), Yamazaki, Sakasegawa and Shantikumar (1992) and others). The bowl phenomenon is exhibited in situations where the optimal order has "worse" stations at the two ends and "better" ones in-between. Actually, as will become evident in section 4, our system exhibits the *saw teeth phenomenon* rather than the bowl phenomenon. In the saw teeth phenomenon an optimal order is obtained by alternating between slow and fast service positions (Patterson (1964)).

b. Poisson Input

Suppose the jobs arrive in a Poisson stream with rate λ, service times $\overline{S}^{(n)}$, $n = 1, 2, \cdots$, are *i.i.d.* where $S_j^{(n)}$ is distributed as S_j, with given distribution $F_j(\cdot)$, $\forall n \geq 1$ and $j = 1, 2, \cdots, k+1$, and $\lambda E(T) < 1$ for T as defined in (3). We define service position 1 as idle whenever it is ready to provide service but the waiting line is empty, and as busy otherwise. A busy period is initiated by an arriving job whose processing in position 1 starts upon its arrival. The busy period then proceeds until the first point in time where position 1 becomes idle, e.g., the busy period ends at the moment that the only job in the system is entering position $k+1$. Clearly, in this sub-system, "service times" of jobs arriving during a busy period are distributed as T and "service time" of a job initiating a busy period is distributed as T_0, where

$$T_0 = \begin{cases} S_1 + S_2 + \cdots + S_k & \text{w. p.} \quad \alpha = \int_0^\infty e^{-\lambda t} dF_{k+1}(t), \\ \max\left(S_1,\, S^*_{k+1}\right) + S_2 + S_3 + \cdots + S_k & \text{w. p.} \quad 1 - \alpha, \end{cases} \tag{5}$$

and the distribution of S^*_{k+1} is given by

$$F^*_{(k+1)}(t) = \int_t^\infty \left(1 - e^{-\lambda(x-t)}\right) dF_{k+1}(x). \tag{6}$$

Note that the service time, T_0, of a job who initiates a busy period and the preceding idle period are not independent. In fact, if the arrival process is not Poisson, T_0 and the preceding busy period will be dependent through the remaining interarrival time at the end of the preceding busy period. The form of T, though, will remain the same as in equation (3) for any arbitrary process of arrivals.

Let Q_1 be the equilibrium queueing time (from arrival to departure) of a job in the subsystem and let Q be its queueing time in the actual tandem system, then

$$Q = Q_1 + S_{k+1}. \tag{7}$$

For simplicity assume that $S_1, S_2, \cdots, S_{k+1}$ are mutually independent resulting in independence of Q_1 and S_{k+1}. (Note that independence of S_{k+1} and $\sum_{j=1}^{k} S_j$ will suffice). Then

$$\mathcal{L}_Q(u) = \mathcal{L}_{Q_1}(u)\mathcal{L}_{S_{k+1}}(u), \qquad u \geq 0 \tag{8}$$

where we denote the Laplace transform of a random variable, say Y, by $\mathcal{L}_Y(u) = E\left(e^{-uY}\right)$, $u \geq 0$.

The Laplace transform of the equilibrium queueing time in an $M/G/1$ type system identical to our sub-system is of a known form, (see Avi-Itzhak and Yadin (1965), Welch (1964)). Substituting this form in (7) yields

$$\mathcal{L}_Q(u) = \frac{1 - \lambda E(T)}{1 - \lambda E(T) + \lambda E(T_0)} \, \frac{(\lambda - u)\mathcal{L}_{T_0}(u) - \lambda \mathcal{L}_T(u)}{\lambda - u - \lambda \mathcal{L}_T(u)} \mathcal{L}_{S_{k+1}}(u). \tag{9}$$

Moments of the equilibrium queueing time, and in some cases its explicit distribution, are readily obtainable from (9). The generating function of the equilibrium queue size (number-in-the-system) is also easily obtainable from (9) through the argument that the queue size at a departure point is the number of Poisson arrivals during a queueing time.

Special Case: Regular (Constant) Processing times

Let the service times at all positions be constants given by the non-negative numbers $s_1, s_2, \cdots$, s_{k+1} and define $s = \max(s_1 + s_2 + \cdots + s_k, s_2 + s_3 + \cdots + s_{k+1})$. Then $T = s$ and

$$\mathcal{L}_{T_0}(u) = \frac{\lambda e^{-u \max(s_{k+1} - s_1, 0)} - u e^{-\lambda \max(s_{k+1} - s_1, 0)}}{\lambda - u} \; e^{-u \sum_{j=1}^{k} s_j}. \tag{10}$$

Substitution in (9) yields

$$\mathcal{L}_Q(u) = (1 - \lambda s) \frac{u e^{-u(s_1 + s_2 + \cdots + s_{k+1})}}{u - \lambda + \lambda e^{-us}}. \tag{11}$$

When $k = 1$, the manufacturing blocking case, equation (11) reduces to the result obtained by Avi-Itzhak and Yadin (1965). When $k = 2$, the communication blocking case, we have from (11)

$$\mathcal{L}_Q(u) = (1 - \lambda \max(s_1 + s_2,\, s_2 + s_3)) \frac{u e^{-u(s_1 + s_2 + s_3)}}{u - \lambda + \lambda e^{-u \max(s_1 + s_2, s_2 + s_3)}} \tag{12}$$

Setting $s_2 = 0$ in this result, we again obtain the manufacturing blocking case, $k = 1$. This means that two positions with one intermediate waiting space under communication blocking behave the same as two positions with no intermediate queue under manufacturing blocking.

The equilibrium waiting time, W, of a job is defined as the time it spends in the system waiting or blocked. by this definition

$$Q = W + s_1 + s_2 + \cdots + s_{k+1}. \tag{13}$$

¿From (13) and (11) we obtain

$$\mathcal{L}_W(u) = \frac{(1-\lambda s)u}{u - \lambda - \lambda e^{-us}} \tag{14}$$

which we immediately recognize as being the L.T. of the waiting time in an $M/D/1$ queue with arrival rate λ and service length s. We conclude then that the waiting time, in the case of constant processing times $s_1, s_2, \cdots, s_{k+1}$, is identical in distribution to the waiting time in an $M/D/1$ queue with the same Poisson rate of arrivals and service of length

$$s = \max(s_1 + s_2 + \cdots + s_k, s_2 + s_3 + \cdots + s_{k+1}).$$

This property extends to the more general system with arbitrary input and any number of service positions $N \geq k+1$, as is shown in the next section. It holds for the n-th arriving job, $n = 1, 2, \cdots$, and does not require equilibrium assumptions.

3 N STATIONS WITH ARBITRARY INPUT AND REGULAR PROCESSING TIMES

Suppose now that we have a sequence of N tandem single server positions with constant processing times, $s_1, s_2, \cdots, s_N$, and some arbitrary arrival process, $\{A^{(n)}, n = 1, 2, \cdots\}$, operating under the k-stage blocking scheme, where $k < N$.

For a given realization (sample path) of the arrival process, say $\{a^{(n)}\}$ where $a^{(n)}$ is the arrival time of job number n, we denote the departure time of job n from position N by $d^{(n)}$, $n \geq 1$, and define

$$q^{(n)} = d^{(n)} - a^{(n)} = w^{(n)} + s_1 + s_2 + \cdots + s_N, \qquad n \geq 1, \tag{15}$$

where $q^{(n)}$ is the queueing time of job n and $w^{(n)}$ is its waiting time in line and in blocking.

The basic theorem that we would like to prove in this section is the following:

Theorem 1 *Let*

$$s = \max_{1 \leq j \leq N-k+1} \{s_j + s_{j+1} + \cdots + s_{j+k-1}\} \tag{16}$$

then for any given realization of arrivals, say $\{a^{(n)}\}$, the waiting time of job n, $w^{(n)}$, is identical to the waiting time of arrival number n in a FIFO single server queue, $G/D/1$, with constant service time s and arrival process realization $\{a^{(n)}\}$, $n \geq 1$.

The *bottleneck* of the system is located with the k adjacent stations whose service times sum to s, in the sense that the inter-departure times in this system cannot be shorter than s,

$$d^{(n)} - d^{(n-1)} \geq s, \qquad n \geq 2. \tag{17}$$

Theorem 1 states that the queueing times in this system are determined solely by the size of the bottleneck and are insensitive to its location. The proof is provided through a sequence of propositions.

Proposition 1 *If* $a^{(i)} - a^{(i-1)} \geq s,\ i = 2, 3, \cdots, n$ *then* $w^{(i)} = 0,\ i = 1, 2, \cdots, n$. *(If all interarrival times are greater than* s *there is no waiting in this system).*

Proof: Assume then that $a^{(i)} - a^{(i-1)} \geq s,\ i = 2, 3, \cdots, n$ and for simplicity assume also that the system is empty at time zero and therefore $w^{(1)} = 0$.

Suppose $w^{(n-1)} = 0$ and let t_j denote the cumulative amount of service job $(n-1)$ could have received from its time of arrival up to time $a^{(n)} + s_0 + s_1 + \cdots + s_j$, $j = 0, 1, 2, \cdots, N-k$, where $s_0 = 0$. We have

$$t_0 = a^{(n)} - a^{(n-1)} \geq s \geq s_1 + s_2 + \cdots + s_k, \tag{18}$$

and therefore at time $a^{(n)}$ the first k servers are free to start serving job n upon its arrival. Also

$$t_j = s_1 + s_2 + \cdots + s_j + t_0 \geq s_1 + s_2 + \cdots + s_j + s \geq s_1 + s_2 + \cdots s_j + (s_{j+1} + \cdots + s_{j+k}), \tag{19}$$

$j = 1, 2, \cdots, N-k$, meaning that servers $j+1, j+2, \cdots, j+k$ are free at time $a^{(n)} + s_1 + s_2 + \cdots + s_j$ and job n can start its processing at position $j+1$ immediately upon entry to that position . Since $w^{(1)} = 0$ it follows that $w^{(i)} = 0,\ i = 2, 3, \cdots, n$, which completes the proof. □

Two immediate results of proposition 1 are:

Corollary 1 *No blocking is possible in service positions located after the bottleneck.*

The departure process from the bottleneck is the arrival process to the remainder of the tandem sequence of servers. However, the inter-departure times from the bottleneck cannot be shorter than s which satisfies the conditions of proposition 1 regarding the remainder of the sequence. The reader can also easily verify that blocking is not possible in the service positions of the bottleneck itself, except for the first position of the bottleneck. This means that all the waiting in this system is before start of service of the jobs in the first service position of the bottleneck.

Corollary 2 *The maximal throughput rate of the system is equal to* $1/s$.

Interdeparture times from this system cannot be shorter than s and therefore the throughput rate cannot exceed $1/s$. If we place on the system an arrival process with constant interarrival times equaling s we have, from proposition 1, constant inter-departure times equaling s and a throughput rate $1/s$. It can be readily verified that this will also be the throughput rate under JIT loading.

Proposition 2 *Let $\Delta_i \geq s_{i-1} + s_{i-2} + \cdots + s_{i-k}$ be the difference between the total cumulative amounts of processing already given to jobs n and $n-1$ at the time that job $n-1$ moves into position i, $i = k, k+1, \cdots, N$. Then*

$$\Delta_{i+1} \leq \max(\Delta_i,\, s_i + s_{i-1} + \cdots + s_{i-k+1}). \tag{20}$$

Proof: Denote the blocking time of job $n-1$ in position i by $b_i^{(n-1)}$. Then, from the dynamics of the process we have

$$\Delta_{i+1} = \begin{cases} s_i + s_{i-1} + \cdots + s_{i-k+1} & \text{if} \quad \Delta_i - b_i^{(n-1)} < \sum_{r=0}^{k-1} s_{i-r} \\ \\ \Delta_i - b_i^{(n-1)} & \text{otherwise,} \end{cases} \tag{21}$$

which can be re-written as

$$\Delta_{i+1} = \max(\Delta_i - b_i^{(n-1)},\, s_i + s_{i-1} + \cdots + s_{i-k+1}).$$

Substituting zero for $b_i^{(n-1)}$ yields relation (20). □

In particular if $b_i^{(n-1)} = 0$, i.e., job $n-1$ is not blocked in position i, then

$$\Delta_{i+1} = \max(\Delta_i,\, s_i + s_{i-1} + \cdots + s_{i-k+1}).$$

Proposition 3

$$d^{(n)} = \max(d^{(n-1)} + s,\, a^{(n)} + s_1 + s_2 + \cdots + s_N), \qquad n \geq 2. \tag{22}$$

Proof: From equation (15) we have

$$d^{(n)} \geq a^{(n)} + s_1 + s_2 + \cdots + s_N, \qquad n \geq 1, \tag{23}$$

and from corollary 1 we have

$$d^{(n)} \geq d^{(n-1)} + s, \tag{24}$$

which we combine into

$$d^{(n)} \geq \max(d^{(n-1)} + s,\, a^{(n)} + s_1 + s_2 + \cdots + s_N), \qquad n \geq 2. \tag{25}$$

¿From equation (15) we also have

$$d^{(n)} = a^{(n)} + s_1 + s_2 + \cdots + s_N \qquad \text{if} \quad w^{(n)} = 0. \tag{26}$$

To prove the proposition it is sufficient to show that

$$d^{(n)} = d^{(n-1)} + s \qquad \text{if} \quad w^{(n)} > 0. \tag{27}$$

Suppose then that $w^{(n)} > 0$ and that the bottleneck is located at positions $m, m+1, \cdots, m+k-1$, i.e.,

$$s = s_m + s_{m+1} + \cdots + s_{m+k-1}. \tag{28}$$

Since $w^{(n)} > 0$ then job n must experience delay in the waiting line and/or while blocked. Say job n experiences delay at position j, $j = 1, 2, \cdots, m-1$,where delay in the waiting line counts as delay at position 1. Then

$$\Delta_{j+k} = s_j + s_{j+1} + \cdots + s_{j+k-1} \leq s. \tag{29}$$

¿From proposition 2 we have that

$$\Delta_{i+1} \leq \max(\Delta_i,\, s_i + s_{i+1} + \cdots + s_{i-k+1}).$$

But because of (29) and the definition of s we have

$$\Delta_{i+1} \leq s, \qquad j+k \leq i \leq m+k-2, \tag{30}$$

and in particular

$$\Delta_{m+k-1} \leq s. \tag{31}$$

¿From proposition 1 we know that no blocking is possible in position $m+k$, namely $b_{m+k}^{(n-1)} = 0$, and therefore, from proposition 2 we have

$$\Delta_{m+k} = \max(\Delta_{m+k-1},\, s) = s. \tag{32}$$

Since no blocking is possible from position m on (proposition 1) we have

$$d^{(n)} - d^{(n-1)} = s,$$

which completes the proof. □

Since $d^{(1)} = s_1 + s_2 + \cdots + s_N$ we see, from (22), that the departure time of job n is the same for all orderings of the service positions in which the value of s is the same. From the definition of $w^{(n)}$, equation (15), it immediately follows that the same holds for $w^{(n)}$ too.

Corollary 3 (Simple reversibility): *Reversing the order of the servers does not affect $d^{(n)}$ and $w^{(n)}$, $n \geq 1$. This is true since reversing the order does not change the value of s.*

Proposition 4 (Equivalence): *Suppose the bottleneck is located at positions $m, m+1, \cdots, m+k-1$ then introducing unlimited waiting space between positions $m+k-1$ and $m+k$ and then reversing the order of the first $m+k-1$ positions does not affect $d^{(n)}$ and $w^{(n)}$, $n \geq 1$.*

Proof: From corollary 1 we know that no blocking is possible after the bottleneck, i.e., in positions $m+k, m+k+1, \cdots, N$.

From corollary 3 we know that reversing the order of positions $1, 2, \cdots, m+k-1$ will not affect the departure times from this subsystem of $m+k-1$ servers. However, these departure times are the unchanged arrival times to the second subsystem consisting of positions $m+k, m+k+1, \cdots, N$, which completes the proof. $\square$

Proof of Theorem 1: Using proposition 4 we reverse the order of the first $m+k-1$ servers and add unlimited waiting space before position $m+k$. This does not affect $w^{(n)}$. In the new, equivalent system, the first k servers, whose service times sum to s, are the bottleneck of the first subsystem. From corollary 1 we know that waiting in this subsystem is possible only before the start the first service. Also, the first position requires s time units between two consecutive service starts. This completes the proof of the theorem. $\square$

Corollary 4 *A k-stage blocking scheme can be reduced in its behavior to a k'-stage scheme, $k' < k$, by insertion of intermediate buffers (service positions of zero processing time).*

For example, communication blocking (2-stage) is reduced to manufacturing blocking (1-stage) by adding one waiting position in between any two adjacent service positions. Doing the same in a k-stage scheme, $k > 2$, will result in a $\frac{k}{2}$-stage behavior if k is even or a $\frac{k+1}{2}$-stage behavior if k is odd.

Clearly, theorem 1 does not seem to support the *bowl phenomenon* theory. Rather it re-enforces the *saw teeth phenomenon* theory, at least for communication blocking, where alternating long and short services seems to minimize the waiting times.

The reader will note that the notion of passive intermediate buffer spaces is superfluous to the fundamentals of the k-stage blocking framework where each server, also those stationed at zero-service-time positions, must actively participate in the service at other stations and are allowed to do so only when their zero-service-time (ZST) position is empty. Suppose one makes the distinction between ZST positions and non-ZST ones and requires at each non-ZST position the joint attendance of the server of that position together with the servers of the next $k-1$ non-ZST positions while at ZST positions no attendance of servers is required (these are passive buffer positions). Also assume that any order of customer serving is allowed. This defines a class G of infinitely many service schemes, only one of which is the k-stage blocking in which a server can attend to another station only if its buffer is empty. (If the k-stage blocking scheme is implemented, the ZST positions, namely the intermediate buffers, are kept empty all the times and are superfluous.)

Proposition 5 *Under JIT input the throughput rate of the k-stage blocking scheme is at least as high as any other scheme in G.*

Proof: Let $TH(g)$ denote the throughput rate of scheme $g \in G$. Since the server of non-ZST position $i+k-1$ spends $s_i + s_{i+1} + \cdots + s_{i+k-1}$ time units to serve one customer, the throughput rate is bounded from above by

$$TH(g) \leq \frac{1}{\max(s_i + s_{i+1} + \cdots + s_{i+k-1})}, \qquad i = 1, 2, \cdots, N-k+1, \ \forall g \in G.$$

But from Corollary 2 we know that this upper bound is reached for the k-stage blocking scheme. □

Thus, no scheme $g \in G$ is superior to the k-stage blocking scheme and addition of "passive" intermediate buffers is not going to improve the performance of the system. The investigation of the optimal performance under arbitrary input is left to the reader.

4 OPTIMAL ORDERING

A particular ordering of the servers is defined here as optimal if it minimizes the waiting times of the jobs or, in the case of JIT input, if it minimizes the departure times of the jobs. Assume we have n servers with service times $s_1 \geq s_2 \geq \cdots \geq s_N$ which we identify by their indices $i = 1, 2, \cdots, N$. Let J be the set of all $N!$ possible orders and for each order $j \in J$, represented by the indices $(j_1, j_2, \cdots, j_N)$, and a positive integer $k < N$ define

$$c_j = \max_{1 \leq i \leq N-k+1} \left(s_{j_i} + s_{j_{i+1}} + \cdots + s_{j_{i+k-1}} \right). \tag{33}$$

Then, an order, say $r \in J$, is optimal if

$$c_r = \min_{j \in J} c_j. \tag{34}$$

a. The case of $k = 2$

Assume first, for simplicity of notation, that N is even and let

$$j_* = (1, N, 2, N-1, \cdots, \frac{N}{2}, \frac{N}{2}+1), \tag{35}$$

and

$$c^* = c_{j_*} = \max_{i=1,2,\cdots,\frac{N}{2}} (s_i + s_{(i)}) = s_{i_*} + s_{(i_*)}, \tag{36}$$

where

$$(i) = N - i + 1, \qquad i = 1, 2, \cdots, \frac{N}{2}.$$

Theorem 2 *j_* is an optimal order: $c_{j_*} = \min_{j \in J} c_j$.*

Proof: Suppose there exists an order $j \in J$ for which $c_j < c_{j*}$, and consider the set $B = \{1, 2, \cdots, i_*\}$. Since $s_1 \geq s_2 \geq \cdots \geq s_{i_*}$ and $c_j < c_{j*}$ the elements of B cannot be adjacent to each other in j. Let A then be the set of all the elements adjacent (in j) to the elements of B. Clearly $|A| \geq |B|$, namely, the number of elements in A cannot be smaller than the number of elements in B. On the other hand since $s_b \geq s_{i_*}$, $\forall b \in B$ then $s_a < s_{(i_*)}$, $\forall a \in A$ (to allow $c_j < c_{j_*}$) and therefore $A \subset \{(1), (2), \cdots, (i_* - 1)\}$ resulting in $|A| \leq i_* - 1 < i_* = |B|$, which contradicts $|A| \geq |B|$. □

If N is not even we add a dummy position with $s_{N+1} = 0$ and redefine

$$(i) = N - i + 1, \quad i = 1, 2, \cdots, \frac{N-1}{2} \qquad \text{and} \qquad \left(\frac{N+1}{2}\right) = N + 1.$$

The dummy position is always placed as last in the sequence so that J remains virtually unchanged. The proof of theorem 2, with some notation adjustments, immediately holds for this case too.

Theorem 2 clearly supports the *saw teeth* ordering. It is not difficult to show that j_* may be the only optimal ordering and in such a case the *saw teeth* ordering is the only optimal one.

b. The Case of $k = 3$.

The optimal ordering problem in the case of $k = 3$ is a hard (NP-complete) problem. To show this we will use the following theorem due to M. Troyon (see Mahader and Peled Chapter 7 (1993)).

Theorem (M. Troyon): *Monotone 3-Partition is NP-Complete.*

The Monotone 3-Partition Problem, which we will call $MON3P$, is the following :

Given a set U of $3M$ non-negative integers

$$U = \{u_1, u_2, \cdots, u_{3M}\}$$

define

$$T = \frac{1}{M} \sum_{i=1}^{3M} u_i. \tag{37}$$

Question: Can U be partitioned into M ordered triplets $U_i = (\alpha_i, \beta_i, \gamma_i)$, $i = 1, 2, \cdots, M$ such that

(i) $\alpha_i + \beta_i + \gamma_i = T$, $i = 1, 2, \cdots, M$,

(ii) $\alpha_i \leq \alpha_{i+1}$, $i = 1, 2, \cdots, M - 1$,

(iii) $\alpha_i + \beta_i \leq \alpha_{i+1} + \beta_{i+1}$, $i = 1, 2, \cdots, M - 1$? (38)

Our problem, which we will call 3-LS (3-Linear Scheduling), is stated as follows:

Given a set S of $N \geq 3$ non-negative integers

$$S = \{s_1, s_2, \cdots, s_N\}$$

and a positive number B.

Question: Is there a linear ordering of the elements of S such that the sum of any 3 consecutive ones is at most B?

Theorem 3 *3-LS is NP-complete.*

Proof: (by reduction from $MON3P$) Suppose $(\alpha_i, \beta_i, \gamma_i)$, $i = 1, 2, \cdots, M$ is a solution of $MON3P$. Take the elements of this solution in the order

$$\alpha_M \beta_M \gamma_M \, \alpha_{M-1} \beta_{M-1} \gamma_{M-1} \, \cdots \, \alpha_1 \beta_1 \gamma_1. \tag{39}$$

¿From conditions (i), (ii) and (iii) of (38) it can be shown that the sum of any three consecutive numbers in this order does not exceed T. Clearly then, this is also a solution of the instance of 3-LS where $N = 3M$, $s_i = t_i$, $i = 1, 2, \cdots, N$ and $B = T$.

On the other hand any solution to the above defined instance of 3-LS induces a solution of $MON3P$. Suppose the order (39) is a solution of the 3-LS instance then it is easily verifiable that the triplets $U_i = (\alpha_i, \beta_i, \gamma_i)$, $i = 1, 2, \cdots, M$, satisfy the three conditions of (38). $\square$

This type of reduction from the Monotone 3-Partition problem can be further pursued to show NP-completeness for $k > 3$. The argument follows the same lines as the one provided by Mahader and Peled (1993) in treating the cyclic ordering problem.

5 DISCUSSION

The property of ordering-invariance, as demonstrated in the case of $k = 1$, does not carry over to the more general case of $k > 1$. From proposition 3 we know that the set J of the $N!$ possible orders can be partitioned into a finite number of sub-sets so as to maintain waiting times invariance within each of these sub-sets. On the other hand the $G/D/1$ equivalence holds true for all $k \geq 1$. Clearly, most of the results of this paper will hold, also at least as a good approximation, for the case of non-overlapping, arbitrarily distributed service times in place of constant service times.

References

[1] Avi-Itzhak, B. and M. Yadin, "A Sequence of Two Servers With No Intermediate Queue," *Management Sci.,* 11, 5 (1965), 553–564.

[2] Avi-Itzhak, B., "A Sequence of Service Stations with Arbitrary Input and Regular Service Times," *Management Sci.,* 11, 5 (1965), 565–571.

[3] Avi-Itzhak, B. and S. Halfin, "Servers in Tandem with Communications and Manufacturing Blocking," *Appl. Prob.,* **30,** (1993), 429–437.

[4] Avi-Itzhak, B., "Servers in Tandem with k-Stage Blocking and Communications Type Flow," *Appl. Prob.,* December, 1994.

[5] Buzacott, J., "Queueing Models of Kanban and MRP Controlled Production Systems,", *Engin. Costs Prod. Econ.* 17, (1989), 3–20.

[6] Dattatreya, E. S., *Tandem Queueing Systems with Blocking,* Ph. D. Dissertation, University of California, Berkeley, 1978.

[7] Ding, J. and B. S. Greenberg, "Bowl Shapes are Better with Buffers-Sometimes," *Prob. in the Eng. and Info. Sciences,* 5 (1991), 159–169.

[8] El-Rayah, T. E., "The Efficiency of Balanced and Unbalanced Production Lines," *Int. J. Prod. Res.,* 17 (1979), 61–75.

[9] Hillier, F. S. and R. W. Boling, "Finite Queues in Series with Exponential or Erlang Service Times – A Numerical Approach," *Oper. Res.,* 15 (1967), 286–303.

[10] Hillier, F. S. and R. W. Boling, "On the Optimal Allocation of Work in Symmetrically Unbalanced Production Line Systems with Variable Operation Times," *Management Sci.,* 25 (1979), 721–728.

[11] Huang, C. C. and G. Weiss, "On the Optimal Order of M Machines in Tandem," *Oper. Res. Letters,* 9 (1990), 299–303.

[12] Kelly, F. P., "The Throughput of a Service of Buffers," *Adv. Appl. Prob.,* 14 (1982), 633-653.

[13] Mahader, N. V. R. and V. N. Peled, *Threshold Graphs and Related Topics.* To be published by North Holland.

[14] Massey, W. A., "Balanced Queues in Series with Communication Blocking," *Math. of Oper. Res.* (1993). (To appear).

[15] Mitra, D. and I. Mitrani, "Analysis of a Kanban Discipline for Cell Coordination in Production Lines I," *Management Sci.,* 36, 12 (1990), 1548–1566.

[16] Mitra, D. and I. Mitrani, "Analysis of a Kanban Discipline for Cell Coordination in Production Lines II: Stochastic Demands," *Oper. Res.,* 39, 5 (1991), 807–823.

[17] Monden, Y., *Toyota Production System,* Industrial Engineering and Management Press, Norcross, GA, 1983.

[18] Muth, E. J., "The Reversibility Property of Production Lines," *Management Sci.,* 25 (1979), 152–158.

[19] Patterson, R. L., "Markov Processes Occurring in the Theory of Traffic Flow through an n-stage Stochastic Service System," *J. Ind. Engineering,* 15 (1964), 188–193.

[20] Perros, H. G., "A Bibliography of Papers on Queueing Networks with Finite Capacity Queues," *Performance Evaluation,* **10,** (1989), 255–260.

[21] Welch, P. D., "On a Generalized $M/G/1$ Queueing Process in Which the First Customer of Each busy Period Receives Exceptional Service," *Oper. Res.,* 12 (1964), 736–752.

[22] Whitt, W., "the Best Order for Queues in Series," *Management Sci.,* 31 (1985), 475–487.

[23] Yamazaki, G. and H. Sakasegawa, "Properties of Duality in Queueing Systems," *Ann. Instit. Stat. Math.,* 27 (1975), 201–212.

[24] Yamazaki, G., T. Kawashima and H. Sakasegawa, "Reversibility of Tandem Blocking Queueing Systems," *Management Sci.,* 31 (1985), 78–83.

[25] Yamazaki, G., H. Sakasegawa and J. G. Shanthikumar, "On Optimal Arrangement of Stations in a Tandem Queueing System with Blocking," *Management Sci.,* 38 (1992), 137–153.

[26] Zipkin, P. "A Kanban-like Production Control System: Analysis of Simple Model," *Research working paper* 89-1, Graduate School of Business, Columbia University, (1989).