Deadlock Properties of Queueing Networks with Finite Capacities and Multiple Routing Chains [*]

Jörg Liebeherr [†]   *Ian F. Akyildiz* [‡]

[†] Department of Computer Science
University of Virginia
Charlottesville, VA 22903

[‡] School of Electrical and Computer Engineering
Georgia Institute of Technology
Atlanta, GA 30332

Abstract

Blocking in queueing network models with finite capacities can lead to deadlock situations. In this paper, deadlock properties are investigated in queueing networks with multiple routing chains. The necessary and sufficient conditions for deadlock-free queueing networks with blocking are provided. An optimization algorithm is presented for finding deadlock-free capacity assignments with the least total capacity. The optimization algorithm maps the queueing network into a directed graph and obtains the deadlock freedom conditions from a specified subset of cycles in the directed graph. In certain network topologies, the number of deadlock freedom conditions can be large, thus, making any optimization computationally expensive. For a special class of topologies, so-called *tandem networks*, it is shown that a minimal capacity assignment can be directly obtained without running an optimization algorithm. Here, the solution to the minimal capacity assignment takes advantage of the regular topology of tandem networks.

[*] This work was supported by the National Science Foundation under Grant No. CCR-90-11981.

1 Introduction

Queueing network models are frequently applied for performance evaluation of computer systems and communication networks. Numerous methods are available for analyzing queueing networks under the assumption that all stations have infinite capacities. However, in actual systems the resources have finite capacities, and queueing networks with finite capacities should be used for performance analysis.

In a queueing network with finite capacities, each station has only a finite waiting room for buffering jobs. Blocking arises due to the limitations imposed by the capacity of these stations. In particular, blocking occurs when the flow of jobs through one station is interrupted due to another station that has reached its full capacity. The set of rules that dictate when a station becomes blocked or unblocked is commonly referred as the blocking mechanism. In this work, we consider the so-called *blocking-after-service* or BAS mechanism, also referred to as Type 1 or manufacturing blocking mechanism [1]. In BAS, a job which has completed service at a station i and attempts to proceed to station j must find an empty buffer space in station j. If station j is full, the job is blocked and forced to wait in station i's server until it can enter destination station j. A server which contains a blocked job cannot serve other jobs waiting in the queue.

Finite station capacities and blocking can lead to a deadlock situation in the queueing network. As an example, suppose a job has finished service at a station, say station 1, and wants to proceed to some other station, say station 2. If the waiting room of station 2 is full, the job is blocked in the server of station 1. Suppose another job has finished service at station 2 and has selected station 1 as its next station. If station 1's waiting room is also full, this job is blocked at station 1. In this situation, the jobs in the servers of both stations 1 and 2 are permanently blocked. As a result, a deadlock situation arises.

There are two approaches to solve deadlock problems in finite capacity queueing networks. First, one can extend the blocking mechanism by providing additional algorithms that dynamically resolve a deadlock situation. For example, some deadlocks can be resolved by allowing blocked jobs to select an alternate destination station. Note however, that adding a deadlock resolution mechanism significantly increases the complexity of the queueing network model, and, as a result, may render an analytical solution of the model intractable. Second, one can select the waiting room at each station sufficiently large such that deadlocks cannot occur. This solution requires knowledge of so-called *deadlock freedom conditions*, that is, conditions on the size of the waiting room of the stations which prevent deadlock situations. An advantage of the second approach is that it does not involve changes to the blocking mechanism.

In this study, we take a preventive approach to deadlocks and derive *deadlock freedom conditions* for finite capacity queueing networks with BAS blocking and multiple routing chains. For queueing networks with a single routing chain, Kundu and Akyildiz [4] proved that a network is deadlock-free if the number of jobs in the network is less than the capacity of the directed cycle with minimal waiting room. However, these conditions for deadlock freedom cannot be straightforwardly extended to networks with multiple routing chains since a deadlock may result from dependencies between jobs from different routing chains.

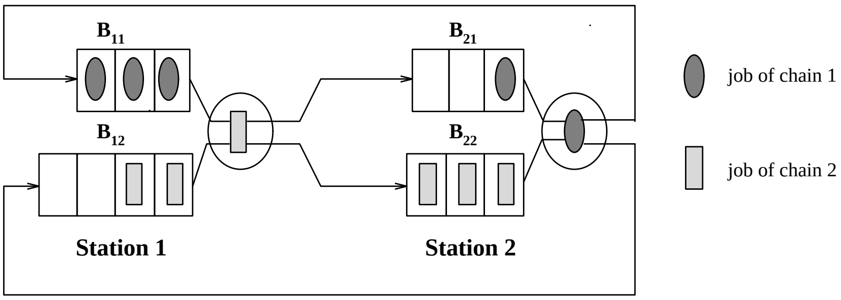

Figure 1: Blocking Network with 2 chains.

We demonstrate these dependencies in the network model shown in Figure 1. The Figure depicts a queueing network with two stations and two routing chains. We denote by B_{ir} the waiting room, referred to as *buffer*, of chain r at station i and by K_r the total number of jobs in chain r. Let the parameters be given by:

$$\begin{array}{ll} K_1 = 5; & K_2 = 6 \\ \text{Capacity of } B_{11} = 3; & \text{Capacity of } B_{21} = 3 \\ \text{Capacity of } B_{12} = 4; & \text{Capacity of } B_{22} = 3 \end{array}$$

In Figure 1, the job from chain 2 residing in the server of station 1 cannot proceed to the full buffer B_{22}. Thus, the job waits in the server until space in B_{22} becomes available. On the other hand, the job in the server of station 2 cannot enter buffer B_{11}, since B_{11} is full. Thus, a deadlock situation occurs, even though the conditions for deadlock freedom in [4] are satisfied for each routing chain in isolation.

We show that deadlock situations in multiple-chain queueing networks always occur in so-called *buffer cycles*. A buffer cycle is a cyclic sequence of buffers in the networks where the buffers may belong to different routing chains. The set of feasible buffer cycles is obtained from the transition probability matrices of the routing chains. We show that a queueing network with multiple routing chains is deadlock-free if and only if each buffer cycle is deadlock-free.

Once the deadlock freedom conditions are available they can be constructively applied to find an assignment of capacities to the buffers of each station such that deadlocks cannot occur. Of particular interest are capacity assignments which yield a deadlock-free network with the least total capacity. We refer to these assignments as *minimal capacity assignments*.

We present an optimization algorithm which yields a minimal capacity assignment for multiple-chain queueing networks with arbitrary topology. The set of buffer cycles is obtained by mapping the queueing network into a directed graph such that each cycle in the graph corresponds to a buffer cycle in the queueing network. The minimal capacity assignment is obtained with standard linear optimization technique.

A potential drawback of the optimization algorithm is its computational complexity which makes the algorithm impractical for networks with a large number of buffer cycles. As a worst case, we consider so-called *tandem networks*, that is, networks where all stations are connected in a sequence. In tandem networks with N stations and R routing chains, the number of buffer cycles is given by R^N. By taking advantage of the regular topology in a tandem network we can provide a minimal capacity assignment without running an optimization algorithm.

The remaining sections of this study are structured as follows. In Section 2 we describe the class of queueing models which is considered in this study. The conditions for deadlock freedom are stated and proved in Section 3. In Section 4 we define a capacity assignment to be minimal if it achieves deadlock freedom with the least total capacity. Then we present an optimization algorithm which generates a minimal capacity assignment for arbitrary network topologies. In Section 5 we show that a minimal capacity assignment can be directly given if the network has a so-called *tandem topology.* In Section 6 we conclude our results.

2 Model Description

We consider a closed queueing network $\Gamma = (\mathcal{N}, \mathcal{R}, \mathcal{P})$ with the following properties:

- The network contains a finite set $\mathcal{N}$ of stations and a finite set $\mathcal{R}$ of disjoint routing chains. Each job in the network belongs to exactly one routing chain. Let $\mathcal{R}_i \subseteq \mathcal{R}$ denote the set of routing chains whose jobs visit station i. Let $\mathcal{N}_r \subseteq \mathcal{N}$ denote the set of stations visited by jobs from routing chain r. $\mathcal{P}$ is a set of matrices $\mathcal{P} = (P_1, P_2, \ldots, P_{|\mathcal{R}|})$ where P_r is the $|\mathcal{N}| \times |\mathcal{N}|$ transition probability matrix for routing chain r. The elements of P_r are denoted by $p_{ij,r}$ with the following interpretation. A job of routing chain r which has received service by station i proceeds to station j with probability $p_{ij,r}$. Throughout the paper, we assume that $p_{ij,r} > 0$ implies that $r \in \mathcal{R}_i$ and $r \in \mathcal{R}_j$.

- Each station i has a single server. The service time distribution and the scheduling discipline of a station is arbitrary, but non-preemptive.

- Each station keeps separate buffers for jobs from different routing chains. B_{ir} denotes the buffer at station i (excluding the server) for jobs from routing chain r. Each buffer may accommodate only a finite number of jobs. Let Φ be an assignment of capacities to the buffers of Γ, i.e.,
$$\Phi : \{B_{ir} | i \in \mathcal{N},\;\; r \in \mathcal{R}_i\} \to \{0, 1, 2, \ldots\ldots\} \tag{1}$$
A buffer can have infinite capacity, i.e., $\Phi(B_{ir}) = \infty$, or no capacity at all, i.e., $\Phi(B_{ir}) = 0$. The total capacity of station i is computed by $\sum_{r \in \mathcal{R}_i} \Phi(B_{ir}) + 1$.

- The jobs from chain r in Γ is fixed and given by K_r. The total number of jobs in the network is denoted by
$$K = \sum_{r \in \mathcal{R}} K_r \tag{2}$$

- The number chain-r jobs in buffer B_{jr} cannot exceed its capacity $\Phi(B_{jr})$. Assume a job of chain r has completed service at some station i and wants to proceed to a station j. If B_{jr} is *saturated*, i.e., $\Phi(B_{jr})$ jobs are waiting in B_{ir}, the job is *blocked* at buffer B_{jr} and must reside in the server of station i until a place in B_{jr} becomes available. A server which contains a blocked job cannot serve other jobs. The described blocking mechanism is referred to as blocking-after-service or BAS.

3 Conditions For Deadlock Freedom

In this section we present the deadlock freedom conditions for a queueing network as described in Section 2. The theorem for deadlock freedom (DLF Theorem) is stated in terms of conditions that must hold for each *buffer cycle* in the network. A buffer cycle is defined as a cyclic sequence of buffers such that each buffer in the sequence belongs to a station which can have a blocked job at the next buffer of the sequence. A formal definition of a buffer cycle is given as follows.

Definition 1 *A* buffer cycle *is a sequence of buffers* $C = (B_{i_1 r_1}, B_{i_2 r_2}, \ldots, B_{i_{M-1} r_{M-1}}, B_{i_M r_M})$ *such that*

$$\begin{array}{llll} p_{i_m i_{m+1}, r_{m+1}} > 0 & \text{and} & r_m \in \mathcal{R}_{i_m} & \text{for all } 1 \leq m < M \\ p_{i_M i_1, r_1} > 0 & \text{and} & r_M \in \mathcal{R}_{i_M} & \end{array} \tag{3}$$

Let $\mathcal{C}$ *denote the set of all buffer cycles in* Γ.

The next definition provides the set of stations in buffer cycle C that may contain a job from routing chain r which is blocked at a buffer in the same cycle.

Definition 2 *The set of stations with buffers from routing chain* r *in buffer cycle* C *is defined by*

$$S_r^{(C)} = \{ i \mid B_{ir} \in C \} \tag{4}$$

For example, $C_1 = (B_{11}, B_{21})$ and $C_2 = (B_{11}, B_{22})$ are two buffer cycles in the network shown in Figure 1. From Definition 2 we obtain the sets of stations with potentially blocked jobs to be $S_1^{(C_1)} = \{1, 2\}$, $S_2^{(C_1)} = \emptyset$ and $S_1^{(C_2)} = \{1\}$, and $S_2^{(C_2)} = \{2\}$, for cycles C_1 and C_2, respectively.

Next we state the conditions for deadlock freedom in a queueing network with multiple routing chains. Theorem 1 states that for each buffer cycle C there must exist at least one routing chain r which, at the same time, cannot saturate all its buffers in cycle C and have $S_r^{(C)}$ jobs blocked at some other buffers in the cycle.

Theorem 1 (DLF Theorem) *A multiple chain queueing network* Γ *is* deadlock-free *if and only if for all* $C \in \mathcal{C}$ *there exists a routing chain* $r \in \mathcal{R}$ *such that:*

$$\sum_{i \in S_r^{(C)}} \Phi(B_{ir}) + |S_r^{(C)}| > K_r \tag{5}$$

For a given buffer cycle C and a given routing chain r, the term on the left of (5) is said to be the *DLF term* of routing chain r in buffer cycle C. We say that routing chain r satisfies the *DLF condition* in buffer cycle C, if inequality (5) holds for chain r in buffer cycle C. A capacity assignment Φ is said to satisfy the *DLF conditions for buffer cycle* C, if there is at least one routing chain which satisfies the DLF condition in cycle C. A capacity assignment Φ is said to be deadlock-free if Φ satisfies the DLF conditions for all buffer cycles $C \in \mathcal{C}$.

Proof:
Necessity: Assume there exists a cycle $C = (B_{i_1 r_1}, B_{i_2 r_2}, \ldots, B_{i_{M-1} r_{M-1}}, B_{i_M r_M})$ such that for all $r \in \mathcal{R}$ we obtain:

$$\sum_{i_m \in S_r^{(C)}} \Phi(B_{i_m r}) + |S_r^{(C)}| \leq K_r \tag{6}$$

Then, a state is feasible where all buffers in C are saturated, i.e., $B_{i_m r_m}$ holds $\Phi(B_{i_m r_m})$ jobs of chain r_m for all $1 \leq m \leq M$, the server of each station i_m contains a job from chain r_{m+1} if $m < M$, and the server of station i_M contains a job from chain r_1. Note that the entire cycle will have $|S_r^{(C)}|$ servers which contain a job from chain r. There exists a positive probability that each job in the server of station i_m $(m < M)$ has picked station i_{m+1} as destination station, and the job in the server of station i_M has selected station station i_1 as its next station. In this state, no server can release a job and eventually, each station i_m is blocked. Thus, a deadlock persists.

Sufficiency: Assume that assignment Φ satisfies the DLF condition for all buffer cycles, but the queueing network is in a deadlock state. Then there must exist a permanently blocked job in the server of a station, say i_1. Assume that the blocked job is from routing chain r_2 $(r_2 \in \mathcal{R}_{i_1})$. Assume that the blocked job in the server of station i_1 is blocked at a buffer, say $B_{i_2 r_2}$ of station i_2, that is, $B_{i_2 r_2}$ contains $\Phi(B_{i_2 r_2})$ jobs. Station i_2 itself must be blocked, otherwise a space in $B_{i_2 r_2}$ will eventually become available and station i_1 would not be permanently blocked. The job in the server of station i_2 is blocked at a saturated buffer, say $B_{i_3 r_3}$ of station i_3. We can continue to apply this argument. Since there is only a finite number of stations in the network, we will eventually encounter a job from routing chain r_{M+1} in the server of some station, say station i_M, which is blocked at a saturated buffer $B_{i_k i_{M+1}}$ of a previously considered station i_k. Then, buffers $B_{i_k r_{M+1}}, B_{i_{k+1} r_{k+1}}, \ldots, B_{i_M r_M}$ define a buffer cycle $C = (B_{i_k r_{M+1}}, B_{i_{k+1} r_{k+1}}, \ldots, B_{i_{M-1} r_{M-1}}, B_{i_M r_M})$. Note that a job from chain r_m is blocked in the server of station i_{m-1} if $B_{i_m r_m}$ is a buffer in the cycle. Therefore, $|S_r^{(C)}|$ jobs from chain r are blocked in servers of stations which have a buffer in the cycle. Since all buffers in the cycle are saturated, there are $\sum_{i_m \in S_r^{(C)}} \Phi(B_{i_m r})$ jobs from routing chain r in the buffers of the cycle. The sum for each chain r must be less or equal the total number of jobs K_r, otherwise the shown construction would not have been feasible. This implies that for all routing chains r with buffers in cycle C we have:

$$\sum_{i_m \in S_r^{(C)}} \Phi(B_{i_m r}) + |S_r^{(C)}| \leq K_r \tag{7}$$

However, this contradicts our assumption that Φ satisfies the DLF condition for all cycles. □

If the DLF conditions (5) are satisfied for a buffer cycle C, they are clearly satisfied for any cycle which contains the buffers of C as a subset. Therefore, to guarantee deadlock freedom of a network, it is not required to test the DLF conditions for all buffer cycles. This observation is formalized in the following lemma:

Lemma 1 *Given a multiple chain queueing network Γ and the set of all buffer cycles $\mathcal{C}$ of Γ. Let $\tilde{\mathcal{C}}$ be a maximal subset of $\mathcal{C}$ such that for any two cycles $C \in \tilde{\mathcal{C}}$ and $C' \in \tilde{\mathcal{C}}$ we have $C \not\subseteq C'$.* [1] *Then, Γ is deadlock-free if the DLF condition (5) is satisfied for all $C \in \tilde{\mathcal{C}}$.*

Proof: If $C \subseteq C'$, then $S_r^{(C)} \subseteq S_r^{(C')}$ for all $r \in \mathcal{R}$. Therefore, if a routing chain r satisfies the DLF condition in cycle C, that is, $\sum_{i_m \in S_r^{(C)}} \Phi(B_{i_m r}) + |S_r^{(C)}| > K_r$, then chain r also satisfies the DLF condition in all cycles C' with $C \subseteq C'$. Now the claim follows immediately since $\tilde{\mathcal{C}}$ is a maximal subset of $\mathcal{C}$. □

4 Deadlock-Free Capacity Assignment Algorithm

With the deadlock freedom conditions of Theorem 1, we can decide whether a given capacity assignment may result in deadlock situations. In this section, we will show that Theorem 1 can be applied in a constructive manner. In particular, we will use Theorem 1 to develop an algorithm which finds a deadlock-free capacity assignment with the least total capacity. Such a capacity assignment is referred to as *minimal.*

Definition 3 *A deadlock-free capacity assignment Φ^* for a queueing network Γ is* <u>minimal</u> *if for all deadlock-free assignments Φ for Γ it holds that*

$$\sum_{i \in \mathcal{N}} \sum_{r \in \mathcal{R}_i} \Phi^*(B_{ir}) \leq \sum_{i \in \mathcal{N}} \sum_{r \in \mathcal{R}_i} \Phi(B_{ir}) \tag{8}$$

Hence, any capacity assignment which allocates less total capacity to the buffers of the queueing network than a minimal capacity assignment will have a deadlock. In the remaining part of this section we will present an algorithm which finds a minimal capacity assignment. The algorithm is executed in two steps:

1. Find a maximal subset $\tilde{\mathcal{C}}$ of buffer cycles, as defined in Lemma 1, and establish the DLF conditions for each cycle.

2. Formulate an optimization problem of minimizing the total number of buffer capacities, subject to the constraints that the DLF conditions be satisfied for all cycles $C \in \tilde{\mathcal{C}}$.

In the following subsections we will discuss the steps of the algorithm in detail. At the end of the section, we present an example of the optimization algorithm.

[1] For two buffer cycles C and C' we say that $C \subseteq C'$ if the set of buffers in C is a subset of the set of buffers in C'.

4.1 Finding Buffer Cycles

We approach the problem of finding the buffer cycles of a queueing network as the problem of finding cycles in a directed graph. The following lemma allows to map the queueing network into a directed graph.

Lemma 2 *Given a multiple chain queueing network* Γ. *Obtain from* Γ *a directed graph* $G_\Gamma = (V_\Gamma, E_\Gamma)$ *with set of vertices* V_Γ *and set of arcs* E_Γ *by:*

(i) $V_\Gamma = \{B_{ir} \mid i \in \mathcal{N}, \quad r \in \mathcal{R}_i\}$,

(ii) $E_\Gamma = \{(B_{ir}, B_{js}) \mid p_{ij,s} > 0\}$.

Then, there exists a one-to-one mapping between the buffer cycles in Γ *and the cycles in* G_Γ.

Proof: The proof follows immediately from the definition of a buffer cycle in Definition 1 and the construction of the directed graph. □

The cycles in the directed graph can be obtained with any cycle-finding algorithm for directed graphs [3]. Note that with Lemma 1 we do not need to find *all* cycles in G_Γ. Rather, motivated by Lemma 1, we are interested in only those cycles which correspond to set $\tilde{\mathcal{C}}$ as given in Lemma 1, that is, a maximal subset of cycles where no cycle is fully contained in another cycle. In [2], we present an efficient algorithm which finds the subset $\tilde{\mathcal{C}}$ of relevant buffer cycles.

4.2 Optimization

From the set of buffer cycles as obtained in Subsection 4.1 we can generate the maximal subset of buffer cycles $\tilde{\mathcal{C}}$ that must be examined for deadlock freedom of the queueing network. If the DLF condition is satisfied for all buffer cycles in $\tilde{\mathcal{C}}$, then the network is deadlock-free. Here, we present two approaches for finding a minimal capacity assignment for queueing networks. The first approach is based on integer programming techniques and guarantees a minimal capacity assignment. The second approach is a heuristic method which always provides a deadlock-free capacity assignment, but may yield a suboptimal solution. The advantage of the heuristic method is that it is computationally less demanding than solving the integer program.

4.2.1 Optimization with Integer Programming

A minimal capacity assignment Φ^* satisfies the DLF condition in all buffer cycles with the least total number of buffer capacities. Thus, for a given network Γ, a minimal capacity assignment is obtained by solving the following optimization problem:

$$\begin{array}{l} \text{Find } \Phi^* \text{ which minimizes} \displaystyle\sum_{i\in\mathcal{N}} \sum_{r\in\mathcal{R}_i} \Phi^*(B_{ir}) \text{ subject to the constraints:} \\ (\forall C \in \tilde{\mathcal{C}})(\exists r \in \mathcal{R}) : \displaystyle\sum_{i\in S_r^{(C)}} \Phi^*(B_{ir}) + |S_r^{(C)}| > K_r \end{array} \tag{9}$$

By indexing the cycles in $\tilde{\mathcal{C}}$, i.e., $\tilde{\mathcal{C}} = \{C_1, C_2, \ldots, C_{|\tilde{\mathcal{C}}|}\}$, the optimization problem can be written as [2]:

$$\begin{array}{lll}
\textbf{minimize} & \sum_{i \in \mathcal{N}} \sum_{r \in \mathcal{R}_i} \Phi^*(B_{ir}) & \\
\textbf{subject to} & & \\
& \bigvee_{S_r^{(C_l)} \neq \emptyset} \sum_{i \in S_r^{(C_l)}} \Phi^*(B_{ir}) + |S_r^{(C_l)}| > K_r & l = 1, 2, \ldots, |\tilde{\mathcal{C}}| \qquad (10) \\
& \Phi^*(B_{ir}) \geq 0 & i \in \mathcal{N},\ r \in \mathcal{R}_i
\end{array}$$

By introducing additional $(0, 1)$-variables which allow us to replace the disjunctions in the constraints, we can state the optimization problem as an integer program:

$$\begin{array}{lll}
\textbf{minimize} & \sum_{i \in \mathcal{N}} \sum_{r \in \mathcal{R}_i} \Phi^*(B_{ir}) & \\
\textbf{subject to} & & \\
& \left(\sum_{i \in S_r^{(C_l)}} \Phi^*(B_{ir}) \right) - \mathrm{y}_{lr} \cdot (K_r - S_r^{(C_l)} + 1) \geq 0 & l = 1, 2, \ldots, |\tilde{\mathcal{C}}|,\ S_r^{(C_l)} \neq \emptyset \\
& \sum_{S_r^{(C_l)} \neq \emptyset} \mathrm{y}_{lr} \geq 1 & l = 1, 2, \ldots, |\tilde{\mathcal{C}}| \\
& \Phi^*(B_{ir}) \geq 0 & i \in \mathcal{N},\ r \in \mathcal{R}_i \qquad (11) \\
& \mathrm{y}_{lr} \in \{0, 1\} & l = 1, 2, \ldots, |\tilde{\mathcal{C}}|,\ S_r^{(C_l)} \neq \emptyset
\end{array}$$

The optimization of the equation system in (12) can be solved with any program package for integer linear programs. The solution of the integer program provides a minimal capacity assignment Φ^* for network Γ. Note that, in general, the system will have more than one optimal solution, since an optimization problem with the above structure will show considerable degeneracy.

4.2.2 Heuristic Approach

The number of constraints and variables in the optimization may become too large for a solution by integer programming. As an alternative, we present in Algorithm 1 an approximate method for allocating buffer capacities. Algorithm 1 consecutively assigns capacities to the buffer B_{ir} which eliminates the most DLF conditions with the smallest buffer capacity. Since the algorithm terminates only if all DLF conditions are satisfied, we obtain a deadlock-free assignment. However, there is no guarantee that the solution is optimal. In the example given in the Subsection 4.3 we will see that the approximate solution yields very accurate results, often identical with an optimal solution.

[2] We use the symbol '$\vee$' to denote the disjunction operator.

```
Input:   $\tilde{\mathcal{C}}$, the maximal subset of buffer cycles in $\Gamma$ where no cycle is a subset
         of another cycle.
Output:  $\Phi^*$, a deadlock-free capacity assignment for $\Gamma$.

1.       $\mathcal{C}_{ir} := \{C \in \tilde{\mathcal{C}} \mid B_{ir} \in C\}$
2.       $\Phi^{min}(B_{ir}) := \min\{K_r - |S_r^{(C)}| + 1 \mid C \in \mathcal{C}_{ir},\ S_r^{(C)} \neq \emptyset\}$
3.       Select $(i', r')$ such that $\frac{|\mathcal{C}_{i'r'}|}{\Phi^{min}(B_{i'r'})} = \min\left\{\frac{|\mathcal{C}_{ir}|}{\Phi^{min}(B_{ir})} \mid \mathcal{C}_{ir} \subseteq \tilde{\mathcal{C}}\right\}$.
4.       $\Phi^*(B_{i'r'}) := \Phi^{min}(B_{i'r'})$
5.       $\tilde{\mathcal{C}} := \tilde{\mathcal{C}} \setminus \mathcal{C}_{i'r'}$.
6.       if $\tilde{\mathcal{C}} \neq \emptyset$ then goto Step 1.
```

Algorithm 1.

4.3 Example

Next we apply the minimal capacity assignment algorithm to an example[3]. Figure 2 depicts a queueing network with three routing chains. The network contains 429 elementary buffer cycles, that is, buffer cycles where no buffer appears twice. However, there exists a maximal subset of buffer cycles where no cycle is contained in another cycle with 19 elements. Thus, with Lemma 1, only 19 buffer cycles must be considered for finding a minimal capacity assignment. These cycles are as follows:

$$
\begin{array}{rcl}
C_1 & = & (B_{1\,1}, B_{2\,1}) \\
C_2 & = & (B_{9\,2}, B_{10\,2}) \\
C_3 & = & (B_{5\,3}, B_{8\,3}, B_{10\,3}, B_{6\,3}) \\
C_4 & = & (B_{2\,1}, B_{4\,1}, B_{10\,1}) \\
C_5 & = & (B_{2\,1}, B_{4\,1}, B_{5\,1}, B_{10\,2}) \\
C_6 & = & (B_{2\,1}, B_{4\,1}, B_{5\,2}, B_{8\,3}, B_{10\,3}) \\
C_7 & = & (B_{2\,1}, B_{4\,1}, B_{5\,2}, B_{10\,2}) \\
C_8 & = & (B_{2\,1}, B_{4\,1}, B_{7\,2}, B_{10\,2}) \\
C_9 & = & (B_{4\,2}, B_{5\,1}, B_{8\,3}, B_{10\,3}, B_{3\,3}) \\
C_{10} & = & (B_{4\,2}, B_{5\,1}, B_{8\,3}, B_{10\,3}, B_{9\,2}, B_{3\,2}) \\
C_{11} & = & (B_{4\,2}, B_{5\,1}, B_{10\,2}, B_{3\,3}) \\
C_{12} & = & (B_{4\,2}, B_{5\,2}, B_{8\,3}, B_{10\,3}, B_{3\,3})
\end{array}
$$

[3] More examples of minimal capacity assignment for various network topologies are given in [5].

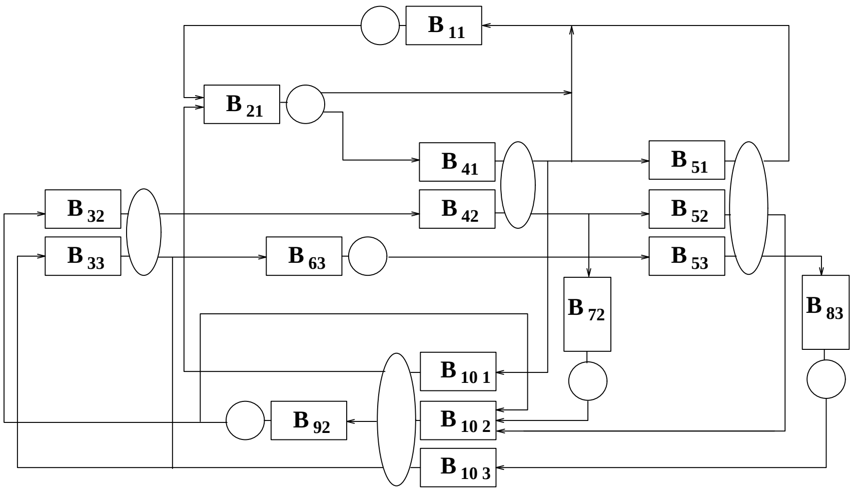

Figure 2: Multiple Chain Queueing Network.

$$
\begin{aligned}
C_{13} &= (B_{4\,2}, B_{5\,2}, B_{8\,3}, B_{10\,3}, B_{9\,2}, B_{3\,2}) \\
C_{14} &= (B_{4\,2}, B_{5\,2}, B_{10\,2}, B_{3\,3}) \\
C_{15} &= (B_{4\,2}, B_{10\,1}, B_{3\,3}) \\
C_{16} &= (B_{4\,2}, B_{10\,1}, B_{9\,2}, B_{3\,2}) \\
C_{17} &= (B_{4\,2}, B_{7\,2}, B_{10\,2}, B_{3\,3}) \\
C_{19} &= (B_{5\,3}, B_{10\,2}, B_{6\,3})
\end{aligned}
$$

In Table 1 we show the solution of the optimization for different values of $\underline{K} = (K_1, K_2, K_3)$ [4]. We also include the results obtained with the heuristic method given in Algorithm 1. Table 1 only depicts non-zero capacity assignments. Note that for all values of $\underline{K}$, the heuristic method of Algorithm 1 provides results which are very close to or identical with an optimal solution.

	$K_1 = 10$ $K_2 = 10$ $K_3 = 10$		$K_1 = 46$ $K_2 = 4$ $K_3 = 125$		$K_1 = 3$ $K_2 = 3$ $K_3 = 3$	
	minimal	heuristic	minimal	heuristic	minimal	heuristic
$\sum_{i\in\mathcal{N}} \sum_{r\in\mathcal{R}_i} \Phi^*(B_{ir})$	36	37	175	175	8	8
$\Phi^*(B_{2\,1})$	9	9	45	45	2	2
$\Phi^*(B_{4\,2})$	10	10	4	4	3	3
$\Phi^*(B_{10\,2})$	9	10	4	4	3	3
$\Phi^*(B_{6\,3})$	7	9	122	122	0	0

Table 1: Results for Example.

[4] The optimizations were solved with the programming package LINDO [8]

5 Deadlock-Free Capacity Assignments in Tandem Networks

The computational complexity of the capacity assignment algorithm presented in Section 4 increases with the size of $\tilde{\mathcal{C}}$, the maximal subset of cycles where no buffer cycle is contained in another cycle. For some networks the size of $\tilde{\mathcal{C}}$ can make the optimization algorithm impractical. As a worst case, consider the network in Figure 3. Here $\tilde{\mathcal{C}}$ is identical with the set of all buffer cycles in the network. Note that in the queueing network in Figure 3, the set $\tilde{\mathcal{C}}$ contains R^N cycles. In the following we show that the effort to obtain a minimal capacity assignment can be greatly reduced, if we take advantage of networks with a regular topology such as the network shown in Figure 3.

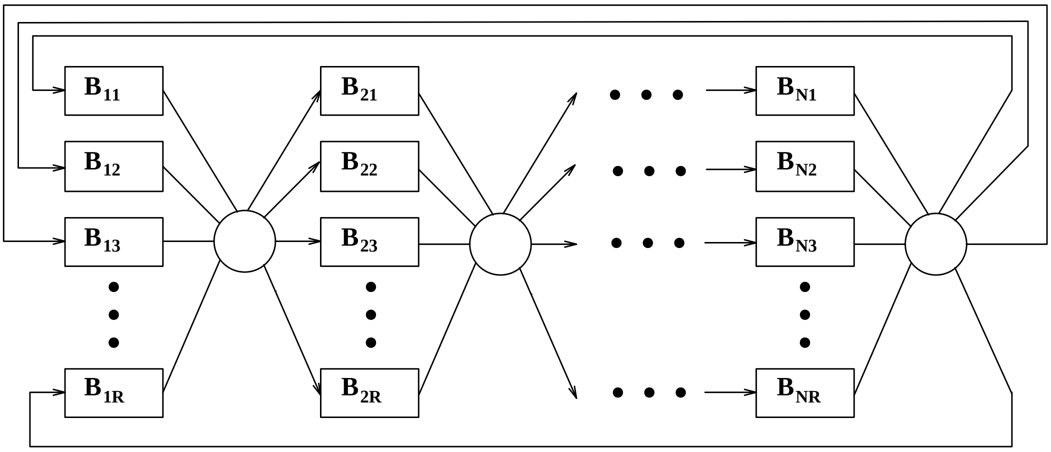

Figure 3: Tandem Network.

5.1 Minimal Capacity Assignments for Tandem Networks

We refer to networks which have a topology as shown in Figure 3 as *tandem networks.* Tandem networks can be formally defined as follows.

Definition 4 *A* <u>tandem network</u> $\Gamma = (\mathcal{N}, \mathcal{R})$ *is a multiple chain queueing network* $\Gamma = (\mathcal{N}, \mathcal{R}, \mathcal{P})$ *with*

$$\mathcal{R}_i = \mathcal{R} \qquad \text{for all } i \in \mathcal{N} \tag{12}$$

and for each station $i \in \mathcal{N}$ *there exists a station* $i' \in \mathcal{N}$ $(i \neq i')$ *such that for all* $r \in \mathcal{R}$:

$$p_{ij,r} = \begin{cases} 1 & \text{if } j = i' \\ 0 & \text{if } j \neq i' \end{cases} \tag{13}$$

In a tandem network, each station has a buffer for all routing chains, i.e., $\mathcal{R}_i = \mathcal{R}$ for all $i \in \mathcal{N}$, and jobs from each routing chain can visit all stations, i.e., $\mathcal{N}_r = \mathcal{N}$ for all $r \in \mathcal{R}$. Each buffer cycle in a tandem network contains one buffer from each station, and the the length of each buffer cycle is equal to $|\mathcal{N}|$.

Our main result in this section is that minimal capacity assignments in tandem networks can be provided without running any optimization algorithm. The result is presented in the following theorem where we state that a minimal assignment for tandem networks is obtained by allocating non-zero capacities to the buffers of only one station.

Theorem 2 *For a tandem network $\Gamma = (\mathcal{N}, \mathcal{R})$, the following capacity assignments Φ_i^* $(i \in \mathcal{N})$ are minimal:*

$$\Phi_i^*(B_{jr}) = \begin{cases} K_r & \textit{if } j = i \textit{ and } K - K_r \geq |\mathcal{N}| \\ K - |\mathcal{N}| + 1 & \textit{if } j = i \textit{ and } K \geq |\mathcal{N}| > K - K_r \\ 0 & \textit{if } j = i \textit{ and } |\mathcal{N}| > K \\ 0 & \textit{if } j \neq i \end{cases} \tag{14}$$

$$\textit{for } j \in \mathcal{N} \textit{ and } r \in \mathcal{R}.$$

Note that Theorem 2 defines a set of $|\mathcal{N}|$ different capacity assignments, that is, there is one capacity assignment Φ_i^* for each station $i \in \mathcal{N}$. The remainder of this section contains the proof of Theorem 2. The proof consists of two parts. First we must show that the capacity assignments given above do not allow deadlock situations. Secondly, we have to show that any capacity assignment, which assigns less total buffer capacities to the stations of the tandem network will result in a deadlock. The second part of the proof requires considerable effort since we must effectively construct a deadlock for a large class of capacity assignments. In the next subsection, we present a set of technical lemmas which simplify the second part of the proof. Following is the proof of Theorem 2.

REMARK: Note that the network shown in Figure 1 also is a tandem network. Thus, with Theorem 2 we can give two minimal capacity assignments, Φ_1^* and Φ_2^*. Assuming the same network parameters as in Section 1 we have $|\mathcal{N}| = 2$ and $K = K_1 + K_2 = 11$. Therefore, Φ_1^* and Φ_2^* assign capacities as shown here:

$\Phi_1^*(B_{11}) = 5$	$\Phi_2^*(B_{11}) = 0$
$\Phi_1^*(B_{12}) = 6$	$\Phi_2^*(B_{12}) = 0$
$\Phi_1^*(B_{21}) = 0$	$\Phi_2^*(B_{21}) = 5$
$\Phi_1^*(B_{22}) = 0$	$\Phi_2^*(B_{22}) = 6$

It can be easily verified that Φ_1^* and Φ_2^* are the only minimal capacity assignment for the tandem network in Figure 1. However, for some tandem networks, there may exist minimal capacity assignments which are different from the assignments given in Theorem 2. For example, in a tandem network with $|\mathcal{N}| = 3$, $\mathcal{R} = \{1, 2\}$, and $K_1 = K_2 = 3$, a capacity assignment Φ' with $\Phi'_{ir} = 1$ (for $i = 1, 2, 3$ and $r = 1, 2$) is minimal.

5.2 Properties of Tandem Networks

In tandem networks, a violation of a DLF condition by a routing chain r can be easily detected if the number of stations in the tandem network is equal to the number of jobs of the routing chain, that is, $K_r = |\mathcal{N}|$. This assertion is shown in Lemmas 3 and 4. In Lemmas 5 and 6, we show how to modify the structure of tandem networks such that we can take advantage of Lemmas 3 and 4, yet, any deadlock in the modified network corresponds to a deadlock in the original network. Lemma 7 presents a modification of a tandem network which has a routing chains r such that $K_r < |\mathcal{N}|$. The modification is deadlock preserving, that is, any deadlock in the modified network can be extended to a deadlock situation in the original network.

Lemma 3 *Given a tandem network $\Gamma = (\mathcal{N}, \mathcal{R})$. Let Φ be a capacity assignment for Γ such that $\sum_{i\in\mathcal{N}} \Phi(B_{ir}) \leq K_r + 1$ and $K_r = |\mathcal{N}|$ for some routing chain $r \in \mathcal{R}$. Let C be a buffer cycle in Γ such that $|S_r^{(C)}| > 0$. If routing chain r satisfies the DLF condition for cycle C then it does not satisfy the DLF condition in any buffer cycle C' with $S_r^{(C')} = \mathcal{N} \setminus S_r^{(C)}$.* ([5])

Thus, if the number of jobs in a routing chain is equal to the number of stations, and the total capacity which is assigned to the buffers of this routing chain exceeds the number of stations by at most one, we can always find a buffer cycle such that the routing chain does not satisfy the DLF condition.

Proof: Assume a capacity assignment Φ and a routing chain $r \in \mathcal{R}$ with $\sum_{i\in\mathcal{N}} \Phi(B_{ir}) \leq K_r + 1$. Let C be a buffer cycle in Γ such that chain r satisfies the DLF condition, i.e.,

$$\sum_{i\in S_r^{(C)}} \Phi(B_{ir}) + |S_r^{(C)}| > K_r \tag{15}$$

Now assume a buffer cycle C' with $S_r^{(C')} = \mathcal{N} \setminus S_r^{(C)}$. Then, we obtain for the DLF term of chain r in cycle C':

$$\begin{aligned}
\sum_{i\in S_r^{(C')}} \Phi(B_{ir}) + |S_r^{(C')}| &= \sum_{i\in S_r^{(C')}} \Phi(B_{ir}) + \left(|\mathcal{N}| - |S_r^{(C)}|\right) && (16)\\
&\leq \left(K_r + 1 - \sum_{i\in S_r^{(C)}} \Phi(B_{ir})\right) + \left(K_r - |S_r^{(C)}|\right) && (17)\\
&< K_r + 1 && (18)
\end{aligned}$$

The equality in (16) follows from $S_r^{(C')} = \mathcal{N} \setminus S_r^{(C)}$. For (17) we use $\sum_{i\in\mathcal{N}} \Phi(B_i r) \leq K_r + 1$ and $K_r = |\mathcal{N}|$. Finally, we obtain (18) from (15). Thus, the DLF condition of chain r in cycle C' is not satisfied. □

[5] For two sets A and B, we use $A \setminus B$ to denote the relative complement of B with respect to A, i.e., $A \setminus B := \{x \in A \mid x \notin B\}$.

The following lemma provides an even stronger result. Similar to Lemma 3, we assume that the number of stations in the tandem network is equal to the number of jobs of a particular routing chain. Here, however, we assume that the total capacities assigned to the buffers of the routing chain is less than the number of stations. Then we can divide the stations of the tandem network into two sets such that the DLF condition of the routing chain is violated in both sets.

Lemma 4 *Given a tandem network* $\Gamma = (\mathcal{N}, \mathcal{R})$*. Let* Φ *be a capacity assignment for* Γ *such that* $\sum_{i \in \mathcal{N}} \Phi(B_{ir}) < K_r$ *and* $K_r = |\mathcal{N}|$ *for some routing chain* $r \in \mathcal{R}$*. Then* $\mathcal{N}$ *can be partitioned into two disjoint sets* $\mathcal{N}_1$ *and* $\mathcal{N}_2$ *such that*

(a) $\displaystyle\sum_{i \in \mathcal{N}_1} \Phi(B_{ir}) + |\mathcal{N}_1| \leq K_r$, *and*

(b) $\displaystyle\sum_{i \in \mathcal{N}_2} \Phi(B_{ir}) + |\mathcal{N}_2| \leq K_r$.

Proof: We partition $\mathcal{N}$, the set of all stations, into two subsets $\mathcal{Z}$ and $\mathcal{NZ}$. $\mathcal{Z}$ denotes the set of stations to which Φ assigns no buffers to chain r. $\mathcal{NZ}$ is the set of stations with non-zero buffer capacities for chain r. $\mathcal{Z}$ and $\mathcal{NZ}$ are given by:

$$\begin{aligned} \mathcal{Z} &= \{i \in \mathcal{N} \mid \Phi(B_{ir}) = 0\} \\ \mathcal{NZ} &= \{i \in \mathcal{N} \mid \Phi(B_{ir}) > 0\} \end{aligned}$$

```
Input:   A set $\mathcal{NZ} = \{i \in \mathcal{N} \mid \Phi(B_{ir}) > 0\}$.
Output:  Two sets $\mathcal{X}$ and $\mathcal{Y}$ such that $\mathcal{X} \cap \mathcal{Y} = \emptyset$, $\mathcal{X} \cup \mathcal{Y} = \mathcal{NZ}$,
         and $\sum_{i \in \mathcal{X}} \Phi(B_{ir}) + |\mathcal{X}| \leq |\mathcal{N}|$.

         begin
         $\mathcal{X} := \emptyset$;
            $\mathcal{Y} := \emptyset$;
            while ($\mathcal{NZ} \neq \emptyset$) do
               Select $v$ such that $\Phi(B_{vr}) = \max_{j \in \mathcal{NZ}}\{\Phi(B_{jr})\}$;
               if $\sum_{i \in \mathcal{X}} \Phi(B_{ir}) + |\mathcal{X}| + \Phi(B_{vr}) + 1 \leq |\mathcal{N}|$ then
                  $\mathcal{X} := \mathcal{X} \cup \{v\}$;
               else
                  $\mathcal{Y} := \mathcal{Y} \cup \{v\}$;
               endif
               $\mathcal{NZ} := \mathcal{NZ} \setminus \{v\}$;
            endwhile
         end
```

Algorithm 2.

Next we apply Algorithm 2 to partition $\mathcal{N}\mathcal{Z}$ into two disjoint sets $\mathcal{X}$ and $\mathcal{Y}$. Algorithm 2 is constructed such that after termination of the algorithm the following holds:

$$\sum_{i\in\mathcal{X}} \Phi(B_{ir}) + |\mathcal{X}| \leq |\mathcal{N}| \tag{19}$$

Since $\sum_{i\in\mathcal{N}} \Phi(B_{ir}) < |\mathcal{N}|$ we have $|\mathcal{Z}| > 0$. Therefore, we can always select a subset $\mathcal{Z}_1 \subseteq \mathcal{Z}$ such that:

$$\sum_{i\in\mathcal{X}} \Phi(B_{ir}) + |\mathcal{X}| + |\mathcal{Z}_1| = |\mathcal{N}| \tag{20}$$

Note that by construction, sets $\mathcal{X}$, $\mathcal{Y}$, and $\mathcal{Z}$ have the following properties:

$$\begin{aligned} |\mathcal{N}| &= |\mathcal{X}| + |\mathcal{Y}| + |\mathcal{Z}| && (21)\\ |\mathcal{N}| &> \sum_{i\in\mathcal{X}} \Phi(B_{ir}) + \sum_{i\in\mathcal{Y}} \Phi(B_{ir}) && (22) \end{aligned}$$

By setting $\mathcal{Z}_2 := \mathcal{Z} \setminus \mathcal{Z}_1$ we obtain from (20), (21), and (22):

$$\sum_{i\in\mathcal{Y}} \Phi(B_{ir}) + |\mathcal{Y}| + |\mathcal{Z}_2| < |\mathcal{N}| \tag{23}$$

Next we define

$$\begin{aligned} \mathcal{N}_1 &:= \mathcal{X} \cup \mathcal{Z}_1 && (24)\\ \mathcal{N}_2 &:= \mathcal{Y} \cup \mathcal{Z}_2 && (25) \end{aligned}$$

Then, equation (20) shows the correctness of claim (a) in Lemma 4, and equation (23) shows the correctness of claim (b) in Lemma 4. □

Lemmas 3 and 4 allow to make statements regarding the deadlock properties of routing chains where the number of jobs is equal to the number of stations in the tandem network. The next two lemmas, Lemmas 5 and 6, show how to modify a tandem network network such that we can take advantage of Lemmas 3 and 4.

Lemma 5 *Given an arbitrary network $\Gamma = (\mathcal{N}, \mathcal{R}, \mathcal{P})$, and Φ, a capacity assignment for Γ. Consider a routing chain $r \in \mathcal{R}$ with K_r jobs. Assume reduce the number of jobs in chain r by one and define a new capacity assignment Φ' which differs from Φ in the capacity assignment to only one buffer B_{jr} with $\Phi(B_{jr}) > 0$ as follows:*

$$\Phi'(B_{iq}) = \begin{cases} \Phi(B_{iq}) - 1 & \textit{if } q = r \textit{ and } i = j \\ \Phi(B_{iq}) & \textit{otherwise} \end{cases}$$

Then Φ violates the DLF condition for chain r with K_r jobs in a buffer cycle, if Φ' violates the DLF condition for chain r with $K_r - 1$ jobs in the same buffer cycle.

Lemma 5 says that given a routing chain with a large capacity assigned to its buffers, i.e., $\sum_{i \in \mathcal{N}} \Phi(B_{ir}) > K_r$, we can reduce both the capacity assignments and the number of jobs such that any deadlock in the modified network also results in a deadlock in the unmodified network.

Proof: Assume that chain r has K_r jobs, and suppose a buffer cycle C in Γ such that capacity assignment Φ' results in a violation of the DLF condition of routing chain r in cycle C, i.e.,

$$\sum_{i \in S_r^{(C)}} \Phi'(B_{ir}) + |S_r^{(C)}| \leq K_r - 1 \tag{26}$$

If $j \notin S_r^{(C)}$ we obtain for assignment Φ:

$$\sum_{i \in S_r^{(C)}} \Phi(B_{ir}) + |S_r^{(C)}| = \sum_{i \in S_r^{(C)}} \Phi'(B_{ir}) + |S_r^{(C)}| < K_r \tag{27}$$

Obviously, the DLF condition is also violated with capacity assignment Φ, and K_r jobs in chain r. On the other hand, if $j \in S_r^{(C)}$, the following holds:

$$\sum_{i \in S_r^{(C)}} \Phi(B_{ir}) + |S_r^{(C)}| = \sum_{i \in S_r^{(C)}} \Phi'(B_{ir}) + 1 + |S_r^{(C)}| \tag{28}$$

Equations (26) and (28) result in:

$$\sum_{i \in S_r^{(C)}} \Phi(B_{ir}) + |S_r^{(C)}| \leq K_r \tag{29}$$

Again, the DLF condition for cycle C with capacity assignment Φ is violated. □

In Lemma 6 we show how to modify a tandem network such that at least one routing has more jobs than the modified network has stations. Lemma 6 uses Algorithm 3 to perform the desired modifications to the tandem network.

Lemma 6 *Given a tandem network $\Gamma = (\mathcal{N}, \mathcal{R})$ and a capacity assignment Φ for Γ. Assume the following conditions hold:*

(α1) $\sum_{i \in \mathcal{N}} \Phi(B_{ir}) \leq K_r$ *for all $r \in \mathcal{R}$, and*

(β1) $\sum_{r \in \mathcal{R}} K_r \geq |\mathcal{N}|$.

Algorithm 3 generates from Γ a tandem network $\Gamma' = (\mathcal{N}', \mathcal{R}')$ with $\mathcal{N}' \subseteq \mathcal{N}$ and $\mathcal{R}' \subseteq \mathcal{R}$, such that

(γ1) $K_r \geq |\mathcal{N}'|$ *for at least one $r \in \mathcal{R}'$.*

(δ1) *If Γ' is not deadlock-free, then Γ is not deadlock-free.*

```
Input:   Tandem network $\Gamma = (\mathcal{N}, \mathcal{R})$ which satisfies properties $(\alpha 1)$ and $(\beta 1)$
         in Lemma 6.
Output:  Tandem network $\Gamma' = (\mathcal{N}', \mathcal{R}')$ with $\mathcal{N}' \subseteq \mathcal{N}$ and $\mathcal{R}' \subseteq \mathcal{R}$.
         $\Gamma'$ satisfies $(\gamma 1)$ and $(\delta 1)$ in Lemma 6.

         begin
1.           $\mathcal{N}^{(1)} := \mathcal{N}$;
2.           $\mathcal{R}^{(1)} := \mathcal{R}$;
4.           $n := 1$;
5.           while $\max_{q \in \mathcal{R}^{(n)}} \{K_q\} < |\mathcal{N}^{(n)}|$ do
6.               Select $r$ with $K_r = \min_{q \in \mathcal{R}^{(n)}} \{K_q\}$;
7.               $\mathcal{Z}^{(n)} := \left\{ j \in \mathcal{N}^{(n)} \mid \Phi(B_{jr}) = 0 \right\}$;
8.               if $K_r > |\mathcal{N}^{(n)}|/2$ then
9.                   Select $\overline{\mathcal{Z}}^{(n)} \subseteq \mathcal{Z}^{(n)}$ with $|\overline{\mathcal{Z}}^{(n)}| = |\mathcal{N}^{(n)}| - K_r$;
10.              else
11.                  Select $\overline{\mathcal{Z}}^{(n)} \subseteq \mathcal{Z}^{(n)}$ with $|\overline{\mathcal{Z}}^{(n)}| = K_r$;
12.              endif
13.              $\mathcal{N}^{(n+1)} := \mathcal{N}^{(n)} \setminus \overline{\mathcal{Z}}^{(n)}$;
14.              $\mathcal{R}^{(n+1)} := \mathcal{R}^{(n)} \setminus \{r\}$;
15.              $n := n + 1$;
16.          endwhile
         end
```

Algorithm 3.

Proof: The proof is performed in three steps. First we show that the set $\overline{\mathcal{Z}}^{(n)}$ can always be selected as shown in Steps 9 and 11 in Algorithm 3. In the second and third steps, we show that conditions $(\gamma 1)$ and $(\delta 1)$, respectively, are satisfied when the algorithm terminates.

1. *Selecting* $|\overline{\mathcal{Z}}^{(n)}|$ *in Steps 9 and 11 of Algorithm 3 is always feasible.*

   Consider the ith iteration of Algorithm 3, and suppose that in Step 6, routing chain r_i is selected. Since property $(\alpha 1)$ holds and since $\mathcal{N}^{(i)} \subseteq \mathcal{N}^{(0)}$, we have $\sum_{j \in \mathcal{N}^{(i)}} \Phi(B_{jr_i}) \leq K_{r_i}$. Therefore, there can be at most K_{r_i} buffers from chain r_i with a non-zero capacity among the stations of $\mathcal{N}^{(i)}$, and there are at least $|\mathcal{N}^{(i)}| - K_{r_i}$ stations without buffer capacities for chain r_i. Thus, the number of elements in $\mathcal{Z}^{(i)}$ (Step 7) has the following lower bound:

   $$|\mathcal{Z}^{(i)}| \geq |\mathcal{N}^{(i)}| - K_{r_i} \tag{30}$$

Now we distinguish two cases:

- *Case 1:* $K_{r_i} \le |\mathcal{N}^{(i)}|/2$
  In this case, Algorithm 3 executes Step 11 in the ith iteration. With $K_{r_i} \le |\mathcal{N}^{(i)}|/2$, equation (30) can be rewritten as:
  $$|\mathcal{Z}^{(i)}| \ge K_{r_i} \tag{31}$$
  We follow that there are at least K_{r_i} stations without capacities assigned to routing chain r among the stations of $\mathcal{N}^{(i)}$. Hence, the selection of $\overline{\mathcal{Z}}^{(i)}$ in Step 11 is feasible.
- *Case 2:* $K_{r_i} > |\mathcal{N}^{(i)}|/2$
  Here, Algorithm 3 executes Step 9 in the ith iteration. Since $\mathcal{Z}^{(i)}$ satisfies (30), the selection of $\overline{\mathcal{Z}}^{(i)}$ with $|\overline{\mathcal{Z}}^{(i)}| = |\mathcal{N}^{(i)}| - K_r$ as in Step 9 is feasible.

2. *After termination, Algorithm 3 satisfies condition (γ1).*
Let us assume, that in the ith iteration $K_{r_i} > |\mathcal{N}^{(i)}|/2$. Recall that in this case, Step 9 is executed. If we count the number of stations in $\mathcal{N}^{(i+1)}$ after Step 13 we obtain:

$$\begin{aligned} |\mathcal{N}^{(i+1)}| &= |\mathcal{N}^{(i)}| - |\overline{\mathcal{Z}}^{(i)}| && (32)\\ &= |\mathcal{N}^{(i)}| - (|\mathcal{N}^{(i)}| - K_{r_i}) && (33)\\ &= K_{r_i} && (34)\end{aligned}$$

Since r_i was selected such that K_{r_i} is minimal among all routing chains in $\mathcal{R}^{(i)}$ (Step 6), all routing chains $r \in \mathcal{R}^{(i+1)}$ (Step 14) satisfy:

$$K_r \ge |\mathcal{N}^{(i+1)}| \tag{35}$$

Thus, condition (γ1) is satisfied after the ith iteration whenever $K_{r_i} > |\mathcal{N}^{(i)}|/2$ in the ith iteration.

Now assume that $K_{r_i} \le |\mathcal{N}^{(i)}|/2$ holds in each iteration of the *while*-loop. Then Step 11 is executed in each iteration. After i iterations, the set $\mathcal{N}^{(i+1)}$ is given by:

$$\begin{aligned} \mathcal{N}^{(i+1)} &= \mathcal{N}^{(i)} \setminus \overline{\mathcal{Z}}^{(i)} && (36)\\ &= (\ldots((\mathcal{N}^{(1)} \setminus \overline{\mathcal{Z}}^{(1)}) \setminus \overline{\mathcal{Z}}^{(2)}) \ldots) \setminus \overline{\mathcal{Z}}^{(i)}) && (37)\\ &= \mathcal{N}^{(1)} \setminus (\overline{\mathcal{Z}}^{(1)} \cup \overline{\mathcal{Z}}^{(2)} \cup \ldots \cup \overline{\mathcal{Z}}^{(i)}) && (38)\end{aligned}$$

Since all $\overline{\mathcal{Z}}^{(j)}$ are mutually disjoint, we obtain for the number of stations in $\mathcal{N}^{(i+1)}$ that

$$\begin{aligned} |\mathcal{N}^{(i+1)}| &= |\mathcal{N}^{(1)}| - (|\overline{\mathcal{Z}}^{(1)}| + |\overline{\mathcal{Z}}^{(2)}| + \ldots + |\overline{\mathcal{Z}}^{(i)}|) && (39)\\ &= |\mathcal{N}| - \sum_{r \notin \mathcal{R}^{(i+1)}} K_r && (40)\end{aligned}$$

For equation (40) we have used $\mathcal{N}^{(1)} = \mathcal{N}$ from Step 1, and $|\overline{\mathcal{Z}}^{(j)}| = K_{r_j}$ from Step 11 of Algorithm 3 (assuming that chain r_j is selected in the jth iteration). Since with property (β1) we have $\sum_{r \in \mathcal{R}} K_r > |\mathcal{N}|$, there must exist an n with $1 \le n < |\mathcal{R}|$ such that $K_r \ge |\mathcal{N}^{(n)}|$ for at least one $r \in \mathcal{R}^{(n)}$. Then, property (γ1) is satisfied after the nth iteration.

3. *If* Γ' *is not deadlock-free, then* Γ *is not deadlock-free.*
   If Γ' contains a deadlock there exists a buffer cycle C' in Γ' which consists of buffers from routing chains in $\mathcal{R}'$, and no routing chain in $\mathcal{R}'$ satisfies the DLF condition. We will complete buffer cycle C' in Γ' to a buffer cycle C in Γ, and we show that no DLF condition is satisfied in C. Then a deadlock situation occurs in cycle C of Γ.

   Assume that Algorithm 3 eliminates routing chains $r_1, r_2, \ldots, r_k$ from network Γ in the indicated sequence, i.e., in the first iteration r_1 is eliminated, in the second iteration r_2 is eliminated, and so on. Then, $\mathcal{R}'$ and $\mathcal{N}'$ are given by:

$$\begin{aligned} \mathcal{R}' &= \mathcal{R}^{(k+1)} = \mathcal{R} \setminus \{r_1, r_2, \ldots, r_k\} && (41) \\ \mathcal{N}' &= \mathcal{N}^{(k+1)} = \mathcal{N} \setminus (\overline{\mathcal{Z}}^{(1)} \cup \overline{\mathcal{Z}}^{(2)} \cup \ldots \cup \overline{\mathcal{Z}}^{(k)}) && (42) \end{aligned}$$

   Per construction, the sets $\overline{\mathcal{Z}}^{(i)}$ are mutually disjunct. Also, per construction, $\mathcal{N}^{(k+1)} \cap \overline{\mathcal{Z}}^{(i)} = \emptyset$ for all $1 \leq i \leq k$. Thus, the sets $\mathcal{N}^{(k+1)}, \overline{\mathcal{Z}}^{(1)}, \overline{\mathcal{Z}}^{(2)}, \ldots, \overline{\mathcal{Z}}^{(k)}$ are a partition of the set of stations $\mathcal{N}$ in Γ. Next we construct a buffer cycle C in Γ by expanding the buffer cycle C' of Γ'. We set

$$S_r^{(C)} := \begin{cases} S_r^{(C')} & \text{if } r \in \mathcal{R}' \\ |\overline{\mathcal{Z}}^{(i)}| & \text{if } r \notin \mathcal{R}' \end{cases} \tag{43}$$

   Per assumption, the DLF condition for any chain $r \in \mathcal{R}'$ is not satisfied. On the other hand, if $r \notin \mathcal{R}'$, or equivalently, $r \in \{r_1, r_2, \ldots, r_k\}$, then $r = r_i$ for some $1 \leq i \leq k$. Note that in this case, the DLF condition cannot be satisfied since $S_{r_1}^{(C'')} = |\overline{\mathcal{Z}}^{(i)}|$ implies

$$\sum_{j \in S_{r_i}^{(C)}} \Phi(B_{jr_i}) = \sum_{j \in \overline{\mathcal{Z}}^{(i)}} \Phi(B_{jr_i}) = 0$$

   and

$$|S_{r_i}^{(C)}| = |\overline{\mathcal{Z}}^{(i)}| = K_{r_i}$$

   Thus, all $r \notin \mathcal{R}'$ do not satisfy the DLF condition for cycle C. Since no routing chain satisfies the DLF condition, cycle C in Γ contains a deadlock. □

The next lemma is applicable to tandem networks where the total number of jobs in all routing chains is less than the number of stations in the tandem network. We use Algorithm 4 to reduce the tandem network to a network with at most one routing chain. If this routing chain contains a deadlock, then the original network also contains a deadlock.

Lemma 7 *Given a tandem network* $\Gamma = (\mathcal{N}, \mathcal{R})$ *and a capacity assignment* Φ *for* Γ. *Assume the following conditions hold:*

(α2) $\sum_{i \in \mathcal{N}} \Phi(B_{ir}) < K_r$ *for all* $r \in \mathcal{R}$, *and*

(β2) $\sum_{r \in \mathcal{R}} K_r < |\mathcal{N}|$.

```
Input:    Tandem network $\Gamma = (\mathcal{N}, \mathcal{R})$ which satisfies properties ($\alpha 2$) and ($\beta 2$)
          in Lemma 7.
Output:   Tandem network $\Gamma' = (\mathcal{N}', \mathcal{R}')$ with $\mathcal{N}' \subseteq \mathcal{N}$ and $\mathcal{R}' \subseteq \mathcal{R}$.
          $\Gamma'$ satisfies ($\gamma 2$) and ($\delta 2$) in Lemma 7.

          begin
1.            $\mathcal{N}^{(1)} := \mathcal{N}$;
2.            $\mathcal{R}^{(1)} := \mathcal{R}$;
4.            $n := 1$;
5.            while $\min_{q \in \mathcal{R}^{(n)}} \{K_q\} < |\mathcal{N}^{(n)}|/2$ do
6.                Select $r$ with $K_r = \min_{q \in \mathcal{R}^{(n)}} \{K_q\}$;
7.                $\mathcal{Z}^{(n)} := \Big\{ j \in \mathcal{N}^{(n)} \mid \Phi(B_{jr}) = 0 \Big\}$;
8.                Select $\overline{\mathcal{Z}}^{(n)} \subseteq \mathcal{Z}^{(n)}$ with $|\overline{\mathcal{Z}}^{(n)}| = K_r$;
9.                $\mathcal{N}^{(n+1)} := \mathcal{N}^{(n)} \setminus \overline{\mathcal{Z}}^{(n)}$;
10.               $\mathcal{R}^{(n+1)} := \mathcal{R}^{(n)} \setminus \{r\}$;
11.               $n := n + 1$;
12.           endwhile
          end
```

Algorithm 4.

Algorithm 4 generates from Γ *a tandem network* $\Gamma' = (\mathcal{N}', \mathcal{R}')$ *with* $\mathcal{N}' \subseteq \mathcal{N}$ *and* $\mathcal{R}' \subseteq \mathcal{R}$*, and* Γ' *satisfies*

($\gamma 2$) $|\mathcal{R}'| = 1$ *and* $K_r \geq |\mathcal{N}'|/2$ *for* $r \in \mathcal{R}'$*, or* $|\mathcal{N}'| = |\mathcal{N}| - \sum_{r \in \mathcal{R}} K_r$.

($\delta 2$) *If* Γ' *is not deadlock-free, then* Γ *is not deadlock-free.*

Proof: The proof is conducted similarly to the proof of Lemma 6. As in the the proof of Lemma 6 (equation (31)) we can show that a selection of $\overline{\mathcal{Z}}^{(n)}$ as in Step 6 of Algorithm 4 is always feasible. Showing that condition ($\delta 2$) holds when Algorithm 4 terminates is equivalent to showing that condition ($\delta 1$) holds when Algorithm 3 terminates. Note however, that Γ' may not contain any routing chain at all, i.e., $\Gamma' = (\mathcal{N}', \emptyset)$ is a feasible outcome. In this case Γ' will never contain a deadlock. To complete the proof we have to show that condition ($\gamma 2$) is satisfied when Algorithm 4 terminates.

Let us first assume that $K_r \leq |\mathcal{N}^{(i)}|/2$ in all iterations of the algorithm. Then $\mathcal{N}' = \mathcal{N}^{(|\mathcal{R}|+1)}$ and

$$|\mathcal{N}^{(|\mathcal{R}|+1)}| \quad = \quad |\mathcal{N}^{(1)}| - (|\overline{\mathcal{Z}}^{(1)}| + |\overline{\mathcal{Z}}^{(2)}| + \ldots + |\overline{\mathcal{Z}}^{(|\mathcal{R}|)}|) \qquad (44)$$

$$= \quad |\mathcal{N}| - \sum_{r \in \mathcal{R}} K_r \tag{45}$$

which yields the first condition of ($\gamma 2$). On the other hand, if $K_r > |\mathcal{N}^{(i)}|/2$ before the ith iteration of the algorithm, then Algorithm 4 terminates after Step 5. Suppose $|\mathcal{R}^{(i)}| > 1$ after termination of the algorithm. Since Algorithm 4 always selected the routing chain with the least number of jobs, we obtain with $|\mathcal{R}^{(i)}| > 1$ and $\min_{q \in \mathcal{R}^{(i)}}\{K_q\} > |\mathcal{N}^{(i)}|/2$:

$$\sum_{q \in \mathcal{R}^{(i)}} K_q > |\mathcal{N}^{(i)}| \tag{46}$$

Since

$$|\mathcal{N}^{(i)}| = |\mathcal{N}| - \sum_{r \notin \mathcal{R}^{(i)}} K_r \tag{47}$$

we obtain with (46) and (47) that

$$\sum_{r \in \mathcal{R}} K_r > |\mathcal{N}| \tag{48}$$

which contradicts our initial assumption ($\beta 2$). Therefore, $|\mathcal{R}^{(i)}| \leq 1$. Since $K_r > |\mathcal{N}^{(i)}|/2$ results in $|\mathcal{R}^{(i)}| \geq 1$, we obtain $|\mathcal{R}^{(i)}| = 1$. Hence, in this case the second condition of ($\gamma 2$) holds. □

5.3 Proof of Theorem 2

Now we are ready to prove Theorem 2. The proof will take advantage of the lemmas in the previous subsection. As previously indicated, we first show that the capacity assignments given in Theorem 2 yield a deadlock-free network. Then we prove the minimality of the assignments Φ_i^* [6] by showing that any capacity assignment which allocates fewer capacities to any routing chain in the tandem network will result in a deadlock situation. The proof of minimality is tedious since we must distinguish a considerable number of cases, each requiring a different set of arguments.

5.3.1 Proof of Deadlock Freedom

Here we show that the capacity assignment Φ_i^* as defined in Theorem 2 yield a deadlock-free assignment. Note that in a tandem network each buffer cycle C must contain a buffer from station i. In the following we will assume without loss of generality that the buffer from station i belongs to routing chain r, that is, $i \in S_r^{(C)}$.

To prove freedom of deadlocks, we distinguish three cases: (a) $|\mathcal{N}| > K$, (b) $K - K_r \geq |\mathcal{N}|$, and (c) $K \geq |\mathcal{N}| > K - K_r$.

(a) $|\mathcal{N}| > K$

In this case, it is not possible that the servers of all stations in Γ are occupied by jobs. Hence, for each buffer cycle C there must exist a chain $r \in \mathcal{R}$ with:

$$|S_r^{(C)}| > K_r \tag{49}$$

[6] Recall that Theorem 2 defines $|\mathcal{N}|$ capacity assignments Φ_i^*, one for each station $i \in \mathcal{N}$. In the following, we assume that the index i of capacity assignment Φ_i^* is arbitrary, but fixed.

With (49), routing r satisfies the DLF condition for cycle C.

(b) $K - K_r \geq |\mathcal{N}|$
Here, Theorem 2 assigns capacities such that $\Phi_i^*(B_{ir}) = K_r$. Since per assumption we have $|S_r^{(C)}| \geq 1$, the DLF term for routing chain r in cycle C evaluates to:

$$\sum_{j \in S_r^{(C)}} \Phi_i^*(B_{jr}) + |S_r^{(C)}| \quad = \quad K_r + |S_r^{(C)}| > K_r \tag{50}$$

Clearly, the DLF condition is satisfied.

(c) $K \geq |\mathcal{N}| > K - K_r$
We first consider all cycles C with

$$i \in S_r^{(C)} \quad \text{and} \quad |S_r^{(C)}| < |\mathcal{N}| - (K - K_r) \tag{51}$$

Since all cycles in a tandem network have a length of $|\mathcal{N}|$, i.e., $\sum_{r \in \mathcal{R}} |S_r^{(C)}| = |\mathcal{N}|$, we obtain from (51) that there must exist a routing chain q $(q \neq r)$ such that $|S_q^{(C)}| > K_q$. But this satisfies the DLF condition for chain q in cycle C.
Now we consider $K \geq |\mathcal{N}| > K - K_r$ and all cycles C with

$$i \in S_r^{(C)} \quad \text{and} \quad |S_r^{(C)}| \geq |\mathcal{N}| - (K - K_r) \tag{52}$$

With $i \in S_r^{(C)}$ and $\Phi_i^*(B_{ir}) = K - |\mathcal{N}| + 1$ we obtain:

$$\begin{aligned} \sum_{j \in S_r^{(C)}} \Phi_i^*(B_{jr}) + |S_r^{(C)}| \quad &\geq \quad (K - |\mathcal{N}| + 1) + (|\mathcal{N}| - (K - K_r)) &(53)\\ &= \quad K_r + 1 &(54)\\ &> \quad K_r &(55) \end{aligned}$$

Thus, the DLF condition for routing chain r in cycle C is satisfied. □

5.3.2 Proof of Minimality

For $K < |\mathcal{N}|$, no capacity is assigned to any buffer. Therefore, Φ_i^* is certainly minimal. In the following, we investigate the minimality of Φ_i^* for $K \geq |\mathcal{N}|$.

To show minimality, we have to prove that any capacity assignment which assigns less total capacity to the stations of the tandem network than Φ_i^* will result in a deadlock situation. Let us select an arbitrary routing chain $r \in \mathcal{R}$ and consider a capacity assignment $\overline{\Phi}$ which satisfies:

$$\sum_{j \in \mathcal{N}} \overline{\Phi}(B_{jq}) = \begin{cases} \sum_{j \in \mathcal{N}} \Phi_i^*(B_{jq}) - 1 & \text{if } q = r \\ \sum_{j \in \mathcal{N}} \Phi_i^*(B_{jq}) & \text{if } q \neq r \end{cases} \tag{56}$$

Thus, $\overline{\Phi}$ assigns one buffer space less to the buffers of routing chain r than capacity assignment Φ_i^*. The total capacity assigned to buffers from chains $q \neq r$ by $\overline{\Phi}$ remains unchanged. Note that we do not have any assumptions on how $\overline{\Phi}$ distributes the capacities to the stations of the tandem network. In particular, we do not assume that $\overline{\Phi}$ assigns non-zero capacities to only one station. For $K \geq |\mathcal{N}|$, we will show that assignment $\overline{\Phi}$ results in a deadlock in at least one buffer cycle in Γ.

Recall that in Theorem 2, the capacities assigned to buffers from routing chain r are different for $K - K_r \geq |\mathcal{N}|$ and $K - K_r < |\mathcal{N}|$. We will prove each case separately.

(a) $K - K_r \geq |\mathcal{N}|$

Consider the tandem network $\Gamma' = (\mathcal{N}, \mathcal{R} \setminus \{r\})$ which is obtained from Γ by eliminating routing chain r. Further consider capacity assignment $\overline{\Phi}$ as given in (56). Next we execute Algorithm 3 on Γ'. Since Γ' satisfies conditions ($\alpha 1$) and ($\beta 1$) in Lemma 6, the output of Algorithm 3 is a network $\Gamma'' = (\mathcal{N}'', \mathcal{R}'')$ with $\mathcal{N}'' \subseteq \mathcal{N}$ and $\mathcal{R}'' \subseteq \mathcal{R} \setminus \{r\}$. Per Lemma 6 there exists a routing chain $q \in \mathcal{R}''$ with $K_q \geq |\mathcal{N}''|$.

Now consider the tandem network $\Gamma''' = (\mathcal{N}'', \{r, q\})$. We will construct a deadlock in Γ'''. With Lemma 6, any deadlock in Γ''' can be extended to a deadlock in Γ. Dependent on the size of K_r relative to $|\mathcal{N}''|$, the construction of the deadlock in Γ''' will be different for $K_r \geq |\mathcal{N}''|$ and $K_r < |\mathcal{N}''|$.

$\underline{K_r \geq |\mathcal{N}''|}$:

Note that $K - K_q \geq |\mathcal{N}|$ must hold in this case. ($K - K_q < |\mathcal{N}|$ implies $\sum_{s \in \mathcal{R}''} K_s + K_r - K_q < |\mathcal{N}''|$ which contradicts $K_r \geq |\mathcal{N}''|$.) Therefore, capacity assignment $\overline{\Phi}$ assigns the following total capacities to buffers from routing chains r and q:

$$\sum_{j \in \mathcal{N}} \overline{\Phi}(B_{jr}) = K_r - 1 \tag{57}$$

$$\sum_{j \in \mathcal{N}} \overline{\Phi}(B_{jq}) = K_q \tag{58}$$

We apply Lemma 5 repeatedly to chains r and q until $K_r = |\mathcal{N}''|$ and $K_q = |\mathcal{N}''|$. Then, by applying Lemma 4 we can effectively partition $\mathcal{N}''$ into two sets $\mathcal{N}_1''$ and $\mathcal{N}_2''$ such that chain r does not satisfy the DLF condition in either subset.

Now consider two cycles C_1 and C_2 with:

$$\begin{array}{ll} S_r^{(C_1)} = |\mathcal{N}_1''| & S_q^{(C_1)} = |\mathcal{N}_2''| \\ S_r^{(C_2)} = |\mathcal{N}_2''| & S_q^{(C_2)} = |\mathcal{N}_1''| \end{array} \tag{59}$$

Per construction of C_1 and C_2, the DLF condition for chain r is not satisfied in either cycle. With Lemma 3 the DLF condition for chain q cannot be satisfied in both cycles C_1 and C_2. Therefore, the DLF conditions for both routing chains are violated in C_1 or C_2. Thus, either cycle C_1 or C_2 contains a deadlock. Since the modifications to Γ have preserved any possibly existing deadlock (see Lemmas 5 and 6), one of the two cycles C_1 or C_2 can be extended to a buffer cycle in Γ such that no routing chain $r \in \mathcal{R}$ satisfies the DLF condition of this cycle. As a result, we have constructed a deadlock in Γ.

$\underline{K_r < |\mathcal{N}''|}$:

If $K_r < |\mathcal{N}''|$ holds, both $K - K_q \geq |\mathcal{N}|$ and $K - K_q < |\mathcal{N}|$ are feasible depending on the outcome of Algorithm 3. We will only show the proof $K - K_q < |\mathcal{N}|$ and assume that Algorithm 3 has executed Step 9 in its last iteration (The proof is similar if Step 11 is executed in the last iteration of Algorithm 3). With these assumptions, the following holds for the number of stations in Γ'' and Γ''':

$$|\mathcal{N}''| = |\mathcal{N}| - \sum_{s \neq r,q} K_s \tag{60}$$

Since per our assumptions, $K - K_r \geq |\mathcal{N}|$ and $K - K_q < |\mathcal{N}|$, capacity assignment $\overline{\Phi}$ assigns the following total capacities to routing chains r and q:

$$\begin{aligned} \sum_{i \in \mathcal{N}} \overline{\Phi}(B_{ir}) &= K_r - 1 & (61) \\ \sum_{i \in \mathcal{N}} \overline{\Phi}(B_{iq}) &= K - |\mathcal{N}| + 1 & (62) \\ &= K_r + K_q - |\mathcal{N}''| + 1 & (63) \end{aligned}$$

The equality in (63) is obtained with (60). Next we apply Lemma 5 repeatedly to chain q until $K_q = |\mathcal{N}''|$. Thus, Lemma 5 is applied exactly $K_q - |\mathcal{N}''|$ times. Applying Lemma 5 changes the capacity assignment to chain q, and we use $\overline{\Phi}'$ to refer to the modified capacity assignment. $\overline{\Phi}'$ is such that:

$$\begin{aligned} \sum_{j \in \mathcal{N}} \overline{\Phi}'(B_{jr}) &= \sum_{j \in \mathcal{N}} \overline{\Phi}(B_{jr}) & (64) \\ \sum_{j \in \mathcal{N}} \overline{\Phi}'(B_{jq}) &= \sum_{j \in \mathcal{N}} \overline{\Phi}(B_{jq}) - (K_q - |\mathcal{N}''|) & (65) \\ &= (K_r + K_q - |\mathcal{N}''| + 1) - (K_q - |\mathcal{N}''|) & (66) \\ &= K_r + 1 & (67) \end{aligned}$$

Let us define $\mathcal{NZ}_q$ as the set of stations in Γ''' with non-zero capacities assigned to buffers of chain q. With (67) we obtain the following upper bound for $|\mathcal{NZ}_q|$:

$$|\mathcal{NZ}_q| \quad \leq \quad K_q + 1 \tag{68}$$

We select a set $\overline{\mathcal{NZ}}_q$ with $\mathcal{NZ}_q \subseteq \overline{\mathcal{NZ}}_q$ such that

$$|\overline{\mathcal{NZ}}_q| = K_r + 1 \tag{69}$$

Since $\sum_{j \in \mathcal{N}} \overline{\Phi}(B_{jr}) = K_r - 1$, and since $\sum_{j \in \overline{\mathcal{NZ}}_q} \overline{\Phi}(B_{jr}) \leq \sum_{j \in \mathcal{N}} \overline{\Phi}(B_{jr})$ we have:

$$\sum_{j \in \overline{\mathcal{NZ}}_q} \overline{\Phi}(B_{jr}) = K_r - 1 \tag{70}$$

Hence, we can apply Lemma 4 and partition $\overline{\mathcal{NZ}}_q$ into two sets $\overline{\mathcal{NZ}}_{q1}$ and $\overline{\mathcal{NZ}}_{q2}$ such that

$$\sum_{j \in \overline{\mathcal{NZ}}_{q1}} \overline{\Phi}'(B_{jr}) + |\overline{\mathcal{NZ}}_{q1}| \leq K_r \quad \text{and} \quad \sum_{j \in \overline{\mathcal{NZ}}_{q2}} \overline{\Phi}'(B_{jr}) + |\overline{\mathcal{NZ}}_{q2}| \leq K_r \tag{71}$$

For routing chain q, we have with (67) that $\sum_{j\in\overline{\mathcal{NZ}}_q}\overline{\Phi}'(B_{jq}) = K_r + 1$. Thus, we obtain with Lemma 3 that either

$$\sum_{j\in\overline{\mathcal{NZ}}_{q1}}\overline{\Phi}'(B_{jq}) + |\overline{\mathcal{NZ}}_{q1}| \le K_r + 1 \quad \text{or} \quad \sum_{j\in\overline{\mathcal{NZ}}_{q2}}\overline{\Phi}'(B_{jq}) + |\overline{\mathcal{NZ}}_{q2}| \le K_r + 1 \tag{72}$$

Without loss of generality we will assume that the first inequality in (72) holds. Then we can perform the following manipulations:

$$\begin{aligned}
\sum_{j\in\mathcal{N}''\setminus\overline{\mathcal{NZ}}_{q2}}\overline{\Phi}'(B_{jq}) + |\mathcal{N}''\setminus\overline{\mathcal{NZ}}_{q2}| &= \sum_{j\in\mathcal{N}''\setminus\overline{\mathcal{NZ}}_{q2}}\overline{\Phi}'(B_{jq}) + |\overline{\mathcal{NZ}}_{q1}| + |\mathcal{N}''| - |\overline{\mathcal{NZ}}_q| && (73)\\
&= \sum_{j\in\overline{\mathcal{NZ}}_{q1}}\overline{\Phi}'(B_{jq}) + |\overline{\mathcal{NZ}}_{q1}| + |\mathcal{N}''| - (K_r+1) && (74)\\
&\le (K_r+1) + |\mathcal{N}''| - (K_r+1) && (75)\\
&= |\mathcal{N}''| && (76)
\end{aligned}$$

Equation (73) follows from $|\overline{\mathcal{NZ}}_q| = |\overline{\mathcal{NZ}}_{q1}| + |\overline{\mathcal{NZ}}_{q2}|$. Equation (74) follows with (69) and $\sum_{j\in\mathcal{N}''\setminus\overline{\mathcal{NZ}}_q}\overline{\Phi}'(B_{jq}) = 0$. The inequality in (74) is obtained with (72), and (76) is obtained by canceling terms. With the above derivations, we can define a buffer cycle C in Γ''' such that:

$$S_r^{(C)} = \overline{\mathcal{NZ}}_{q2} \qquad S_q^{(C)} = \mathcal{N}''\setminus\mathcal{NZ}_{q2} \tag{77}$$

It follows from (71), and (72) – (76), that the DLF conditions for both routing chains r and q are not satisfied. Therefore, a deadlock results in network Γ'''. With the same arguments as used before, the deadlock in Γ''' results in a deadlock in Γ.

(b) $K - K_r < |\mathcal{N}|$

Again, we consider the capacity assignment $\overline{\Phi}$ for Γ as given in (56). Recall that $\overline{\Phi}$ assigns the same total capacities to the buffers of routing chains $q \ne r$ as Φ_i^*, but assigns one buffer capacity less to routing chain r. Since $K - K_r < |\mathcal{N}|$ we have that:

$$\sum_{j\in\mathcal{N}}\overline{\Phi}(B_{jq}) = K - |\mathcal{N}| \tag{78}$$

Consider the tandem network $\Gamma' = (\mathcal{N}, \mathcal{R}\setminus\{r\})$, which is obtained from Γ by eliminating routing chain r. Since Γ' satisfies conditions ($\alpha 2$) and ($\beta 2$) from Lemma 7, we execute Algorithm 4 on Γ' and obtain a tandem network $\Gamma'' = (\mathcal{N}'', \mathcal{R}'')$. Recall from Lemma 7 that any deadlock in Γ'' can be extended to a deadlock in Γ'. Algorithm 4 has two feasible outcomes (see condition ($\gamma 2$) in Lemma 7):

(a) $|\mathcal{N}''| = |\mathcal{N}| - \sum_{q\in\mathcal{R}\setminus\{r\}} K_q$, or
(b) $|\mathcal{R}''| = 1$ and $K_q \ge |\mathcal{N}'|/2$ for $q \in \mathcal{R}''$.

In the following we construct a deadlock situation for chains r and q in the tandem network $\Gamma''' = (\mathcal{N}'', \mathcal{R}''\cup\{r\})$. First we assume outcome (a) of Algorithm 4. Here, we construct a buffer cycle C in Γ''' which consists exclusively of buffers from routing chain r, i.e., $S_r^{(C)} = \mathcal{N}''$.

The DLF term for routing chain r in cycle C can be evaluated as follows:

$$\begin{aligned}\sum_{j\in\mathcal{N}''}\overline{\Phi}(B_{jr})+|\mathcal{N}''| &\leq (K-|\mathcal{N}|)+|\mathcal{N}''| \qquad (79)\\ &= (K-|\mathcal{N}|)+(|\mathcal{N}|-\sum_{q\in\mathcal{R}\setminus\{r\}}K_q) \qquad (80)\\ &= K_r \qquad (81)\end{aligned}$$

To obtain (79) we have used

$$\sum_{j\in\mathcal{N}''}\overline{\Phi}(B_{jr})\leq\sum_{j\in\mathcal{N}}\overline{\Phi}(B_{jr})=K-|\mathcal{N}| \qquad (82)$$

Applying the assumption on the outcome of Algorithm 4, that is, $|\mathcal{N}''|=|\mathcal{N}|-\sum_{q\in\mathcal{R}\setminus\{r\}}K_q$, yields (80). Finally, (81) follows from $K=\sum_{q\in\mathcal{R}}K_q$. The derivations in equations (79) – (81) show that chain r does not satisfy the DLF condition in cycle C, and we have constructed a deadlock in Γ'''. With Lemma 7 we also obtain a deadlock in Γ.

Now we consider the outcome (b) of Algorithm 4, i.e., $\mathcal{R}''=\{q\}$ and $K_q\geq|\mathcal{N}''|/2$. Note that Algorithm 4 yields the following equalities:

$$\begin{aligned}|\mathcal{N}''| &= |\mathcal{N}|-\sum_{s\neq q,r}K_s \qquad (83)\\ K-|\mathcal{N}| &= K_r+K_q-|\mathcal{N}''| \qquad (84)\end{aligned}$$

From equation (84) and $K-K_r<|\mathcal{N}|$, it follows that $K_q<|\mathcal{N}|$. With Lemma 7 we can construct a deadlock in Γ by finding a buffer cycle in $\Gamma'''=(\mathcal{N}'',\{q,r\})$ where neither chain r nor chain q satisfy the DLF condition. The construction of this buffer cycle is dependent on the size of K_r relative to $|\mathcal{N}''|$. Here we only show the proof for $K_r\geq|\mathcal{N}''|$.

If $K_r\geq|\mathcal{N}''|$, then $K-K_q\geq|\mathcal{N}|$ must hold. Otherwise, we obtain a contradiction with $K_r>K_r$. Hence, capacity assignment $\overline{\Phi}$ assigns the following capacities to routing chains r and q:

$$\begin{aligned}\sum_{j\in\mathcal{N}}\overline{\Phi}(B_{jr}) &= K-|\mathcal{N}|=K_r+K_q-|\mathcal{N}''| \qquad (85)\\ \sum_{j\in\mathcal{N}}\overline{\Phi}(B_{jq}) &= K_q \qquad (86)\end{aligned}$$

We apply Lemma 5 repeatedly to chain r until $K_r=|\mathcal{N}''|$. Note that Lemma 5 is applied exactly $K_r-|\mathcal{N}''|$ times. Let us refer to the capacity assignment obtained by applying Lemma 5 as $\overline{\Phi}'$. The assignments of capacities to chain q have not changed, i.e., $\overline{\Phi}'(B_{iq})=\overline{\Phi}(B_{iq})$ for all $i\in\mathcal{N}$. However, $\overline{\Phi}'$ changes the total capacities assigned to buffers from chain r to:

$$\begin{aligned}\sum_{j\in\mathcal{N}}\overline{\Phi}'(B_{jr}) &= (K_r+K_q)-|\mathcal{N}''|-(K_r-|\mathcal{N}''|) \qquad (87)\\ &= K_q \qquad (88)\end{aligned}$$

Summarizing the last steps, we have constructed a network $\Gamma''' = (\mathcal{N}'', \{q, r\})$ and a capacity assignment $\overline{\Phi}'$ with

$$\begin{array}{ll} K_r = |\mathcal{N}''| & \sum_{j \in \mathcal{N}} \overline{\Phi}'(B_{jr}) = K_q \\ |\mathcal{N}''|/2 < K_q < |\mathcal{N}''| & \sum_{j \in \mathcal{N}} \overline{\Phi}'(B_{jq}) = K_q \end{array} \tag{89}$$

Recall that any deadlock in this network will result in a deadlock in Γ with capacity assignment $\overline{\Phi}$. Now we select a subset $\mathcal{N}_1 \subset \mathcal{N}$ of stations with $|\mathcal{N}_1| = |\mathcal{N}''| - K_q$ such that:

$$\sum_{j \in \mathcal{N}_1} \overline{\Phi}'(B_{jr}) = 0 \tag{90}$$

$$\sum_{j \in \mathcal{N}_1} \overline{\Phi}'(B_{jq}) > 0 \tag{91}$$

From (89), it follows that this selection is always feasible. We denote by $\mathcal{N}_2$ the set of stations not in $\mathcal{N}_1$, i.e., $\mathcal{N}_2 = \mathcal{N}'' \setminus \mathcal{N}_1$. Since $|\mathcal{N}_2| = K_q$ holds, and since equation (91) implies that $\sum_{i \in \mathcal{N}_2} \overline{\Phi}'(B_{iq}) < K_q$, we can use Lemma 4 to partition $\mathcal{N}_2$ into two subsets $\mathcal{N}_{21}$ and $\mathcal{N}_{22}$ such that:

$$\sum_{j \in \mathcal{N}_{21}} \overline{\Phi}'(B_{jq}) + |\mathcal{N}_{21}| \leq K_q \tag{92}$$

$$\sum_{j \in \mathcal{N}_{22}} \overline{\Phi}'(B_{jq}) + |\mathcal{N}_{22}| \leq K_q \tag{93}$$

Now consider the following two buffer cycles C_1 and C_2 in Γ''':

$$\begin{array}{ll} S_q^{(C_1)} = |\mathcal{N}_{21}| & S_r^{(C_1)} = |\mathcal{N}'' \setminus \mathcal{N}_{21}| = |\mathcal{N}_1 \cup \mathcal{N}_{22}| \\ S_q^{(C_2)} = |\mathcal{N}_{22}| & S_r^{(C_2)} = |\mathcal{N}'' \setminus \mathcal{N}_{22}| = |\mathcal{N}_1 \cup \mathcal{N}_{21}| \end{array} \tag{94}$$

With (92) and (93), routing chain q does not satisfy the DLF condition in either cycle. Therefore, routing chain r must satisfy the DLF condition in both C_1 and C_2. Let us assume chain r satisfies the DLF condition in cycle C_1, that is,

$$\sum_{j \in \mathcal{N}_1 \cup \mathcal{N}_{22}} \overline{\Phi}'(B_{jr}) + |\mathcal{N}_1| + |\mathcal{N}_{22}| > K_r \tag{95}$$

Note that with (90) we have

$$\sum_{i \in \mathcal{N}_1 \cup \mathcal{N}_{22}} \overline{\Phi}'(B_{ir}) = \sum_{i \in \mathcal{N}_{22}} \overline{\Phi}'(B_{ir}) \tag{96}$$

With $|\mathcal{N}_1| = |\mathcal{N}''| - K_q$, $K_r = |\mathcal{N}''|$, and (96), equation (95) is equivalent to

$$\sum_{i \in \mathcal{N}_{22}} \overline{\Phi}'(B_{ir}) + |\mathcal{N}_{22}| > K_q \tag{97}$$

Then, the DLF term for routing chain r in cycle C_2 evaluates as follows:

$$\begin{aligned} \sum_{i \in \mathcal{N}_1 \cup \mathcal{N}_{21}} \overline{\Phi}'(B_{ir}) + |\mathcal{N}_1| + |\mathcal{N}_{21}| &= (K_q - \sum_{i \in \mathcal{N}_{22}} \overline{\Phi}'(B_{ir})) + (|\mathcal{N}''| - |\mathcal{N}_{22}|) && (98) \\ &< |\mathcal{N}''| && (99) \\ &= K_r && (100) \end{aligned}$$

Equality in (98) follows from (86) and $|\mathcal{N}''| = |\mathcal{N}_{21}| + |\mathcal{N}_{22}| + |\mathcal{N}_1|$. Equation (99) is obtained with (97), and (100) is correct since (89) implies $K_r = |\mathcal{N}''|$. Hence, if routing chain r satisfies the DLF condition in cycle C_1, it does not satisfy the DLF condition in cycle C_2. On the other hand, if we assume that chain r satisfies the DLF condition in cycle C_2, equivalent derivations as in (95) – (100) yield that chain r cannot satisfy the DLF condition in cycle C_1. Therefore, either cycle C_1 or cycle C_2 contains a deadlock in Γ'''. Since all manipulations of Γ where such that any deadlock Γ''' can be extended to a deadlock in Γ, we follow that Γ also contains a deadlock. □

6 Conclusions

Deadlock situations in blocking networks with multiple routing chains are difficult to detect since deadlocks may occur due to interdependencies between jobs from different routing chains. In this study, we presented necessary and sufficient conditions for deadlock-free blocking networks with multiple routing chains. We addressed the problem of finding a deadlock-free capacity assignment for a network with the least number of buffer spaces (*minimal capacity assignment*). We presented an algorithm which generates a minimal assignment for multiple chain networks without restrictions on the network topology. A drawback of the optimization algorithm is that it involves the solution of an integer program. To avoid the high computational complexity involved in solving integer programs, we offered an efficient heuristic algorithm which always provides a deadlock-free capacity assignment, but does not guarantee minimality. For a special class of network topologies, so-called *tandem networks*, we proved that a minimal capacity assignment can be given without running any optimization algorithm. We proved the correctness of a minimal capacity assignment for tandem networks which assigns non-zero capacities to only one station in the network.

Our work can be extended in several directions. In our study, we assume the most common blocking mechanism, i.e., blocking-after-service or BAS. Many systems are more realistically modeled with different blocking mechanisms, e.g., blocking-before-service or repeated-service [1]. For these blocking mechanisms, deadlock freedom conditions – different from the conditions in this study – must be found. We also assume that all stations have a single server, and that each routing chain has a separate buffer at a station. Extensions of our work could consider stations with multiple servers, and stations where the available buffers are shared by all routing chains.

References

[1] I. F. Akyildiz and H. G. Perros (Editors). Queueing Networks with Finite Capacity Queues. *Special Issue of Performance Evaluation*, 10(4), 1989.

[2] I. F. Akyildiz and J. Liebeherr. Optimal Deadlock Free Buffer Allocation in Multiple Chain Blocking Networks of Queues. In *Proc. IFIP TC7/WG 7.3 International Conference on Performance of Distributed Systems and Integrated Communication Networks*, pages 217–236, Kyoto, Japan, 1991.

[3] B. Johnson. Finding All the Elementary Circuits in a Directed Graph. *SIAM Journal on Computing*, 4(1):77 – 84, 1975.

[4] S. Kundu and I. F. Akyildiz. Deadlock Free Buffer Allocation in Closed Queueing Networks. *Queueing Systems*, 4(1):47 – 56, 1989.

[5] J. Liebeherr. *Performance Evaluation of Synchronized Systems.* PhD thesis, Georgia Institute of Technology, College of Computing, 1991.

[6] J. Liebeherr. Deadlock-Free Buffer Allocations in Blocking Networks with Tandem Sequences. In *Proc. 2nd Workshop on Queueing Networks with Finite Capacities*, pages 41 – 54. North-Holland, 1993.

[7] R. O. Onvural. Survey of Closed Queueing Networks with Blocking. *ACM Computing Surveys*, 22(2):83–121, 1990.

[8] L. Schrage. *Linear, Integer, and Quadratic Programming with LINDO.* The Scientific Press, Third Edition, 1986.