Otten, Sonja; Krenzler, Ruslan; Xie, Lin; Daduna, Hans; Kruse, Karsten

Analysis of semi-open queueing networks using lost customers approximation with an application to robotic mobile fulfilment systems

OR Spectrum

Provided in Cooperation with:
Springer Nature

Suggested Citation: Otten, Sonja; Krenzler, Ruslan; Xie, Lin; Daduna, Hans; Kruse, Karsten (2021) : Analysis of semi-open queueing networks using lost customers approximation with an application to robotic mobile fulfilment systems, OR Spectrum, ISSN 1436-6304, Springer, Berlin, Heidelberg, Vol. 44, Iss. 2, pp. 603-648, https://doi.org/10.1007/s00291-021-00662-9

OR Spectrum (2022) 44:603–648
https://doi.org/10.1007/s00291-021-00662-9

ORIGINAL ARTICLE

Analysis of semi-open queueing networks using lost customers approximation with an application to robotic mobile fulfilment systems

Sonja Otten[1,2] · **Ruslan Krenzler**[1] · **Lin Xie**[1] · **Hans Daduna**[3] · **Karsten Kruse**[2]

Received: 20 October 2020 / Accepted: 15 November 2021 / Published online: 16 December 2021

Abstract

We consider a semi-open queueing network (SOQN), where one resource from a resource pool is needed to serve a customer. If on arrival of a customer some resource is available, the resource is forwarded to an inner network to complete the customer's order. If no resource is available, the new customer waits in an external queue until one becomes available ("backordering"). When a resource exits the inner network, it is returned to the resource pool. We develop a new solution approach. In a first step we modify the system such that new arrivals are lost if the resource pool is empty ("lost customers"). We adjust the arrival rate of the modified system such that the throughputs in all nodes of the inner network are pairwise identical to those in the original network. Using queueing theoretical methods, in a second step we reduce this inner network to a two-station system including the resource pool. For this two-station systems, we invert the first step and obtain a standard SOQN which can be solved analytically. We apply our results to storage and delivering systems with robotic mobile fulfilment systems (RMFSs). Instead of sending pickers to the storage area to search for the ordered items and pick them, robots carry shelves with ordered items from the storage area to picking stations. We model the RMFS as an SOQN to determine the minimal number of robots.

Keywords Semi-open queueing network · Backordering · Lost customers · Product form approximation · Robotic mobile fulfilment system · Warehousing

✉ Sonja Otten
sonja.otten@tuhh.de

[1] Leuphana University of Lüneburg, Lüneburg, Germany

[2] Hamburg University of Technology, Hamburg, Germany

[3] Universität Hamburg, Hamburg, Germany

1 Introduction

Queueing networks can be classified as follows according to (Chen and Yao 2001, p. 21ff.), Roy (2016), Azadeh et al. (2017b): open queueing networks (OQN), where customers arrive from the exterior, request service at several nodes and then leave the system; closed queueing networks (CQN), without external arrivals and departures, and a fixed number of customers; semi-open queueing networks (SOQN), which have characteristics of both OQNs and CQNs (Jia and Heragu 2009). An SOQN resembles an OQN because customers arrive from the exterior and leave the system after service. It resembles a CQN because there is an overall capacity constraint for the network, which is realized as follows: Any customer needs a resource from an associated resource pool for service. If on arrival of a customer a resource is available, the resource enters the network (which for definiteness will be termed "inner network") to complete the customer's order. If there is no resource available, the new customer has to wait in an external queue until one becomes available. When a resource exits the inner network, it returns to the resource pool.

SOQNs are adopted for performance analysis of manufacturing systems and service systems, e.g. in logistics, communication, warehousing, health care (Roy 2016).

We focus on SOQNs where the inner network consists of exponential single server nodes with infinite waiting room under either FCFS regime or processor sharing regime. The literature on SOQNs is overwhelming, so we point only to the most relevant sources for our present investigation. An overview of SOQNs and solution methods is available in Jia and Heragu (2009), Ekren et al. (2014) and Roy (2016). Roy (2016) also compares numerical accuracy of several methods. A recent article, not included in these reviews, is Kim et al. (2018). The most common solution approaches are matrix-geometric method, aggregation method, network decomposition approach, parametric decomposition method and performance bounds. For SOQNs with an inner network consisting of more than one node, closed-form expressions for steady-state distributions are not available.

We develop a new solution approach, visualised in Fig. 1: (i) We modify the system such that new arrivals are lost if the resource pool is empty ("lost customers"). For this modification, closed-form expressions for the steady-state distribution in product form are available. We adjust the arrival rate at the modified system such that the throughputs in all nodes of the resource network (= inner network + resource pool) are pairwise identical to those in the original network. We also prove that idling probabilities for nodes with constant service rates are pairwise identical. Moreover, we provide closed-form expressions for throughputs and for these idling probabilities. (ii) Using queueing theoretical methods (Norton's Theorem), we reduce this inner network to a two-station system including the resource pool. (iii) For this system, we invert step (i) and obtain a standard SOQN with a two-station resource network and an external queue which can be solved analytically.

Step (ii) resembles the standard application of matrix-geometrical methods (MGM) as surveyed in Roy (2016). The inner network is converted into a

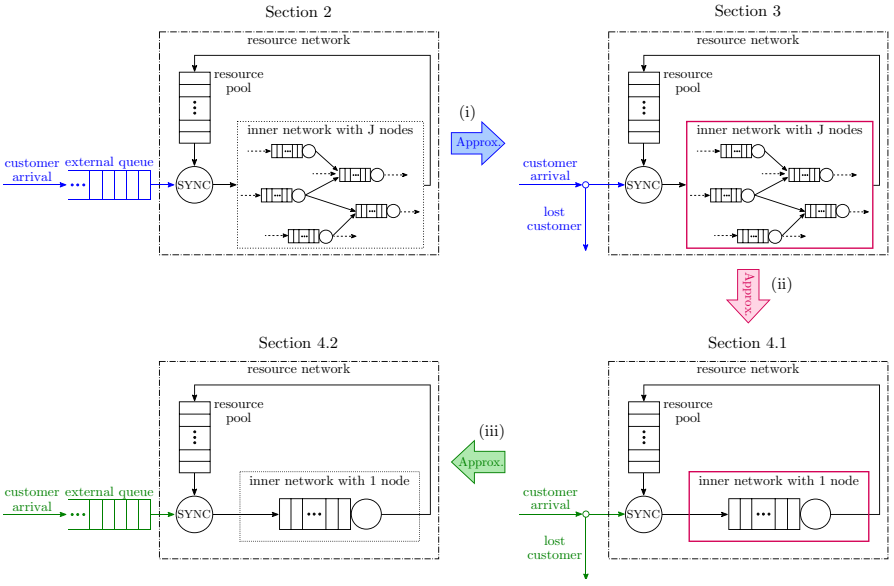

Fig. 1 Overview of the models. We use the same colour for parts, which we change in a single approximation step

two-station network by aggregation. "The challenge...lies in decomposing the original subnetwork...into two subnetworks" (Roy 2016, p. 1742), to be aggregated. Several recipes are provided. While the aggregations use heuristics, our use of Norton's Theorem is exact because of the product form distribution behind, and we escape from the decomposition problem because we end up with a single node. The MGM as developed and used (e.g.) in Jia and Heragu (2009), Ekren et al. (2014), Lamballais et al. (2017) and Buitenhek et al. (2000) requires as final step a numerical approach to solve for steady-state distributions. Our final step uses steady-state distributions obtained in closed form for computing performance indices.

Convention To distinguish the versions of SOQNs which we elaborate on, we use terms borrowed from inventory theory, where unsatisfied demand can be either "backordered" or is called "lost sales" if backordering is not possible. SOQN-BO denotes a standard SOQN with external queue for waiting customers when the resource pool is empty (BO = backordering), and SOQN-LC an SOQN with lost sales (LC = lost customers). The first version is often called SOQN with infinite buffer, the second one SOQN with finite buffer.

Remark The approximation of BO-systems by LC-systems follows (Krenzler 2016, Section 2.1.4, p. 34ff.). In the literature, results for queueing networks with lost customers are available, e.g. lost sales models in Otten (2018), Schwarz et al. (2006) and Krishnamoorthy et al. (2011). For a review of systems with lost sales, we refer to Bijvank and Vis (2011).

We apply our new solution approach to the investigation of robotic mobile fulfilment systems (RMFSs), such as the Kiva System (nowadays Amazon Robotics), the GreyOrange Butler or the Swisslog CarryPick. RMFSs are a new type of warehousing system

which have received attention recently, due to increasing growth in the e-commerce sector. The principle of RMFS (see Fig. 7) is, instead of sending pickers to the storage area to search for ordered items and pick them, to carry shelves—called *pods*—with ordered items by robots from the storage area to picking stations. At every picking station resides a picker, a person who takes items from the pods and packs them into boxes according to customers' orders. When the picker does not need the pod any longer, the robot either transports the pod directly back to the storage area or first makes a stopover at a replenishment station. RMFSs pose many hard decision problems at strategic, tactical and operational levels. An overview is included in (Merschformann et al. 2019, Section 4). The literature on RMFS is, similar to that on SOQNs, already overwhelming, so we only point to some references closely related to our investigations. RMFSs are modelled as SOQNs in several articles. For an overview, we refer to (Azadeh et al. 2017a, Section 7.2 and Table 4) and (Azadeh et al. 2017b, Section 6). There articles are classified according to the decision problem of interest and the relevant methodology. Recent overviews are presented in Lamballais et al. (2019) and (Boysen et al. 2019, Section 7). Most articles analyse decision problems different from what we look at in this paper: Decisions on optimal number of robots. A similar problem is tackled in Yuan and Gong (2017) where two protocols are compared, pooled and dedicated robots using OQNs instead of SOQNs, for a model without replenishment station. In Zou et al. (2018), the optimal number of robots is determined for different battery recovery strategies using a nested SOQN. Different types of warehousing systems use also semi-open or open queueing models, e.g. Ekren and Akpunar (2021) and Ekren et al. (2013).

Main contributions of the paper

- We determine closed-form expressions for stability, throughputs and some idling probabilities in an SOQN.
- We develop a new approximation method for an SOQN to determine general performance metrics.
- We model an RMFS as an SOQN and apply our method to calculate the optimal number of robots. We develop a simulation tool for a queueing model of the RMFS to analyse the quality of our approximation method.

Structure of the paper

In Sect. 2, we describe a general SOQN (=SOQN-BO) and analyse its stability. In Sect. 2.3, we determine throughputs and idle probabilities in steady state. In Sect. 3, we introduce our new approximation method for SOQNs, the approach is depicted in Fig. 1. In Sect. 3.1, we analyse a modification of the SOQN-BO, where newly arriving customers are lost, if the resource pool is empty (SOQN-LC). In Sect. 3.2, we adjust the input of the SOQN-LC to meet the throughput of the original SOQN-BO. In Sect. 3.3, we calculate node-throughputs and idle probabilities in steady state. We reduce the complexity of the modified SOQN-LC using Norton's theorem in Sect. 4.1, and we reinvent the external queue in Sect. 4.2. In Sect. 5, we model an RMFS as an SOQN and apply our results to determine the optimal number of robots. We formulate an algorithm to calculate the minimal number of robots for stability and present numerical examples. Furthermore, we discuss the limitations of the approximation. Section 6 summarises our conclusions.

Notation and preliminaries

- $\mathbb{N} := \{1, 2, 3, \dots\}, \mathbb{N}_0 := \{0\} \cup \mathbb{N}, \mathbb{R}_0^+ := [0, \infty)$ and $\mathbb{R}^+ := (0, \infty)$.
- The vector $\mathbf{0}$ is a row vector of appropriate size with all entries equal to 0. The vector $\mathbf{e}$ is a column vector of appropriate size with all entries equal to 1. The vector $\mathbf{e}_i = (0, \dots, 0, \underbrace{1}_{i\text{-th element}}, 0, \dots, 0)$ is a vector of appropriate size.
- The function $1_{\{expression\}}$ is 1 if *expression* is true and 0 otherwise.
- Empty sums are 0, and empty products are 1.
- We call a "generator" a matrix $M \in \mathbb{R}^{K \times K}$ with countable index set K if all its off-diagonal elements are non-negative and all its row sums are zero.
- "Markov process" means a continuous-time homogeneous strong Markov process with discrete state space, which is regular with cadlag paths (right-continuous, left limits everywhere). A Markov process is regular if it is non-explosive (i.e. the sequence of jump times of the process diverges almost surely) and its transition intensity matrix is a generator.

2 SOQN with backordering

2.1 Description of the model

An SOQN (*SOQN-BO*) consists of a queueing network ("inner network"), a resource pool, and an external queue (Fig. 2). Customers arrive according to a Poisson process with rate $\lambda_{\mathrm{BO}} > 0$. Every customer requires exactly one resource from the resource pool for service. If on arrival of a customer a resource is available, the resource enters the inner network to complete the customer's order. If on arrival no resource is available, the new customer waits in the external queue under the first-come, first-served (FCFS) regime until a resource becomes available (backordering).

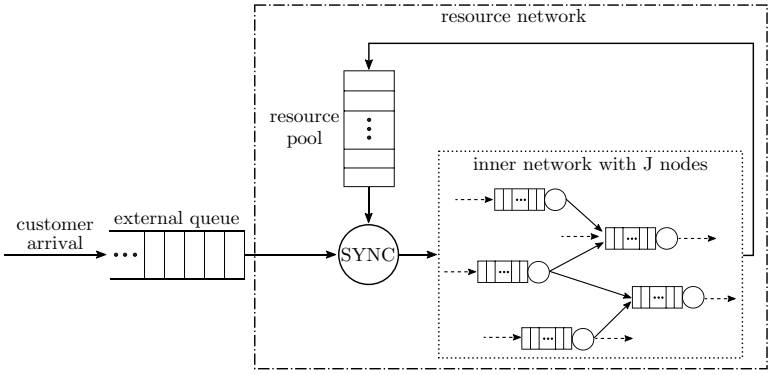

Fig. 2 An SOQN with backordering and external queue

When the resource exits the inner network, it returns to the resource pool (referred to as node 0) and waits for the next customer. Whenever the external queue is not empty and a resource item is returned to the resource pool, this item is instantaneously synchronised with the customer at the head of the line. The resources therefore move in a closed network, called *resource network*. The maximal number of resources in the resource pool is N.

The inner network consists of $J \geq 1$ numbered service stations (nodes), denoted by $\overline{J} := \{1, \ldots, J\}$. Each station j consists of a single server with infinite waiting room under FCFS regime or processor sharing regime. Customers in the network are indistinguishable. The service times are exponentially distributed random variables with mean 1. If there are $n_j > 0$ customers present at node j, service at node j is provided with intensity $\nu_j(n_j) > 0$. All service and inter-arrival times constitute an independent family of random variables.

Movements of resources in the inner network are governed by a Markovian routing mechanism: After synchronisation with a customer, a resource visits node j with probability $r(0,j) \geq 0$. A resource, when leaving node i, selects with probability $r(i,j) \geq 0$ to visit node j next, and then enters node j immediately. It starts service if it finds the server idle, otherwise it joins the tail of the queue at node j. This resource can also leave the inner network with probability $r(i,0) \geq 0$. It holds $\sum_{j=0}^{J} r(i,j) = 1$ with $r(0,0) := 0$ for all $i \in \overline{J}_0 := \{0, 1, \ldots, J\}$. Given the departure node i, the resource's routing decision is made independently of the network's history. We assume that the routing matrix $\mathcal{R} := \left(r(i,j) : i,j \in \overline{J}_0\right)$ is irreducible.

To obtain a Markovian process description, we denote by $X_{\text{ex}}(t)$ the number of customers in the external queue at time $t \geq 0$, by $Y_0(t)$ the number of resources in the resource pool at time $t \geq 0$ and by $Y_j(t)$, $j \in \overline{J}$, the number of resources present at node j in the inner network at time $t \geq 0$, either waiting or in service. We call this $Y_j(t)$ queue length at node $j \in \overline{J}_0$ at time $t \geq 0$. Then $\mathbf{Y}(t) := \left(Y_j(t) : j \in \overline{J}_0\right)$ is the queue length vector of the resource network at time $t \geq 0$. We define the joint queue length process of the semi-open network with **b**ack**o**rdering by $Z_{\text{BO}} := \left(\left(X_{\text{ex}}(t), \mathbf{Y}(t)\right) : t \geq 0\right)$. Due to the independence and memorylessness assumptions, Z_{BO} is an irreducible Markov process with state space

$$E := \left\{ \left(0, n_0, n_1, \ldots, n_J\right) : n_j \in \{0, \ldots, N\}\ \forall j \in \overline{J}_0, \sum_{j \in \overline{J}_0} n_j = N \right\}$$

$$\cup \left\{ \left(n_{\text{ex}}, 0, n_1, \ldots, n_J\right) : n_{\text{ex}} \in \mathbb{N},\ n_j \in \{0, \ldots, N\}\ \forall j \in \overline{J}, \sum_{j \in \overline{J}} n_j = N \right\}.$$

2.2 Stability

In this section, we analyse the stability of the system and Z_{BO}, which has infinitesimal generator $\mathbf{Q} := (q(z;\tilde{z}) : z, \tilde{z} \in E)$ with the following transition rates for $\left(n_{\text{ex}}, \mathbf{n}\right)$, $(0, \mathbf{n}) \in E$, where $\mathbf{n} := \left(n_j : j \in \overline{J}_0\right)$:

$$
\begin{aligned}
&q\big((n_{\text{ex}},\mathbf{n});(n_{\text{ex}}+1,\mathbf{n})\big) = \lambda_{\text{BO}}\cdot 1_{\{n_0=0\}},\ n_{\text{ex}}\geq 0,\\
&q\big((0,\mathbf{n});(0,\mathbf{n}-\mathbf{e}_0+\mathbf{e}_i)\big) = \lambda_{\text{BO}}\cdot r(0,i)\cdot 1_{\{n_0>0\}},\ i\in\overline{J},\\
&q\big((n_{\text{ex}},\mathbf{n});(n_{\text{ex}},\mathbf{n}-\mathbf{e}_i+\mathbf{e}_j)\big) = \nu_i(n_i)\cdot r(i,j)\cdot 1_{\{n_i>0\}},\ n_{\text{ex}}\geq 0,\ i,j\in\overline{J},\\
&q\big((0,\mathbf{n});(0,\mathbf{n}-\mathbf{e}_i+\mathbf{e}_0)\big) = \nu_i(n_i)\cdot r(i,0)\cdot 1_{\{n_i>0\}},\ i\in\overline{J},\\
&q\big((n_{\text{ex}},\mathbf{n});(n_{\text{ex}}-1,\mathbf{n}-\mathbf{e}_i+\mathbf{e}_j)\big) = \nu_i(n_i)\cdot r(i,0)\cdot r(0,j)\cdot 1_{\{n_{\text{ex}}>0\}}\cdot 1_{\{n_i>0\}},\\
&\quad n_{\text{ex}}>0,\ i\in\overline{J}.
\end{aligned}
$$

Furthermore, $q(z;\tilde{z}) = 0$ for any other pair $z\neq\tilde{z}$, and

$$
q(z;z) = -\sum_{\substack{\tilde{z}\in E,\\ \tilde{z}\neq z}} q(z;\tilde{z}) \qquad \forall z\in E.
$$

Z_{BO} is a level-independent quasi-birth-and-death process in the sense of (Latouche and Ramaswami 1999, Def. 1.3.1, p. 12). The level is the length n_{ex} of the external queue. The phase is the state of the resource network. For level zero, the phase space is

$$
\widetilde{E}_0 := \left\{ \big(n_0,n_1,\ldots,n_J\big)\ :\ n_j\in\{0,\ldots,N\}\ \forall j\in\overline{J}_0, \sum_{j\in\overline{J}_0} n_j = N \right\}.
$$

When level $n_{\text{ex}}>0$, the resource pool is empty. Hence, for positive levels the phase space is

$$
\widetilde{E}_+ := \left\{ \big(0,n_1,\ldots,n_J\big)\ :\ n_j\in\{0,\ldots,N\}\ \forall j\in\overline{J}, \sum_{j\in\overline{J}} n_j = N \right\}.
$$

Arranging states by level, the infinitesimal generator $\mathbf{Q}$ of Z_{BO} is

$$
\mathbf{Q} = \begin{pmatrix}
\mathbf{B}_0 & \mathbf{B}_1 & & & \\
\mathbf{B}_2 & \mathbf{A}_0 & \mathbf{A}_1 & & \\
 & \mathbf{A}_{-1} & \mathbf{A}_0 & \mathbf{A}_1 & \\
 & & \ddots & \ddots & \ddots
\end{pmatrix},
$$

where $\mathbf{B}_0\in\mathbb{R}^{\widetilde{E}_0\times\widetilde{E}_0}$, $\mathbf{B}_1\in\mathbb{R}^{\widetilde{E}_0\times\widetilde{E}_+}$, $\mathbf{B}_2\in\mathbb{R}^{\widetilde{E}_+\times\widetilde{E}_0}$ and $\mathbf{A}_{-1}$, $\mathbf{A}_0$, $\mathbf{A}_1\in\mathbb{R}^{\widetilde{E}_+\times\widetilde{E}_+}$ are matrices.

$\mathbf{A}_1$ is a non-negative matrix with the following positive elements

$$
a_1\big((0,n_1,\ldots,n_J);(0,n_1,\ldots,n_J)\big) = \lambda_{\text{BO}}.
$$

$\mathbf{A}_{-1}$ is a non-negative matrix with at most the following non-negative elements

$$a_{-1}\big((0,n_1,\ldots,n_J);(0,n_1,\ldots,n_i-1,\ldots,n_j+1,\ldots,n_J)\big) = \nu_i(n_i)\cdot r(i,0)\cdot r(0,j)\cdot 1_{\{n_i>0\}},\quad i,j\in\overline{J}.$$

$\mathbf{A}_0$ has non-negative off-diagonal elements and strictly negative diagonals. The off-diagonal elements are

$$a_0\big((0,n_1,\ldots,n_J);(0,n_1,\ldots,n_i-1,\ldots,n_j+1,\ldots,n_J)\big) = \nu_i(n_i)\cdot r(i,j)\cdot 1_{\{n_i>0\}},\quad i,j\in\overline{J}.$$

Let $\boldsymbol{\pi}_{\mathrm{BO}} := \big(\pi_{\mathrm{BO}}(n_{\mathrm{ex}},\mathbf{n}) : (n_{\mathrm{ex}},\mathbf{n})\in E\big)$ be the steady-state distribution of the Markov process Z_{BO}. The global balance equations $\boldsymbol{\pi}_{\mathrm{BO}}\cdot\mathbf{Q}=\mathbf{0}$ are:

For $n_{\mathrm{ex}}=0$

$$\begin{aligned}
&\pi_{\mathrm{BO}}(0,\mathbf{n})\\
&\quad\cdot\Big(\lambda_{\mathrm{BO}}+\sum_{i\in\overline{J}}\sum_{j\in\overline{J}\setminus\{i\}}\nu_i(n_i)\cdot r(i,j)\cdot 1_{\{n_i>0\}}+\sum_{i\in\overline{J}}\nu_i(n_i)\cdot r(i,0)\cdot 1_{\{n_i>0\}}\Big)\\
&=\sum_{i\in\overline{J}}\pi_{\mathrm{BO}}\big(0,\mathbf{n}+\mathbf{e}_0-\mathbf{e}_i\big)\cdot\lambda_{\mathrm{BO}}\cdot r(0,i)\cdot 1_{\{n_i>0\}}\\
&\quad+\sum_{i\in\overline{J}}\sum_{j\in\overline{J}\setminus\{i\}}\pi_{\mathrm{BO}}\big(0,\mathbf{n}+\mathbf{e}_i-\mathbf{e}_j\big)\cdot\nu_i(n_i+1)\cdot r(i,j)\cdot 1_{\{n_j>0\}}\\
&\quad+\sum_{i\in\overline{J}}\pi_{\mathrm{BO}}\big(0,\mathbf{n}+\mathbf{e}_i-\mathbf{e}_0\big)\cdot\nu_i(n_i+1)\cdot r(i,0)\cdot 1_{\{n_0>0\}}\\
&\quad+\sum_{i\in\overline{J}}\sum_{j\in\overline{J}\setminus\{i\}}\pi_{\mathrm{BO}}\big(1,\mathbf{n}+\mathbf{e}_i-\mathbf{e}_j\big)\cdot\nu_i(n_i+1)\cdot r(i,0)\cdot r(0,j)\cdot 1_{\{n_j>0\}}\cdot 1_{\{n_0=0\}}\\
&\quad+\sum_{i\in\overline{J}}\pi_{\mathrm{BO}}(1,\mathbf{n})\cdot\nu_i(n_i)\cdot r(i,0)\cdot r(0,i)\cdot 1_{\{n_i>0\}}\cdot 1_{\{n_0=0\}}.
\end{aligned}$$

For $n_{\mathrm{ex}}>0$, which implies $n_0=0$,

$$\begin{aligned}
&\pi_{\mathrm{BO}}\big(n_{\mathrm{ex}},\mathbf{n}\big)\\
&\quad\cdot\Big(\lambda_{\mathrm{BO}}+\sum_{i\in\overline{J}}\sum_{j\in\overline{J}\setminus\{i\}}\nu_i(n_i)\cdot r(i,j)\cdot 1_{\{n_i>0\}}+\sum_{i\in\overline{J}}\nu_i(n_i)\cdot r(i,0)\cdot 1_{\{n_i>0\}}\Big)\\
&=\pi_{\mathrm{BO}}\big(n_{\mathrm{ex}}-1,\mathbf{n}\big)\cdot\lambda_{\mathrm{BO}}\\
&\quad+\sum_{j\in\overline{J}}\sum_{i\in\overline{J}\setminus\{i\}}\pi_{\mathrm{BO}}\big(n_{\mathrm{ex}},\mathbf{n}+\mathbf{e}_i-\mathbf{e}_j\big)\cdot\nu_i(n_i+1)\cdot r(i,j)\cdot 1_{\{n_j>0\}}\\
&\quad+\sum_{j\in\overline{J}}\sum_{i\in\overline{J}\setminus\{i\}}\pi_{\mathrm{BO}}\big(n_{\mathrm{ex}}+1,\mathbf{n}+\mathbf{e}_i-\mathbf{e}_j\big)\cdot\nu_i(n_i+1)\cdot r(i,0)\cdot r(0,j)\cdot 1_{\{n_j>0\}}\\
&\quad+\sum_{i\in\overline{J}}\pi_{\mathrm{BO}}\big(n_{\mathrm{ex}}+1,\mathbf{n}\big)\cdot\nu_i(n_i)\cdot r(i,0)\cdot r(0,i)\cdot 1_{\{n_i>0\}}.
\end{aligned}$$

No closed-form expression is known for $\boldsymbol{\pi}_{\mathrm{BO}}$ in case of $J>1$. Latouche and Ramaswami developed a logarithmic reduction algorithm for level-independent

quasi-birth-and-death processes to compute the steady-state distribution (Latouche and Ramaswami 1993, Latouche and Ramaswami 1999, Theorem 6.4.1 and Lemma 6.4.3, p. 142ff.). For $J = 1$, we calculate a closed-form expression for the steady-state distribution, see Sect. 2.4.

To determine the stability condition of the system, we define traffic equations:

$$\eta_j = \sum_{i \in \overline{J}_0} \eta_i \cdot r(i,j), \qquad j \in \overline{J}_0. \tag{1}$$

Denote by $\lambda_{\mathrm{BO,max}}$ the throughput of node 0 in the closed network depicted in Fig. 3, which is obtained from the original SOQN when infinitely many customers reside in the external queue. In Lavenberg (1978) this network is called *saturated*—we will call this network *stability network* ("stb"). In this network, resources which enter node 0 spend zero time there and jump to the next node according to the branching vector $(r(0,j) : j \in \overline{J})$. In Lavenberg (1978), it is proved that an SOQN (with back-ordering) is stable if $\lambda_{\mathrm{BO}} < \lambda_{\mathrm{BO,max}}$, and that $\lambda_{\mathrm{BO}} > \lambda_{\mathrm{BO,max}}$ implies instability. We use matrix geometrical methods to show that $\lambda_{\mathrm{BO}} < \lambda_{\mathrm{BO,max}}$ is sufficient *and* necessary for stability. To simplify notation, we define

$$C^{\mathrm{stb}}(\overline{J}, N) := \sum_{\sum_{j \in \overline{J}} n_j = N} \prod_{j=1}^{J} \left(\prod_{i=1}^{n_j} \frac{\eta_j}{\nu_j(i)} \right).$$

Proposition 1 *The system is stable if and only if* $\lambda_{\mathrm{BO}} < \lambda_{\mathrm{BO,max}}$ *with*

$$\lambda_{\mathrm{BO,max}} = \eta_0 \cdot \frac{C^{\mathrm{stb}}(\overline{J}, N-1)}{C^{\mathrm{stb}}(\overline{J}, N)}. \tag{2}$$

Proof We apply matrix-geometric methods using (Latouche 2011, Theorem 1): Given the irreducible inter-level generator matrix $\mathbf{A} := \mathbf{A}_{-1} + \mathbf{A}_0 + \mathbf{A}_1$ of Z_{BO} and the stochastic solution $\boldsymbol{\alpha} := (\alpha(\widetilde{\mathbf{n}}) : \widetilde{\mathbf{n}} \in \widetilde{E}_+)$ of $\boldsymbol{\alpha} \cdot \mathbf{A} = \mathbf{0}$, the process Z_{BO} is stable if and only if

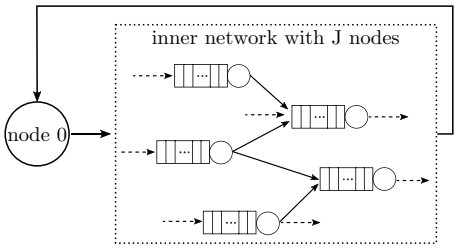

Fig. 3 Stability network – A closed network described by the inter-level generator matrix $\mathbf{A}$ from the proof of Proposition 1. Interpretation I: a generalised Gordon–Newell network with zero service time at node 0. Interpretation II: a classical Gordon–Newell network obtained after rerouting at node 0

$$\boldsymbol{\alpha} \cdot \mathbf{A}_1 \cdot \mathbf{e} < \boldsymbol{\alpha} \cdot \mathbf{A}_{-1} \cdot \mathbf{e}. \tag{3}$$

Because $\mathbf{A}_1$ is a diagonal matrix with λ_{BO} on its diagonal and $\boldsymbol{\alpha}$ is a stochastic vector, the left-hand side of (3) is λ_{BO}. We define

$$\lambda_{\mathrm{BO,max}} := \boldsymbol{\alpha} \cdot \mathbf{A}_{-1} \cdot \mathbf{e}.$$

To determine $\boldsymbol{\alpha}$, we note that the non-negative non-diagonal elements of the generator $\mathbf{A}$ are for $i \neq j$ of the form

$$\begin{aligned} &a\big((0, n_1, \ldots, n_J); (0, n_1, \ldots, n_i - 1, \ldots, n_j + 1, \ldots, n_J)\big) \\ &\quad = \nu_i(n_i) \cdot \Big(r(i,j) + r(i,0) \cdot r(0,j)\Big) \cdot 1_{\{n_i > 0\}}. \end{aligned}$$

The diagonal elements are chosen to set row sums to zero.

We now solve for all $\widetilde{\mathbf{n}} := \big(0, n_1, \ldots, n_J\big) \in \widetilde{E}_+$

$$\begin{aligned} &\alpha\big(\widetilde{\mathbf{n}}\big) \cdot \sum_{i \in \overline{J}} \nu_i(n_i) \cdot \sum_{j \in \overline{J}} \Big(r(i,j) + r(i,0) \cdot r(0,j)\Big) \cdot 1_{\{n_i > 0\}} \\ &\quad = \sum_{j \in \overline{J}} \sum_{i \in \overline{J}} \alpha\big(\widetilde{\mathbf{n}} + \mathbf{e}_i - \mathbf{e}_j\big) \cdot \nu_i(n_i + 1) \cdot \Big(r(i,j) + r(i,0) \cdot r(0,j)\Big) \cdot 1_{\{n_j > 0\}}. \end{aligned} \tag{4}$$

Equation (4) is the global balance equation of a generalised Gordon–Newell network with node set $\overline{J}_0$, N customers and zero service time at node 0.

Equation (4) has another interpretation, which allows us to use standard algorithms for Gordon–Newell networks for performance analysis. Note that the status of node 0 is invariant ($= 0$), therefore we define $\alpha'(n_1, \ldots, n_J) := \alpha\big(0, n_1, \ldots, n_J\big)$ and the routing matrix $\mathcal{R}' := \Big(r'(i,j) : i,j \in \overline{J}\Big)$ with

$$r'(i,j) := r(i,j) + r(i,0) \cdot r(0,j). \tag{5}$$

$\mathcal{R}'$ is obtained from $\mathcal{R}$ when every resource directed to 0 skips this node and jumps to the next node according to $(r(0,j) : j \in \overline{J})$. So Eq. (4) can be written as

$$\begin{aligned} &\alpha'\big(n_1, \ldots, n_J\big) \cdot \sum_{i \in \overline{J}} \nu_i(n_i) \sum_{j \in \overline{J}} r'(i,j) \cdot 1_{\{n_i > 0\}} \\ &\quad = \sum_{j \in \overline{J}} \sum_{i \in \overline{J}} \alpha'\big((n_1, \ldots, n_J) + \mathbf{e}_i - \mathbf{e}_j\big) \cdot \nu_i(n_i + 1) \cdot r'(i,j) \cdot 1_{\{n_j > 0\}}. \end{aligned} \tag{6}$$

Equation (6) is the global balance equation of a Gordon–Newell network with nodes $\overline{J} = \{1, 2, \ldots, J\}$, N customers, routing matrix $\mathcal{R}'$ and steady-state distribution

$$\alpha'(n_1, \ldots, n_J) = \Big[C_{\mathrm{BO}}^{\mathrm{stb}'}(\overline{J}, N)\Big]^{-1} \prod_{j=1}^{J} \left(\prod_{i=1}^{n_j} \frac{\eta'_j}{\nu_j(i)} \right) \tag{7}$$

where $\boldsymbol{\eta}' := \Big(\eta'_j : j \in \overline{J}\Big)$ is a solution of the traffic equation $\boldsymbol{\eta}' \cdot \mathcal{R}' = \boldsymbol{\eta}'$ and

$$C_{\mathrm{BO}}^{\mathrm{stb}'}(\bar{J},N) := \sum_{\sum_{j\in\bar{J}} n_j = N} \prod_{j=1}^{J} \left(\prod_{i=1}^{n_j} \frac{\eta_j'}{\nu_j(i)} \right)$$

is the normalisation constant. Because of the special structure (5) of $\mathcal{R}'$, $\eta_j' := \eta_j$ for all $j \in J$ is a solution of $\boldsymbol{\eta}' \cdot \mathcal{R}' = \boldsymbol{\eta}'$, see (Krenzler et al. 2016, Proposition 2.1). Consequently, $C_{\mathrm{BO}}^{\mathrm{stb}'}(\bar{J},N) = C^{\mathrm{stb}}(\bar{J},N)$, and we can switch between both interpretations without recalculating $\boldsymbol{\eta}'$ and $C_{\mathrm{BO}}^{\mathrm{stb}'}(\bar{J},N)$, and obtain, e.g.

$$\alpha'(n_1,\ldots,n_J) = \left[C^{\mathrm{stb}}(\bar{J},N)\right]^{-1} \prod_{j=1}^{J} \left(\prod_{i=1}^{n_j} \frac{\eta_j}{\nu_j(i)} \right).$$

We now calculate $\lambda_{\mathrm{BO,max}}$ explicitly.

$$\begin{aligned}
\lambda_{\mathrm{BO,max}} &= \alpha \cdot \mathbf{A}_{-1} \cdot \mathbf{e} = \sum_{(0,m_1,\ldots,m_J)\in\widetilde{E}_+} \big(\alpha \cdot \mathbf{A}_{-1}\big)\big(0,m_1,\ldots,m_J\big) \\
&= \sum_{(0,m_1,\ldots,m_J)\in\widetilde{E}_+} \Bigg[\sum_{(0,n_1,\ldots,n_J)\in E_1} \alpha\big(0,n_1,\ldots,n_J\big) \\
&\quad \cdot a_{-1}\big(\big(0,n_1,\ldots,n_J\big);\big(0,m_1,\ldots,m_J\big)\big)\Bigg] \\
&= \sum_{(0,n_1,\ldots,n_J)\in\widetilde{E}_+} \Bigg[\alpha\big(0,n_1,\ldots,n_J\big) \\
&\quad \cdot \sum_{(0,m_1,\ldots,m_J)\in\widetilde{E}_+} a_{-1}\big(\big(0,n_1,\ldots,n_J\big);\big(0,m_1,\ldots,m_J\big)\big)\Bigg] \\
&= \sum_{(0,n_1,\ldots,n_J)\in\widetilde{E}_+} \alpha\big(0,n_1,\ldots,n_J\big) \cdot \Bigg[\sum_{i=1}^{J}\sum_{j=1}^{J} \nu_i(n_i)\cdot r(i,0)\cdot r(0,j)\cdot 1_{\{n_i>0\}}\Bigg] \\
&= \sum_{(0,n_1,\ldots,n_J)\in\widetilde{E}_+} \underbrace{\alpha\big(0,n_1,\ldots,n_J\big)}_{=\alpha'(n_1,\ldots,n_J)} \cdot \Bigg[\sum_{i=1}^{J} \nu_i(n_i)\cdot r(i,0)\cdot 1_{\{n_i>0\}} \cdot \underbrace{\sum_{j=1}^{J} r(0,j)}_{=1}\Bigg] \\
&= \sum_{(n_1,\ldots,n_J)\in\widetilde{E}_+} \alpha'(n_1,\ldots,n_J) \cdot \Bigg[\sum_{i=1}^{J} \nu_i(n_i)\cdot 1_{\{n_i>0\}} \cdot r(i,0)\Bigg] \\
&= \sum_{i=1}^{J} \underbrace{\Bigg[\sum_{n_i=0}^{N} \sum_{\substack{n_j\in\{0,\ldots,N\},\ j\in\bar{J}\setminus\{i\} \\ \sum_{j\in\bar{J}\setminus\{i\}} n_j = N-n_i}} \alpha'(n_1,\ldots,n_J)\cdot\nu_i(n_i)\cdot 1_{\{n_i>0\}}\Bigg]}_{(*)} \cdot r(i,0).
\end{aligned}$$

The expression (∗) is the throughput through node i in the Gordon–Newell network with routing matrix $\mathcal{R}'$:

$$TH_i^{\mathrm{stb}}(N) := \sum_{n_i=0}^{N} \sum_{\substack{n_j \in \{0,\ldots,N\},\ j \in \bar{J}\setminus\{i\} \\ \sum_{j\in\bar{J}\setminus\{i\}} n_j = N-n_i}} \alpha'(n_1,\ldots,n_J) \cdot \nu_i(n_i) \cdot 1_{\{n_i>0\}}, \qquad i \in \bar{J}.$$

This yields

$$\lambda_{\mathrm{BO,max}} = \sum_{i=1}^{J} TH_i^{\mathrm{stb}}(N) \cdot r(i,0). \tag{8}$$

According to (Bolch et al. 1998, p. 374, (8.14)) it holds $TH_i^{\mathrm{stb}}(N) = \eta_i \cdot \frac{C^{\mathrm{stb}}(\bar{J},N-1)}{C^{\mathrm{stb}}(\bar{J},N)}$. Therefore,

$$\lambda_{\mathrm{BO,max}} = \sum_{i=1}^{J} \eta_i \cdot \frac{C^{\mathrm{stb}}(\bar{J},N-1)}{C^{\mathrm{stb}}(\bar{J},N)} \cdot r(i,0) = \frac{C^{\mathrm{stb}}(\bar{J},N-1)}{C^{\mathrm{stb}}(\bar{J},N)} \underbrace{\sum_{i=1}^{J} \eta_i \cdot r(i,0)}_{=\eta_0}.$$

□

Remark 1 The right-hand side of Eq. (8) is the throughput of node 0 in the stability network in Fig. 3. Formally, this is

$$\lambda_{\mathrm{BO,max}} = TH_0^{\mathrm{stb}}(N) \text{ with } TH_0^{\mathrm{stb}}(N) := \sum_{i=1}^{J} TH_i^{\mathrm{stb}}(N) \cdot r(i,0). \tag{9}$$

The advantage of representation (9) is that it uses throughputs $TH_i^{\mathrm{stb}}(N)$, $i \in \bar{J}$, of a classical Gordon–Newell network with routing matrix $\mathcal{R}'$. We can calculate these throughputs efficiently with standard methods, e.g. mean value analysis (MVA). Using Eq. (2), another representation of $TH_0^{\mathrm{stb}}(N)$ is

$$TH_0^{\mathrm{stb}}(N) = \eta_0 \cdot \frac{C^{\mathrm{stb}}(\bar{J},N-1)}{C^{\mathrm{stb}}(\bar{J},N)}. \tag{10}$$

To calculate efficiently the constants on the right-hand side of Eq. (10), we can use, for example, the convolution algorithm. Both algorithms are illustrated in (Bolch et al. 1998, p. 371ff., Section 8.1 and p. 384ff., Section 8.2).

2.3 Throughputs and idle times

We consider an SOQN-BO in steady state. Let $\boldsymbol{\eta} := (\eta_j : j \in \bar{J}_0)$ be a solution of Eq. (1). $\boldsymbol{\eta}$ is unique up to a constant, which implies that the following formula (11) does

not depend on that constant. Equation (11) occurs as an approximation in a slightly different setting in Dallery (1990) as (26).

Proposition 2 *The local throughput at the nodes $j \in \overline{J}_0$ is*

$$TH_{\mathrm{BO},j} = \lambda_{\mathrm{BO}} \cdot \frac{\eta_j}{\eta_0}. \tag{12}$$

Proof We define for $j \in \overline{J}_0$ in steady state:

- the mean number of departures from j per time unit is D_j, and
- the mean number of arrivals at j per time unit is V_j.

From the steady state assumption follows $TH_{\mathrm{BO},j} = V_j = D_j$. For any $j \in \overline{J}_0$ it holds $V_j = \sum_{i \in \overline{J}_0} D_i \cdot r(i,j)$. Therefore, the vector $V := (V_j : j \in \overline{J}_0)$ fulfils the set of equations $V_j = \sum_{i \in \overline{J}_0} V_i \cdot r(i,j)$, $j \in \overline{J}_0$, which is $V = V \cdot \mathcal{R}$. This implies that $V_j = \eta_j \cdot K$ for some constant $K > 0$. Because of $\lambda_{\mathrm{BO}} = V_0 = \eta_0 \cdot K$ we have $K = \frac{\lambda_{\mathrm{BO}}}{\eta_0}$, and therefore, $V_j = \lambda_{\mathrm{BO}} \cdot \frac{\eta_j}{\eta_0}$, $j \in \overline{J}_0$. □

Corollary 1 *Let $Y_{\mathrm{BO}} := (Y_{\mathrm{BO},j} : j \in \overline{J})$ denote a random vector which is distributed according to the stationary queue length at the nodes in $\overline{J}$ of the SOQN-BO. If the service rate at node j is independent of the queue length, i.e. $\nu_j(\cdot) = \nu_j, j \in \overline{J}$, then the probability that node j is idling is*

$$P(Y_{\mathrm{BO},j} = 0) = 1 - \lambda_{\mathrm{BO}} \cdot \frac{\eta_j}{\eta_0} \cdot \nu_j^{-1}.$$

This is also the proportion of time that node j is idling.

Proof We define for $j \in \overline{J}$ in steady state:

- the mean number of customers in service is B_j,
- the mean service time is S_j, and
- the arrival intensity is λ_j.

According to Little's formula, $B_j = \lambda_j \cdot S_j$ for every node j. In steady state, the arrival rate λ_j at node j equals its throughput $TH_{\mathrm{BO},j}$. Hence, from Proposition 2, we have $\lambda_j = TH_{\mathrm{BO},j} = \lambda_{\mathrm{BO}} \cdot \left(\eta_j/\eta_0\right)$. For node j, with constant service rate ν_j, the mean service time S_j is ν_j^{-1}. Inserting these λ_j and S_j into Little's formula yields $B_j = \lambda_{\mathrm{BO}} \cdot \left(\eta_j/\eta_0\right) \cdot \nu_j^{-1}$ for the mean number of customers in service. Consequently, the probability that node j is idling is

$$\begin{aligned} P(Y_{\mathrm{BO},j} = 0) &= 1 - P(Y_{\mathrm{BO},j} > 0) = 1 - E\left[1_{\{Y_{\mathrm{BO},j} > 0\}}\right] = 1 - B_j \\ &= 1 - \lambda_{\mathrm{BO}} \cdot \frac{\eta_j}{\eta_0} \cdot \nu_j^{-1}. \end{aligned}$$

□

2.4 Special case: $J = 1$

We consider an SOQN-BO with inner network consisting of one node only.

Theorem 1 *For $J = 1$, the joint queue length process Z_{BO} is stable if and only if $\lambda_{\mathrm{BO}} < \nu_1(N)$. For stable Z_{BO} the stationary distribution is $\boldsymbol{\pi}_{\mathrm{BO}} := \left(\pi_{\mathrm{BO}}(n_{\mathrm{ex}}, n_0, n_1) : (n_{\mathrm{ex}}, n_0, n_1) \in E\right)$ with*

$$\begin{aligned}\pi_{\mathrm{BO}}(n_{\mathrm{ex}}, n_0, n_1) &= \pi_{\mathrm{BO}}(n_{\mathrm{ex}}, N - n_1, n_1) \\ &= \left[C_{\mathrm{BO}}(\{1\}, N)\right]^{-1} \cdot \left(\frac{\lambda_{\mathrm{BO}}}{\nu_1(N)}\right)^{n_{\mathrm{ex}}} \cdot \prod_{m=1}^{n_1} \frac{\lambda_{\mathrm{BO}}}{\nu_1(m)}\end{aligned} \tag{12}$$

and normalisation constant

$$C_{\mathrm{BO}}(\{1\}, N) := \sum_{m=0}^{N-1} \prod_{\ell=1}^{m} \frac{\lambda_{\mathrm{BO}}}{\nu_1(\ell)} + \left(\frac{1}{1 - \frac{\lambda_{\mathrm{BO}}}{\nu_1(N)}}\right) \cdot \prod_{\ell=1}^{N} \frac{\lambda_{\mathrm{BO}}}{\nu_1(\ell)}. \tag{13}$$

Proof Let $(\widehat{X}_{\mathrm{ex}}, \widehat{Y}_0, \widehat{Y}_1)$ denote a random vector with distribution $\boldsymbol{\pi}_{\mathrm{BO}}$. In (Avi-Itzhak and Heyman 1973, equations (20) and (21)) the authors calculated $P(\widehat{X}_{\mathrm{ex}} + \widehat{Y}_1 = m)$. From this we obtain the probabilities for all queues, because for $m \leq N$ it holds $\widehat{X}_{\mathrm{ex}} + \widehat{Y}_1 = m \Leftrightarrow [\widehat{X}_{\mathrm{ex}} = 0 \wedge \widehat{Y}_1 = m \wedge \widehat{Y}_0 = N - m]$ and for $m > N$ it holds $\widehat{X}_{\mathrm{ex}} + \widehat{Y}_1 = m \Leftrightarrow [\widehat{X}_{\mathrm{ex}} = m - N \wedge \widehat{Y}_1 = N \wedge \widehat{Y}_0 = 0]$. □

Proposition 3 *Let $(\widehat{X}_{\mathrm{ex}}, \widehat{Y}_0, \widehat{Y}_1)$ denote a random vector which is distributed according to the steady-state distribution of Z_{BO} in the SOQN-BO with $J = 1$.*

(i) For the marginal distributions it holds:

$$P(\widehat{X}_{\mathrm{ex}} = 0) = \left[C_{\mathrm{BO}}(\{1\}, N)\right]^{-1} \cdot \sum_{n_1=0}^{N} \prod_{m=1}^{n_1} \frac{\lambda_{\mathrm{BO}}}{\nu_1(m)}, \tag{14}$$

$$P(\widehat{X}_{\mathrm{ex}} = n_{\mathrm{ex}}) = \left[C_{\mathrm{BO}}(\{1\}, N)\right]^{-1} \left(\frac{\lambda_{\mathrm{BO}}}{\nu_1(N)}\right)^{n_{\mathrm{ex}}} \prod_{m=1}^{N} \frac{\lambda_{\mathrm{BO}}}{\nu_1(m)}, \; n_{\mathrm{ex}} > 0, \tag{15}$$

$$P(\widehat{Y}_0 = N - n_1, \widehat{Y}_1 = n_1) = \left[C_{\mathrm{BO}}(\{1\}, N)\right]^{-1} \prod_{m=1}^{n_1} \frac{\lambda_{\mathrm{BO}}}{\nu_1(m)}, \; 0 \leq n_1 < N, \tag{16}$$

$$P(\widehat{Y}_0 = 0, \widehat{Y}_1 = N) = \left[C_{\mathrm{BO}}(\{1\}, N)\right]^{-1} \frac{1}{1 - \frac{\lambda_{\mathrm{BO}}}{\nu_1(N)}} \prod_{m=1}^{N} \frac{\lambda_{\mathrm{BO}}}{\nu_1(m)}. \tag{17}$$

(ii) The average external queue length is

$$\widehat{L_{\text{ex}}} = P(\widehat{Y}_0 = 0, \widehat{Y}_1 = N) \cdot \frac{\lambda_{\text{BO}}}{\nu_1(N) - \lambda_{\text{BO}}} \tag{18}$$

and the average waiting time of customers in the external queue is

$$\widehat{W_{\text{ex}}} = \frac{\widehat{L_{\text{ex}}}}{\lambda_{\text{BO}}} = P(\widehat{Y}_0 = 0, \widehat{Y}_1 = N) \cdot \frac{1}{\nu_1(N) - \lambda_{\text{BO}}}. \tag{19}$$

Proof (i) All probabilities can be expressed in terms of Eq. (12), resulting in routine calculations.

(ii) The stationary average external queue length is obtained directly from the marginal distribution (15) in (i). Equation (19) follows by Little's law, see e.g. Little and Graves (2008). □

An important question about SOQN is whether more resources yield more throughput. Proposition 4 shows that even for $J = 1$ this is not the case.

Proposition 4 *Let* $J = 1$. *Then the following are equivalent:*

(i) $\lambda_{\text{BO,max}} = TH_0^{\text{stb}}(\cdot)$ *is non-decreasing on* $\mathbb{N}$.
(ii) ν_1 *is non-decreasing on* $\mathbb{N}$.

Proof Because of Eq. (2) and (9), (i) is equivalent to

$$\forall\, N \in \mathbb{N}: \; C^{\text{stb}}(\{1\}, N-1) \cdot C^{\text{stb}}(\{1\}, N+1) \le C^{\text{stb}}(\{1\}, N)^2. \tag{20}$$

We note

$$C^{\text{stb}}(\{1\}, N+1) = C^{\text{stb}}(\{1\}, N) \cdot \frac{\eta_1}{\nu_1(N+1)},$$
$$C^{\text{stb}}(\{1\}, N-1) = C^{\text{stb}}(\{1\}, N) \cdot \frac{\nu_1(N)}{\eta_1},$$

implying

$$(20) \Leftrightarrow \forall\, N \in \mathbb{N}: \; \frac{\nu_1(N)}{\nu_1(N+1)} \cdot C^{\text{stb}}(\{1\}, N)^2 \le C^{\text{stb}}(\{1\}, N)^2$$

and by $C^{\text{stb}}(\{1\}, N)^2 > 0$ this is equivalent to $\forall\, N \in \mathbb{N}: \; \nu_1(N) \le \nu_1(N+1)$, which is (ii). □

Van der Wal (1989) proved that in general it holds (ii) ⇒ (i).

Proposition 5 *Let* $J \ge 1$ *and all service rates be non-decreasing in the number of customers, i.e.* $\nu_j(n+1) \ge \nu_j(n)$, $n \in \{0, \ldots, N-1\}$, *for all* $j \in \overline{J}$. *Then* $\lambda_{\text{BO,max}} = TH_0^{\text{stb}}(\cdot)$ *is non-decreasing on* $\mathbb{N}$.

3 Lost customer approximation of SOQN-BO

The steady-state distribution of an SOQN-BO is unknown for $J > 1$. For a modified system, we get closed product-form results for $J \geq 1$.

3.1 SOQN with lost customers

We consider a modification of the SOQN-BO model from Sect. 2.1: Newly arriving customers are rejected and lost for the system if the resource pool is empty (Fig. 4). This property is termed, e.g. "lost customers", "lost sales", "lost arrivals" or "loss systems". We denote this system *SOQN-LC*. Customers arrive in a Poisson stream with rate $\lambda_{\mathrm{LC}} > 0$. Because of loss of customers, the effective arrival rate $\lambda_{\mathrm{eff}}(\lambda_{\mathrm{LC}})$ is smaller.

Such SOQN-LC can be investigated using Gordon–Newell network theory for the resource network (after suitable modification), see (Chen and Yao 2001, p. 21).

To obtain a Markovian process description, we denote by $Y_0(t)$ the number of resources in the resource pool at time $t \geq 0$ and by $Y_j(t)$, $j \in \bar{J}$, the number of resources at node j in the inner network at time $t \geq 0$, either waiting or in service (queue length at node $j \in \bar{J}$). Then $\mathbf{Y}(t) := \left(Y_j(t) : j \in \bar{J}_0\right)$ is the local queue length vector of the resource network at time $t \geq 0$. We define the joint queue length process of this semi-open network by

$$Z_{\mathrm{LC}} := (\mathbf{Y}(t) : t \geq 0).$$

Due to the usual independence and memorylessness assumptions (see Sect. 2.1), Z_{LC} is an irreducible Markov process on state space

$$E_{LC} := \left\{ \left(n_0, n_1, \ldots, n_J\right) : n_j \in \{0, \ldots, N\}\ \forall j \in \bar{J}_0,\ \sum_{j \in \bar{J}_0} n_j = N \right\}.$$

The stationary distribution $\boldsymbol{\pi}_{\mathrm{LC}} := \left(\pi_{\mathrm{LC}}(\mathbf{n}) : \mathbf{n} \in E_{LC}\right)$ of Z_{LC} in product form is available (Chen and Yao 2001, p. 22, Theorem 2.5): For $\mathbf{n} := \left(n_j : j \in \bar{J}_0\right) \in E_{LC}$

$$\pi_{\mathrm{LC}}(\mathbf{n}) = \left[C_{\mathrm{LC}}(\bar{J}_0, N)\right]^{-1} \cdot \left(\frac{\eta_0}{\lambda_{\mathrm{LC}}}\right)^{n_0} \cdot \prod_{j=1}^{J} \left(\prod_{i=1}^{n_j} \frac{\eta_j}{\nu_j(i)} \right) \tag{21}$$

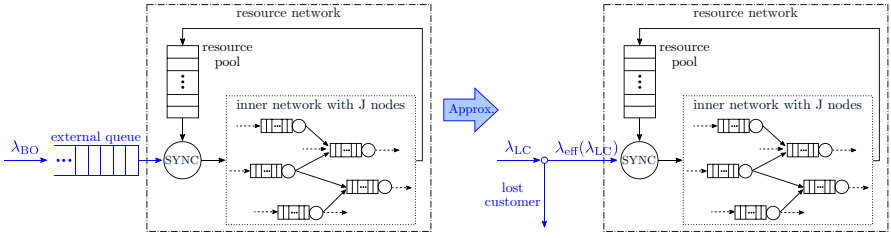

Fig. 4 Transition from SOQN-BO (left) to SOQN-LC (right)

with normalisation constant

$$\begin{aligned} C_{\mathrm{LC}}(\bar{J}_0, N) &:= \sum_{\sum_{j\in\bar{J}_0} n_j = N} \left(\frac{\eta_0}{\lambda_{\mathrm{LC}}}\right)^{n_0} \cdot \prod_{j=1}^{J} \left(\prod_{i=1}^{n_j} \frac{\eta_j}{\nu_j(i)} \right) \\ &= \sum_{n_0=0}^{N} \left(\frac{\eta_0}{\lambda_{\mathrm{LC}}}\right)^{n_0} \sum_{\sum_{j\in\bar{J}} n_j = N-n_0} \prod_{j=1}^{J} \left(\prod_{\ell=1}^{n_j} \frac{\eta_j}{\nu_j(\ell)} \right). \end{aligned} \tag{22}$$

3.2 Adjustment

We use the modified system (SOQN-LC) to approximate the SOQN-BO. First, we ensure that both systems process in the mean the same number of customers, i.e. they have the same throughput at (synchronisation node) 0. Our main idea is: To compensate customer losses, we increase the input rate λ_{LC} of the modified system until the desired throughput is reached. We will prove this in Theorem 2. Before, we calculate the throughput of both systems.

Lemma 1 *The throughput of the SOQN-BO in steady state is λ_{BO}. The throughput of the SOQN-LC in steady state is*

$$\lambda_{\mathrm{eff}}(\lambda_{\mathrm{LC}}) = \lambda_{\mathrm{LC}} \cdot \Bigg(1 - \underbrace{\frac{C^{\mathrm{stb}}(\bar{J}, N)}{C_{\mathrm{LC}}(\bar{J}_0, N)}}_{=\pi_{\mathrm{LC},0}(0)}\Bigg)$$

where $\pi_{\mathrm{LC},0}(0)$ is the probability of an empty resource pool in the SOQN-LC.

Proof Because all customers pass the SOQN-BO, in steady state its throughput is λ_{BO}. In the SOQN-LC, a portion $\pi_{\mathrm{LC},0}(0)$ of the arrivals is lost. From the steady-state distribution (21), we find the idling probability $\pi_{\mathrm{LC},0}(0)$ of the resource pool in the SOQN-LC as

$$\pi_{\mathrm{LC},0}(0) := \sum_{\sum_{j\in\bar{J}} n_j = N} \pi_{\mathrm{LC}}\big(0, n_1, \ldots, n_J\big) \overset{(21)}{=} \frac{C^{\mathrm{stb}}(\bar{J}, N)}{C_{\mathrm{LC}}(\bar{J}_0, N)}. \tag{23}$$

Then the effective arrival rate $\lambda_{\mathrm{eff}}(\lambda_{\mathrm{LC}})$, which coincides with the throughput of the system, is $\lambda_{\mathrm{eff}}(\lambda_{\mathrm{LC}}) = \lambda_{\mathrm{LC}} \cdot \big(1 - \pi_{\mathrm{LC},0}(0)\big)$. □

We adjust λ_{LC} so that both systems have the same throughput. We assume that both systems are stable. For the SOQN-BO, by Proposition 1 stability is equivalent to $\lambda_{\mathrm{BO}} \in \big(0, \lambda_{\mathrm{BO,max}}\big)$. For the SOQN-LC, stability is granted for any arrival rate $\lambda_{\mathrm{LC}} \in (0, \infty)$, because the state space E_{LC} is finite.

Theorem 2 *For every stable SOQN-BO, there exists an SOQN-LC with arrival rate λ_{LC} such that both systems have the same throughput in steady state. Formally, with $\lambda_{\mathrm{BO,max}}$ from (2) this means*

$$\text{For all } \lambda_{\mathrm{BO}} \in \left(0, \lambda_{\mathrm{BO,max}}\right) \text{ exists } \lambda_{\mathrm{LC}} \in (0, \infty) \text{ with } \lambda_{\mathrm{eff}}(\lambda_{\mathrm{LC}}) = \lambda_{\mathrm{BO}}.$$

Proof We show that for any $\lambda_{\mathrm{BO}} \in \left(0, \lambda_{\mathrm{BO,max}}\right)$, the function $\lambda_{\mathrm{eff}}(\cdot)$ from Lemma 1 takes values larger and smaller than a prescribed λ_{BO}, and is continuous. By the intermediate value theorem, there exists λ_{LC} with $\lambda_{\mathrm{eff}}(\lambda_{\mathrm{LC}}) = \lambda_{\mathrm{BO}}$.

(i) $\lambda_{\mathrm{eff}}(\lambda_{\mathrm{LC}}) = \lambda_{\mathrm{LC}} \cdot \left(1 - \frac{C^{\mathrm{stb}}(\bar{J},N)}{C_{\mathrm{LC}}(\bar{J}_0,N)}\right)$ can be larger than any λ_{BO}. We have

$$\begin{aligned} C_{\mathrm{LC}}(\bar{J}_0, N) &= \sum_{\sum_{j\in\bar{J}_0} n_j = N} \left(\frac{\eta_0}{\lambda_{\mathrm{LC}}}\right)^{n_0} \cdot \prod_{j=1}^{J} \left(\prod_{i=1}^{n_j} \frac{\eta_j}{\nu_j(i)}\right) \\ &= \sum_{n_0=0}^{N} \left(\frac{\eta_0}{\lambda_{\mathrm{LC}}}\right)^{n_0} \cdot \sum_{\sum_{j\in\bar{J}} n_j = N - n_0} \prod_{j=1}^{J} \left(\prod_{i=1}^{n_j} \frac{\eta_j}{\nu_j(i)}\right) \\ &= \sum_{n_0=0}^{N} \left(\frac{\eta_0}{\lambda_{\mathrm{LC}}}\right)^{n_0} \cdot C^{\mathrm{stb}}(\bar{J}, N - n_0). \end{aligned} \tag{24}$$

To simplify notation, we define $b(n_0) := C^{\mathrm{stb}}(\bar{J}, N - n_0)$. Then

$$\begin{aligned} \lambda_{\mathrm{eff}}(\lambda_{\mathrm{LC}}) &= \lambda_{\mathrm{LC}} \cdot \left(1 - \frac{C^{\mathrm{stb}}(\bar{J},N)}{C_{\mathrm{LC}}(\bar{J}_0,N)}\right) \\ &= \lambda_{\mathrm{LC}} \cdot \left(1 - \frac{b(0)}{\sum_{n_0=0}^{N} \left(\frac{\eta_0}{\lambda_{\mathrm{LC}}}\right)^{n_0} b(n_0)}\right) \\ &= \lambda_{\mathrm{LC}} \cdot \left(\frac{\sum_{n_0=1}^{N} \left(\frac{\eta_0}{\lambda_{\mathrm{LC}}}\right)^{n_0} b(n_0)}{\sum_{n_0=0}^{N} \left(\frac{\eta_0}{\lambda_{\mathrm{LC}}}\right)^{n_0} b(n_0)}\right) \\ &= \eta_0 \cdot \left(\frac{\sum_{n_0=1}^{N} \left(\frac{\eta_0}{\lambda_{\mathrm{LC}}}\right)^{n_0-1} b(n_0)}{b(0) + \sum_{n_0=1}^{N} \left(\frac{\eta_0}{\lambda_{\mathrm{LC}}}\right)^{n_0} b(n_0)}\right) \\ &= \eta_0 \cdot \left(\frac{b(1) + \sum_{n_0=2}^{N} \left(\frac{\eta_0}{\lambda_{\mathrm{LC}}}\right)^{n_0-1} b(n_0)}{b(0) + \sum_{n_0=1}^{N} \left(\frac{\eta_0}{\lambda_{\mathrm{LC}}}\right)^{n_0} b(n_0)}\right). \end{aligned}$$

Hence, it holds

$$\lim_{\lambda_{\mathrm{LC}} \to \infty} \lambda_{\mathrm{eff}}(\lambda_{\mathrm{LC}}) = \eta_0 \cdot \frac{b(1)}{b(0)} = \lambda_{\mathrm{BO,max}}.$$

Therefore, $\lambda_{\text{eff}}(\lambda_{\text{LC}})$ can be larger than any arrival rate $\lambda_{\text{BO}} \in \big(0, \lambda_{\text{BO,max}}\big)$.

(ii) $\lambda_{\text{eff}}(\lambda_{\text{LC}})$ can be smaller than any $\lambda_{\text{BO}} \in \big(0, \lambda_{\text{BO,max}}\big)$, because

$$\lim_{\lambda_{\text{LC}} \to 0} \lambda_{\text{eff}}(\lambda_{\text{LC}}) = \lim_{\lambda_{\text{LC}} \to 0} \lambda_{\text{LC}} \cdot \underbrace{\big(1 - \pi_{\text{LC},0}(0)\big)}_{>0 \text{ and } <1} = 0.$$

(iii) From $\lambda_{\text{eff}}(\lambda_{\text{LC}}) = \lambda_{\text{LC}} \cdot \left(1 - \frac{b(0)}{\sum_{n_0=0}^{N} \left(\frac{\eta_0}{\lambda_{\text{LC}}}\right)^{n_0} \cdot b(n_0)}\right)$ follows that λ_{eff} is a continuous function of $\lambda_{\text{LC}} \in (0, \infty)$, which proves our claim by the intermediate value theorem. □

Henceforth, we call λ_{LC} with $\lambda_{\text{eff}}(\lambda_{\text{LC}}) = \lambda_{\text{BO}}$ *adjusted* arrival rate for λ_{BO}.

Explicit results for adjusted λ_{LC} for $N = 1$ and $N = 2$ can be found in Remark 3 in Appendix A. The proof of Proposition 6 is given in Appendix A.

Proposition 6 *If the service rates $\nu_j(\cdot)$, $j \in \overline{J}$, are non-decreasing, λ_{LC} in Theorem 2 is unique.*

3.3 Throughputs and idle times

Theorem 2 only guarantees that for a stable SOQN-BO with arrival rate λ_{BO} an SOQN-LC with the same resource network and adjusted arrival rate exists such that the resource pools have the same throughput. Because an SOQN-LC can be investigated with standard Gordon–Newell network techniques, all local throughputs can be computed. We shall prove, that these local throughputs are the same as those in the SOQN-BO. This suggests to use local performance characteristics of the queues in the resource network of the SOQN-LC as approximation for performance measures of the SOQN-BO.

Proposition 7 *The local throughput $TH_{\text{LC},j}$ at nodes $j \in \overline{J}_0$ in the SOQN-LC with adjusted arrival rate is pairwise the same as that of the respective nodes in the SOQN-BO given in Proposition 2. With*

$$C_{\text{LC}}(\overline{J}_0, N) = \sum_{n_0=0}^{N} \left(\frac{\eta_0}{\lambda_{\text{LC}}}\right)^{n_0} \sum_{\sum_{j \in \overline{J}} n_j = N - n_0} \prod_{j=1}^{J} \left(\prod_{\ell=1}^{n_j} \frac{\eta_j}{\nu_j(\ell)}\right)$$

it holds:

$$TH_{\text{LC},j} = \eta_j \cdot \frac{C_{\text{LC}}(\overline{J}_0, N-1)}{C_{\text{LC}}(\overline{J}_0, N)} = \lambda_{\text{BO}} \cdot \frac{\eta_j}{\eta_0}, \quad j \in \overline{J}_0.$$

Proof It was shown in the proof of Theorem 2 that with λ_{LC} as adjusted arrival rate for λ_{BO} it holds

$$TH_{\mathrm{LC},0} = \lambda_{\mathrm{eff}}(\lambda_{\mathrm{LC}}) = \eta_0 \cdot \frac{C_{\mathrm{LC}}(\overline{J}_0, N-1)}{C_{\mathrm{LC}}(\overline{J}_0, N)}.$$

It is well known (Chen and Yao 2001, Section 2.3), that the joint queue length vector $Z_{\mathrm{LC}} := \left(\left(Y_j(t) : j \in \overline{J}_0 \right) : t \geq 0 \right)$ behaves stochastically as the joint queue length vector of a Gordon–Newell network consisting of nodes in $\overline{J}_0$, where node 0 has service rate λ_{LC} and nodes in $\overline{J}$ have the same characteristics as in the SOQN-LC. Because $\lambda_{\mathrm{eff}}(\lambda_{\mathrm{LC}}) = \lambda_{\mathrm{BO}}$ we have $\lambda_{\mathrm{BO}} = \eta_0 \cdot \frac{C_{\mathrm{LC}}(\overline{J}_0, N-1)}{C_{\mathrm{LC}}(\overline{J}_0, N)}$ and from the formula for throughputs in Gordon–Newell networks it follows

$$TH_{\mathrm{BO},j} = \lambda_{\mathrm{BO}} \cdot \frac{\eta_j}{\eta_0} = \eta_j \cdot \frac{C_{\mathrm{LC}}(\overline{J}_0, N-1)}{C_{\mathrm{LC}}(\overline{J}_0, N)} = TH_{\mathrm{LC},j}, \quad j \in \overline{J}_0.$$

□

The explicit formulas for the throughputs in Proposition 7 allow to determine efficiently the steady-state marginal distribution of the queue length at every node $j \in \overline{J}$ without knowing the adjusted value $\lambda_{\mathrm{eff}}(\lambda_{\mathrm{LC}})$ of λ_{LC}. This leads especially to

Proposition 8 *Let $Y_{\mathrm{LC}} := (Y_{\mathrm{LC},j} : j \in \overline{J})$ denote a random vector which is distributed according to the stationary queue length at the nodes in $\overline{J}$ of the SOQN-LC with adjusted arrival rate.*

If the service rate at node j does not depend on the queue length, i.e. $\nu_j(\cdot) = \nu_j, j \in \overline{J}$, then the probabilities that the nodes $j \in \overline{J}_0$ in the SOQN-LC with adjusted arrival rate are idling are pairwise the same as those of the respective nodes in the SOQN-BO given in Corollary 1:

$$P(Y_{\mathrm{LC},j} = 0) = 1 - \lambda_{\mathrm{BO}} \cdot \frac{\eta_j}{\eta_0} \cdot \nu_j^{-1}.$$

Proof According to Proposition 7, the throughputs are equal. The rest of the proof is the same as the proof of Corollary 1. □

4 Approximation of the external queue

We have shown that, after adjusting λ_{LC}, the behaviour of the resource network of the original SOQN-BO can be approximated well by the behaviour of the resource network of the modified SOQN-LC. The behaviour of the external queue of the SOQN-BO is represented only by the modified arrival intensity of the associated SOQN-LC. On the other side characteristics of the external queue are important performance measures of the original system. We solve this problem in a two-step approach to approximate the external queue:

Step 1 In Sect. 4.1, we reduce the modified system to a simpler SOQN-LC.
Step 2 In Sect. 4.2, we combine the results from Sects. 4.1 and 2.4, obtaining a simple SOQN-BO to approximate the external queue.

4.1 Reduced SOQN with lost customers

Because the joint queue length vector $Z_{\mathrm{LC}} := \left(\left(Y_j(t) : j \in \bar{J}_0\right) : t \geq 0\right)$ of the SOQN-LC from Sect. 3.1 can be studied via a Gordon–Newell network, we can reduce complexity further by applying Norton's theorem (Chandy et al. 1975) to construct a two-node Gordon–Newell network as shown in Fig. 5, with the same throughput.

The inner network is replaced by a composite node ($\bar{J} := \{1\}$): A single exponential-1-server with infinite waiting room under FCFS regime with a queue-length-dependent service intensity $\varphi(\cdot)$ (Chandy et al. 1975, p. 39, eq. (20)), with

$$\varphi(0) = 0, \qquad \varphi(m) = \eta_0 \cdot \frac{C^{\mathrm{stb}}(\bar{J}, m-1)}{C^{\mathrm{stb}}(\bar{J}, m)}, \quad m \in \{1, \ldots, N\}. \tag{25}$$

Remarkably, $\varphi(N)$ equals $\lambda_{\mathrm{BO,max}}$ in (2). We deduce from (10) that

$$\varphi(m) = TH_0^{\mathrm{stb}}(m), \quad m \in \{1, \ldots, N\}, \tag{26}$$

which is independent of λ_{LC}. The normalisation constants $C^{\mathrm{stb}}(\bar{J}, m)$, $m \in \{0, \ldots, N\}$, can be calculated by the convolution algorithm or mean value analysis (MVA), see (Bolch et al. 1998, p. 371ff., Section 8.1 and p. 384ff., Section 8.2).

4.2 Back to backordering

Recall that Theorem 2 guarantees, that for every stable SOQN-BO there exists an SOQN-LC with adjusted arrival rate λ_{LC} such that both systems have the same throughput in steady state. Our next step is to apply an "inversion" of that construction to the reduced SOQN-LC on the right side of Fig. 5. This results in removing the lost-customer property to get again the backordering property as shown in Fig. 6. The recipe is simple: We reopen the external queue and reduce the arrival intensity λ_{LC} to λ_{BO} while the service rate at the single queue of the inner network is $\nu_1(m) := \varphi(m) = TH_0^{\mathrm{stb}}(m)$ from Eq. (26). The final result is: The external queue of the large SOQN-BO with $J > 1$ approximated by a reduced SOQN-BO with $J = 1$. This yields the following approximating performance characteristics for the external queue of the SOQN-BO of Sect. 2.

Denote by $\left(X_{\mathrm{ex}}, \mathbf{Y}\right)$ a vector distributed according to the stationary distribution of Z_{BO}, L_{ex} the mean external queue length, and W_{ex} the mean waiting time at the external queue length in steady state. These characteristics are approximated by the respective quantities from the system at the right side of Fig. 6, given in Proposition 3.

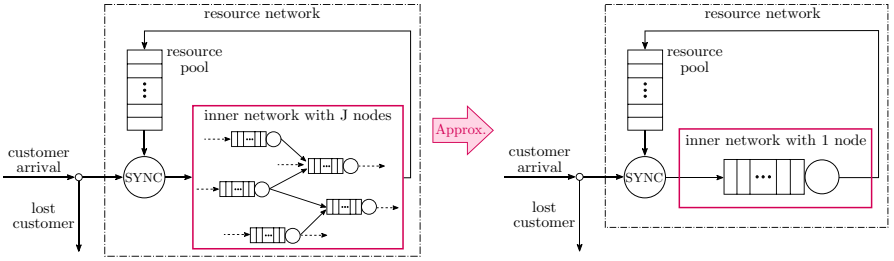

Fig. 5 Step 1: Reduction of complexity

$$P(X_{\text{ex}} = 0) \approx P(\widehat{X}_{\text{ex}} = 0) \overset{(14)}{=} \left[C_{\text{BO}}(\{1\}, N)\right]^{-1} \cdot \sum_{n_1=0}^{N} \prod_{m=1}^{n_1} \frac{\lambda_{\text{BO}}}{TH_0^{\text{stb}}(m)}$$

and for $n_{\text{ex}} > 0$:

$$\begin{aligned} P(X_{\text{ex}} = 0) &\approx P(\widehat{X}_{\text{ex}} = n_{\text{ex}}) \\ &\overset{(15)}{=} \left[C_{\text{BO}}(\{1\}, N)\right]^{-1} \cdot \left(\frac{\lambda_{\text{BO}}}{TH_0^{\text{stb}}(N)}\right)^{n_{\text{ex}}} \cdot \prod_{m=1}^{N} \frac{\lambda_{\text{BO}}}{TH_0^{\text{stb}}(m)} \end{aligned}$$

with

$$C_{\text{BO}}(\{1\}, N) \overset{(13)}{=} \sum_{n_1=0}^{N-1} \prod_{m=1}^{n_1} \frac{\lambda_{\text{BO}}}{TH_0^{\text{stb}}(m)} + \frac{1}{1 - \frac{\lambda_{\text{BO}}}{TH_0^{\text{stb}}(N)}} \cdot \prod_{m=1}^{N} \frac{\lambda_{\text{BO}}}{TH_0^{\text{stb}}(m)}.$$

The mean external queue length is approximated with Eq. (17), (18) by

$$\begin{aligned} L_{\text{ex}} &\approx \widehat{L_{\text{ex}}} \\ &= \frac{1}{C_{\text{BO}}(\{1\}, N)} \cdot \frac{1}{1 - \frac{\lambda_{\text{BO}}}{TH_0^{\text{stb}}(N)}} \cdot \prod_{m=1}^{N} \frac{\lambda_{\text{BO}}}{TH_0^{\text{stb}}(m)} \cdot \frac{\lambda_{\text{BO}}}{TH_0^{\text{stb}}(N) - \lambda_{\text{BO}}}. \end{aligned} \tag{27}$$

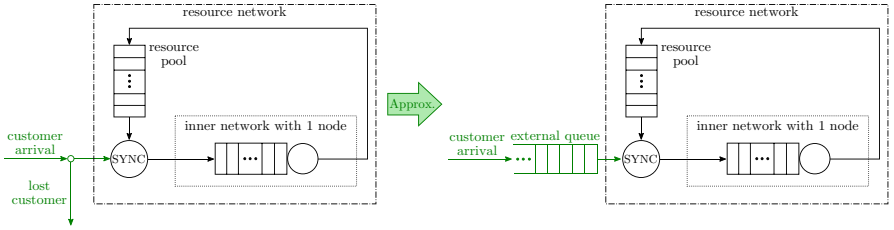

Fig. 6 Step 2: Transition from reduced SOQN-LC to reduced SOQN-BO

With our approximation we arrive at the same formula for L_{ex} as (Dallery 1990, eq. (22)) obtained by aggregation technique in (Dallery 1990, Section 6). Using (19) the average waiting time of customers in the external queue is approximated as

$$W_{\mathrm{ex}} \approx \widehat{W_{\mathrm{ex}}} = \frac{\widehat{L_{\mathrm{ex}}}}{\lambda_{\mathrm{BO}}}. \tag{28}$$

Remark: We expect that the results are close to the true values, but at present we do not have strict error bounds.

5 Application to RMFS

We evaluate analytically the performance of a robotic mobile fulfilment system (RMFS) modelled as an SOQN-BO. In an RMFS, robots are expensive resources. Therefore, we determine the minimal number of robots needed to stabilize the system or to maintain a required quality of service. From Sect. 2, an SOQN-BO can be described by a level-independent quasi-birth-death-process, so numerical schemes are at hand. Due to the large state space direct application of these matrix-geometric methods is not practical. For example, for 10 robots, we need to calculate ca. $\left(9 \cdot 10^4\right)^2$ entries of a special matrix. Therefore, approximative matrix-geometric methods are developed.

5.1 Description of RMFS

The components and the order fulfilment processes in an RMFS with an illustrated example are depicted in Fig. 7. Central components are:

- movable shelves, called *pods*, on which items are stored,
- *storage area*—the area where the pods are stored,
- workstations, where
  - the items are picked from pods by pickers (*picking stations*) or
  - the items are stored to pods (*replenishment stations*),
- mobile *robots*, which can move underneath pods and carry them to workstations.

Figure 7 illustrates order fulfilment processes in an RMFS. On the upper left hand we have three customers' orders which contain different items, distinguished by colours. To fulfil orders, we send them or parts of them to picking stations. To these stations we send pods with the necessary items. Each pod is carried by a robot. In this way, customers' orders generate tasks for robots. The robots, with their pods, queue up in front of the picking stations. A picker takes all the necessary items from the pod at the head of the queue. Then he sends the pod with its robot back to the storage area. As soon as the customer's order or part of it is fulfilled, we remove it from the picking station. The order in which we send the customers' orders, how we

split them apart, and which pod we send, is a complex topic, see Xie et al. (2021). In the present paper, we focus on the generated robots' tasks, which we will call just *tasks*.

In the example, each customer's order is split into two parts. Three parts are sent to picking station 1, and three to picking station 2. To fulfil these partial orders, a robot transports one pod to picking station 1, and another robot transports one pod to picking station 2, from the storage area.

From time to time, we need to refill pods. To do this, we send these pods to the replenishment station. There, employees refill the pods and send them back to the storage area. In the example, after picking, pod 2 is sent to the replenishment station to refill it with blue items.

5.2 Modelling as SOQN

An SOQN-BO as model for the RMFS from Fig. 7 is depicted in Fig. 8. The RMFS is open with respect to tasks and closed with respect to robots, which are the resources. It has two picking and one replenishment station.

Customers' orders arrive at the RMFS with rate λ_{CO} and generate tasks. The number of tasks, which a single customer's orders can generate, depends on many parameters. In particular, it depends on the efficiency of the algorithm which

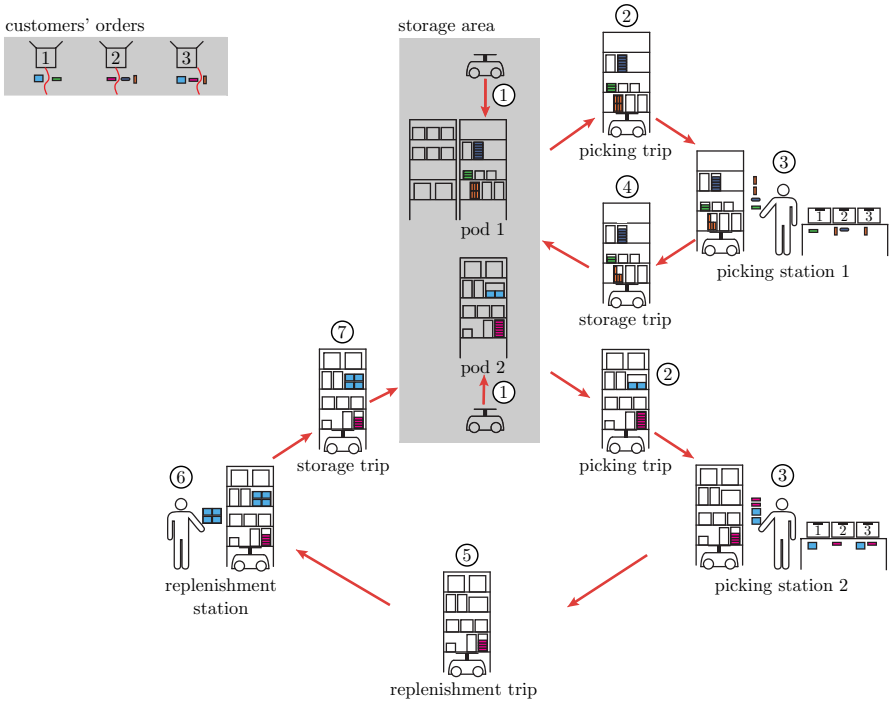

Fig. 7 Order fulfilment processes in RMFS. The circled numbers refer to the processes in Fig. 8

tries to find an optimal match between customers' orders and pods. The matching problem is NP-hard, and, to the best of our knowledge, there are no formulas known to determine how many pods an order will need. Therefore, we assume that there exists an average pod/order ratio $\sigma_{\text{pod/order}}$ which we find empirically for an RMFS. We assume this ratio depends only on the pods' contents and customers' order contents and is independent of the number of robots.

The matching algorithm adds some delay in order to assign pods to orders. We assume that this delay depends only on the pods' contents, customers' order contents and order input rate and is independent of the number of robots. We assume that we can find the average delay W_{alg} empirically for our RMFS.

Thus, the customers' orders generate a stream of "bring a pod to a picking station" tasks with rate $\lambda_{\text{BO}} = \lambda_{\text{CO}} \cdot \sigma_{\text{pod/order}} > 0$. The delay, introduced by the matching algorithm, does not change this rate.

We reduce the complexity of creating a task and model the task stream as a Poisson stream with rate $\lambda_{\text{BO}} = \lambda_{\text{CO}} \cdot \sigma_{\text{pod/order}}$. To be processed (= to enter the inner network), each such task requires exactly one idle robot from the robot pool (resource pool), which is henceforth referred to as node 0. If there is no idle robot available, the new task has to wait in an external queue until a robot becomes available ("backordering"). The maximal number of robots in the resource pool is N. The inner network in the example in Fig. 8 consists of 11 nodes, denoted by

$$\overline{J} := \left\{\text{sp}, \text{pp}_1, \text{pp}_2, \text{p}_1, \text{p}_2, \text{p}_1\text{s}, \text{p}_2\text{s}, \text{p}_1\text{r}, \text{p}_2\text{r}, \text{r}, \text{rs}\right\}.$$

The meaning and notations of nodes are given in Table 1. The robot with assigned task moves through the network. The following processes occur from the perspective of a robot:

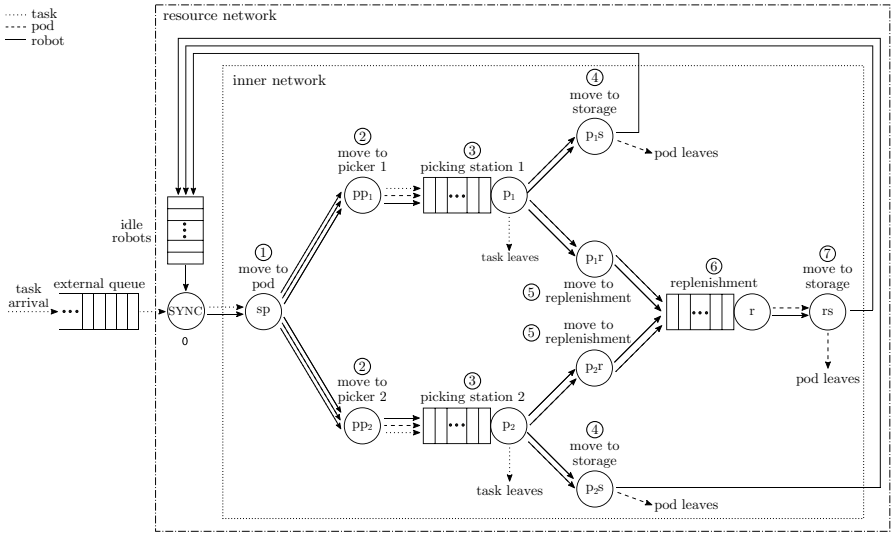

Fig. 8 RMFS modelled as an SOQN-BO. The circled numbers refer to the processes in Fig. 7

- The idle robot waits to be assigned to a task (bring a particular pod).
- The robot moves with the assigned task to a pod.
- With this pod the robot moves with probability $q_{\mathrm{pp}_1} \in (0,1)$ to picking station 1 and with probability $q_{\mathrm{pp}_2} \in (0,1)$ to station 2, $q_{\mathrm{pp}_1} + q_{\mathrm{pp}_2} = 1$.
- The robot queues with the pod at the picking stations.
- After picking at picking station 1 resp. picking station 2, the robot:
  - either
    - carries the pod directly back to the storage area with probability $q_{\mathrm{p}_1\mathrm{s}} \in (0,1)$ resp. $q_{\mathrm{p}_2\mathrm{s}} \in (0,1)$, or
    - moves to the replenishment station with probability $q_{\mathrm{p}_1\mathrm{r}} \in (0,1)$ resp. $q_{\mathrm{p}_2\mathrm{r}} \in (0,1)$, whereby $q_{\mathrm{p}_1\mathrm{s}} + q_{\mathrm{p}_1\mathrm{r}} = 1$ resp. $q_{\mathrm{p}_2\mathrm{s}} + q_{\mathrm{p}_2\mathrm{r}} = 1$,
  - queues at the replenishment station, and
  - carries the pod back to the storage area and waits for the next task.

Each of these processes is modelled as a queue. Movements of the robots are modelled by processor-sharing nodes with exponential service times with intensities $\nu_j(n_j) := \mu_j \cdot \phi_j(n_j)$, $j \in \overline{J} \setminus \{\mathrm{p}_1, \mathrm{p}_2, \mathrm{r}\}$, presented in Table 1.

The two picking stations and the replenishment station, which are referred to as node p_1, node p_2 resp. node r, consist of a single server with waiting room under the FCFS regime. The picking times and the replenishment times are exponentially distributed with rates ν_{p_1}, ν_{p_2} resp. ν_r.

The robots travel among the nodes following an irreducible routing matrix $\mathcal{R} := \left(r(i,j) : i,j \in \overline{J}_0\right)$, whereby $\overline{J}_0 := \{0\} \cup \overline{J}$, which is given by

$$\mathcal{R} = \left(\begin{array}{c|cccccccccccc}
 & 0 & \mathrm{sp} & \mathrm{pp}_1 & \mathrm{pp}_2 & \mathrm{p}_1 & \mathrm{p}_2 & \mathrm{p}_1\mathrm{s} & \mathrm{p}_2\mathrm{s} & \mathrm{p}_1\mathrm{r} & \mathrm{p}_2\mathrm{r} & \mathrm{r} & \mathrm{rs} \\ \hline
0 & & 1 & & & & & & & & & & \\
\mathrm{sp} & & & q_{\mathrm{pp}_1} & q_{\mathrm{pp}_2} & & & & & & & & \\
\mathrm{pp}_1 & & & & & 1 & & & & & & & \\
\mathrm{pp}_2 & & & & & & 1 & & & & & & \\
\mathrm{p}_1 & & & & & & & q_{\mathrm{p}_1\mathrm{s}} & & q_{\mathrm{p}_1\mathrm{r}} & & & \\
\mathrm{p}_2 & & & & & & & & q_{\mathrm{p}_2\mathrm{s}} & & q_{\mathrm{p}_2\mathrm{r}} & & \\
\mathrm{p}_1\mathrm{s} & 1 & & & & & & & & & & & \\
\mathrm{p}_2\mathrm{s} & 1 & & & & & & & & & & & \\
\mathrm{p}_1\mathrm{r} & & & & & & & & & & & 1 & \\
\mathrm{p}_2\mathrm{r} & & & & & & & & & & & 1 & \\
\mathrm{r} & & & & & & & & & & & & 1 \\
\mathrm{rs} & 1 & & & & & & & & & & &
\end{array}\right).$$

We define the joint stochastic process Z of this system by

$$Z := \Big(\Big(X_{\mathrm{ex}}(t), Y_0(t), Y_{\mathrm{sp}}(t), Y_{\mathrm{pp}_1}(t), Y_{\mathrm{pp}_2}(t), Y_{\mathrm{p}_1}(t), Y_{\mathrm{p}_2}(t), Y_{\mathrm{p}_1\mathrm{s}}(t), Y_{\mathrm{p}_2\mathrm{s}}(t), Y_{\mathrm{p}_1\mathrm{r}}(t), Y_{\mathrm{p}_2\mathrm{r}}(t), Y_{\mathrm{r}}(t), Y_{\mathrm{rs}}(t)\Big) : t \geq 0\Big).$$

Table 1 Overview of the nodes in the network

Node	Service intensity	Random variable	State	Description (number of robots at time t)
sp	$\mu_{\mathrm{sp}} \cdot \phi_{\mathrm{sp}}(n_{\mathrm{sp}})$	$Y_{\mathrm{sp}}(t)$	n_{sp}	Moving in the storage area to a pod
pp_1	$\mu_{\mathrm{pp}_1} \cdot \phi_{\mathrm{pp}_1}(n_{\mathrm{pp}_1})$	$Y_{\mathrm{pp}_1}(t)$	n_{pp_1}	Moving a pod from the storage area to picking station 1
pp_2	$\mu_{\mathrm{pp}_2} \cdot \phi_{\mathrm{pp}_2}(n_{\mathrm{pp}_2})$	$Y_{\mathrm{pp}_2}(t)$	n_{pp_2}	Moving a pod from the storage area to picking station 2
p_1	ν_{p_1}	$Y_{\mathrm{p}_1}(t)$	n_{p_1}	In the queue of picking station 1
p_2	ν_{p_2}	$Y_{\mathrm{p}_2}(t)$	n_{p_2}	In the queue of picking station 2
$\mathrm{p}_1\mathrm{s}$	$\mu_{\mathrm{p}_1\mathrm{s}} \cdot \phi_{\mathrm{p}_1\mathrm{s}}(n_{\mathrm{p}_1\mathrm{s}})$	$Y_{\mathrm{p}_1\mathrm{s}}(t)$	$n_{\mathrm{p}_1\mathrm{s}}$	Moving a pod from picking station 1 to the storage area and entering node 0
$\mathrm{p}_2\mathrm{s}$	$\mu_{\mathrm{p}_2\mathrm{s}} \cdot \phi_{\mathrm{p}_2\mathrm{s}}(n_{\mathrm{p}_2\mathrm{s}})$	$Y_{\mathrm{p}_2\mathrm{s}}(t)$	$n_{\mathrm{p}_2\mathrm{s}}$	Moving a pod from picking station 2 to the storage area and entering node 0
$\mathrm{p}_1\mathrm{r}$	$\mu_{\mathrm{p}_1\mathrm{r}} \cdot \phi_{\mathrm{p}_1\mathrm{r}}(n_{\mathrm{p}_1\mathrm{r}})$	$Y_{\mathrm{p}_1\mathrm{r}}(t)$	$n_{\mathrm{p}_1\mathrm{r}}$	Moving a pod from picking station 1 to the replenishment station
$\mathrm{p}_2\mathrm{r}$	$\mu_{\mathrm{p}_2\mathrm{r}} \cdot \phi_{\mathrm{p}_2\mathrm{r}}(n_{\mathrm{p}_2\mathrm{r}})$	$Y_{\mathrm{p}_2\mathrm{r}}(t)$	$n_{\mathrm{p}_2\mathrm{r}}$	Moving a pod from picking station 2 to the replenishment station
r	ν_r	$Y_{\mathrm{r}}(t)$	n_{r}	In the queue of the replenishment station
rs	$\mu_{\mathrm{rs}} \cdot \phi_{\mathrm{rs}}(n_{\mathrm{rs}})$	$Y_{\mathrm{rs}}(t)$	n_{rs}	Moving a pod from the replenishment station to the storage area and entering node 0

Due to the usual independence and memoryless assumptions, Z is an irreducible Markov process with state space

$$E := \left\{ \left(0, k_{\text{idle robots}}, n_{\text{sp}}, n_{\text{pp}_1}, n_{\text{pp}_2}, n_{\text{p}_1}, n_{\text{p}_2}, n_{\text{p}_1\text{s}}, n_{\text{p}_2\text{s}}, n_{\text{p}_1\text{r}}, n_{\text{p}_2\text{r}}, n_{\text{r}}, n_{\text{rs}}\right) : \right.$$
$$\left. n_j \in \{0, \dots, N\}\ \forall j \in \overline{J}_0,\ \sum_{j \in \overline{J}_0} n_j = N \right\}$$
$$\cup \left\{ \left(n_{\text{ex}}, 0, n_{\text{sp}}, n_{\text{pp}_1}, n_{\text{pp}_2}, n_{\text{p}_1}, n_{\text{p}_2}, n_{\text{p}_1\text{s}}, n_{\text{p}_2\text{s}}, n_{\text{p}_1\text{r}}, n_{\text{p}_2\text{r}}, n_{\text{r}}, n_{\text{rs}}\right) : \right.$$
$$\left. n_{\text{ex}} \in \mathbb{N},\ n_j \in \{0, \dots, N\}\ \forall j \in \overline{J},\ \sum_{j \in \overline{J}} n_j = N \right\}.$$

5.3 Determine the minimal number of robots

The throughput $TH_0^{\text{stb}}(N)$ in Eq. (9) depends on the number of robots N. To find the minimal number of robots that stabilises the system, we check the stability criterion from Proposition 1. The maximal number of robots $N^{\max}$ is the number of pods or is determined by financial restrictions. Algorithm 1 determines the set $\overline{N}^*$ of feasible numbers of robots for a stable system.

Algorithm 1 Calculate set of feasible numbers of robots for a stable system

```
function StableRobotsSet
    N̄* = {N ∈ {1, ..., N^max} : λ_BO < TH_0^stb(N)}
    return N̄*
end function
```

Remark 2 If $TH_0^{\text{stb}}(\cdot)$ is non-decreasing in $\mathbb{N}$, the algorithm can be shortened: Starting with one robot and adding a new robot in each step until the stability criterion is satisfied for the first time. Sufficient conditions for non-decreasing $TH_0^{\text{stb}}(\cdot) = \lambda_{\text{BO,max}}$ are given in Propositions 5 and 4 .

Stability does not guarantee acceptable turnover times of orders. Therefore, we consider additionally the turnover time of customers' order. The turnover time of a customer's order can be split into three main parts:

1. Waiting time until the matching algorithm has assigned all required pods to that order. By assumption, this time does not depend on the number of robots and is on average $W_{\text{alg}} > 0$.

2. Waiting time of the first matched pod for an idle robot, time for transport to the picking station, waiting time for the picker at the picking station. During all these times, the order is coupled with at least one task. We call this turnover time for the task $TO_{\text{task}}(\lambda_{\text{LC}}, N)$.
3. Time of an order between start of picking and its completion. This time depends on many factors, for example: How many orders can a picker complete with the same pod? Will all completed orders wait until a pod leaves? Is the order's content in multiple pods? Will these pods arrive right after each other, or will there be many pods for other orders in between? Is there any complex merging procedure outside of the picking station? In our model, we use a simplifying assumption that the order needs on average $W_{\text{assembled}} > 0$ from the time its first pod arrives at the picking station until the time picking for this order is completed.

With these assumptions, we can assume that the turnover time of an order is

$$TO_{\text{order}}(\lambda_{\text{LC}}, N) := W_{\text{alg}} + TO_{\text{task}}(\lambda_{\text{LC}}, N) + W_{\text{assembled}}.$$

Even if W_{alg} and $W_{\text{assembled}}$ are unknown, we can still use $TO_{\text{task}}(\lambda_{\text{LC}}, N)$ as a lower bound for $TO_{\text{order}}(\lambda_{\text{LC}}, N)$.

Because of the simplifying assumption about W_{alg} and $W_{\text{assembled}}$, only the turnover time $TO_{\text{task}}(\lambda_{\text{LC}}, N)$ of a task depends on N, and for the minimal number of robots we can focus on this.

The turnover time $TO_{\text{task}}(\lambda_{\text{LC}}, N)$ of a task is measured from the time the task is received to the time the picker starts to process it:

$$TO_{\text{task}}(\lambda_{\text{LC}}, N) := W_{\text{ex}}(N) + W_{\text{in}}(\lambda_{\text{LC}}, N).$$

$W_{\text{ex}}(N)$ is the average time a task spends waiting in the external queue until it enters the inner network. We can calculate it with Eq. (28). $W_{\text{in}}(\lambda_{\text{LC}}, N)$ is the average time a task spends in the inner network until a picker starts to process it at one of the picking stations. Given the average waiting times $W_j(\lambda_{\text{LC}}, N)$ at nodes $j \in \overline{J} = \{\text{sp}, \text{pp}_1, \text{pp}_2, \text{p}_1, \text{p}_2, \text{p}_1\text{s}, \text{p}_2\text{s}, \text{p}_1\text{r}, \text{p}_2\text{r}, \text{r}, \text{rs}\}$ from arrival until service completion, and constant service rates ν_j at nodes $j \in \{\text{p}_1, \text{p}_2\}$, then

$$\begin{aligned} W_{\text{in}}(\lambda_{\text{LC}}, N) := {} & W_{\text{sp}}(\lambda_{\text{LC}}, N) \\ & + r(\text{sp}, \text{pp}_1) \cdot \left(W_{\text{pp}_1}(\lambda_{\text{LC}}, N) + W_{\text{p}_1}(\lambda_{\text{LC}}, N) - 1/\nu_{\text{p}_1}\right) \\ & + r(\text{sp}, \text{pp}_2) \cdot \left(W_{\text{pp}_2}(\lambda_{\text{LC}}, N) + W_{\text{p}_2}(\lambda_{\text{LC}}, N) - 1/\nu_{\text{p}_2}\right). \end{aligned}$$

We calculate $W_j(\lambda_{\text{LC}}, N)$, $j \in \overline{J}$, with MVA.

In Algorithm 2, we determine the minimal number of robots for an acceptable turnover time of a task. We will call this time TO_{task}^{max}.

Algorithm 2 Calculate the minimal number of robots for acceptable turnover time of a task

```
function MinimalRobots(N̄*, TO_task^max)
    while N̄* ≠ {} do
        N ← min(N̄*)
        calculate λ_LC with λ_eff(λ_LC) = λ_BO
        if TO_task(λ_LC, N) ≤ TO_task^max then
            return N
        else
            N̄* ← N̄* \ {N}
        end if
    end while
    return "no solution"
end function
```

5.4 Numerical experiments

In this section, we show by example (with data taken from the literature) how to apply our approximation for SOQNs. We compare the results with simulations of the original SOQN, considering more details of the network. We investigate the performance of the system under prescribed utilisation levels for robots and discuss the quality of the approximation. Eventually, we investigate the effect of interarrival time variability on waiting times at some stations of the SOQN.

In our experiments, we take parameters from (Lamballais et al. 2019, Table 5.3 and Table 5.4). The maximal number of pods is $N^{\max} = 550$, arrival rate of tasks is $468\,\frac{\text{tasks}}{\text{h}} = 0.13\,\frac{\text{tasks}}{\text{s}}$ (in (Lamballais et al. 2019, Table 5.3 and Table 5.4), arrival rates are given in [order/hour]. Because every order generates one task, we use [task/hour] directly). Average travel time at node sp: $\mu_{\text{sp}}^{-1} = 18.4$ s, at node pp_1: $\mu_{\text{pp}_1}^{-1} = 34.5$ s, at node pp_2: $\mu_{\text{pp}_2}^{-1} = 34.5$ s, at node p_1s: $\mu_{\text{p}_1\text{s}}^{-1} = 34.5$ s, at node p_2s: $\mu_{\text{p}_2\text{s}}^{-1} = 34.5$ s, at node p_2r: $\mu_{\text{p}_2\text{r}}^{-1} = 34.5$ s, at node p_1r: $\mu_{\text{p}_1\text{r}}^{-1} = 34.5$ s, at node rs: $\mu_{\text{rs}}^{-1} = 34.5$ s. Average picking time of picking station 1: $\nu_{\text{p}_1}^{-1} = 10$ s, of picking station 2: $\nu_{\text{p}_2}^{-1} = 10$ s, average replenishment time (node r): $\nu_r^{-1} = 30$ s.

We assume that moving robots do not interfere. Hence, our processor-sharing queues are infinite server queues, i.e. $\phi_j(n_j) = n_j$ for all $j \in \bar{J} \setminus \{\text{p}_1, \text{p}_2, \text{r}\}$.

We implemented our algorithm in R and used the package *queueing*, see (Canadilla 2017). In the worst case scenario—when we need to try all $(550 - 18 + 1)$ of the robots—our implementation needs on average 83 seconds on a notebook with an i7-7600U CPU processor, 2.80GHz and 16GB RAM. We plotted important system parameters in Figs. 9, 10, 11 and 12. For better readability, we plotted data for a limited number of robots (until stabilising of curves).

Figure 9 shows maximal arrival rates λ_{BO} for given numbers N of robots to keep the system stable. In particular, the minimal number of robots for the system to be stable is 18 and for more than 40 robots, additional robots do not allow significantly higher arrival rates.

Figure 10 shows the throughputs of the nodes. By Proposition 2, these throughputs do not depend on N, and are pairwise the same for the original SOQN-BO and the adjusted approximation SOQN-LC. Idling probabilities of nodes p_1, p_2 and r are 0.35, 0.35, and 0.22, obtained with Corollary 1.

Figure 11 shows adjusted arrival rates λ_{LC} for an SOQN-LC to obtain an effective arrival rate $\lambda_{eff}(\lambda_{LC}) = \lambda_{BO} = 468\,h^{-1}$. Because in an SOQN-LC with many robots the probability of an empty resource pool is ≈ 0, only few customers are lost: λ_{LC} has to be adjusted only slightly, so $\lim_{N\to\infty} \lambda_{LC} \approx \lambda_{BO}$.

Figure 12 shows average turnover times. From arrival of an order (= one task) at the system until it is completed at a picking station. With only 18 robots, turnover times are extremely large, although the system is stable. Remarkably, turnover times decrease dramatically with only one additional robot.

Comparison with simulation. We simulated the SOQN-BO model of the RMFS for 365 days 20 times for each number of robots using SimPy 3.0. Figure 13 shows mean waiting times in the external queue starting with 19 robots. The simulated SOQN-BO and the SOQN-LC approximation show the same qualitative behaviour. Although the approximation under-estimates the "true values" (obtained by simulation) in the region of 19-25 robots, the approximation answers the question "how many robots do we need to obtain a target turnover time" quite well. From 26 robots

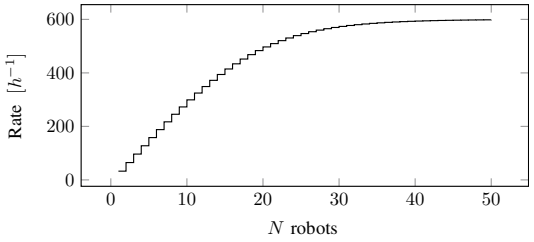

Fig. 9 Maximal arrival rates λ_{BO} for given numbers of robots to keep the system stable

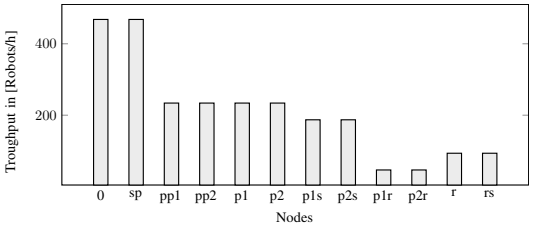

Fig. 10 Throughputs for each node of the RMFS example

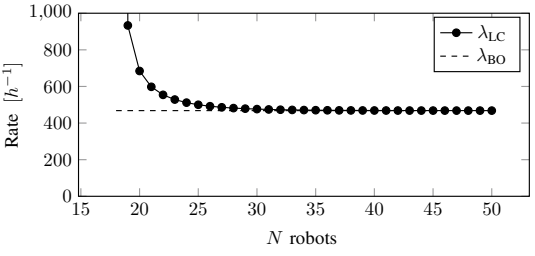

Fig. 11 Adjusted arrival rate λ_{LC} for a system with lost customers such that the effective arrival rate is $\lambda_{BO} = 468\,h^{-1}$

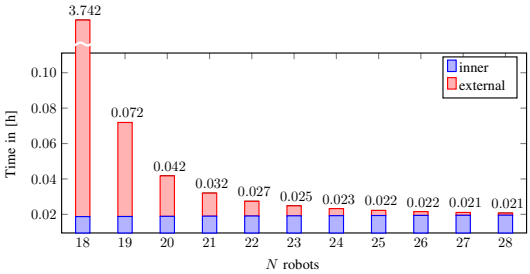

Fig. 12 Average turnover time $TO_{\text{task}}(\lambda_{\text{LC}}, N)$ of a task, which is the average delay of a task until it is completed

on, the approximation reflects the behaviour of the original system well. We omitted results for 18 robots because then the system operates on the edge of instability (utilisation ≈ 1) with extremely large mean value and standard deviation. We ran 200 additional simulations for this setting. Figure 14 shows the large variability of the average waiting time. So, operating a system under such conditions cannot be recommended. More details on utilisation for the system are shown in Fig. 15.

Figure 16 shows that our approximation works well for turnover times. To assess the quality of approximation better, recall that turnover times consist of transportation and waiting times for a picker. Average transportation times are equal in the original system and in the approximation. We calculate them from service times at appropriate nodes: $\mu_{\text{sp}}^{-1} + r(\text{sp}, \text{pp}_1) \cdot \mu_{\text{pp}_1}^{-1} + r(\text{sp}, \text{pp}_2) \cdot \mu_{\text{pp}_1}^{-1} = 0.0147\text{h}$. The hard part is to estimate average waiting times for pickers. Figure 17 shows that the approximation is good.

Although mean waiting times for replenishments are not needed for our optimisation, Fig. 18 demonstrates that our algorithm estimates further network characteristics well. (We admit that less impressive results may occur.)

Utilisation-dependent comparisons. Figures 13 and 15 emphasize the importance of systems' utilisation for a good fit of our lost-customers approximation of SOQNs. In this section, we study the goodness-of-fit for the approximation under prescribed utilisations of the robots.

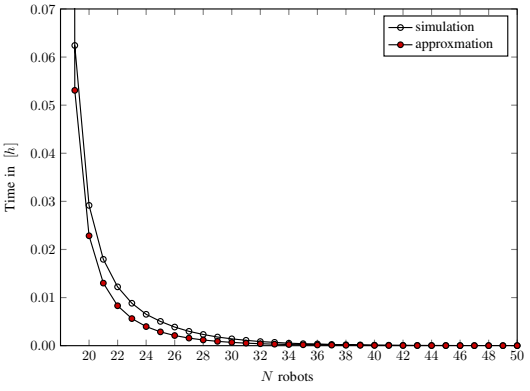

Fig. 13 Average waiting times in the external queue, simulation vs. approximation

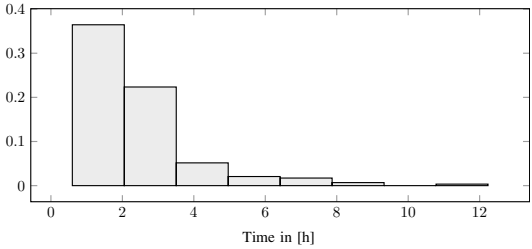

Fig. 14 Distribution of waiting times in the external queue for a system with 18 robots, obtained with 200 simulations

The network's structure (inner network and resource pool) is the same as before. We consider robot utilisations 0.95, 0.90, 0.80, 0.70 and 0.60 and determine for given number of robots $N = 1, 2, \ldots, 50$ the external arrival rate to the SOQN-BO which yields the required utilisation. Because simulation of the SOQN-BO is (especially for high utilisations) extremely time consuming we used an iteration based on the approximation procedure described in Sects. 2 and 3. For a guess of λ_{BO} we determine in the associated lost customer approximation with adjusted arrival rate the robots' utilisation. This is iterated until we meet the prescribed utilisation. So, by construction the utilisation of the approximation is in any

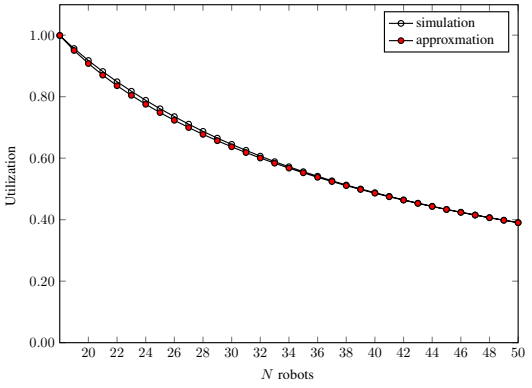

Fig. 15 Robots' utilisation: Proportion of time robots do not wait for assigned tasks, simulation vs. approximation

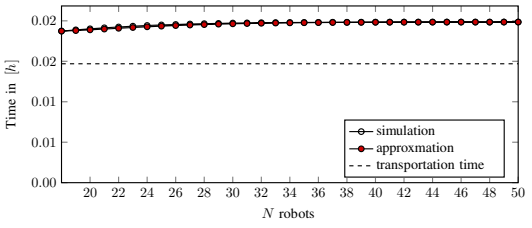

Fig. 16 Average turnover times, simulation vs. approximation. The graphs for approximation and simulation coincide. The large fraction of turnover times for transportation is the same in both systems by construction

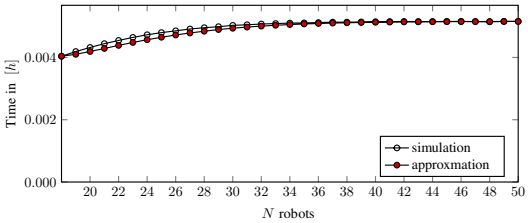

Fig. 17 Average waiting times for pickers, simulation vs. approximation

case precisely of the required size. We checked whether the utilisation of the SOQN-BO is the same with that λ_{BO} (depending on N) for $N = 1, 2, \ldots, 50$. The results are presented in Fig. 19 and confirm that deviations are moderate: For $N \leq 20$ the results fit well, and even in the worst cases (for $N = 50$) the relative deviation is below 10%.

The deviation of the approximation of the average turnover times from the respective simulations are shown in Fig. 20 and are almost neglectable.

For approximations of the external queue the situation is different. In Fig. 21 we see that our method reproduces the behaviour of the system qualitatively pretty well: The shapes of the average waiting time curves for fixed utilisation are nearly the same. But the differences between the respective values of the simulated and approximated average waiting times for fixed N are in parts large, especially for high utilisations and high numbers of robots. (We presented only results for average waiting times up to 0.07h^{-1}. Behaviour of curves for larger delay is similar.)

An explanation for the deviations of the approximated external mean waiting times from the simulated values may be as follows. Consider the resource network (inner network + resource pool) as a black box server with complicated service time structure for an approximately Poissonian arrival stream (Fig. 2). From the Pollaćek-Khintchin formula for mean waiting times we know that even in a simple

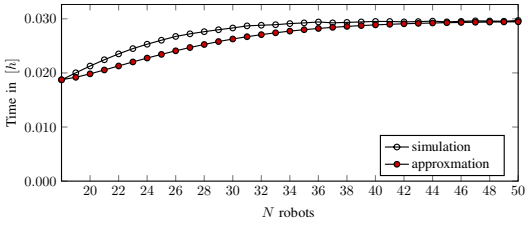

Fig. 18 Average waiting times for replenishment, simulation vs. approximation

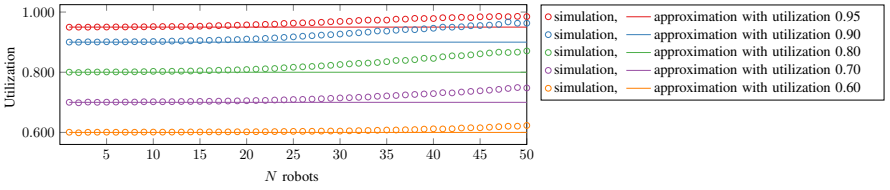

Fig. 19 Robots' utilisation, simulation vs. approximation

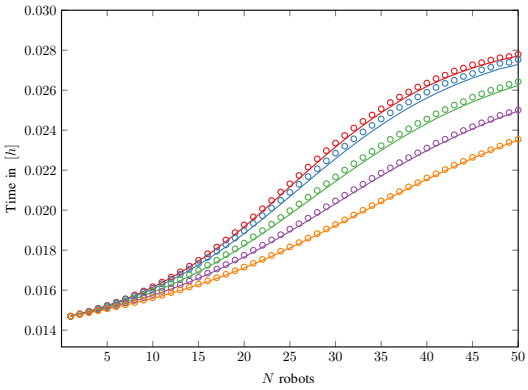

Fig. 20 Average turnover times, simulation vs. approximation

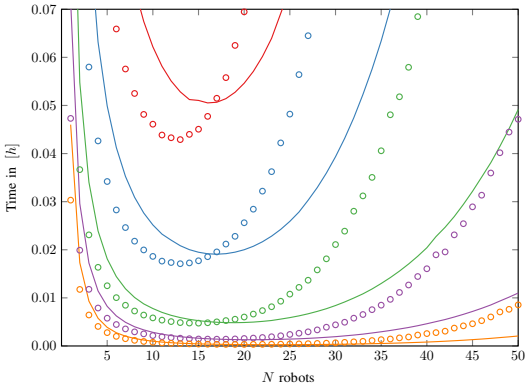

Fig. 21 Average waiting times in the external queue, simulation vs. approximation

$M/G/1/\infty$ queue (with state-independent service rate) the variability of the service time is crucial for the size of the mean waiting time at the system. We conjecture that our last approximation step which reduces the complex inner network to a single server queue with state dependent service rates does not encapsulate the variability of the passage time through the inner network sufficiently precise. We studied this problem theoretically in a simple model which fits into the class of systems described in Sect. 2. Surprisingly, we found all cases: Increasing, decreasing, and invariant variability of passage time variance in the reduction step, see the comment at the end of Appendix A.

Arrival streams We assumed so far that the arrival streams to the original SOQN and to the approximating system are Poissonian. If only mean interarrival time $\lambda^{-1} > 0$ is available, this is the classical assumption based on entropy arguments: Exponential distribution has maximal entropy among continuous distributions on $[0, \infty)$ with this mean. On the other side the deterministic interarrival variable with mean $\lambda^{-1} > 0$ has entropy 0. We performed experiments comparing SOQNs with internal structure as in the introductory example in the beginning of Sect. 5.4, now with deterministic and exponential interarrival times, comparing this with our approximation using Poissonian arrivals.

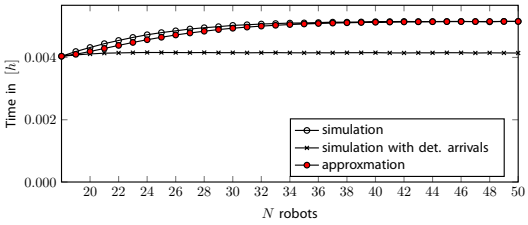

Fig. 22 Average waiting time for the picker. Simulation with Poisson and deterministic arrival times and approximation

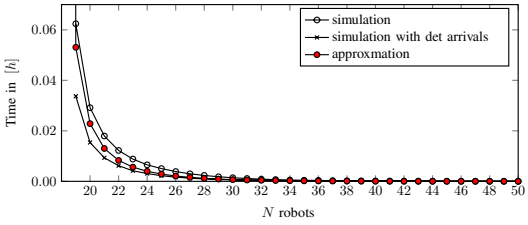

Fig. 23 Average waiting times in the external queue. Simulation with Poisson and deterministic arrivals vs. approximation

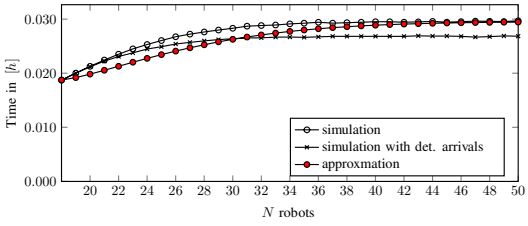

Fig. 24 Average waiting times for the replenisher. Simulation with Poisson and deterministic arrivals vs. approximation

As worst case we observed results for average waiting times at the picker, see Fig. 22, with largest difference between simulation of the SOQN-BO with deterministic interarrival times and approximation using Poisson arrivals. While up to 25 robots the deviation of the approximation is below $\approx 10\%$, for 50 robots the deviation of the approximation from the simulation result is $\approx 20\%$. We included in Fig. 22 simulation results for the SOQN-BO with Poisson arrivals which fit very well with the approximation.

Figure 23 shows average waiting times at the external queue similar to Fig. 13. The approximation provides results in the midst between the simulation results with Poisson and deterministic arrival streams. We conclude that approximation of the external queue works well in both cases.

This in-between-property holds for the average waiting times at the picker as well, see Fig. 22, but it is not a universal property as can be seen from Fig. 24. We

investigated waiting times at the replenisher. In this case for less than 32 robots the approximation under-estimates the average waiting time at the replenisher under deterministic arrivals, while for larger number of robots this average waiting time is over-estimated, although in the worst case for 50 robots for less than $\approx 10\,\%$, which seems to be satisfying.

6 Conclusion

Our contribution to modelling and performance evaluation of SOQNs is based on an interplay of exact procedures, partly originating from queueing network theory, and heuristic transformation of SOQNs with infinite external queue into SOQNs with finite external queue and vice-versa. Before realising this transformation we have investigated performance indices of the original SOQN to obtain exact stability conditions and some directly accessible mean values (throughputs) and steady-state probabilities. The advantage of the procedure is to obtain a closed form steady-state distribution for the external queue and the total population size in the inner network.

The subsequent application of our results to performance analysis of an RMFS validates the applicability of the obtained performance evaluation methods. An advantage of our procedure is the possibility to use in applications well-established performance algorithms from queueing network theory.

Proof of Proposition 6 and further details

Proof of Proposition 6 We show strict isotonicity of $\lambda_{\text{eff}}(\lambda_{\text{LC}})$ by induction in N. We write $\lambda_{\text{eff}}^{(N)}(\lambda_{\text{LC}})$ in a system with N resources. Recall from Eq. (22) for $L = 1, 2, \ldots, N$

$$C_{\text{LC}}(\bar{J}_0, L) = \sum_{n_0=0}^{L} \left(\frac{\eta_0}{\lambda_{\text{LC}}}\right)^{n_0} \sum_{\sum_{j\in\bar{J}} n_j = L-n_0} \prod_{j=1}^{J} \left(\prod_{i=1}^{n_j} \frac{\eta_j}{\nu_j(i)}\right).$$

We set $C_{\lambda_{\text{LC}}}(\bar{J}_0, L) := C_{\text{LC}}(\bar{J}_0, L)$ to indicate functional dependence on λ_{LC}. It holds

$$\begin{aligned}
\lambda_{\text{eff}}^{(N)}(\lambda_{\text{LC}}) &= \lambda_{\text{LC}} \cdot \left(1 - \frac{C^{\text{stb}}(\bar{J}, N)}{C_{\text{LC}}(\bar{J}_0, N)}\right) \\
&= \lambda_{\text{LC}} \cdot \left(1 - \frac{\sum_{\sum_{j\in\bar{J}} n_j = N} \prod_{j=1}^{J} \left(\prod_{i=1}^{n_j} \frac{\eta_j}{\nu_j(i)}\right)}{\sum_{\sum_{j\in\bar{J}_0} n_j = N} \left(\frac{\eta_0}{\lambda_{\text{LC}}}\right)^{n_0} \cdot \prod_{j=1}^{J} \left(\prod_{i=1}^{n_j} \frac{\eta_j}{\nu_j(i)}\right)}\right) \\
&= \eta_0 \cdot \frac{\sum_{n_0=1}^{N} \left(\frac{\eta_0}{\lambda_{\text{LC}}}\right)^{n_0-1} \sum_{\sum_{j\in\bar{J}} n_j = N-n_0} \prod_{j=1}^{J} \left(\prod_{i=1}^{n_j} \frac{\eta_j}{\nu_j(i)}\right)}{\sum_{n_0=0}^{N} \left(\frac{\eta_0}{\lambda_{\text{LC}}}\right)^{n_0} \sum_{\sum_{j\in\bar{J}} n_j = N-n_0} \prod_{j=1}^{J} \left(\prod_{i=1}^{n_j} \frac{\eta_j}{\nu_j(i)}\right)} \\
&= \eta_0 \cdot \frac{\sum_{n_0=0}^{N-1} \left(\frac{\eta_0}{\lambda_{\text{LC}}}\right)^{n_0} \sum_{\sum_{j\in\bar{J}} n_j = N-1-n_0} \prod_{j=1}^{J} \left(\prod_{i=1}^{n_j} \frac{\eta_j}{\nu_j(i)}\right)}{\sum_{n_0=0}^{N} \left(\frac{\eta_0}{\lambda_{\text{LC}}}\right)^{n_0} \sum_{\sum_{j\in\bar{J}} n_j = N-n_0} \prod_{j=1}^{J} \left(\prod_{i=1}^{n_j} \frac{\eta_j}{\nu_j(i)}\right)} \\
&= \eta_0 \cdot \frac{C_{\lambda_{\text{LC}}}(\bar{J}_0, N-1)}{C_{\lambda_{\text{LC}}}(\bar{J}_0, N)}.
\end{aligned}$$

We further define for

$$L = 1, 2, \ldots, N: \quad C(\bar{J}, L) := \sum_{\sum_{j\in\bar{J}} n_j = L} \prod_{j=1}^{J} \left(\prod_{i=1}^{n_j} \frac{\eta_j}{\nu_j(i)}\right)$$

and obtain

$$C_{\lambda_{\text{LC}}}(\bar{J}_0, L) = C(\bar{J}, L) + \frac{\eta_0}{\lambda_{\text{LC}}} \cdot C_{\lambda_{\text{LC}}}(\bar{J}_0, L-1), \quad L = 1, 2, \ldots, N. \tag{29}$$

We prove by induction

$$\lambda_{\text{eff}}^{(N)}(\lambda_{\text{LC}} + \varepsilon) > \lambda_{\text{eff}}^{(N)}(\lambda_{\text{LC}}) \qquad \forall \lambda_{\text{LC}} > 0,\ \varepsilon > 0,\ \forall N = 1, 2, \ldots$$

Base step: For $N = 1$:

$$\lambda_{\text{eff}}^{(1)}(\lambda_{\text{LC}} + \varepsilon) - \lambda_{\text{eff}}^{(1)}(\lambda_{\text{LC}}) = \frac{\eta_0}{\frac{\eta_0}{\lambda_{\text{LC}}+\varepsilon} + \sum_{j=1}^{J} \frac{\eta_j}{\nu_j(1)}} - \frac{\eta_0}{\frac{\eta_0}{\lambda_{\text{LC}}} + \sum_{j=1}^{J} \frac{\eta_j}{\nu_j(1)}} > 0.$$

Induction step: Assume the inequality holds for $1 \le L \le N-1$. Then for $L = N$ holds

$$\begin{aligned}
\frac{1}{\eta_0} \cdot \left(\lambda_{\text{eff}}^{(N)}(\lambda_{\text{LC}} + \varepsilon) - \lambda_{\text{eff}}^{(N)}(\lambda_{\text{LC}})\right) &= \frac{C_{\lambda_{\text{LC}}+\varepsilon}(\bar{J}_0, N-1)}{C_{\lambda_{\text{LC}}+\varepsilon}(\bar{J}_0, N)} - \frac{C_{\lambda_{\text{LC}}}(\bar{J}_0, N-1)}{C_{\lambda_{\text{LC}}}(\bar{J}_0, N)} \\
&= \frac{C_{\lambda_{\text{LC}}+\varepsilon}(\bar{J}_0, N-1) \cdot C_{\lambda_{\text{LC}}}(\bar{J}_0, N) - C_{\lambda_{\text{LC}}}(\bar{J}_0, N-1) \cdot C_{\lambda_{\text{LC}}+\varepsilon}(\bar{J}_0, N)}{C_{\lambda_{\text{LC}}+\varepsilon}(\bar{J}_0, N) \cdot C_{\lambda_{\text{LC}}}(\bar{J}_0, N)}.
\end{aligned}$$

Because the denominator is strictly positive, it suffices to show that the numerator is strictly positive. Using the induction assumption, we obtain from Eq. (29):

$$
\begin{aligned}
&C_{\lambda_{\mathrm{LC}}+\varepsilon}(\bar{J}_0, N-1)\cdot C_{\lambda_{\mathrm{LC}}}(\bar{J}_0, N) - C_{\lambda_{\mathrm{LC}}}(\bar{J}_0, N-1)\cdot C_{\lambda_{\mathrm{LC}}+\varepsilon}(\bar{J}_0, N)\\
&\quad=\left(C(\bar{J}, N-1) + \frac{\eta_0}{\lambda_{\mathrm{LC}}+\varepsilon}\cdot C_{\lambda_{\mathrm{LC}}+\varepsilon}(\bar{J}_0, N-2)\right)\\
&\qquad\cdot\left(C(\bar{J}, N) + \frac{\eta_0}{\lambda_{\mathrm{LC}}}\cdot C_{\lambda_{\mathrm{LC}}}(\bar{J}_0, N-1)\right)\\
&\qquad-\left(C(\bar{J}, N-1) + \frac{\eta_0}{\lambda_{\mathrm{LC}}}\cdot C_{\lambda_{\mathrm{LC}}}(\bar{J}_0, N-2)\right)\\
&\qquad\cdot\left(C(\bar{J}, N) + \frac{\eta_0}{\lambda_{\mathrm{LC}}+\varepsilon}\cdot C_{\lambda_{\mathrm{LC}}+\varepsilon}(\bar{J}_0, N-1)\right)\\
&\quad= C(\bar{J}, N-1)\cdot C(\bar{J}, N) + C(\bar{J}, N-1)\cdot\frac{\eta_0}{\lambda_{\mathrm{LC}}}\cdot C_{\lambda_{\mathrm{LC}}}(\bar{J}_0, N-1)\\
&\qquad+\frac{\eta_0}{\lambda_{\mathrm{LC}}+\varepsilon}\cdot C_{\lambda_{\mathrm{LC}}+\varepsilon}(\bar{J}_0, N-2)\cdot C(\bar{J}, N)\\
&\qquad+\frac{\eta_0}{\lambda_{\mathrm{LC}}+\varepsilon}\cdot C_{\lambda_{\mathrm{LC}}+\varepsilon}(\bar{J}_0, N-2)\cdot\frac{\eta_0}{\lambda_{\mathrm{LC}}}\cdot C_{\lambda_{\mathrm{LC}}}(\bar{J}_0, N-1)\\
&\qquad- C(\bar{J}, N-1)\cdot C(\bar{J}, N) - C(\bar{J}, N-1)\cdot\frac{\eta_0}{\lambda_{\mathrm{LC}}+\varepsilon}\cdot C_{\lambda_{\mathrm{LC}}+\varepsilon}(\bar{J}_0, N-1)\\
&\qquad-\frac{\eta_0}{\lambda_{\mathrm{LC}}}\cdot C_{\lambda_{\mathrm{LC}}}(\bar{J}_0, N-2)\cdot C(\bar{J}, N)\\
&\qquad-\frac{\eta_0}{\lambda_{\mathrm{LC}}}\cdot C_{\lambda_{\mathrm{LC}}}(\bar{J}_0, N-2)\cdot\frac{\eta_0}{\lambda_{\mathrm{LC}}+\varepsilon}\cdot C_{\lambda_{\mathrm{LC}}+\varepsilon}(\bar{J}_0, N-1)\\
&\quad=\underbrace{\frac{\eta_0}{\lambda_{\mathrm{LC}}+\varepsilon}\cdot\frac{\eta_0}{\lambda_{\mathrm{LC}}}\cdot\Big[C_{\lambda_{\mathrm{LC}}+\varepsilon}(\bar{J}_0, N-2)\cdot C_{\lambda_{\mathrm{LC}}}(\bar{J}_0, N-1)
-C_{\lambda_{\mathrm{LC}}}(\bar{J}_0, N-2)\cdot C_{\lambda_{\mathrm{LC}}+\varepsilon}(\bar{J}_0, N-1)\Big]}_{=:A_N}\\
&\qquad+\frac{\eta_0}{\lambda_{\mathrm{LC}}}\cdot C(\bar{J}, N-1)\cdot C_{\lambda_{\mathrm{LC}}}(\bar{J}_0, N-1) + \frac{\eta_0}{\lambda_{\mathrm{LC}}+\varepsilon}\cdot C_{\lambda_{\mathrm{LC}}+\varepsilon}(\bar{J}_0, N-2)\\
&\qquad\cdot C(\bar{J}, N)\\
&\qquad-\frac{\eta_0}{\lambda_{\mathrm{LC}}+\varepsilon}\cdot C(\bar{J}, N-1)\cdot C_{\lambda_{\mathrm{LC}}+\varepsilon}(\bar{J}_0, N-1) - \frac{\eta_0}{\lambda_{\mathrm{LC}}}\cdot C_{\lambda_{\mathrm{LC}}}(\bar{J}_0, N-2)\\
&\qquad\cdot C(\bar{J}, N)\\
&\quad= A_N + \frac{\eta_0}{\lambda_{\mathrm{LC}}}\cdot\Big[C(\bar{J}, N-1)\cdot C_{\lambda_{\mathrm{LC}}}(\bar{J}_0, N-1) - C_{\lambda_{\mathrm{LC}}}(\bar{J}_0, N-2)\cdot C(\bar{J}, N)\Big]\\
&\qquad+\frac{\eta_0}{\lambda_{\mathrm{LC}}+\varepsilon}\cdot\Big[C_{\lambda_{\mathrm{LC}}+\varepsilon}(\bar{J}_0, N-2)\cdot C(\bar{J}, N) - C(\bar{J}, N-1)\\
&\qquad\cdot C_{\lambda_{\mathrm{LC}}+\varepsilon}(\bar{J}_0, N-1)\Big].
\end{aligned}
$$

A_N is positive: $\frac{\eta_0}{\lambda_{\mathrm{LC}}+\varepsilon} \cdot \frac{\eta_0}{\lambda_{\mathrm{LC}}}$ multiplied by the numerator of $\lambda_{\mathrm{eff}}^{(N-1)}(\lambda_{\mathrm{LC}}+\varepsilon) - \lambda_{\mathrm{eff}}^{(N-1)}(\lambda_{\mathrm{LC}})$. Therefore, $A_N > 0$ by induction assumption. Thus, it suffices to prove

$$\begin{aligned}
&\frac{\eta_0}{\lambda_{\mathrm{LC}}} \cdot \left[C(\bar{J}, N-1) \cdot C_{\lambda_{\mathrm{LC}}}(\bar{J}_0, N-1) - C_{\lambda_{\mathrm{LC}}}(\bar{J}_0, N-2) \cdot C(\bar{J}, N) \right] \\
&\quad + \frac{\eta_0}{\lambda_{\mathrm{LC}}+\varepsilon} \cdot \left[C_{\lambda_{\mathrm{LC}}+\varepsilon}(\bar{J}_0, N-2) \cdot C(\bar{J}, N) - C(\bar{J}, N-1) \right. \\
&\quad \left. \cdot C_{\lambda_{\mathrm{LC}}+\varepsilon}(\bar{J}_0, N-1) \right] \geq 0,
\end{aligned}$$

which is equivalent to

$$\begin{aligned}
&\frac{\eta_0}{\lambda_{\mathrm{LC}}} \cdot \left[\frac{C(\bar{J}, N-1)}{C(\bar{J}, N)} - \frac{C_{\lambda_{\mathrm{LC}}}(\bar{J}_0, N-2)}{C_{\lambda_{\mathrm{LC}}}(\bar{J}_0, N-1)} \right] \cdot C(\bar{J}, N) \cdot C_{\lambda_{\mathrm{LC}}}(\bar{J}_0, N-1) \\
&\quad - \frac{\eta_0}{\lambda_{\mathrm{LC}}+\varepsilon} \cdot \left[\frac{C(\bar{J}, N-1)}{C(\bar{J}, N)} - \frac{C_{\lambda_{\mathrm{LC}}+\varepsilon}(\bar{J}_0, N-2)}{C_{\lambda_{\mathrm{LC}}+\varepsilon}(\bar{J}_0, N-1)} \right] \cdot C(\bar{J}, N) \\
&\quad \cdot C_{\lambda_{\mathrm{LC}}+\varepsilon}(\bar{J}_0, N-1) \\
&= \Bigg\{ \underbrace{\left[\frac{C(\bar{J}, N-1)}{C(\bar{J}, N)} - \frac{C_{\lambda_{\mathrm{LC}}}(\bar{J}_0, N-2)}{C_{\lambda_{\mathrm{LC}}}(\bar{J}_0, N-1)} \right]}_{=:B} \cdot \underbrace{\frac{\eta_0}{\lambda_{\mathrm{LC}}} \cdot C_{\lambda_{\mathrm{LC}}}(\bar{J}_0, N-1)}_{=:D} \\
&\quad - \underbrace{\left[\frac{C(\bar{J}, N-1)}{C(\bar{J}, N)} - \frac{C_{\lambda_{\mathrm{LC}}+\varepsilon}(\bar{J}_0, N-2)}{C_{\lambda_{\mathrm{LC}}+\varepsilon}(\bar{J}_0, N-1)} \right]}_{=:C} \cdot \underbrace{\frac{\eta_0}{\lambda_{\mathrm{LC}}+\varepsilon} \cdot C_{\lambda_{\mathrm{LC}}+\varepsilon}(\bar{J}_0, N-1)}_{=:E} \Bigg\} \\
&\quad \cdot C(\bar{J}, N) \geq 0.
\end{aligned}$$

(i) We show that $B \geq 0, C \geq 0$. Because increasing population increases throughput (Van der Wal 1989)

$$\frac{C(\bar{J}, N-1)}{C(\bar{J}, N)} \geq \frac{C(\bar{J}, N-2)}{C(\bar{J}, N-1)}.$$

Next, we show:

$$\frac{C(\bar{J}, N-2)}{C(\bar{J}, N-1)} \geq \frac{C_{\lambda_{\mathrm{LC}}}(\bar{J}_0, N-2)}{C_{\lambda_{\mathrm{LC}}}(\bar{J}_0, N-1)}.$$

We consider the right-hand side as throughput of a cyclic Gordon–Newell network with node set $\bar{J}_0 := \{0, 1, \ldots, J\}$, service rates $\mu_j(n) := \frac{\nu_j(i)}{\eta_j}$, $j = 1, \ldots, J$, $n = 0, 1, \ldots, N$ and $\mu_0(n) := \frac{\lambda_{\mathrm{LC}}}{\eta_0}$ and solution $\underbrace{(1, 1, \ldots, 1)}_{J+1\text{-times}}$ of the routing matrix for

the cycle. $\frac{C(\bar{J},N-2)}{C(\bar{J},N-1)}$ and $\frac{C_{\lambda_{\mathrm{LC}}}(\bar{J},N-2)}{C_{\lambda_{\mathrm{LC}}}(\bar{J},N-1)}$ as given in our derivations are the (average) throughput of a cycle following (Daduna et al. 2008, Definition 2.6).

The left-hand side is the throughput of a cyclic Gordon–Newell network which is obtained from the first cycle by deleting node 0 and skipping the gap by cycling customers.

Lemma 2.8 in Daduna et al. (2008) states that deleting a node of a cycle with skipping the gap increases the throughput. Consequently, $B \geq 0$ in a two-step conclusion, and similarly $C \geq 0$.

(ii) From the definitions of D and E it follows by direct comparison $D > E$.

(iii) The proof will be finished if we can show $B \geq C$. It is sufficient to prove with Shanthikumar and Yao (1986)

$$\frac{C_{\lambda_{\mathrm{LC}}}(\bar{J}_0,N-2)}{C_{\lambda_{\mathrm{LC}}}(\bar{J}_0,N-1)} \leq \frac{C_{\lambda_{\mathrm{LC}}+\varepsilon}(\bar{J}_0,N-2)}{C_{\lambda_{\mathrm{LC}}+\varepsilon}(\bar{J}_0,N-1)}. \tag{30}$$

The left-hand side is the throughput of the Gordon–Newell network according to Definition 2.2 in Shanthikumar and Yao (1986). The right-hand side is the throughput of the Gordon–Newell network with service rate at node 0 increased to $\lambda_{\mathrm{LC}}+\varepsilon$. Corollary 3.1(i) in Shanthikumar and Yao (1986) states

$$\eta_0 \cdot \frac{C_{\lambda_{\mathrm{LC}}}(\bar{J}_0,N-2)}{C_{\lambda_{\mathrm{LC}}}(\bar{J}_0,N-1)} \leq \eta_0 \cdot \frac{C_{\lambda_{\mathrm{LC}}+\varepsilon}(\bar{J}_0,N-2)}{C_{\lambda_{\mathrm{LC}}+\varepsilon}(\bar{J}_0,N-1)},$$

which verifies (30). □

Remark 3 For λ_{LC} with $\lambda_{\mathrm{eff}}(\lambda_{\mathrm{LC}}) = \lambda_{\mathrm{BO}} \in \left(0, \lambda_{\mathrm{BO,max}}\right) = \left(0, \eta_0 \cdot \frac{b(1)}{b(0)}\right)$ it holds

$$\lambda_{\mathrm{LC}} = \begin{cases} \frac{\eta_0 \cdot \lambda_{\mathrm{BO}}}{\eta_0 - \lambda_{\mathrm{BO}} \cdot b(0)}, & N = 1, \\ -\frac{\eta_0}{2 \cdot \left(\eta_0 \cdot b(1) - \lambda_{\mathrm{BO}} \cdot b(0)\right)} \\ \cdot \left(\eta_0 - \lambda_{\mathrm{BO}} \cdot b(1) - \sqrt{\left(\eta_0 + \lambda_{\mathrm{BO}} \cdot b(1)\right)^2 - 4 \cdot \lambda_{\mathrm{BO}}^2 \cdot b(0)}\right), & N = 2. \end{cases}$$

Proof Due to Theorem 2 we have for $\lambda_{\mathrm{LC}} \in (0, \infty)$

$$\begin{aligned} \lambda_{\mathrm{eff}}(\lambda_{\mathrm{LC}}) &= \lambda_{\mathrm{LC}} \cdot \left(1 - \frac{b(0)}{\sum_{n=0}^{N} \left(\frac{\lambda_{\mathrm{LC}}}{\eta_0}\right)^n \cdot b(n)}\right) \\ &= \lambda_{\mathrm{LC}} \cdot \left(\frac{\sum_{n=1}^{N} b(n) \cdot \lambda_{\mathrm{LC}} \cdot \left(\frac{\lambda_{\mathrm{LC}}}{\eta_0}\right)^{N-n}}{\sum_{n=0}^{N} b(n) \cdot \left(\frac{\lambda_{\mathrm{LC}}}{\eta_0}\right)^{N-n}}\right) \end{aligned}$$

and

$$b(N) = 1, \quad b(N-1) = \sum_{j=1}^{J} \frac{\eta_j}{\nu_j(1)},$$
$$b(N-2) = \sum_{j=1}^{J} \frac{\eta_j}{\nu_j(1) \cdot \nu_j(2)} \quad + \sum_{j=1}^{J-1} \sum_{k=j+1}^{J} \frac{\eta_j \cdot \eta_k}{\nu_j(1) \cdot \nu_k(1)}.$$

Let $\lambda_{\mathrm{BO}} \in \left(0, \eta_0 \cdot \frac{b(1)}{b(0)}\right)$. First, we note that

$$\eta_0 \cdot b(1) - \lambda_{\mathrm{BO}} \cdot b(0) > \eta_0 \cdot b(1) - \eta_0 \cdot \frac{b(1)}{b(0)} \cdot b(0) = 0. \tag{31}$$

$\lambda_{\mathrm{eff}}(\lambda_{\mathrm{LC}}) = \lambda_{\mathrm{BO}}$ is equivalent to

$$\frac{\sum_{n=1}^{N} \left(\frac{\lambda_{\mathrm{LC}}}{\eta_0}\right)^{N+1-n} \cdot \eta_0 \cdot b(n)}{\sum_{n=0}^{N} \left(\frac{\lambda_{\mathrm{LC}}}{\eta_0}\right)^{N-n} \cdot b(n)} = \lambda_{\mathrm{BO}},$$

which is equivalent to:

$$\sum_{n=1}^{N} \left(\eta_0 \cdot b(n) - \lambda_{\mathrm{BO}} \cdot b(n-1)\right) \cdot \left(\frac{\lambda_{\mathrm{LC}}}{\eta_0}\right)^{N+1-n} - \lambda_{\mathrm{BO}} \cdot b(N) = 0. \tag{32}$$

If $N = 1$, then Eq. (32) is equivalent to

$$\lambda_{\mathrm{LC}} = \frac{\eta_0 \cdot \lambda_{\mathrm{BO}} \cdot b(1)}{\eta_0 \cdot b(1) - \lambda_{\mathrm{BO}} \cdot b(0)} = \frac{\eta_0 \cdot \lambda_{\mathrm{BO}}}{\eta_0 - \lambda_{\mathrm{BO}} \cdot b(0)}.$$

If $N = 2$, then Eq. (32) is equivalent to

$$\left(\eta_0 \cdot b(1) - \lambda_{\mathrm{BO}} \cdot b(0)\right) \cdot \left(\frac{\lambda_{\mathrm{LC}}}{\eta_0}\right)^2 + \left(\eta_0 \cdot b(2) - \lambda_{\mathrm{BO}} \cdot b(1)\right) \cdot \left(\frac{\lambda_{\mathrm{LC}}}{\eta_0}\right) - b(2) \cdot \lambda_{\mathrm{BO}} = 0.$$

Since the discriminant fulfils

$$\left(\eta_0 \cdot b(2) - \lambda_{\mathrm{BO}} \cdot b(1)\right)^2 + 4 \cdot \left(\eta_0 \cdot b(1) - \lambda_{\mathrm{BO}} \cdot b(0)\right) \cdot b(2) \cdot \lambda_{\mathrm{BO}} > 0$$

by Eq. (31), it follows that

$$\frac{\lambda_{\mathrm{LC}}}{\eta_0} = -\frac{1}{2 \cdot \left(\eta_0 \cdot b(1) - \lambda_{\mathrm{BO}} \cdot b(0)\right)} \cdot \Big(\eta_0 \cdot b(2) - \lambda_{\mathrm{BO}} \cdot b(1) \pm \sqrt{\left(\eta_0 \cdot b(2) - \lambda_{\mathrm{BO}} \cdot b(1)\right)^2 + 4 \cdot \left(\eta_0 \cdot b(1) - \lambda_{\mathrm{BO}} \cdot b(0)\right) \cdot b(2) \cdot \lambda_{\mathrm{BO}}}\Big).$$

Due to Eq. (31), the solution λ_{LC} is positive only if

$$\eta_0 \cdot b(2) - \lambda_{\mathrm{BO}} \cdot b(1) \pm \sqrt{\left(\eta_0 \cdot b(2) - \lambda_{\mathrm{BO}} \cdot b(1)\right)^2 + 4 \cdot \left(\eta_0 \cdot b(1) - \lambda_{\mathrm{BO}} \cdot b(0)\right) \cdot b(2) \cdot \lambda_{\mathrm{BO}}} < 0.$$

Let $c := \eta_0 \cdot b(2) - \lambda_{\mathrm{BO}} \cdot b(1)$ and $d := 4 \cdot \left(\eta_0 \cdot b(1) - \lambda_{\mathrm{BO}} \cdot b(0)\right) \cdot b(2) \cdot \lambda_{\mathrm{BO}}$, which implies $d > 0$. If $c \geq 0$, then $c + \sqrt{c^2 + d} > 0$. If $c < 0$, then $c + \sqrt{c^2 + d} > c + \sqrt{c^2} = c + |c| = c - c = 0$. Hence, the only positive solution (existence guaranteed by Theorem 2) is

$$\begin{aligned}\lambda_{\mathrm{LC}} &= -\frac{\eta_0}{2 \cdot \left(\eta_0 \cdot b(1) - \lambda_{\mathrm{BO}} \cdot b(0)\right)} \cdot \Big(\eta_0 \cdot b(2) - \lambda_{\mathrm{BO}} \cdot b(1) \\ &\quad - \sqrt{\left(\eta_0 \cdot b(2) - \lambda_{\mathrm{BO}} \cdot b(1)\right)^2 + 4 \cdot \left(\eta_0 \cdot b(1) - \lambda_{\mathrm{BO}} \cdot b(0)\right) \cdot b(2) \cdot \lambda_{\mathrm{BO}}}\Big) \\ &= \frac{-\eta_0}{2 \cdot \left(\eta_0 \cdot b(1) - \lambda_{\mathrm{BO}} \cdot b(0)\right)} \\ &\quad \cdot \left(\eta_0 - \lambda_{\mathrm{BO}} \cdot b(1) - \sqrt{\left(\eta_0 + \lambda_{\mathrm{BO}} \cdot b(1)\right)^2 - 4 \cdot \lambda_{\mathrm{BO}}^2 \cdot b(0)}\right).\end{aligned}$$

□

Transformation of passage time through the inner network. We consider the SOQN-BO from Sect. 2 and its approximation by an SOQN-LC in Sect. 3. Within the SOQN-LC the resource network behaves stochastically as a Gordon–Newell network with $N \geq 1$ customers. We take the inner network as a complex black-box server, which is able to serve several customers in parallel. It is well known that the distribution of customers' passage time through that black box is in general not known. Only mean values are accessible, although complicated as well. At a first glance one expects to have passage times, say Y, with high variability, measured e.g. by the squared coefficient of variation $C^2(Y) = Var(Y)/E^2(Y)$. (Note, even Var(Y) is in general not known.) The problem is to evaluate the effect of the transformation which reduces the inner network to a single substitute station with state dependent service rates exploiting Norton's Theorem. We have not been able to resolve the problem of transforming the variability in general, but we can show by examples why there will be no general answer to that question.

We consider three cases of SOQN-BO with Poisson-λ arrival stream and only one resource. Then at most one customer can be served by the inner network, and the passage time through the inner network has a Phase-type distribution (PH). So the system behaves like a single server $M/PH/1/\infty$ queue and for simple PH-distributions we can compute e.g. the mean waiting time at the external queue explicitly. We fix in advance a parameter $1/\mu > 0$.

(i) The inner system consists of only one single server station with service rate μ. Then the reduced system is trivially the same and the coefficient of variation of the passage time through the inner network and the substitute is 1.

(ii) The inner system consists of $k > 1$ single server stations in line (tandem queue) each with service rate $k \cdot \mu$. Then the passage time through the inner network is a k-stage Erlang distribution with mean $1/\mu$ and coefficient of variation $1/k < 1$. The reduced system according to Norton's Theorem is a single server station with service rate μ and coefficient of variation 1.

(iii) Let $C^2 > 1$. The inner system consists of two parallel single server stations with service rates μ_1 and μ_2. A customer traversing the inner network chooses station 1 with probability $\alpha \in (0, 1)$, station 2 with probability $1 - \alpha$. We take $\mu_1 := 2\alpha\mu$ and $\mu_2 := 2(1 - \alpha)\mu$, and $\alpha := (1/2) \cdot [1 - \sqrt{(C^2 - 1)/(C^2 + 1)}]$. This yields a hyperexponential distribution with mean $1/\mu$ and coefficient of variation $C^2 > 1$. The reduced system according to Norton's Theorem is a single server station with service rate μ and coefficient of variation 1.

The conclusion is that the variability of passage time through the inner network under the transformation can be (i) maintained, (ii) decreased, (iii) increased.

Acknowledgements Ruslan Krenzler and Sonja Otten are funded by the industrial project "Robotic Mobile Fulfillment System", which is financially supported by Ecopti GmbH (Paderborn, Germany) and Beijing Hanning Tech Co., Ltd. (Beijing, China).

Funding Open Access funding enabled and organized by Projekt DEAL.

Data availability The datasets generated during the current study are available in the TORE repository, https://doi.org/10.15480/336.3943.

References

Avi-Itzhak B, Heyman D (1973) Approximate queuing models for multiprogramming computer systems. Oper Res 21(6):1212–1230. https://doi.org/10.1287/opre.21.6.1212

Azadeh K, de Koster R, Roy D (2017a). Robotized and automated warehouse systems: Review and recent developments. https://doi.org/10.2139/ssrn.2977779

Azadeh K, de Koster R, Roy D (2017b) Robotized warehouse systems: Developments and research opportunities. Research paper (no. ERS-2017-009-LIS), ERIM Report Series Research in Management

Bijvank M, Vis F (2011) Lost-sales inventory theory: a review. Eur J Oper Res 215:1–13. https://doi.org/10.1016/j.ejor.2011.02.004

Bolch G, Greiner S, de Meer H, Trivedi K (1998) Queueing Networks and Markov Chains: Modeling and Performance Evaluation with Computer Science Applications. Wiley-Interscience, New York. https://doi.org/10.1002/0471791571

Boysen N, de Koster R, Weidinger F (2019) Warehousing in the e-commerce area: a survey. Eur J Oper Res 177:396–411. https://doi.org/10.1016/j.ejor.2018.08.023

Buitenhek R, van Houtum G, Zijm H (2000) AMVA-based solution procedures for open queueing networks with population constraints. Ann Oper Res 93:15–40. https://doi.org/10.1023/A:1018967622069

Canadilla P (2017) R package queueing. Analysis of queueing networks and models, version 0.2.11

Chandy K, Herzog U, Woo L (1975) Parametric analysis of queueing networks. IBM J Res Develop 19:43–49. https://doi.org/10.1147/rd.191.0036

Chen H, Yao D (2001) Fundamentals of Queueing Networks: Performance, Asymptotics, and Optimization. Applications of mathematics : stochastic modelling and applied probability, Springer, New York,. https://doi.org/10.1007/978-1-4757-5301-1

Daduna H, Pestien V, Ramakrishnan S (2008) Throughput limits from the asymptotic profile of cyclic networks with state-dependent service rates. Queueing Syst 58(3):191. https://doi.org/10.1007/s11134-008-9067-8

Dallery Y (1990) Approximate analysis of general open queuing networks with restricted capacity. Perform Eval 11(3):209–222. https://doi.org/10.1016/0166-5316(90)90013-9

Ekren B, Akpunar A (2021) An open queuing network-based tool for performance estimations in a shuttle-based storage and retrieval system. Appl Math Modell 89:1678–1695. https://doi.org/10.1016/j.apm.2020.07.055

Ekren B, Heragu S, Krishnamurthy A, Malmborg C (2013) An approximate solution for semi-open queueing network model of an autonomous vehicle storage and retrieval system. IEEE Trans Autom Sci Eng 10(1):205–215. https://doi.org/10.1109/TASE.2012.2200676

Ekren B, Heragu S, Krishnamurthy A, Malmborg C (2014) Matrix-geometric solution for semi-open queuing network model of autonomous vehicle storage and retrieval system. Comput Ind Eng 68:78–86. https://doi.org/10.1016/j.cie.2013.12.002

Jia J, Heragu S (2009) Solving semi-open queuing networks. Oper Res 57(2):391–401. https://doi.org/10.1287/opre.1080.0627

Kim J, Dudin A, Dudin S, Kim C (2018) Analysis of a semi-open queueing network with Markovian arrival process. Perform Eval 120:1–19. https://doi.org/10.1016/j.peva.2017.12.005

Krenzler R (2016) Queueing systems in a random environment. PhD thesis, University of Hamburg, Department of Mathematics

Krenzler R, Daduna H, Otten S (2016) Jackson networks in nonautonomous random environments. Adv Appl Probab 48(2):315–331. https://doi.org/10.1017/apr.2016.2

Krishnamoorthy A, Lakshmy B, Manikandan R (2011) A survey on inventory models with positive service time. OPSEARCH 48(2):153–169. https://doi.org/10.1007/s12597-010-0032-z

Lamballais T, Roy D, de Koster R (2017) Estimating performance in a robotic mobile fulfillment system. Eur J Oper Res 256:976–990

Lamballais T, Merschformann M, Roy D, Suhl L (2019) Dynamic policies for resource reallocation in a robotic mobile fulfillment system with time-varying demand. In: Controlling Robotic Mobile Fulfillment Systems and further topics in decision support, PhD thesis, University of Paderborn, chap 5, pp 129–175, https://doi.org/10.17619/UNIPB/1-695

Latouche G (2011) Level-independent quasi-birth-and-death processes. In: Cochran J, Cox L, Keskinocak P, Kharoufeh J, Smith J (eds) Wiley encyclopedia of operations research and management science. American Cancer Society, Atlanta. https://doi.org/10.1002/9780470400531.eorms0461

Latouche G, Ramaswami V (1993) A logarithmic reduction algorithm for quasi-birth-death processes. J Appl Probab 30:650–674. https://doi.org/10.2307/3214773

Latouche G, Ramaswami V (1999) Introduction to matrix analytic methods in stochastic modeling. Soc Ind Appl Math. https://doi.org/10.1137/1.9780898719734

Lavenberg SS (1978) Stability and maximum departure rate of certain open queuing networks having finite capacity constraints (stma v21 1124). RAIRO: Informatique RAIRO: Computer Science 12(4):353–370

Little J, Graves S (2008) Little's Law. In: Chhajed D, Lowe TJ (eds) Building Intuition: Insights From Basic Operations Management Models and Principles, Springer US, Boston, MA, pp 81–100, https://doi.org/10.1007/978-0-387-73699-0_5

Merschformann M, Lamballais T, de Koster R, Suhl L (2019) Decision rules for robotic mobile fulfillment systems. Oper Res Perspect 6:100128. https://doi.org/10.1016/j.orp.2019.100128

Otten S (2018) Integrated models for performance analysis and optimization of queueing-inventory-systems in logistic networks. PhD thesis, University of Hamburg, Department of Mathematics

Roy D (2016) Semi-open queuing networks: a review of stochastic models, solution methods and new research areas. Int J Product Res 54(6):1735–1752. https://doi.org/10.1080/00207543.2015.1056316

Schwarz M, Sauer C, Daduna H, Kulik R, Szekli R (2006) M/M/1 queueing systems with inventory. Queueing Syst Theory Appl 54:55–78. https://doi.org/10.1007/s11134-006-8710-5

Shanthikumar J, Yao D (1986) The preservation of likelihood ratio ordering under convolution. Stoch Processes Appl 23(2):259–267. https://doi.org/10.1016/0304-4149(86)90039-6

van der Wal J (1989) Monotonicity of the throughput of a closed exponential queueing network in the number of jobs. Operations-Research-Spektrum 11(2):97–100. https://doi.org/10.1007/BF01746005

Xie L, Thieme N, Krenzler R, Li H (2021) Introducing split orders and optimizing operational policies in robotic mobile fulfillment systems. Eur J Oper Res 288(1):80–97. https://doi.org/10.1016/j.ejor.2020.05.032

Yuan Z, Gong Y (2017) Bot-in-time delivery for robotic mobile fulfillment systems. IEEE Trans Eng Manage 64(1):83–93. https://doi.org/10.1109/TEM.2016.2634540

Zou B, Xu X, Gong Y, de Koster R (2018) Evaluating battery charging and swapping strategies in a robotic mobile fulfillment system. Eur J Oper Res 267(2):733–753. https://doi.org/10.1016/j.ejor.2017.12.008