A family of multiclass LCFS networks with order-dependent product-form solution

Giuliano Casale[1*]

[1*]Department of Computing, Imperial College London, 180 Queen's Gate, London, SW7 2AZ, England, United Kingdom.

Corresponding author(s). E-mail(s): g.casale@imperial.ac.uk;

Abstract

We study two-station closed queueing network models where jobs cyclically visit a non-preemptive last-come first-serve (LCFS-NP) station and a last-come first-serve preemptive-resume (LCFS-PR) station. Jobs belong to multiple classes and receive at both stations exponential service times with arbitrary means. Even though multiclass non-preemptive LCFS-NP stations are not quasi-reversible, we show that the considered models still admit a product-form solution. A key feature of the new product-form is to be order-dependent, as it includes factors that relate the mean service times at the LCFS-NP queue with the positions occupied by the jobs at both the LCFS-NP station and the LCFS-PR station. To account for job positions, we propose a strategy to compute the normalizing constant of the state probabilities using permanents which, for a fixed number of classes, solves the model exactly in polynomial time as the total number of jobs grows. A mean-value analysis algorithm is also derived, using recursion on networks where the job holding the last position at the LCFS-PR queue is removed from the model.

Keywords: Queueing network, multiclass, product-form, last-come first-serve, permanent.

1 Introduction

Since the seminal works by Baskett, Chandy, Muntz, Palacios (BCMP) [1] and Kelly [2], multiclass product-form queueing networks have found numerous applications in computer science, telecommunications, logistics, and several other domains [3–6]. However, limited progress has been made towards improving the understanding of queueing networks that include non-preemptive last-come first-serve (LCFS-NP) queues, which under multiclass workloads do not satisfy existing product-form assumptions [1, 7].

In this paper, we find that two-station closed networks where jobs are restricted to cycle between a multiclass LCFS-NP station and a last-come first-serve preemptive-resume (LCFS-PR) station admit a product-form solution that significantly differs in structure from the ordinary BCMP product-form. The new product-form expression is *order-dependent*, i.e., states that only differ for job order do *not* have in general the same probability. Another property of the new product-form is that the mean service times of the LCFS-NP station also appear in the factor describing the job positions at the LCFS-PR station, highlighting the complex interplay between these two scheduling disciplines.

To characterize this class of models, we discuss in depth throughout the paper the CTMC of small models with two and three circulating jobs. We show in particular the presence of certain cycles that describe residence times of jobs scheduled to execute last at the LCFS-PR station. We also find that the probability distribution on such cycles and their satellite states is proportional to the equilibrium distribution of the model obtained after removing the resident job at the LCFS-PR station, enabling a recursive evaluation.

Using these observations, we develop computational techniques for the normalizing constant of state probabilities. To cope with order-dependence, we represent the normalizing constant as a sum of permutations of the job positions, which can be expressed using permanents. Although computing the permanent is in general #P-hard [8], the specific representation we propose benefits from computational savings when job populations grow under a fixed number of classes, achieving a computational complexity for an exact solution similar to the one of the multiclass convolution algorithm for BCMP networks [9]. Moreover, we show that computational algorithms for the normalizing constant of BCMP networks can also be expressed using permanents, enabling a similar structure in the computational algorithms of the two families of models. This is further leveraged in the paper to develop a mean-value analysis method for LCFS networks.

In summary, the paper provides insights on multiclass LCFS models, showing how the order of jobs at each station plays a role in determining the analytical expression of the product-form solutions. The proposed product-form also appears to be fairly different from common classes of product-forms, such as those arising from time-reversibility, ρ-reversibility, quasi-reversibility, and order-independence [7, 10, 11].

The remainder of the paper is organized as follows. Section 2 gives related work, followed by the model and running case definitions in Section 3. Section 4 presents the novel product-form solution. Section 5 develops computational techniques for performance metrics based on normalizing constants. Methods for mean-value analysis are introduced in Section 6. Section 7 extends our results to models with multiple jobs per class. Extensions of the product-form solution are discussed in Section 8. Numerical examples to understand the qualitative behavior of these models are given in Section 9. Lastly, Section 10 gives closing remarks and outlines possible future research directions.

2 Preliminaries

2.1 BCMP Theorem

Classic product-form queueing networks, which we refer to as BCMP networks, allow composition in arbitrary topologies of quasi-reversible nodes [1, 7]. These models support the inclusion in the network of LCFS-PR, processor sharing (PS), infinite server (IS), and first-come first-serve (FCFS) queues. Except for FCFS scheduling, which requires exponential service times with identical means for all classes, the other queues can feature service distributions with rational Laplace transform, such as Coxian distributions.

The most relevant case for the present work are BCMP models that include M queueing stations and R closed job classes. We do not consider throughout IS stations. With this assumption, the stationary probability of state $\boldsymbol{n} = (n_1 \dots, n_M)$, where $\boldsymbol{n}_i = (n_{i,1}, \dots, n_{i,R})$ is the state of station i and $n_{i,r}$ is the total number of class-r jobs therein, is given by

$$\pi(\boldsymbol{n}) = \frac{1}{G} \prod_{i=1}^{M} \frac{n_i!}{n_{i,1}! \cdots n_{i,R}!} \prod_{r=1}^{R} \theta_{i,r}^{n_{i,r}} \tag{1}$$

where the mean service demand $\theta_{i,r}$ is the product of the mean service time of class-r jobs at station i with their mean visit ratio to the station, and G is a normalizing constant [1]. If we wish to further track the location of individual jobs, it is sufficient to segregate them into different classes, each having a unit population, so that the last expression reduces to the simpler formula

$$\pi(\boldsymbol{n}) = \frac{1}{S} \prod_{i=1}^{M} n_i! \prod_{r=1}^{R} \theta_{i,r}^{n_{i,r}} \tag{2}$$

where now $R = N$, in which N is the total number of jobs, S is a normalizing constant for the model with segregated jobs, and $\sum_i n_{i,r} = 1$. If we additionally wish to track the order of jobs at each station, then the last expression further simplifies to

$$\pi(\boldsymbol{s}) = \frac{1}{S} \prod_{i=1}^{M} \prod_{r \in \boldsymbol{s}_i} \theta_{i,r} \tag{3}$$

where now state $\boldsymbol{s} = (\boldsymbol{s}_1, \dots, \boldsymbol{s}_M)$ belongs to an expanded state space $\mathcal{S}$ in which the state $\boldsymbol{s}_i$ of station i is defined differently according to the chosen scheduling discipline; explicit formulas are given in [1]. The normalizing constants in the above BCMP expressions have in general different values, but using their integral representations [12] they can be shown to be related by

$$S = G \prod_{r=1}^{R} N_r! \tag{4}$$

The LCFS-NP policy does not fit instead in general the assumptions of the BCMP theorem in [1], except in the special case of exponential service times and $R = 1$ class. In this case, since we do not explicitly track the identity of individual jobs and due to the memoryless property of exponential distributions, LCFS-NP and LCFS-PR become stochastically indistinguishable.

2.2 Related work on product-form extensions

Research carried out after the BCMP theorem has shown that service in random order (SIRO) and last-batch processor sharing (LBPS) stations may also be included in a BCMP network retaining a product-form solution [13, 14]. Additional types of stations that are composable within product-form networks include order-independent (OI) queues [11], multiserver centers with concurrent classes of customers (MSCCC) [15], and multiservers with hierarchical concurrency constraints (MSHCC). Such stations all require specialized forms for the product-form factors that appear in (1). Additional extensions of product-form queueing models include solutions for blocking [16], server compatibilities [17], resets [18], pass-and-swap queues [19], and signals [20], among others. Proofs of advanced mechanisms of this kind can be obtained through various analytical balances, reversed processes, and other specialized tools of product-form theory [21–25].

Queueing models with LCFS-PR and non-preemptive LCFS-NP have also been considered in other settings. Due to the tree-structure of the Markov chains that arises when these queues are considered in isolation, matrix product-form solutions have been found [26, 27]. The work in [28] also finds a solution for LCFS-PR networks with dependent service times, but this is not in a product-form. A characterization of the LCFS-PR queue with marked input point process obtained using Palm calculus is given in [29]. An application of LCFS-PR to congestion control in open queueing networks is developed in [30]. Lastly, in [31] LCFS-PR is shown to connect to the theory of product-form generalized stochastic Petri nets. To the best of our knowledge, no prior work has examined two-station networks with non-preemptive LCFS-NP and LCFS-PR stations.

3 Two-station LCFS networks

3.1 Definitions

In the queueing network model under study, jobs visit an LCFS-NP station and an LCFS-PR station, as shown in Figure 1. Throughout, we refer to these two queues as the non-preemptive (NP) station and the preemptive (PR) station. There is a single server in each queue that processes jobs without idle times whenever there is unfinished work at the station. Jobs belong to R service classes. Initially, we assume classes to include a population of a single job, an assumption relaxed later in Section 7. Under this assumption, the total number of jobs N is equal to the total number of classes R. However, if two classes r and $s \neq r$ have identical rates at both stations, then their jobs are stochastically indistinguishable and they may be seen as effectively representing a single class consisting of two jobs. Thus, the assumption of a single job per class does

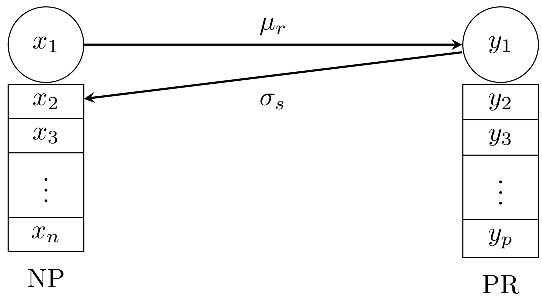

Fig. 1 The family of two-station LCFS networks under study. The NP and PR servers process jobs of classes $x_1 = r$ and $y_1 = s$, respectively, while their buffers $(x_2, \ldots, x_n)$ and $(y_2, \ldots, y_p)$ are waiting stacks. An arrival from NP to the PR station preempts the running job at the destination. Conversely, an arrival from PR to the NP station cannot preempt the running job.

not harm generality, since a model with N_r jobs in class r can be represented in an exact fashion using a model having $N_r = 1$ for all classes and $N = \sum_r N_r$ classes.

Service times are exponentially distributed, with rate $\mu_r > 0$ at the NP station and rate $\sigma_r > 0$ at the PR station, having mean values $\alpha_r = \mu_r^{-1}$ and $\beta_r = \sigma_r^{-1}$, respectively. Routing is cyclic, therefore a class-r job leaving the NP queue joins the PR queue with no option for feedback loops, and similarly a job leaving the PR queue joins the NP queue.

In what follows, it is convenient to describe the state of each station through a vector specifying the scheduled order of service for the jobs residing at each station, including the job in execution. That is, the NP station has state

$$\boldsymbol{x} = (x_1, x_2, \ldots, x_n), \qquad x_i \in \{1, \ldots, R\} \tag{5}$$

where n describes the total queue-length of the NP station and x_i is the class of the job in position i at the station. The state vector is read left-to-right, i.e., job x_i is scheduled for execution earlier than job x_{i+1}; thus, x_1 is the running job and x_n is the job scheduled last for execution. When the NP station is empty, we set $x = (\emptyset)$ and $n = 0$. The state of the PR station is defined identically to the NP case, but indicated with $\boldsymbol{y} = (y_1, y_2, \ldots, y_p)$, where $y_i \in \{1, \ldots, R\}$, and we denote with $p = N - n$ the total queue-length at the PR station.

With the above definitions, a network state is any tuple $\boldsymbol{s} = (\boldsymbol{x}; \boldsymbol{y})$ that describes a valid permutation of the positions of the N jobs across the two stations. Note that our notation uses a semicolon to separate jobs belonging to different stations, as opposed to commas to separate them within the station states $\boldsymbol{x}$ and $\boldsymbol{y}$. The state space is now given by the set

$$\mathcal{E} = \left\{ \boldsymbol{s} = (x; y) \,\middle|\, n + p = N, \boldsymbol{s} \in \Sigma_N \right\} \tag{6}$$

where Σ_N contains all possible permutations of the class indexes $\{1, \ldots, N\}$. The number of states is finite and equal to $|\mathcal{E}| = (N+1)!$, since $|\Sigma_N| = N!$ and there are $N + 1$ ways of choosing non-negative n and p that sum to N.

3.2 Continuous-Time Markov Chain

Using the definitions above, we may specify a continuous-time Markov chain (CTMC) describing the network dynamics. Let $\boldsymbol{Q} = [q_{s,s'}]$ be the infinitesimal generator underlying the CTMC that describes the transition rate between $\boldsymbol{s} = (\boldsymbol{x}, \boldsymbol{y})$ and $\boldsymbol{s}' = (\boldsymbol{x}'; \boldsymbol{y}')$, $\boldsymbol{s}, \boldsymbol{s}' \in \mathcal{E}$. Then

$$q_{s,s'} = \begin{cases} -\sum_{j \neq s} q_{s,j} & \text{if } s = s' \\ \mu_r & \text{if } x_1 = r, \boldsymbol{x}' = (x_2, \dots, x_n), \boldsymbol{y}' = (r, y_1, \dots, y_p) \\ \sigma_r & \text{if } y_1 = r \text{ and } n = 0, \boldsymbol{y}' = (y_2, \dots, y_p), \boldsymbol{x}' = (r) \\ \sigma_r & \text{if } y_1 = r \text{ and } n > 0, \boldsymbol{y}' = (y_2, \dots, y_p), \boldsymbol{x}' = (x_1, r, x_2, \dots, x_n) \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

This definition highlights the difference between the LCFS-NP and LCFS-PR scheduling disciplines. After a class-r job finishes, both disciplines select the job at the top of the buffer – which we refer to as the waiting stack – as the next job to be executed, i.e., x_2 or y_2. The difference arises upon arrival of a class-r job to the queue. Upon joining the PR station, it immediately preempts the running job, which moves to the top of the waiting stack (μ_r transition); in NP, the arriving job either joins the top of the waiting stack ($n > 0$ case), or starts immediately if the NP station is empty ($n = 0$ case).

Throughout, we indicate with the (row) vector $\boldsymbol{\pi} = (\pi(\boldsymbol{s}))$, $\boldsymbol{s} \in \mathcal{E}$, the stationary distribution for this finite CTMC. This can be readily obtained by solving the system of linear equations $\boldsymbol{\pi Q} = 0$, $\pi \boldsymbol{e} = 1$, where $\boldsymbol{e} = (1, \dots, 1)^T$. The problem we address in the paper is finding product-form expressions for each state probability $\pi(\boldsymbol{s})$.

3.2.1 Running case

To summarize the definitions, we introduce a small running case used throughout the next sections. We consider first a model with $N = 2$ jobs with state space

$$\mathcal{E} = \{(\emptyset; 1, 2), (\emptyset; 2, 1), (1; 2), (2; 1), (2, 1; \emptyset), (1, 2; \emptyset)\}$$

having cardinality $|\mathcal{E}| = (N + 1)! = 6$. Taking as an example element $(2, 1; \emptyset)$, this state represents a class-2 job running at the NP station, a class-1 job at the top of the waiting stack, and an empty PR station. Let us assign to the above states, from left to right, indexes 1 to 6. The infinitesimal generator in (7) is then given by

$$\boldsymbol{Q} = \begin{bmatrix} -\sigma_1 & 0 & \sigma_1 & 0 & 0 & 0 \\ 0 & -\sigma_2 & 0 & \sigma_2 & 0 & 0 \\ \mu_1 & 0 & -\mu_1 - \sigma_2 & 0 & 0 & \sigma_2 \\ 0 & \mu_2 & 0 & -\mu_2 - \sigma_1 & \sigma_1 & 0 \\ 0 & 0 & \mu_1 & 0 & -\mu_1 & 0 \\ 0 & 0 & 0 & \mu_2 & 0 & -\mu_2 \end{bmatrix}$$

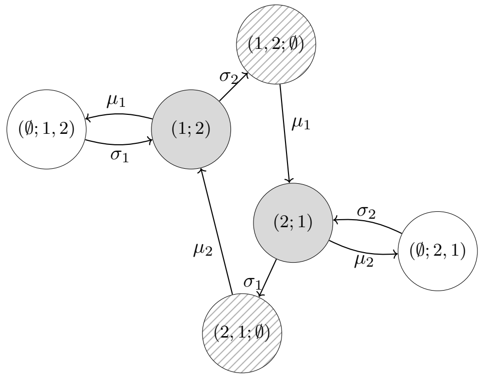

Fig. 2 CMTC state transition diagram for the running case with $N = 2$ jobs. The servers of both stations are busy in solid shaded states. The PR station is empty in striped states. The NP station is empty in white states.

Figure 2 illustrates the state transition diagram for this CTMC. Shading is based on whether both stations are busy running jobs or whether one of them is idle.

To see the general state-space pattern, we also illustrate the struture of this CTMC on a larger population with $N = 3$ jobs. The state space now grows to 24 states and the CTMC state transitions are illustrated in Figure 3; the mapping of node identificators to state vectors is given in Table 1. States 1 to 6 correspond to the NP station becoming empty, while states 19 to 24 correspond to the PR station becoming empty. In the remaining 12 states, both stations are busy.

It is interesting to note that, moving from 2 to 3 jobs, Figure 3 displays new cycles, indicated with solid shading. Each shaded cycle for $N = 3$, when paired with its satellite states shown in white, displays a similar structure as the state space for $N = 2$ in Figure 2, provided one job is removed from the state space. Consider for example the cycle consisting of nodes {10, 14, 12, 16}. The associated transitions describe the closed loop

$$(2; 1, 3) \xrightarrow{\sigma_1} (2, 1; 3) \xrightarrow{\mu_2} (1; 2, 3) \xrightarrow{\sigma_2} (1, 2; 3) \xrightarrow{\mu_1} (2; 1, 3) \xrightarrow{\sigma_1} \cdots$$

that performs an adjacent job interchange between NP states $\boldsymbol{x} = (1, 2)$ and $\boldsymbol{x}' = (2, 1)$ during a period when the class-3 job is resident at the PR station. Noting that the satellite nodes 2 and 4 correspond to states $(\emptyset; 2, 1, 3)$ and $(\emptyset; 1, 2, 3)$, we conclude that the interchange cycle and its satellite states describe the network dynamics when job 3 is conditioned to remain in the last position at the PR station. If we disable transitions for arrival of departure for job 3 and ignore this job, the group of states {2, 4, 10, 14, 12, 16} simply reduces to the CTMC for $N = 2$ in Figure 2.

Note that the last property shows an asymmetry between the two disciplines, i.e., a job holding the last position at the NP station, instead than the PR station, sees the network evolve on a fairly different state space than in Figure 2, see for instance the group of states {7,8,17,18,20,22}. Here, state 8 is not doubly-connected to state 18 precisely due to the NP discipline, as it would instead be under PR. This breaks the state space pattern compared to the one in the model with $N = 2$.

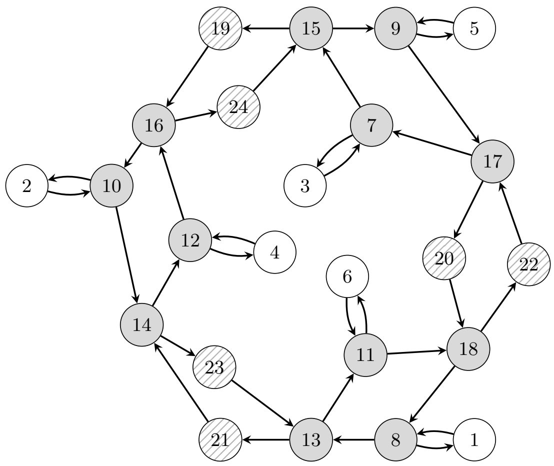

Fig. 3 CMTC state transition diagram for the running case with $N = 3$ jobs. Node shading is defined as in Figure 2. The mapping of indexes to detailed (x, y) states is given in Table 1.

Table 1 State space $\mathcal{E}$ for the running case with $N = 3$ jobs. Each state $(x; y)$ describes the schedules at the NP and PR stations.

Node	State	Node	State	Node	State	Node	State
1	$(\emptyset; 3, 1, 2)$	7	$(3; 2, 1)$	13	$(3, 1; 2)$	19	$(3, 1, 2; \emptyset)$
2	$(\emptyset; 2, 1, 3)$	8	$(3; 1, 2)$	14	$(2, 1; 3)$	20	$(2, 1, 3; \emptyset)$
3	$(\emptyset; 3, 2, 1)$	9	$(2; 3, 1)$	15	$(3, 2; 1)$	21	$(3, 2, 1; \emptyset)$
4	$(\emptyset; 1, 2, 3)$	10	$(2; 1, 3)$	16	$(1, 2; 3)$	22	$(1, 2, 3; \emptyset)$
5	$(\emptyset; 2, 3, 1)$	11	$(1; 3, 2)$	17	$(2, 3; 1)$	23	$(2, 3, 1; \emptyset)$
6	$(\emptyset; 1, 3, 2)$	12	$(1; 2, 3)$	18	$(1, 3; 2)$	24	$(1, 3, 2; \emptyset)$

The above observations can be repeated for jobs 1 and 2, looking at the interchange cycles {7,15,9,17} and {8,13,11,18}, together with their respective satellite states {3,5} and {1,6}. Here too, upon removal of the job under study, the CTMC matches the corresponding transition diagram for $N = 2$ when considering the remaining jobs.

3.2.2 Irreducibility

Based on the intuitions gained from the running example on the CTMC structure, the proposed model can be shown to be irreducible. Assuming finite rates, this is sufficient to ensure ergodicity, and thus have a unique distribution π at equilibrium. This is not an obvious property in LCFS networks, for example the CTMC of a cyclic network consisting of two LCFS-PR stations with $R = N = 2$ is reducible into two non-communicating components and its stationary analysis requires care in assigning the initial state of the two CTMCs[1]. We now show that the models under study do not face these complications.

[1]Numerical solution for the equilibrum of such models requires to initialize the CTMC in each strongly connected component with probability proportional to the sum of the BCMP product-form factors for the states within that component.

Theorem 1 *If $\mu_r > 0$ and $\sigma_r > 0$, $r = 1, \dots, R$, then the CTMC (7) is irreducible.*

Proof We need to show that there exists a path between any pair of states $s \in \mathcal{E}$ and $s' \in \mathcal{E}$. We do this by induction on the number of jobs N, i.e., showing that at each population the state transitions define a single strongly connected component.

Base case, $N = 1$. The CTMC is irreducible as it consists only of two states $(1; \emptyset)$ and $(\emptyset; 1)$, connected back and forth by transitions with rates $\mu_1 > 0$ and $\sigma_1 > 0$.

Induction step. We now assume that models with $N-1$ jobs have an irreducible CTMC, thus featuring a single strongly connected component. Consider the group of states $\mathcal{E}_r$ where a job of class r is scheduled last for execution at the PR station, thus $y_p = r$. Transitions restricted to $\mathcal{E}_r$ form a CTMC for a system without the class-r job, and thus by the induction hypothesis they define a strongly connected component. Since $\sigma_r > 0$ and $\mu_j > 0$, each $\mathcal{E}_r$ is also bidirectionally connected to any another $\mathcal{E}_j$, $j \neq r$, by means of transition sequences

$$(j, x_2, \dots, x_n; r) \xrightarrow{\sigma_r} (j, r, x_2, \dots, x_n; \emptyset) \xrightarrow{\mu_j} (r, x_2, \dots, x_n; j)$$

for all permutations $(j, x_2, \dots, x_n) \in \Sigma_n$, which are all reachable within S_r since this defines a strongly connected component by the induction hypothesis. With a similar argument, there exists a transition sequence connecting back $\mathcal{E}_j$, $j \neq r$ to $\mathcal{E}_r$, i.e.,

$$(r, x_2, \dots, x_n; j) \xrightarrow{\sigma_j} (r, j, x_2, \dots, x_n; \emptyset) \xrightarrow{\mu_r} (j, x_2, \dots, x_n; r)$$

thus proving that each pair of strongly connected components $\mathcal{E}_j$ and $\mathcal{E}_r$ is bidirectionally connected and thus belongs to the same strongly connected component. The two transition sequences described above also show that any state where the PR station is empty has a path to both states in $\mathcal{E}_j$ and $\mathcal{E}_r$. Thus, all states in $\mathcal{E}$ belong to the same strongly connected component.

□

The proof provides an intuition on how the transition diagram generalizes for $N > 3$. The CTMC features partitions of states defined by conditioning on the presence of a class-r job in position p at the PR node. Each such group needs to be reachable from $N-1$ other partitions, one for each choice of the class $j \neq r$ in position p at the PR station and a different permutation of the jobs at the NP station.

Running case. The described pattern is already visible when comparing Figure 2 with Figure 3. Conditioning on the last position p at the PR node corresponds in Figure 3 to the set of solid shaded and white states that are directly connected to each other. In Figure 2, as $(N-1) = 1$, the $\{(\emptyset; 1, 2), (1, 2)\}$ group is bidirectionally connected to the group defined by $\{(\emptyset; 2, 1), (2, 1)\}$. This connection leverages transitions through the striped states $\{1, 2; \emptyset\}$ and $\{2, 1; \emptyset\}$ where the PR station is empty. We see the same in Figure 3, where now $(N-1) = 2$, and thus every group of directly connected solid and white states (e.g., nodes labeled {2,4,10,12,14,16}) is connected to two other similar groups, again by means of transitions through the striped states where the PR station is empty.

4 Main result

4.1 Product-form solution

We are now ready to prove a general product-form solution for the models under study.

Theorem 2 *The model with N jobs and $R = N$ classes defined in Section 3.1 admits for all states $\boldsymbol{s} = (\boldsymbol{x}; \boldsymbol{y})$, $s \in \mathcal{E}$, the product-form stationary distribution*

$$\pi(\boldsymbol{s}) = \pi(\boldsymbol{x}; \boldsymbol{y}) = \frac{1}{S} \left(\prod_{i=1}^{n} \alpha_{x_i}^{n-i+1} \right) \left(\prod_{j=1}^{p} \alpha_{y_j}^{N-p+j-1} \beta_{y_j} \right) \tag{8}$$

where S is a normalizing constant, n is the number of jobs at the NP station, and p is the number of jobs at the PR station.

Proof We show that (8) satisfies the global balance equations of the system under study. We classify the global balance equations according to the busy stations in state s, showing on the left-hand side of each global balance equation the contribution due to outgoing transitions from $\pi(\boldsymbol{s})$ and on the right-hand side the incoming transitions from connected states.
Case 1: NP busy, PR idle. For such states, $y = \emptyset$ and the global balance equations are

$$\mu_{x_1} \pi(x_1, x_2, \ldots, x_N; \emptyset) = \sigma_{x_2} \pi(x_1, x_3, \ldots, x_N; x_2)$$

Plugging (8), and simplifying, we get

$$\mu_{x_1} \left(\prod_{i=1}^{N} \alpha_{x_i}^{N-i+1} \right) = \sigma_{x_2} \alpha_{x_1}^{N-1} \left(\prod_{i=3}^{N} \alpha_{x_i}^{(N-1)-(i-1)+1} \right) \left(\alpha_{x_2}^{N-1} \beta_{x_2} \right)$$

By definition, $\mu_{x_1} = 1/\alpha_{x_1}$ and $\sigma_{x_2} = 1/\beta_{x_2}$, so that the last expression becomes the identity

$$\left(\prod_{i=1}^{N} \alpha_{x_i}^{N-i+1} \right) = \alpha_{x_1}^{N} \alpha_{x_2}^{N-1} \left(\prod_{i=3}^{N} \alpha_{x_i}^{N-i+1} \right)$$

Case 2: NP idle, PR busy. Here the global balance equation is

$$\sigma_{y_1} \pi(\emptyset; y_1, y_2, \ldots, y_N) = \mu_{y_1} \pi(y_1; y_2, y_3, \ldots, y_N)$$

Plugging again (8), we obtain

$$\sigma_{y_1} \left(\prod_{j=1}^{N} \alpha_{y_j}^{j-1} \beta_{y_j} \right) = \mu_{y_1} \alpha_{y_1} \left(\prod_{j=2}^{N} \alpha_{y_j}^{1+(j-1)-1} \beta_{y_j} \right)$$

Plugging $\mu_{y_1} = 1/\alpha_{y_1}$ and $\sigma_{y_1} = 1/\beta_{y_1}$ and simplifying, we now get

$$\left(\prod_{j=1}^{N} \alpha_{y_j}^{j-1} \beta_{y_j} \right) = \beta_{y_1} \left(\prod_{j=2}^{N} \alpha_{y_j}^{j-1} \beta_{y_j} \right)$$

and equality holds since the term for $j = 1$ on the left-hand side evaluates to β_{y_1}.
Case 3: both stations busy. Lastly, we study states where $n > 0$ and $p > 0$. Here two departure types are possible, and the global balance equations take the form

$$(\sigma_{y_1} + \mu_{x_1}) \pi(\boldsymbol{x}; \boldsymbol{y}) = \mu_{y_1} \pi(y_1, \boldsymbol{x}; y_2, y_3, \ldots, y_p) + \sigma_{x_2} \pi(x_1, x_3, \ldots, x_n; x_2, \boldsymbol{y}) \tag{9}$$

Using (8), we can write the left-hand side probability as

$$\pi(\boldsymbol{x}; \boldsymbol{y}) = \frac{1}{S} \left(\prod_{i=1}^{n} \alpha_{x_i}^{n-i+1} \right) \left(\prod_{j=1}^{p} \alpha_{y_j}^{N-p+j-1} \beta_{y_j} \right) \tag{10}$$

The first probability on the right-hand side of (9) is

$$\pi(y_1, \boldsymbol{x}; y_2, y_3, \ldots, y_p) = \frac{1}{S} \alpha_{y_1}^{n+1} \left(\prod_{i=1}^{n} \alpha_{x_i}^{(n+1)-(i+1)+1} \right) \tag{11}$$

$$\times \left(\prod_{j=2}^{p} \alpha_{y_j}^{N-(p-1)+(j-1)-1} \beta_{y_j} \right)$$

$$= \frac{1}{S} \alpha_{y_1}^{N-p+1} \left(\prod_{i=1}^{n} \alpha_{x_i}^{n-i+1} \right) \left(\prod_{j=2}^{p} \alpha_{y_j}^{N-p+j-1} \beta_{y_j} \right)$$

$$= \frac{1}{S} \frac{\alpha_{y_1}}{\beta_{y_1}} \left(\prod_{i=1}^{n} \alpha_{x_i}^{n-i+1} \right) \left(\prod_{j=1}^{p} \alpha_{y_j}^{N-p+j-1} \beta_{y_j} \right) \qquad (12)$$

where we used, upon deriving the second expression, that in $\mathcal{E}$ it is $n = N - p$.

The probability in the second term on the right-hand side of (9) is

$$\pi(x_1, x_3, \ldots, x_n; x_2, \boldsymbol{y}) = \frac{1}{S} \alpha_{x_1}^{n-1} \left(\prod_{i=3}^{n} \alpha_{x_i}^{(n-1)-(i-1)+1} \right) \times$$

$$\times \left(\alpha_{x_2}^{N-(p+1)} \beta_{x_2} \prod_{j=1}^{p} \alpha_{y_j}^{N-(p+1)+(j+1)-1} \beta_{y_j} \right)$$

$$= \frac{1}{S} \alpha_{x_1}^{n-1} \alpha_{x_2}^{n-1} \beta_{x_2} \left(\prod_{i=3}^{n} \alpha_{x_i}^{n-i+1} \right)$$

$$\times \left(\prod_{j=1}^{p} \alpha_{y_j}^{N-p+j-1} \beta_{y_j} \right)$$

$$= \frac{1}{S} \alpha_{x_1}^{-1} \beta_{x_2} \left(\prod_{i=1}^{n} \alpha_{x_i}^{n-i+1} \right) \left(\prod_{j=1}^{p} \alpha_{y_j}^{N-p+j-1} \beta_{y_j} \right) \qquad (13)$$

Combining (12), $\sigma_{y_1} = 1/\beta_{y_1}$, $\mu_{y_1} = 1/\alpha_{y_1}$ it now follows that

$$\sigma_{y_1} \pi(\boldsymbol{x}; \boldsymbol{y}) = \mu_{y_1} \pi(y_1, \boldsymbol{x}; y_2, y_3, \ldots, y_p) \qquad (14)$$

Similarly, using (13), $\sigma_{x_2} = 1/\beta_{x_2}$, $\mu_{x_1} = 1/\alpha_{x_1}$, we find

$$\mu_{x_1} \pi(\boldsymbol{x}; \boldsymbol{y}) = \sigma_{x_2} \pi(x_1, x_3, \ldots, x_n; x_2, \boldsymbol{y}) \qquad (15)$$

Summing (14) and (15) yields (9), which completes the proof. □

We now comment on some general properties of the product-form solution (8) and its relation to existing product-forms.

Order-dependence. Similarly to the BCMP product-form, in (8) the queue-length sizes n and p both bear a strong influence on the product-form factors for the NP and PR stations. However, a known property of BCMP product-forms is that any two states differing only for the order of the jobs in a station have equal probability [32]. Clearly, (8) violates this property as both the NP and PR terms depend in their exponents on the relative position of the jobs within the schedules $\boldsymbol{x}$ and $\boldsymbol{y}$. Thus, our result offers an example of order-dependent product-form in queueing networks.

Dependence on position can be further highlighted if we note that, since $N - p = n$, (8) may also be rewritten as

$$\pi(\boldsymbol{x}, \boldsymbol{y}) = \frac{1}{S} \left(\prod_{k=1}^{N} \alpha_k^n \right) \left(\prod_{i=1}^{n} \alpha_{x_i}^{-i+1} \right) \left(\prod_{j=1}^{p} \alpha_{y_j}^{j-1} \right) \left(\prod_{j=1}^{p} \beta_{y_j} \right) \qquad (16)$$

where now the second and third products capture the position-based factors. Higher α_r powers in the expression appear for class-r jobs scheduled early in the NP station and, viceversa, at the PR station these are associated to jobs scheduled late.

BCMP specialization under identical classes. First, it is worth noting that in a trivial model where $N = 1$, the order-depedent terms are not present and (16) coincides with the BCMP expression for this case

$$\pi(1;\emptyset) = \frac{\alpha_1}{S} \qquad \pi(\emptyset;1) = \frac{\beta_1}{S} \qquad S = \alpha_1 + \beta_1 \tag{17}$$

Next, we consider the case $N > 1$, where we expect (8) to be able to recover the BCMP product-form if $\alpha_r = \alpha$ and $\beta_r = \beta$, $r = 1, \ldots, N$, since this corresponds to a single-class model with N jobs, in which case NP and PR are stochastically identical under exponential service times, as the CTMC does not need to track residual times. Indeed, after algebraic manipulations, and using $n + p = N$, we find that

$$\pi(\boldsymbol{x};\boldsymbol{y}) = \frac{1}{S}\left(\prod_{i=1}^{n} \alpha^{n-i+1}\right)\left(\prod_{j=1}^{p} \alpha^{N-p+j-1}\beta\right) \tag{18}$$

$$= \left(\frac{\alpha^{\frac{N(N-1)}{2}}}{S}\right)\alpha^n\beta^p = \frac{1}{S'}\alpha^n\beta^p \tag{19}$$

where in the last passage we have grouped all constant terms into a new normalizing constant S'. This expression correctly matches the expected value for the BCMP product-form (3) with demands $\theta_{\text{NP}} = \alpha$ and $\theta_{\text{PR}} = \beta$, under the assumption of identical mean service times and visit ratio for all classes.

Relation to other product-forms. Next, we relate the obtained product-form with known product-form solutions arising from reversible, quasi-reversible, and order-independent models [2, 7, 11]. It is immediate to verify that the model is not order-independent, since NP does not belong to this class of models [11]. Similarly, it is possible to verify by the definition that the underpinning CTMC is not reversible, a simple counterexample being the generator shown in Section 3.2.1, for which the time-reversed CTMC has infinitesimal generator

$$\bar{\boldsymbol{Q}} = \begin{bmatrix} -\sigma_1 & 0 & \sigma_1 & 0 & 0 & 0 \\ 0 & -\sigma_2 & 0 & \sigma_2 & 0 & 0 \\ \mu_1 & 0 & -\mu_1 - \sigma_2 & 0 & \sigma_2 & 0 \\ 0 & \mu_2 & 0 & -\mu_2 - \sigma_1 & 0 & \sigma_1 \\ 0 & 0 & 0 & \mu_1 & -\mu_1 & 0 \\ 0 & 0 & \mu_2 & 0 & 0 & -\mu_2 \end{bmatrix}$$

that differs from $\boldsymbol{Q}$. Also note that, in general, there is no valid re-labeling of the states to make the two matrices equal, thus the CTMC is also not ρ-reversible [10].

Concerning quasi-reversibility, while the LCFS-PR station is known to be quasi-reversible, non-preemptive multiclass LCFS-NP in general is neither quasi-reversible nor symmetric [25, §5.5]. This lack of quasi-reversibility can be easily checked by noting that the addition of another LCFS-NP station to the cyclic topology in any position loses the product-form solution. This applies even if the new station is identical to the NP queue already present in the model. In such models, even for small populations N, factors appear in the CTMC solution that depend on the sum of mixed products of mean service times at different stations. Such observations are incompatible with the properties of networks of quasi-reversible queues.

4.1.1 Corollaries

A number of additional characterizations follow from Theorem 2.

4.1.2 Detailed balances

The first two corollaries arise from equations (14) and (15) in the proof, which show the existence of two independent balances that drive the product-form solution.

Corollary 1 *The departure rate of class-r jobs from the PR station with queue-length $p = N - n$ balances the class-r arrival rate from the NP station with queue-length $n+1$, i.e.,*

$$\sigma_r \pi(\boldsymbol{x}; r, y_2, y_3, \dots, y_p) = \mu_r \pi(r, \boldsymbol{x}; y_2, y_3, \dots, y_p) \tag{20}$$

Corollary 2 *The departure rate of class-r jobs from the NP station with queue-length $n = N - p$ balances the class-r arrival rate from the PR station with queue-length $p+1$, i.e.,*

$$\mu_r \pi(r, x_2, \dots, x_n; \boldsymbol{y}) = \sigma_r \pi(x_1, x_3, \dots, x_n; r, \boldsymbol{y}) \tag{21}$$

Running case. If we attempt to relate the above balances to the running case, we note that they both describe states where the job in position p at the PR station is fixed. Thus, they provide a balance locally to the white and solid shaded states in both Figure 2 and Figure 3. Looking at the detailed view in Figure 4, the two corollaries (instantiated for $r = 1$ and $r = 2$) ensure that the probability flow around each closed cycle in the figure is conserved.

4.1.3 Symmetry

Another consequence of Theorem 2 is the following symmetry among the state probabilities, which can be easily verified by direct substitution of (8) into (22).

Corollary 3 *The product of the stationary probabilities of any pair of states, $\boldsymbol{s} = (\boldsymbol{x}; \boldsymbol{y})$ and $\boldsymbol{s}' = (\boldsymbol{y}; \boldsymbol{x})$, with $s, s' \in \mathcal{E}$, yields a constant, namely*

$$\pi(\boldsymbol{x}; \boldsymbol{y}) \pi(\boldsymbol{y}; \boldsymbol{x}) = \frac{\prod_{i=1}^{N} \alpha_i^N \beta_i}{S^2} \tag{22}$$

Based on the last result, it may be sufficient to compute $|\mathcal{E}|/2$ state probabilities to readily obtain the remaining ones by (22).

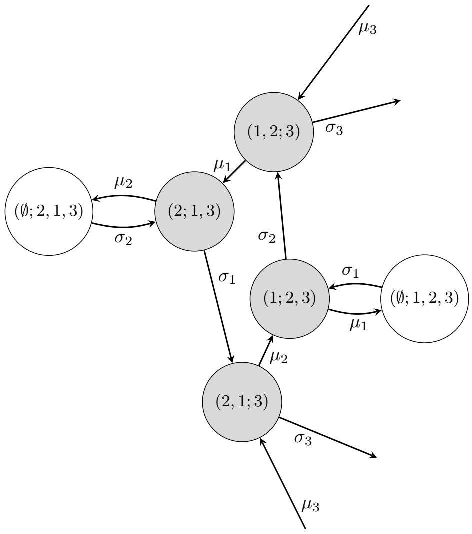

Fig. 4 Detail of the group of states {2,4,10,12,14,16} in the CTMC shown in Figure 3. Note that eliminating transitions for the job of class-3 recovers the same CTMC structure of Figure 2.

Running case. Thanks to its small size when $N = 2$, the CTMC global balance equations can be solved explicitly under the infinitesimal generator $\boldsymbol{Q}$ defined in Section 3.2.1. Recall that $\alpha_r = \mu_r^{-1}$ and $\beta_r = \sigma_r^{-1}$ are the mean service times for class-r at the NP and PR stations, respectively. It is possible to manually check that the stationary distribution is in this example

$$\pi(\emptyset;1,2) = \frac{\alpha_2\beta_1\beta_2}{S} \qquad \pi(\emptyset;2,1) = \frac{\alpha_1\beta_1\beta_2}{S}$$

$$\pi(1;2) = \frac{\alpha_1\alpha_2\beta_2}{S} \qquad \pi(2;1) = \frac{\alpha_2\alpha_1\beta_1}{S}$$

$$\pi(2,1;\emptyset) = \frac{\alpha_2^2\alpha_1}{S} \qquad \pi(1,2;\emptyset) = \frac{\alpha_1^2\alpha_2}{S}$$

where the normalizing constant is given by

$$S = \alpha_2\beta_1\beta_2 + \alpha_1\beta_1\beta_2 + \alpha_1\alpha_2\beta_2 + \alpha_2\alpha_1\beta_1 + \alpha_2^2\alpha_1 + \alpha_1^2\alpha_2 \tag{23}$$

This matches the expression in (8) when $N = 2$. It is interesting to compare this solution to a model in which the NP station is changed to PR, as in this case the assumptions of the BCMP product-form (3) hold. In the example, the mean service demands are $\theta_{1,r} = \alpha_r$ and $\theta_{2,r} = \beta_r$; we call the corresponding rates $\widetilde{\mu}_r$ and $\widetilde{\sigma}_r$.

Noting that the state space remains unchanged, the infinitesimal generator becomes

$$\widetilde{\boldsymbol{Q}} = \begin{bmatrix} -\widetilde{\sigma}_1 & 0 & \widetilde{\sigma}_1 & 0 & 0 & 0 \\ 0 & -\widetilde{\sigma}_2 & 0 & \widetilde{\sigma}_2 & 0 & 0 \\ \widetilde{\mu}_1 & 0 & -\widetilde{\mu}_1 - \widetilde{\sigma}_2 & 0 & \widetilde{\sigma}_2 & 0 \\ 0 & \widetilde{\mu}_2 & 0 & -\widetilde{\mu}_2 - \widetilde{\sigma}_1 & 0 & \widetilde{\sigma}_1 \\ 0 & 0 & \widetilde{\mu}_2 & 0 & -\widetilde{\mu}_2 & 0 \\ 0 & 0 & 0 & \widetilde{\mu}_1 & 0 & -\widetilde{\mu}_1 \end{bmatrix}$$

with the changes being in the transitions back and forth the last two states. It is possible to show that the stationary probabilities, denoted by $\widetilde{\pi}(\boldsymbol{s})$, $\boldsymbol{s} \in \mathcal{E}$, are now given by

$$\widetilde{\pi}(\emptyset; 1, 2) = \frac{\theta_{2,1}\theta_{2,2}}{\widetilde{S}} \qquad \widetilde{\pi}(\emptyset; 2, 1) = \frac{\theta_{2,1}\theta_{2,2}}{\widetilde{S}}$$

$$\widetilde{\pi}(1; 2) = \frac{\theta_{1,1}\theta_{2,2}}{\widetilde{S}} \qquad \widetilde{\pi}(2; 1) = \frac{\theta_{1,2}\theta_{2,1}}{\widetilde{S}}$$

$$\widetilde{\pi}(2, 1; \emptyset) = \frac{\theta_{1,2}\theta_{1,1}}{\widetilde{S}} \qquad \widetilde{\pi}(1, 2; \emptyset) = \frac{\theta_{1,1}\theta_{1,2}}{\widetilde{S}}$$

where

$$\widetilde{S} = 2(\theta_{2,1}\theta_{2,2} + \theta_{1,1}\theta_{2,2} + \theta_{1,1}\theta_{1,2})$$

is again a normalizing constant.

The last solution is notably different from the one obtained for the previous case. Note first that, with mixed disciplines, the order of the terms is higher. Also, even states such as $(\emptyset; 1, 2)$, where all the jobs reside at the PR station, include terms α_r describing the service process of the NP station, despite the latter is empty. Conversely, the BCMP product-form features terms only describing the local populations at each station. These observations hold also in larger models, as visible in Appendix A where we report the product-form equilibrium for the case $N = 3$.

A property that remains similar in the two models is the one captured by Corollary 3, since we see that multiplying the probability of symmetric states yields an identical constant, i.e.,

$$\pi(\emptyset; 1, 2)\pi(1, 2; \emptyset) = \pi(\emptyset; 2, 1)\pi(2, 1; \emptyset) = \pi(1; 2)\pi(2; 1) = \frac{\alpha_1^2\alpha_2^2\beta_1\beta_2}{S^2} \tag{24}$$

and

$$\widetilde{\pi}(\emptyset; 1, 2)\widetilde{\pi}(1, 2; \emptyset) = \widetilde{\pi}(\emptyset; 2, 1)\widetilde{\pi}(2, 1; \emptyset) = \widetilde{\pi}(1; 2)\widetilde{\pi}(2; 1) = \frac{\theta_{1,1}\theta_{1,2}\theta_{2,1}\theta_{2,2}}{\widetilde{S}^2} \tag{25}$$

4.1.4 Reciprocal model

Lastly, we give a corollary that relates networks with mean service times that are reciprocals of each other. This result is important for later developments on computational algorithms for LCFS networks, where the reciprocal model allows us to study the NP station using results for the PR station.

Corollary 4 *Consider a model parameterized with mean service times α_r and β_r at the NP and PR stations, with stationary distribution $\pi(\boldsymbol{s})$, for all states $\boldsymbol{s} \equiv (\boldsymbol{x}; \boldsymbol{y}) \in \mathcal{E}$. Define the reciprocal model as a network with mean service times $\bar{\alpha}_r = \alpha_r^{-1}$ and $\bar{\beta}_r = \beta_r^{-1}$, and call its stationary distribution $\bar{\pi}(\boldsymbol{s})$. Then*

$$\pi(\boldsymbol{x}, \boldsymbol{y}) = \bar{\pi}(\boldsymbol{y}, \boldsymbol{x}) \tag{26}$$

and the normalizing constants of the two models are related by

$$\bar{S} = S \prod_{k=1}^{N} \bar{\alpha}_k^N \bar{\beta}_k \tag{27}$$

$$S = \bar{S} \prod_{k=1}^{N} \alpha_k^N \beta_k \tag{28}$$

Proof

$$\pi(\boldsymbol{x}; \boldsymbol{y}) = \frac{1}{S} \left(\prod_{i=1}^{n} \alpha_{x_i}^{n-i+1} \right) \left(\prod_{j=1}^{p} \alpha_{y_j}^{N-p+j-1} \beta_{y_j} \right) \tag{29}$$

$$= \frac{1}{S} \left(\prod_{i=1}^{n} \bar{\alpha}_{x_i}^{-n+i-1} \right) \left(\prod_{j=1}^{p} \bar{\alpha}_{y_j}^{-N+p-j+1} \beta_{y_j} \right)$$

$$= \frac{1}{(\prod_{k=1}^{N} \bar{\alpha}_k^N \bar{\beta}_k) S} \left(\prod_{i=1}^{n} \bar{\alpha}_{x_i}^{N-n+i-1} \bar{\beta}_{x_i} \right) \left(\prod_{j=1}^{p} \bar{\alpha}_{y_j}^{p-j+1} \right) \tag{30}$$

Requiring $\sum_{(\boldsymbol{x},\boldsymbol{y})} \pi(\boldsymbol{x}, \boldsymbol{y}) = 1$, yields

$$\sum_{(x,y)} \left(\prod_{i=1}^{n} \bar{\alpha}_{x_i}^{N-n+i-1} \bar{\beta}_{x_i} \right) \left(\prod_{j=1}^{p} \bar{\alpha}_{y_j}^{p-j+1} \right) = \left(\prod_{k=1}^{N} \bar{\alpha}_k^N \bar{\beta}_k \right) S$$

but the left-hand side is $\bar{S}$, proving (27). Noting that $(\prod_{k=1}^{N} \bar{\alpha}_k^N \bar{\beta}_k)^{-1} = (\prod_{k=1}^{N} \alpha_k^N \beta_k)$ readily proves also (28). Returning to (30), we also see that, after introducing $\bar{S}$, this also implies $\pi(\boldsymbol{x}; \boldsymbol{y}) = \bar{\pi}(\boldsymbol{y}; \boldsymbol{x})$. □

Thanks to the last corollary, the recursive structure identified from the CTMC for the periods where a job is resident at the PR queue can now be extended to describe a recursion on periods where a job resides at the NP station. This is done by looking at the PR station in the reciprocal model, which since $\pi(\boldsymbol{x}, \boldsymbol{y}) = \bar{\pi}(\boldsymbol{y}, \boldsymbol{x})$ has performance metrics identical to the NP station in the original model. We leverage this observation later in Section 6 to derive recurrence relations for performance metrics analogous to mean-value analysis (MVA) in BCMP networks [33].

Comparison with BCMP networks. It should also be noted that the property above holds also for a two-station BCMP network. It can be easily checked following similar passages as in the proof of Corollary 4 that $\pi(\boldsymbol{x}, \boldsymbol{y}) = \bar{\pi}(\boldsymbol{y}, \boldsymbol{x})$ still holds, but the normalizing constants of the two models are now related by $\bar{S} = S \prod_{k=1}^{N} \bar{\theta}_{1,k}\bar{\theta}_{2,k}$ and $S = \bar{S} \prod_{k=1}^{N} \theta_{1,k}\theta_{2,k}$, where $\bar{\theta}_{i,j} = \theta_{i,j}^{-1}$. The formula does not generalize to BCMP models with $M > 2$ stations.

5 Computing the normalizing constant

5.1 Permanents

As common in closed queueing networks, state space explosion makes a direct computation of the normalizing constant S quickly intractable as the model size grows large. This also applies to the class of models at hand, where the state space has cardinality $|\mathcal{E}| = (N+1)!$. To address this problem, we now propose a method for computing S using permanents [34], which we show later in the paper yields a polynomial-time algorithm for the normalizing constant under a fixed number of job classes R.

Since to the best of our knowledge permanents have not been used before in queueing network theory, we give in this section a brief overview of the main properties of this operator. For an arbitrary real matrix $\boldsymbol{A} = [a_{i,j}]$ of order n, the permanent is defined similarly to the determinant, but with all the elements taken with positive sign, i.e.,

$$\operatorname{perm} \boldsymbol{A} = \sum_{\sigma \in \Sigma_n} \prod_{i=1}^{n} a_{i,\sigma(i)} \tag{31}$$

where Σ_n is the set of all possible permutations for the indexes $\{1, \dots, n\}$. Due to its similarity with the determinant, the permanent therefore admits Laplace expansion by minors [35, p. 16], e.g.,

$$\operatorname{perm} \boldsymbol{A} = \sum_{i=1}^{n} a_{ij} \boldsymbol{A}(i|j) \tag{32}$$

where j is an arbitrarily chosen column and we denote with $\boldsymbol{A}(\mathcal{I}|\mathcal{J})$ the minor obtained by removing rows $i \in \mathcal{I}$ and columns $j \in \mathcal{J}$ from $\boldsymbol{A}$. A similar expansion applies to an arbitrary row by applying the last result to the transpose $\boldsymbol{A}^T$, for which $\operatorname{perm} \boldsymbol{A} = \operatorname{perm} \boldsymbol{A}^T$. Lastly, we note that multiplying the elements of a column (or row) of $\boldsymbol{A}$ by a constant c acts a multiplier on the permanent of $\boldsymbol{A}$, i.e.,

$$\operatorname{perm} \begin{bmatrix} a_{11} & \cdots & ca_{1j} & \cdots & a_{1n} \\ a_{21} & \cdots & ca_{2j} & \cdots & a_{2n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{n1} & \cdots & ca_{nj} & \cdots & a_{nn} \end{bmatrix} = c \operatorname{perm} \boldsymbol{A} \tag{33}$$

In what follows, we show that (32) and (33) are sufficient to derive recurrence relations for normalizing constants and performance metrics in LCFS networks and show how they can also be used to recover the classic BCMP convolution algorithm [9, 36].

Lastly, we remark that (33) further suggests how to improve the numerical stability of computing a single permanent. For given diagonal matrices $\boldsymbol{D} = \mathrm{diag}(d_1, \ldots, d_n)$ and $\boldsymbol{E} = \mathrm{diag}(e_1, \ldots, e_n)$, (33) implies that

$$\operatorname{perm} \boldsymbol{DAE} = \left(\prod_i d_i\right)\left(\prod_j e_j\right) \operatorname{perm} \boldsymbol{A} \tag{34}$$

By suitable choice of $\boldsymbol{D}$ and $\boldsymbol{E}$, the last ressult enables transforming the argument of the permanent into a doubly stochastic matrix $\boldsymbol{P} = \boldsymbol{DAE}$, which tends to improve stability of the numerical evaluation, and recover afterwards perm $\boldsymbol{A}$.

5.2 Application to normalizing constants

The permanent naturally arises in the problem at hand due to the given definitions of the network state $\boldsymbol{s} = (\boldsymbol{x}; \boldsymbol{y})$ that, as noted before, may be seen itself as a permutation of the job class indexes $\{1, \ldots, N\}$ split into the two schedules $\boldsymbol{x}$ and $\boldsymbol{y}$ at the NP and PR stations. In the following, it is convienent to analyze such permutations for fixed values of the queue-lengths n and $p = N - n$, so that $\boldsymbol{x}$ and $\boldsymbol{y}$ are vectors of constant size. With this approach, we may rewrite the normalizing constant in Theorem 2 as

$$S = \sum_{k=0}^{N} \sum_{\substack{(x,y)\in\Sigma_N: \\ n=k}} \left(\prod_{i=1}^{n} \alpha_{x_i}^{n-i+1}\right)\left(\prod_{j=1}^{N-n} \alpha_{y_j}^{n+j-1}\beta_{y_j}\right) = \sum_{n=0}^{N} \operatorname{perm} \boldsymbol{A}_n \tag{35}$$

where $\boldsymbol{A}_n$ is a matrix of order N defined as

$$\boldsymbol{A}_n = \begin{bmatrix} \alpha_1 & \cdots & \alpha_1^{n-1} & \alpha_1^n & \alpha_1^n\beta_1 & \alpha_1^{n+1}\beta_1 & \cdots & \alpha_1^{N-1}\beta_1 \\ \alpha_2 & \cdots & \alpha_2^{n-1} & \alpha_2^n & \alpha_2^n\beta_2 & \alpha_2^{n+1}\beta_2 & \cdots & \alpha_2^{N-1}\beta_2 \\ \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \alpha_N & \cdots & \alpha_N^{n-1} & \alpha_N^n & \alpha_N^n\beta_N & \alpha_N^{n+1}\beta_N & \cdots & \alpha_N^{N-1}\beta_N \end{bmatrix} \tag{36}$$

By its definition, it should be noted that $\boldsymbol{A}_n$ therefore groups states where the NP station has population n and thus the quantities

$$\pi_{\mathrm{NP}}(n|\boldsymbol{N}) = \frac{\operatorname{perm} \boldsymbol{A}_n}{S} = \pi_{\mathrm{PR}}(N - n|\boldsymbol{N})$$

simply defines the marginal queue-length probability in the models at hand. Also note that, in the boundary cases $n = 0$ and $n = N$, the definition (36) gives

$$\boldsymbol{A}_0 = \begin{bmatrix} \beta_1 & \alpha_1\beta_1 & \cdots & \alpha_1^{N-1}\beta_1 \\ \beta_2 & \alpha_2\beta_2 & \cdots & \alpha_2^{N-1}\beta_2 \\ \vdots & \vdots & \ddots & \vdots \\ \beta_N & \alpha_N\beta_N & \cdots & \alpha_N^{N-1}\beta_N \end{bmatrix} \qquad \boldsymbol{A}_N = \begin{bmatrix} \alpha_1 & \cdots & \alpha_1^{N-1} & \alpha_1^N \\ \alpha_2 & \cdots & \alpha_2^{N-1} & \alpha_2^N \\ \vdots & \ddots & \vdots & \vdots \\ \alpha_N & \cdots & \alpha_N^{N-1} & \alpha_N^N \end{bmatrix}$$

The validity of (35) follows by construction, as shown by applying the expansion by minors in (32), e.g., for $n \geq 2$

$$\begin{aligned}\text{perm}\, \boldsymbol{A}_n &= \sum_{x_1=1}^{N} \alpha_{x_1} \,\text{perm}\, \boldsymbol{A}_n(x_1|1) \\ &= \sum_{x_1=1}^{N} \sum_{\substack{x_2=1\\ x_2\neq x_1}}^{N} \alpha_{x_1}\alpha_{x_2}^2 \,\text{perm}\, \boldsymbol{A}_n(x_1,x_2|1,2) = \dots\end{aligned}$$

progressively assigns one job to each position in the state vector $\boldsymbol{x}$ and, after the first n steps, it progressively assigns a job to each position in the state vector $\boldsymbol{y}$. Each assignment removes a job from further consideration in the remaining positions. Thus, the permanent eventually considers all permutations of the positions of the N jobs in the state vectors $\boldsymbol{x}$ and $\boldsymbol{y}$.

The expression in (35) shows that the cost of computing S in the model under study is bounded by the cost of computing $N+1$ permanents on matrices of order N. A popular exact method to do so is Ryser's formula, which for a matrix of order N requires at best $O(2^N(N+1))$ time if implemented by means of Gray code order [37]. A celebrated result in [38] further shows that permanents of non-negative matrices may also be approximated within a given target error using a fully polynomial-time randomized approximation scheme (FPRAS). However, practical exact and approximate methods for the permanent remain presently too slow to compute (35) in models with more than a handful of classes, e.g., $N \geq 10$.

5.3 Recurrence relation for the normalizing constant

As observed in Section 3.2.1, the state space of our models offers a recursive structure when looking at states where a given job r holds the last position at the PR queue. We now note that conditioning on the latter corresponds to selecting the element in the permanent computation in the last (right-most) column of $\boldsymbol{A}_n$ to be $\alpha_r^{N-1}\beta_r$. This is because this term corresponds to position $j=p$ in the state vector $\boldsymbol{y}$. Using this observation, we can recursively relate the normalizing constant S to the solution of models without the job of class-r, as shown next.

Theorem 3 *The normalizing constant in (8) for a model with single-job classes ($N=R$) admits the following recurrence relations*

$$V = \left(\prod_{j=1}^{N} \alpha_j\right) \sum_{r=1}^{N} V_r \tag{37}$$

$$S = V + \sum_{r=1}^{N} \alpha_r^{N-1}\beta_r S_r \tag{38}$$

where V_r and S_r are evaluated on the same model after removing a job of class r. Termination conditions are $V=S=1$ when $N=0$.

Proof Applying an expansion by minors on the last column of $\boldsymbol{A}_n$ when $p > 0$, we see that minors match the structure of the original matrix when instantiated on a model without class r. That is,

$$\text{perm}\, \boldsymbol{A}_n = \sum_{r=1}^{N} \alpha_r^{N-1} \beta_r \boldsymbol{A}_n(r|N)$$

where

$$\boldsymbol{A}_n(r|N) = \begin{bmatrix} \alpha_1 & \alpha_1^2 & \cdots & \alpha_1^n & \alpha_1^n\beta_1 & \cdots & \alpha_1^{N-2}\beta_1 \\ \alpha_2 & \alpha_2^2 & \cdots & \alpha_2^n & \alpha_2^n\beta_2 & \cdots & \alpha_2^{N-2}\beta_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \alpha_{r-1} & \alpha_{r-1}^2 & \cdots & \alpha_{r-1}^n & \alpha_{r-1}^n\beta_{r-1} & \cdots & \alpha_{r-1}^{N-2}\beta_{r-1} \\ \alpha_{r+1} & \alpha_{r+1}^2 & \cdots & \alpha_{r+1}^n & \alpha_{r+1}^n\beta_{r+1} & \cdots & \alpha_{r+1}^{N-1}\beta_{r+1} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \alpha_N & \alpha_N^2 & \cdots & \alpha_N^n & \alpha_N^n\beta_N & \cdots & \alpha_N^{N-1}\beta_N \end{bmatrix}$$

removes row r and column N from $\boldsymbol{A}_n$. The key passage is now to recognize that, by the given definitions, the latter matrix is simply

$$\boldsymbol{A}_n(r|N) = \boldsymbol{A}_{n,r} \tag{39}$$

where $\boldsymbol{A}_{n,r}$ is $\boldsymbol{A}_n$ in a model with the class-r job removed from the network. Hence

$$\text{perm}\, \boldsymbol{A}_n = \sum_{r=1}^{N} \alpha_r^{N-1} \beta_r \,\text{perm}\, \boldsymbol{A}_{n,r}$$

that implies

$$\sum_{n=0}^{N-1} \text{perm}\, \boldsymbol{A}_n = \sum_{r=1}^{N} \alpha_r^{N-1} \beta_r \sum_{n=0}^{N-1} \text{perm}\, \boldsymbol{A}_{n,r} = \sum_{r=1}^{N} \alpha_r^{N-1} \beta_r S_r$$

where the last passage follows applying (35) to the models without the class-r job for each choice of r, which have normalizing constant S_r. This proves (38) by adding to both sides of the last equation $V = \text{perm}\, \boldsymbol{A}_N$.

Lastly, (37) is verified by noting that by (34)

$$V = \text{perm}\, \boldsymbol{A}_N = \left(\prod_{j=1}^{N} \alpha_j \right) \text{perm} \begin{bmatrix} 1 & \alpha_1 & \cdots & \alpha_1^{n-1} & \alpha_1^n & \alpha_1^{n+1} & \cdots & \alpha_1^{N-1} \\ 1 & \alpha_2 & \cdots & \alpha_2^{n-1} & \alpha_2^n & \alpha_2^{n+1} & \cdots & \alpha_2^{N-1} \\ \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \alpha_k & \cdots & \alpha_k^{n-1} & \alpha_k^n & \alpha_k^{n+1} & \cdots & \alpha_k^{N-1} \\ \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \alpha_N & \cdots & \alpha_N^{n-1} & \alpha_N^n & \alpha_N^{n+1} & \cdots & \alpha_N^{N-1} \end{bmatrix}$$

and the result now follows using expansion by minors applied to the first column of the Vandermonde matrix on the right-hand side. □

We remark that that the expansion by minors (32) applied to the last column of $\boldsymbol{A}_N$ for the permanent V gives an alternative form for (37) given by $V = \sum_{r=1}^{N} \alpha_r^N V_r$; we prefer (37) as it is more consistent with later characterizations of the NP station.

Recurrence relations (37)-(38) allow to compute S in $O(N \prod_{r=1}^{N}(1 + N_r)) = O(N 2^N)$, using as termination conditions the empty model ($N = 0$), for which $S = 1$

Table 2 Illustration of the recurrence relations (37)-(38) on the running case for $N = 3$.

Class population				Results	
N_1	N_2	N_3	N	V	S
0	0	0	0	1	1
1	0	0	1	1	3
0	1	0	1	3	7
0	0	1	1	5	11
1	1	0	2	12	62
1	0	1	2	30	142
0	1	1	2	120	462
1	1	1	3	2430	17766

and $V = 1$. Although the algorithm may appear inefficient, this complexity is no different than the one required for a BCMP network solution when $R = N$ using the multiclass convolution algorithm [9] and is more efficient than a direct computation over the state space, which requires $O(N!)$ time. Indeed, the recurrence relation above is similar to dynamic programming over subsets, where complexity reductions from factorial-time enumeration to exponential-time complexity are typical [39]. We discuss later in Section 7 the much more significant computational savings arising from assuming $R < N$ and increasing the populations while keeping a fixed number of classes, which make the complexity of normalizing constant evaluation polynomial in N.

Running case. Relating back the above procedure to the running case for $N = 3$, it may be noted that the V term accounts for the contribution to the normalizing constant of the striped states in Figure 3, which are those in which the PR station is empty. Conversely, the summation in (38) considers the union of the shaded and white states, which, as discussed before, can be related to models with a class-r job less.

Assume mean service times $\alpha_k = 2k-1$ and $\beta_k = 2k$, for $k = 1, \dots, N$. We denote the population vector of each submodel considered by (38) as $\boldsymbol{N} = (N_1, N_2, N_3)$, with N_r being the number of jobs in class r. The recursive evaluation of (38) provides the values shown in Table 2. For example, comparing the table with (23) for the case $N_1 = N_2 = 1$, $N_3 = 0$, it can be immediately verified that $S = 62$. The case for $N = 3$ can be verified either using (35) or summing the unnormalized state probabilities in Appendix A.

Comparison with the BCMP convolution algorithm. The structure of (38) is appealing due to its similarity to the convolution algorithm for a two-station BCMP network [9]. To better see the similarity, we note that the BCMP product-form, when written as (3), defines itself a permutation of the N jobs. Suppose that the first station is also LCFS-PR instead than LCFS-NP, so that the BCMP theorem holds. Taking into account the different expression in (3) compared to (8), we can repeat the argument above provided that the matrix $\boldsymbol{A}_n$ is replaced by

$$\widetilde{\boldsymbol{A}}_n = \begin{bmatrix} \theta_{1,1} & \theta_{1,1} & \cdots & \theta_{1,1} & \theta_{2,1} & \theta_{2,1} & \cdots & \theta_{2,1} \\ \theta_{1,2} & \theta_{1,2} & \cdots & \theta_{1,2} & \theta_{2,2} & \theta_{2,2} & \cdots & \theta_{2,2} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \theta_{1,N} & \theta_{1,N} & \cdots & \theta_{1,N} & \theta_{2,N} & \theta_{2,N} & \cdots & \theta_{2,N} \end{bmatrix} \tag{40}$$

where the mean service demands are $\theta_{1,r} = \alpha_r$ and $\theta_{2,r} = \beta_r$, the first column is repeated n times, and the last column is repeated $N - n$ times.

With this expression, the derivation above of the recurrence relation matches the multiclass convolution algorithm of [9] for a two-station network, yielding

$$S = \widetilde{V} + \sum_{r=1}^{N} \theta_{2,r} S_r \tag{41}$$

where the permanent is now given by

$$\widetilde{V} = \operatorname{perm} \begin{bmatrix} \theta_{1,1} & \theta_{1,1} & \cdots & \theta_{1,1} \\ \theta_{1,2} & \theta_{1,2} & \cdots & \theta_{1,2} \\ \vdots & \vdots & \ddots & \vdots \\ \theta_{1,N} & \theta_{1,N} & \cdots & \theta_{1,N} \end{bmatrix} = N! \prod_{k=1}^{N} \theta_{1,k}$$

The closed-form of $\widetilde{V}$ correctly matches the BCMP product-form factor in (2) when all classes have a single job and these are all residing at the first station. Replacing the last result in (41) yields the usual expression of the convolution algorithm of [9] for two-station networks when $N = R$.

5.4 Performance metrics

We now wish to obtain expressions for commonly used mean performance metrics, namely throughputs, utilizations, responses times, and mean queue-lengths.

Throughput. In order to compute the class-r throughput $X_r(\boldsymbol{N})$, we note that when the NP station is in state x_1 then a class-r job can depart only if $x_1 = r$. We can easily identify such states resorting again to expansion. Namely, we have

$$X_r(\boldsymbol{N}) = \sum_{\substack{(x,y)\in\mathcal{E}: \\ x_1 = r}} \pi(x,y)\mu_r = \mu_r \frac{\sum_{n=1}^{N} \alpha_r^n \operatorname{perm} \boldsymbol{A}(r|n)}{S} \tag{42}$$

where we have used in the last simplification that $\alpha_r = 1/\mu_r$ and $\boldsymbol{A}(r|n)$ is the matrix obtained from $\boldsymbol{A}_n$ removing the column and row of α_r^n, i.e.,

$$\boldsymbol{A}(r|n) = \begin{bmatrix} \alpha_1 & \alpha_1^2 & \cdots & \alpha_1^{n-1} & \alpha_1^n\beta_1 & \alpha_1^{n+1}\beta_1 & \cdots & \alpha_1^{N-1}\beta_1 \\ \alpha_2 & \alpha_2^2 & \cdots & \alpha_2^{n-1} & \alpha_2^n\beta_2 & \alpha_2^{n+1}\beta_2 & \cdots & \alpha_2^{N-1}\beta_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \alpha_{r-1} & \alpha_{r-1}^2 & \cdots & \alpha_{r-1}^{n-1} & \alpha_{r-1}^n\beta_{r-1} & \alpha_{r-1}^{n+1}\beta_{r-1} & \cdots & \alpha_{r-1}^{N-1}\beta_{r-1} \\ \alpha_{r+1} & \alpha_{r+1}^2 & \cdots & \alpha_{r+1}^{n-1} & \alpha_{r+1}^n\beta_{r+1} & \alpha_{r+1}^{n+1}\beta_{r+1} & \cdots & \alpha_{r+1}^{N-1}\beta_{r+1} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \alpha_N & \alpha_N^2 & \cdots & \alpha_N^{n-1} & \alpha_N^n\beta_N & \alpha_N^{n+1}\beta_N & \cdots & \alpha_N^{N-1}\beta_N \end{bmatrix} \tag{43}$$

Thus, the computational cost to obtain the mean performance metrics remains of the same order as the cost of computing S.

It should be noted that, unlike BCMP networks, throughput in general does *not* admit, unless $N = 1$, the representation $T_r(\boldsymbol{N}) = S_r/S$, where S_r is the normalizing constant in a model without a job of class-r appearing in (41). This is because (38) conditions on the class-r job residing in the last position p at the PR station, and since our model is order-dependent the probability of the corresponding states does not match in general the probability of states where the class-r job is at the head of the station running in the PR server and can generate departures.

Utilization. Since there is a single job in each class, having computed the mean class-r throughput $X_r(\boldsymbol{N})$ for the NP station, we may readily obtain the non-preemptive NP utilization from the mean class-r throughput as $U_{\text{NP},r}(\boldsymbol{N}) = X_r(\boldsymbol{N})\alpha_r$. Using the traffic equations, it follows that the utilization of the PR station is $U_{\text{PR},r}(\boldsymbol{N}) = X_r(\boldsymbol{N})\beta_r$.

Mean queue-length. Let $Q_{\text{NP},r}(\boldsymbol{N})$ be the mean queue-length of class-r jobs at the non-preemptive NP station and similarly let $Q_{\text{PR},r}(\boldsymbol{N})$ be the mean queue-length at the PR station. Note that $Q_{\text{PR},r}(\boldsymbol{N}) = 1 - Q_{\text{NP},r}(\boldsymbol{N})$, since class-$r$ has a single job. By definition

$$Q_{\text{NP},r}(\boldsymbol{N}) = \sum_{\substack{(x,y)\in\mathcal{E}:\\ r\in x}} \pi(x,y) = \frac{\sum_{n=1}^{N} \operatorname{perm} \boldsymbol{L}_{n,r}}{S} \tag{44}$$

where $\boldsymbol{L}_{n,r}$ is a modification of matrix $\boldsymbol{A}_n$ that considers only states where the class-r job does not reside at the PR station, which can be obtained by setting the last $N-n$ columns of row r to zero. For example, if $r = 1$, this is

$$\boldsymbol{L}_{n,1} = \begin{bmatrix} \alpha_1 & \alpha_1^2 & \cdots & \alpha_1^n & 0 & 0 & \cdots & 0 \\ \alpha_2 & \alpha_2^2 & \cdots & \alpha_2^n & \alpha_2^n\beta_2 & \alpha_2^{n+1}\beta_2 & \cdots & \alpha_2^{N-1}\beta_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \alpha_N & \alpha_N^2 & \cdots & \alpha_N^n & \alpha_N^n\beta_N & \alpha_N^{n+1}\beta_N & \cdots & \alpha_N^{N-1}\beta_N \end{bmatrix} \tag{45}$$

Thus, similarly to the throughput computation, the computational cost of computing $Q_{\text{NP},r}(\boldsymbol{N})$ remains of the same order of computing S.

Response time. Let $C_{\mathrm{NP},r}(\boldsymbol{N})$ and $C_{\mathrm{PR},r}(\boldsymbol{N})$ be the mean time for a job in class-r to complete a single visit to the NP and PR stations, respectively. By Little's law, we get $C_{\mathrm{NP},r}(\boldsymbol{N}) = Q_{\mathrm{NP},r}(\boldsymbol{N})/X_r(\boldsymbol{N})$ and $C_{\mathrm{PR},r}(\boldsymbol{N}) = Q_{\mathrm{PR},r}(\boldsymbol{N})/X_r(\boldsymbol{N})$. The response time for a class-r job to complete a cycle in the network is $W_r = C_{\mathrm{NP},r}(\boldsymbol{N}) + C_{\mathrm{PR},r}(\boldsymbol{N})$.

6 Mean value analysis

6.1 Overview

In this section, we use the results obtained in Section 5 to obtain a mean-value analysis (MVA) algorithm for the model under study. We still assume throughout that $N_r = 1$ and extend later the derivations to the case $N_r > 0$ in Section 7.

However, scaling recurrence relations such as (38) to mean values requires identifying novel recurrence relations with similar quantities that arise in models with a job less. This is relatively simple in BCMP networks, since the arrival distribution seen by a class-k job matches at *every* station the stationary distribution of the *same* model with a class-k job less. As we argue later, this is not the case for LCFS networks.

In details, the MVA arrival theorem for BCMP networks [33] finds a recurrence relation for the mean queue-lengths in the form

$$Q_{i,k}(\boldsymbol{N}) = U_{i,k}(\boldsymbol{N})\,(1 + Q_i(\boldsymbol{N} - 1_k)) \tag{46}$$

where $Q_{i,k}(\boldsymbol{N})$ and $U_{i,k}(\boldsymbol{N})$ are the mean queue-length and utilization for class-k at station $i = 1, \ldots, M$, and $Q_i(\boldsymbol{N} - 1_k)$ is the total mean queue-length at the same station in a model with a class-k job less. Relevant to the present work is that, under the alternative view proposed in [32], the above expression may also be interpreted as describing the conditional mean-queue length seen by a class-k job during its residence time at station i. Although the residence time view produces the same results as the arrival theorem in BCMP networks with load-independent, this is no longer true in load-dependent models, where a conditional MVA algorithm enables working recursively with mean performance metrics, instead than with the marginal queue-length probabilities as in the classic MVA [40].

The conditional MVA approach is relevant to the LCFS networks under study due to the property that conditioning on the residence period of a class-k job in the last position at the PR station yields (37)-(38). However, this formulation poses three challenges. First, the utilization term in (46) should be replaced by the probability $B_{\mathrm{PR},k}(\boldsymbol{N})$ that the class-k job holds the last position at the PR station. Next, using $B_{\mathrm{PR},k}(\boldsymbol{N})$ appears to lose the property that we can easily relate an equation like (46) to the system throughout $X_r(\boldsymbol{N})$ by means of Little's law [41], i.e., $U_{i,r}(\boldsymbol{N}) = X_r(\boldsymbol{N})\theta_{i,r}$. Hence, in the LCFS networks under study, recurrence relations are also required for computing throughputs. Lastly, we observe that conditioning on the residence times at the PR station does not readily yield (unconditional) performance metrics at the NP station. To address the last issue, we introduce the novel notion of using the reciprocal model defined in Corollary 4 to derive recurrence relations also for the NP station. That is, we characterize recursively the NP station by studying in the proofs the PR station in the reciprocal model.

6.2 Recurrence relations for mean performance metrics

To condition on a job of a given class-k holding the last position at the PR station, we introduce the *back probabilities*

$$B_{\text{NP},k}(\boldsymbol{N}) = \sum_{x_n=k} \pi(\boldsymbol{x};\boldsymbol{y}) \qquad B_{\text{PR},k}(\boldsymbol{N}) = \sum_{y_p=k} \pi(\boldsymbol{x};\boldsymbol{y}) \tag{47}$$

for $k = 1,\ldots,N$. The back probability is also defined at the NP station to use the characteristics of the reciprocal model introduced in Section 4.1.1, where numerically the PR station behaves as the NP station in the original model. We now give expressions for the back probabilities in terms of normalizing constants.

Theorem 4 *The back probability of a class-k job at the NP and PR queues is given by the expressions*

$$B_{\text{NP},k}(\boldsymbol{N}) = \left(\prod_{j=1}^{N} \alpha_j\right) \frac{S_k}{S} \tag{48}$$

$$B_{\text{PR},k}(\boldsymbol{N}) = \alpha_k^{N-1} \beta_k \frac{S_k}{S} \tag{49}$$

for $k = 1,\ldots,N$.

A proof is given in Appendix B.1. Further characterization results for $B_{\text{NP},k}(\boldsymbol{N})$ and $B_{\text{PR},k}(\boldsymbol{N})$ are developed in Appendix B.2, which provides useful identities.

The factorization in (16) illustrates why recurrence relations such as (48) depend on a factor $\prod_{j=1}^{N} \alpha_j$ at the NP station. As these relations describe models with populations $\boldsymbol{N}$ and $\boldsymbol{N} - 1_k$, where the queue-length at the NP station is decreased by one job, the term accounts for the change in the first factor of (16).

We are now ready to formulate the main recurrence relation for the mean queue-lengths in terms of models with a class-r job less.

Theorem 5 *The mean queue-lengths admit the recurrence relations*

$$Q_{\text{NP},k}(\boldsymbol{N}) = B_{\text{NP},k}(\boldsymbol{N}) + \sum_{\substack{r=1 \\ r\neq k}}^{R} B_{\text{NP},r}(\boldsymbol{N}) Q_{\text{NP},k}(\boldsymbol{N} - 1_r) \tag{50}$$

$$Q_{\text{PR},k}(\boldsymbol{N}) = B_{\text{PR},k}(\boldsymbol{N}) + \sum_{\substack{r=1 \\ r\neq k}}^{R} B_{\text{PR},r}(\boldsymbol{N}) Q_{\text{PR},k}(\boldsymbol{N} - 1_r) \tag{51}$$

with termination conditions $Q_{\text{PR},r}(\mathbf{0}) = Q_{\text{NP},r}(\mathbf{0}) = 0$, where $\mathbf{0} = (0,\ldots,0)$, and $B_{\text{NP},r}(1_r) = \alpha_r$, $B_{\text{PR},r}(1_r) = \beta_r$, $B_{\text{NP},k}(1_r) = 0$, $B_{\text{PR},k}(1_r) = 0$, for $r \neq k$.

Proof We begin by proving (51). Note first that, similarly to a BCMP network [42], the derivatives of the normalizing constant yield mean queue-lengths, i.e.,

$$\frac{\beta_k}{S}\frac{\partial S}{\partial \beta_k} = \frac{1}{S}\sum_{\substack{(x;y)\\ k\in y}}\left(\prod_{i=1}^{n}\alpha_{x_i}^{n-i+1}\right)\left(\prod_{j=1}^{p}\alpha_{y_j}^{N-p+j-1}\beta_{y_j}\right) = Q_{\text{PR},k}(\boldsymbol{N})$$

Therefore, taking the derivative of the recurrence relation (38), and noting that V is independent of β_k, we get

$$\frac{\beta_k}{S}\frac{\partial S}{\partial \beta_k} = \frac{\beta_k}{S}\alpha_k^{N-1}S_k + \frac{\beta_k}{S}\sum_{\substack{r=1\\ r\neq k}}^{K}\alpha_r^{N-1}\beta_r\frac{\partial S_r}{\partial \beta_k}$$

$$Q_{\text{PR},k}(\boldsymbol{N}) = B_{\text{PR},k}(\boldsymbol{N}) + \sum_{\substack{r=1\\ r\neq k}}^{K} B_{\text{PR},r}(\boldsymbol{N})\frac{\beta_k}{S_r}\frac{\partial S_r}{\partial \beta_k}$$

$$Q_{\text{PR},k}(\boldsymbol{N}) = B_{\text{PR},k}(\boldsymbol{N}) + \sum_{\substack{r=1\\ r\neq k}}^{K} B_{\text{PR},r}(\boldsymbol{N})Q_{\text{PR},k}(\boldsymbol{N}-1_r)$$

Applying the formula to the reciprocal model then yields also (50) by (76), since $\bar{Q}_{\text{PR},k}(\boldsymbol{N}) = Q_{\text{NP},k}(\boldsymbol{N})$, given that $\pi(\boldsymbol{x};\boldsymbol{y}) = \bar{\pi}(\boldsymbol{y};\boldsymbol{x})$, and the same relationship holds for all populations $\boldsymbol{N}-1_r$, that is, $\bar{Q}_{\text{NP},r}(\boldsymbol{N}-1_r) = Q_{\text{PR},r}(\boldsymbol{N}-1_r)$.

Lastly, we discuss the termination conditions. The case $\boldsymbol{N}=\boldsymbol{0}$ corresponds to an empty model, thus $Q_{\text{PR},r}(\boldsymbol{0}) = Q_{\text{NP},r}(\boldsymbol{0}) = 0$. Instead, the case $\boldsymbol{N}=1_r$ corresponds to the probability distribution given in (17), for which the values of $B_{\text{NP},k}(\boldsymbol{N})$ and $B_{\text{PR},k}(\boldsymbol{N})$ equal the probability of the job being at the NP and PR station, respectively. □

To express compactly the result in the last theorem, let us define

$$Y_k(\boldsymbol{N}) = \frac{S_k}{S} \qquad B_{\text{NP},k}(\boldsymbol{N}) = \left(\prod_j \alpha_j\right)Y_k(\boldsymbol{N}) \qquad B_{\text{PR},k}(\boldsymbol{N}) = \alpha_k^{N-1}\beta_k Y_k(\boldsymbol{N})$$

Using the identities for back probabilities given in Appendix B.2, in particular plugging (72)-(74) into (50)-(51) and requiring $Q_{\text{NP},k}(\boldsymbol{N}) + Q_{\text{PR},k}(\boldsymbol{N}) = N_k = 1$, we obtain

$$Y_k(\boldsymbol{N}) = \left[\left(\prod_{j=1}^{N}\alpha_j\right)\left(1+\sum_{\substack{r=1\\ r\neq k}}^{R}\frac{\alpha_k}{\alpha_r}\frac{B_{\text{NP},r}(\boldsymbol{N}-1_k)}{B_{\text{NP},k}(\boldsymbol{N}-1_r)}Q_{\text{NP},k}(\boldsymbol{N}-1_r)\right)\right.$$
$$\left. + \alpha_k^{N-1}\beta_k\left(1+\sum_{\substack{r=1\\ r\neq k}}^{R}\frac{\alpha_r}{\alpha_k}\frac{B_{\text{PR},r}(\boldsymbol{N}-1_k)}{B_{\text{PR},k}(\boldsymbol{N}-1_r)}Q_{\text{PR},k}(\boldsymbol{N}-1_r)\right)\right]^{-1} \tag{52}$$

that readily yields $B_{\text{NP},k}(\boldsymbol{N})$ and $B_{\text{PR},k}(\boldsymbol{N})$. These probabilities then enable computing the mean queue-lengths $Q_{\text{NP},k}(\boldsymbol{N})$ and $Q_{\text{PR},k}(\boldsymbol{N})$ using Theorem 5, completing the recursive step for population $\boldsymbol{N}$. The computational costs are only slightly increased compared to a direct computation of the normalizing constant, namely $O(K\prod_{r=1}^{K}(1+N_r))$ time and space.

As the recursive evaluation in Theorem 5 proceeds, it is also possible to determine the value of the mean throughput using the next theorem.

Theorem 6 *The class-k throughput admits the recurrence relation*

$$T_k(\boldsymbol{N}) = \mu_k B_{\mathrm{NP},k}(\boldsymbol{N})\pi_{\mathrm{NP}}(0|\boldsymbol{N}-1_k) + \sum_{r=1}^{N} B_{\mathrm{NP},r}(\boldsymbol{N})T_k(\boldsymbol{N}-1_r)$$

where $\pi_{\mathrm{NP}}(0|\boldsymbol{N}-1_k)$ is the probability that the NP station is empty in a model with $\boldsymbol{N}-1_k$ jobs.

Proof As discussed in Section 5.4, computing the throughput $T_k(\boldsymbol{N})$ requires evaluating the permanent for the matrix $\boldsymbol{A}_n(k|n)$. Following a similar argument as in the proof of (37), it follows that, for $n \geq 1$

$$\mathrm{perm}\,\boldsymbol{A}_n(k|n) = \begin{cases} \left(\prod_{j\neq k}\alpha_j\right)\sum_{r=1}^{N}\mathrm{perm}\,\boldsymbol{A}_{n-1,r}(k|n-1) & \text{if } n>1 \\ \left(\prod_{j\neq k}\alpha_j\right)\mathrm{perm}\,\boldsymbol{A}_{0,k} & \text{if } n=1 \end{cases}$$

where, as before, the notation $\boldsymbol{A}_{n,r}$ indicates $\boldsymbol{A}_n$ in a model with a class-r job less. Therefore, computing the throughput at the NP station using (42) yields

$$\begin{aligned} T_k(\boldsymbol{N}) &= \mu_k \sum_{n=1}^{N} \frac{\alpha_k^n\,\mathrm{perm}\,\boldsymbol{A}_n(k|n)}{S} \\ &= \left(\prod_{j\neq k}\alpha_j\right)\frac{\mathrm{perm}\,\boldsymbol{A}_{0,k}}{S} + \mu_k\sum_{r=1}^{N}\frac{S_r}{S}\sum_{n=2}^{N}\left(\prod_{j=1}^{N}\alpha_j\right)\frac{\alpha_k^{n-1}\,\mathrm{perm}\,\boldsymbol{A}_{n-1,r}(k|n-1)}{S_r} \\ &= \mu_k B_{\mathrm{NP},k}(\boldsymbol{N})\pi_{\mathrm{NP}}(0|\boldsymbol{N}-1_k) + \sum_{r=1}^{N} B_{\mathrm{NP},r}(\boldsymbol{N})T_k(\boldsymbol{N}-1_r) \end{aligned}$$

where $\pi_{\mathrm{NP}}(0|\boldsymbol{N}-1_k)$ is the probability that the NP station is empty in a model with $\boldsymbol{N}-1_k$ jobs. □

The last result can be equivalently expressed using throughputs instead of $\pi_{\mathrm{NP}}(0|\boldsymbol{N}-1_k)$. For a model with $R=N$ classes, $\pi_{\mathrm{NP}}(0|\boldsymbol{N}-1_k) = 1 - U_{\mathrm{NP}}(\boldsymbol{N}-1_k)$, so that

$$T_k(\boldsymbol{N}) = \mu_k B_{\mathrm{NP},k}(\boldsymbol{N})\left(1 - \sum_{\substack{r=1\\ r\neq k}}^{R}\alpha_r T_r(\boldsymbol{N}-1_k)\right) + \sum_{r=1}^{N} B_{\mathrm{NP},r}(\boldsymbol{N})T_k(\boldsymbol{N}-1_r) \qquad (53)$$

Using the recurrence relations for mean queue-lengths and throughputs, all the mean performance metrics described in Section 5.4 can now be recursively obtained.

While (53) is typically sufficient to evaluate $T_k(\boldsymbol{N})$, if the NP station utilization is very close to 1.0 then the subtraction may incur round-off errors, resulting in numerical instabilities. In such cases, it may be preferable to compute recursively the $\pi_{\mathrm{NP}}(0|\boldsymbol{N}-1_k)$ term without restoring to subtractions. To this aim, we develop in Appendix C recurrence relations for marginal queue-length probabilities at each station. In particular, since $N_k = 1$, we have that

$$\pi_{\text{NP}}(0|\boldsymbol{N} - 1_k) = \sum_{\substack{r=1 \\ r \neq k}}^{N} \frac{\beta_r}{\alpha_r} B_{\text{NP},r}(\boldsymbol{N} - 1_k)\pi_{\text{NP}}(0|\boldsymbol{N} - 1_k - 1_r) \qquad (54)$$

This expression avoids subtractions and can be evaluated recursively together with the other mean-value analysis equations.

Comparison with BCMP networks. It should be noted that the recurrence relations we have derived until now are more complex than the corresponding MVA recurrence relations for BCMP networks. We argue that this is due to the dependence on the right-hand side of (50)-(51) of probability terms that differ for each class. The additional complexity of (52) does not arise in the corresponding throughput formulas for BCMP networks, since (46), unlike (50)-(51), depends only on the utilization of a single class on its right side. Nevertheless, we show in Appendix B.3 that an analogue of (52) can be derived for BCMP networks, in which the back probabilities are replaced by per-class utilizations and the $Y_k(\boldsymbol{N})$ terms correspond instead to throughputs.

7 Models with multiple jobs per class

Normally, closed queueing networks features, unlike our model, classes with more than a single job in each. As discussed in Section 3.1, if $N_r \leq N$ classes share a common set of rates (μ_r, σ_r), they are stochastically equivalent to a single class with a population of N_r jobs. In this section, we discuss how the product-form and computational methods in the previous sections change when we consider class populations with more than a single job, i.e., the case $R < N$.

7.1 Marginal distribution

Let R be the number of distinct service rate pairs in the model and assume that a pair (μ_r, σ_r) appears $N_r \geq 1$ times, $N_1 + \ldots + N_R = N$. We order classes so that the first R have all distinct pairs (μ_r, σ_r). If $R < N$, we may want to estimate the marginal probability of states $\boldsymbol{n} = (\boldsymbol{n}_{\text{NP}}, \boldsymbol{n}_{\text{PR}})$, where $\boldsymbol{n}_{\text{NP}} = (n_{\text{NP},1}, \ldots, n_{\text{NP},R})$ and $\boldsymbol{n}_{\text{PR}} = (n_{\text{PR},1}, \ldots, n_{\text{PR},R})$ count, respectively, the number of jobs having a given pair of rates (μ_r, σ_r) residing at the NP and PR stations. As before, we set the total queue-lengths as $n = \sum_{r=1}^{R} n_{\text{NP},r}$ and $p = \sum_{r=1}^{R} n_{\text{PR},r}$.

Since our models are order-dependent, we need to sum all permutations of the job positions. Let $F_{\text{NP}}(\boldsymbol{n}_{\text{NP}})$ be a matrix of order n with $n_{\text{NP},r}$ rows equal to $(\alpha_r, \alpha_r^2, \ldots, \alpha_r^n)$ and $F_{\text{PR}}(\boldsymbol{n}_{\text{PR}})$ be a matrix of order p with $n_{\text{PR},r}$ rows equal to $(\alpha_r^n\beta_r, \alpha_r^{n+1}\beta_r, \ldots, \alpha_r^{N-1}\beta_r)$. We now obtain from (8)

$$\pi(\boldsymbol{n}) = C(\boldsymbol{n}) \frac{\text{perm}\, F_{\text{NP}}(\boldsymbol{n}_{\text{NP}})\, \text{perm}\, F_{\text{PR}}(\boldsymbol{n}_{\text{PR}})}{S} \qquad \boldsymbol{n} \in \mathcal{A}$$

where $C(\boldsymbol{n}) = \prod_{r=1}^{R} \binom{N_r}{n_{\mathrm{NP},r}}$ counts the number of ways to choose station populations from the job vector $\boldsymbol{n}$, we set $\operatorname{perm} F_{\mathrm{NP}}(\mathbf{0}) = \operatorname{perm} F_{\mathrm{PR}}(\mathbf{0}) = 1$ and define

$$\mathcal{A} = \left\{ \boldsymbol{n} = (\boldsymbol{n}_{\mathrm{NP}}; \boldsymbol{n}_{\mathrm{PR}}) \,\middle|\, n_{\mathrm{NP},r} + n_{\mathrm{PR},r} = N_r, n_{\mathrm{NP},r} \geq 0, n_{\mathrm{PR},r} \geq 0 \right\} \tag{55}$$

Using that $S = G \prod_{r=1}^{R} N_r!$, this can be further simplified to

$$\pi(\boldsymbol{n}) = \left(\prod_{r=1}^{R} \frac{1}{n_{\mathrm{NP},r}! n_{\mathrm{PR},r}!} \right) \frac{\operatorname{perm} F_{\mathrm{NP}}(\boldsymbol{n}_{\mathrm{NP}}) \operatorname{perm} F_{\mathrm{PR}}(\boldsymbol{n}_{\mathrm{PR}})}{G} \qquad \boldsymbol{n} \in \mathcal{A} \tag{56}$$

Comparison with BCMP networks. The representation (56) readily extends to BCMP networks as

$$\pi(\boldsymbol{n}) = \prod_{i=1}^{M} \left(\prod_{r=1}^{R} \frac{1}{n_{i,r}!} \right) \frac{\operatorname{perm} F_i(\boldsymbol{n}_i)}{G} \qquad \boldsymbol{n} \in \mathcal{A} \tag{57}$$

In the BCMP case, for station i the permanent is computed on matrix $F_i(\boldsymbol{n}_i)$ of order $n_i = \sum_r n_{i,r}$ with $n_{i,r}$ rows equal to $(\theta_{i,r}, \ldots, \theta_{i,r})$. Thus, counting the number of permutations that give a total population vector $\boldsymbol{n}_i$ we get the term

$$\operatorname{perm} F_i(\boldsymbol{n}_i) = n_i! \prod_{r=1}^{R} \theta_{i,r}^{n_{i,r}}$$

Plugging the last expression into (57) recovers the BCMP product-form (1).

Permanents with repeated rows. In the case of repeated rows, which arise in expressions such as (56) and (35), the permanent can be computed at a reduced cost [43]. In details, Reyser's formula with repeated rows applies to an order-N matrix as follows [43]

$$\operatorname{perm} \boldsymbol{A} = (-1)^N \sum_{(u_1,u_2,\ldots,u_R)\in\mathcal{U}} (-1)^{u_1+u_2+\cdots+u_R} \prod_{r=1}^{R} \binom{N_r}{u_r} \prod_{j=1}^{N} \left(\sum_{i=1}^{R} u_i a_{i,j} \right) \tag{58}$$

where $\mathcal{U}$ is the set of of vectors $(u_1, u_2, \ldots, u_R)$ with $0 \leq u_r \leq N_r$, $a_{i,j}$ is the generic element of matrix $\boldsymbol{A}$ in row i and column j, the first R rows of $\boldsymbol{A}$ are assumed distinct and each of them appears a total of N_r times in $\boldsymbol{A}$.

Formula (58) requires $O(KR \prod_{r=1}^{R} (N_r + 1))$ time, thus lowering the total complexity of (35) to $O(RK^{R+2})$. Repeated rows therefore reduce permanent evaluations to a polynomial complexity when we keep fixed the number of distinct rows R.

7.2 Revised normalizing constant evaluation

We now remark that the recurrence relation given in (38) can be adapted to the case $R < N$ by noting that any two right-hand side terms S_r and S_s are identical whenever classes r and s have the same service rates. Thus, we may rewrite the recurrence relations in Theorem 3 for the normalizing constant $G = S \prod_r N_r!$ on the aggregate states as

$$F = \left(\prod_{j=1}^{R} \alpha_j^{N_j} \right) \sum_{r=1}^{R} F_r \tag{59}$$

$$G = F + \sum_{r=1}^{R} \alpha_r^{N-1} \beta_r G_r \tag{60}$$

where $F \equiv F(\boldsymbol{N}) = V \prod_r N_r!$, and both F_r and G_r refer to a model in which the population of class r is decreased by one job. As before, the termination conditions are $F = 1$ and $G = 1$ when $N = 0$. Equations (59)-(60) show that the complexity of the recurrence relation reduces from the $O(N!)$ of (38) to $O(N^R)$, thus becoming polynomial in the total number of jobs.

Mean performance metrics. If the model has $R < N$ distinct classes, indexed by $r = 1, \ldots, R$, then all the mean performance metrics for a group consisting of N_r identical classes can be obtained by multiplying the metric of any single-job class r by N_r. While the expressions given in Section 5.4 rely on permanents for matrices of orders N or $N-1$, when $R < N$ these matrices also feature replicated rows and thus the required permanents can again be computed efficiently by means of (58).

Moreover, we note that it is possible to compute the matrices $\boldsymbol{L}_{n,r}$ recursively. Let $L_r = \sum_{n=1}^{N} \text{perm}\, \boldsymbol{L}_{n,r}$ be the numerator in (44), so that the mean queue-lengths are $Q_{\text{NP},r}(\boldsymbol{N}) = N_r L_r / S$ and $Q_{\text{PR},r}(\boldsymbol{N}) = N_r - Q_{\text{NP},r}(\boldsymbol{N})$. From (45), we see that Y_r may be seen as the limit of the normalizing constant S in a sequence of models where $\beta_r \to 0$. Since S is a multivariate polynomial, it is sufficient to evaluate the recurrence relations (59)-(60) with $\beta_r = 0$ to obtain Y_r. Thus, mean per-class queue-lengths can be obtained by solving (59) $-$ (60) a total of $R+1$ times to obtain G and Y_r, $r = 1, \ldots, R$.

Comparison to BCMP networks. It should be noted that (59)-(60) bear similarities with the multiclass convolution algorithm of BCMP networks [9], which has the form

$$G = G_{(i)} + \sum_{r=1}^{R} \theta_{i,r} G_r \tag{61}$$

where G_r is the normalizing constant for a model with population $\boldsymbol{N} - 1_r$, while $G_{(i)}$ has population $\boldsymbol{N}$ but an arbitrary station i is removed from the network. The last term captures the contributions to G of states where station i is empty. The presence of the terms N_r in (60) that do not appear in the BCMP case is due to (4). A more fundamental difference is that the power α_r^{N-1} changes at every step of the recursion, making the summands that form S dependent on the specific sequence followed through the recursion to compute that term.

In the case with $M = 2$ stations, $G_{(i)}$ becomes the product-form factor for the BCMP network state where every job resides at station $j \neq i$. This is the same

Table 3 Comparison of runtime (in seconds) for (35), evaluated with (58), and (60). All classes have identical populations.

	Runtime of (35)			Runtime of (60)		
	$N = 2R$	$N = 5R$	$N = 10R$	$N = 2R$	$N = 5R$	$N = 10R$
$R = 2$	0.000	0.004	0.0042	0.000	0.000	0.003
$R = 3$	0.001	0.0029	0.0683	0.000	0.001	0.0018
$R = 4$	0.006	0.0425	1.2876	0.001	0.0023	0.0255

interpretation of the V term in (60), which is the permanent associated to the marginal state $\boldsymbol{n}_{\text{NP}} = \boldsymbol{N}$, where the PR station is empty.

Running case. Table 3 shows the time required to compute S for the running case when we set an identical population $N_r = N/R$ for each class $r = 1, \dots, R$. Mean service times for the R distinct classes are set to $\alpha_r = 2r - 1$ and $\beta_r = 2r$. Experiments are run in MATLAB 2025a on an Intel Core i7-14700 machine with 16GB of RAM.

The results give evidence of the computational gains achieved by the recurrence relations (59)-(60) compared to a direct numerical evaluation of the normalizing constant representation (35) with (58). The largest multiclass population we have solved with the recurrence (59)-(60) is a model with $R = 2$, $N_1 = N_2 = N/R$, $N = 500$ jobs, which requires on the same machine $79ms$. Larger populations exceed the available memory needed to store all the intermediate values of V_r and S_r in (59)-(60), which also grows as $O(N^R)$. Dynamic memory management techniques to store only the values needed for later evaluation steps may help in further increasing scalability.

7.3 Revised mean-value analysis

The recurrence relations developed for mean-value analysis in Section 6 also require minor changes. In particular, class-k performance metrics at populations $\boldsymbol{N} - 1_k$ are no longer zero if $N_k > 1$. The revised recurrence relations as follows

$$Y_k(\boldsymbol{N}) = N_k \left[\left(\prod_{j=1}^{R} \alpha_j^{N_j} \right) \left(1 + \sum_{r=1}^{R} \frac{\alpha_k}{\alpha_r} \frac{B_{\text{NP},r}(\boldsymbol{N} - 1_k)}{B_{\text{NP},k}(\boldsymbol{N} - 1_r)} Q_{\text{NP},k}(\boldsymbol{N} - 1_r) \right) \right.$$
$$\left. + \alpha_k^{N-1} \beta_k \left(1 + \sum_{r=1}^{R} \frac{\alpha_r}{\alpha_k} \frac{B_{\text{PR},r}(\boldsymbol{N} - 1_k)}{B_{\text{PR},k}(\boldsymbol{N} - 1_r)} Q_{\text{PR},k}(\boldsymbol{N} - 1_r) \right) \right]^{-1} \tag{62}$$

$$B_{\text{NP},k}(\boldsymbol{N}) = \left(\prod_{j=1}^{R} \alpha_j^{N_j} \right) Y_k(\boldsymbol{N}) \tag{63}$$

$$B_{\text{PR},k}(\boldsymbol{N}) = \alpha_k^{N-1} \beta_k Y_k(\boldsymbol{N}) \tag{64}$$

$$Q_{\text{NP},k}(\boldsymbol{N}) = B_{\text{NP},k}(\boldsymbol{N}) + \sum_{r=1}^{R} B_{\text{NP},r}(\boldsymbol{N}) Q_{\text{NP},k}(\boldsymbol{N} - 1_r) \tag{65}$$

$$Q_{\text{PR},k}(\boldsymbol{N}) = B_{\text{PR},k}(\boldsymbol{N}) + \sum_{r=1}^{R} B_{\text{PR},r}(\boldsymbol{N}) Q_{\text{PR},k}(\boldsymbol{N} - 1_r) \tag{66}$$

$$T_k(\boldsymbol{N}) = \mu_k B_{\text{NP},k}(\boldsymbol{N}) \left(1 - \sum_{r=1}^{R} \alpha_r T_r(\boldsymbol{N} - 1_k)\right) + \sum_{r=1}^{R} B_{\text{NP},r}(\boldsymbol{N}) T_k(\boldsymbol{N} - 1_r) \quad (67)$$

for all classes $k = 1, \dots, R$. As before, the last equation may be revised for increased numerical stability to use a form that does not rely on subtractions, namely

$$T_k(\boldsymbol{N}) = \mu_k B_{\text{NP},k}(\boldsymbol{N}) \pi_{\text{NP}}(0|\boldsymbol{N} - 1_k) + \sum_{r=1}^{R} B_{\text{NP},r}(\boldsymbol{N}) T_k(\boldsymbol{N} - 1_r) \quad (68)$$

$$\pi_{\text{NP}}(0|\boldsymbol{N} - 1_k) = \sum_{r=1}^{R} \frac{\beta_r}{\alpha_r} B_{\text{NP},r}(\boldsymbol{N} - 1_k) \pi_{\text{NP}}(0|\boldsymbol{N} - 1_k - 1_r) \quad (69)$$

8 Extending the product-form solution

In this section, we investigate situations where the product-form (8) is retained after relaxing the model assumptions given in Section 3.1.

8.1 Feedback loops

The product-form solution is, in general, not preserved under Markovian routing. In particular, a product-form solution no longer holds if a feedback loop is added at the LCFS-NP station with the convention that the departing job j joins back the NP queue at the top of the waiting stack, whenever the station has $n > 1$ jobs.

Adding feedback loops exclusively to the LCFS-PR station retains instead the product-form. More generally, if NP and PR use the same looping convention that the departing job immediately re-enters service, then the proof of (8) remains valid with minor modifications. Let $1 - p_r$ be the probability that a class-r job loops at the NP node after completion, and similarly let $1 - q_r$ be the probability of a feedback loop at the PR node. In this model, under the assumption of a single job per class, the product-form solution becomes

$$\pi(\boldsymbol{s}) = \pi(\boldsymbol{x}; \boldsymbol{y}) = \frac{1}{S} \left(\prod_{i=1}^{n} \gamma_{x_i}^{n-i} \theta_{\text{NP},x_i} \right) \left(\prod_{j=1}^{p} \gamma_{y_j}^{N-p+j-1} \theta_{\text{PR},y_j} \right) \quad (70)$$

where S is a normalizing constant, $v_r = q_r/(p_r + q_r)$ and $w_r = p_r/(p_r + q_r)$ are, respectively, the visit ratios of class-r jobs at the NP and PR stations, and we set

$$\theta_{\text{NP},r} = v_r \alpha_r, \quad \theta_{\text{PR},r} = w_r \beta_r, \quad \gamma_r = w_r \alpha_r \quad (71)$$

where $\theta_{\text{NP},r}$ and $\theta_{\text{PR},r}$ match the BCMP definition of service demand. In this expression, it is notable that γ_r depends on both parameters of the NP and PR station, further highlighting the difference in structure compared to a standard BCMP product-form.

8.2 Coxian service distributions

The product-form solution (2) no longer holds if Coxian distributions are used at the NP station. Conversely, it is simple to show that the product-form holds when the service distribution at the PR station is Coxian instead of exponential. This property matches the BCMP product-form, where Coxians can be used at LCFS-PR stations. In particular, the balances studied in the proof of Theorem 2 can be re-derived also in the Coxian model provided that the global balance equations are summed up for all service phases of the job in execution, while keeping fixed the station populations $\boldsymbol{x}$ and $\boldsymbol{y}$. This aggregation effectively derives the product-form distribution (8) for an equivalent model where the Coxians are replaced by exponential distributions with the same mean.

9 Numerical examples

The proposed models combine scheduling disciplines that, to the best of our knowledge, have not been jointly studied before in prior work. Typically, LCFS-PR approximates preemptive priority systems [44]. Its combination in a closed loop with LCFS-NP can therefore produce a situation where a small group of jobs at the top of the NP waiting stack continue to loop for extended periods of time at the PR station, potentially resulting in starvation for the remaining jobs. An application of the product-form networks we have studied may therefore be the analysis of starvation risks when a non-preemptive LCFS queue is connected to a priority-type system.

We illustrate starvation analysis on five model parameterizations. Classes are labelled $r = 1, \dots, R$ and we assume a single job population $N_r = 1$ in each class. In the *Balanced* parameterization ($\alpha_r = \beta_r = r$), jobs within a class are processed at equal rate at both queues and their mean service time grows linearly with the job index r. In the *Opposite* parameterization ($\alpha_r = r$, $\beta_r = R - r + 1$) jobs instead that demand a large service time at the NP queue require a small service time at the PR queue, and viceversa.

In the last parameterization, which consists of two cases, the classes receive heterogeneous times at the two stations and, for any two classes r and s, $\alpha_r > \alpha_s$ implies $\beta_r > \beta_s$. The two cases are termed *NP-heavy* and *PR-heavy* and distinguish whether the bottleneck station is the NP station ($\alpha_r = r+1$, $\beta_r = r$) or the PR station ($\alpha_r = r$, $\beta_r = r + 1$).

We vary through the experiments the number of classes, increasing them in powers of two until $R = 16$, and the results are shown in Table 4. We highlight the differences compared to BCMP networks by repeating the same experiments after changing the NP station to use LCFS-PR, so that both nodes now satisfy the assumptions of the BCMP theorem. In this new model, the NP and PR stations are renamed to PR1 and PR2, respectively, and the numerical results are shown in Table 5. We refer to this new network as the BCMP network and to the original one with the NP station as the LCFS network.

We first consider the *Balanced* experiment. In this case, the usual BCMP behavior remains unchanged also in the LCFS network upon looking at the total number of jobs or the total and per-class throughputs, which are identical in the two stations.

However, the per-class queue-lengths show a significant difference. In the LCFS network, classes with fast service times, such as class 1, spend most of the time at the NP station. Conversely, slow running classes, such as class R, spend most of their time at the PR station.

Considering the *NP-heavy* and *PR-heavy* experiments, we find marginal deviation in the total queue-lengths compared to the BCMP case. However, substantial class imbalances remain. Consistently with the *Balanced* case, fast classes spend more time at the NP queue than in a BCMP network; instead, slow classes spend more time at the PR queue. However, the most noticeable change are the variations in total throughputs compared to the BCMP case, which range from a large decrease in the *NP-heavy* case to a large increase in the *PR-heavy.* The latter is driven by the fast classes starving the slow classes, as visible from the class-1 throughput value $T_1 = 0.120$ that is achieved by simultaneously reducing the throughput of the slow class R to $T_R = 0.001$, about four-times lower than in the BCMP case. We remark that the improvements in throughputs extend beyond class-1, for example $T_2 = 0.0512$ in the LCFS network compared to $T_2 = 0.0219$ in the BCMP network; thus starvation is not always realized through a single dominating class, though it is possible to choose parameterizations that display starvation driven by a single class. The same effect is not present in the *NP-heavy* case, where however the total throughput generally degrades compared to a BCMP network.

A significant degradation in throughput compared to the BCMP case is also visible in the *Opposite* scenario, where the gap in $T = \sum_r T_r$ shows an increasing trend in larger populations. This case is also noticeable in further demonstrating the asymmetric behavior of the NP and PR stations, with total queue-lengths not achieving the same value despite $\sum_r \alpha_r = \sum_r \beta_r$. Under the same setting, a BCMP network achieves instead a symmetric placement of the jobs aross the two stations.

In conclusion, these examples highlight that LCFS networks show some similarities with BCMP networks, but also feature major differences. Two key observations are the asymmetric behavior of the NP and PR stations and that situations exist where a LCFS network can deliver higher total throughput than a BCMP network. However, gains in total throughput are achieved at the expense of a progressive starvation of the slowest-running job classes.

10 Conclusion

In this paper, we have highlighted that two-station closed queueing networks consisting of a non-preemptive LCFS station and an LCFS preemptive resume station admit a product-form solution. The solution differs from the ordinary BCMP expression along two main dimensions: order dependence and the presence of mean service times of the NP station in the product-form factor for the PR station.

Even though the family of models we have studied in this paper is not as flexible as BCMP networks when it comes to creating larger networks, it still offers a glimpse on the exact behavior of multiclass non-preemptive NP in a closed queueing network. Moreover, it emphasizes the existence and need in some queueing network models for order-dependent product-form solutions. Additionally, our results demonstrate

Table 4 Mean performance in the four scenarios. $T = \sum_r T_r$ is the total throughput; Q_{NP} and Q_{PR} are total queue-lengths.

Scenario	R	T	T_1	T_R	Q_{NP}	$Q_{\text{NP},1}$	$Q_{\text{NP},R}$	Q_{PR}	$Q_{\text{PR},1}$	$Q_{\text{PR},R}$
Balanced	2	0.500	0.333	0.167	1.000	0.556	0.444	1.000	0.444	0.556
Balanced	4	0.417	0.200	0.050	2.000	0.698	0.349	2.000	0.302	0.651
Balanced	8	0.302	0.111	0.014	4.000	0.835	0.257	4.000	0.165	0.743
Balanced	16	0.199	0.059	0.004	8.000	0.912	0.185	8.000	0.088	0.815
NP-heavy	2	0.345	0.207	0.138	1.345	0.724	0.621	0.655	0.276	0.379
NP-heavy	4	0.285	0.096	0.054	2.713	0.838	0.542	1.287	0.162	0.458
NP-heavy	8	0.189	0.032	0.019	5.424	0.946	0.432	2.576	0.054	0.568
NP-heavy	16	0.109	0.009	0.006	10.790	0.984	0.327	5.210	0.016	0.673
PR-heavy	2	0.353	0.235	0.118	0.647	0.353	0.294	1.353	0.647	0.706
PR-heavy	4	0.331	0.179	0.030	1.210	0.450	0.192	2.790	0.550	0.808
PR-heavy	8	0.295	0.145	0.006	2.192	0.562	0.106	5.808	0.438	0.894
PR-heavy	16	0.258	0.120	0.001	3.908	0.639	0.059	12.092	0.361	0.941
Opposite	2	0.444	0.222	0.222	1.000	0.444	0.556	1.000	0.556	0.444
Opposite	4	0.297	0.080	0.079	2.008	0.528	0.531	1.992	0.472	0.469
Opposite	8	0.151	0.032	0.025	4.208	0.615	0.508	3.792	0.385	0.492
Opposite	16	0.070	0.018	0.009	8.565	0.567	0.529	7.435	0.433	0.471

Table 5 Results for the four scenarios when both stations are LCFS-PR and thus the model is a BCMP network. The former NP station is now called PR1 and the PR station is renamed PR2.

Scenario	R	T	T_1	T_R	Q_{NP}	$Q_{\text{NP},1}$	$Q_{\text{NP},R}$	Q_{PR}	$Q_{\text{PR},1}$	$Q_{\text{PR},R}$
Balanced	2	0.500	0.333	0.167	1.000	0.500	0.500	1.000	0.500	0.500
Balanced	4	0.417	0.200	0.050	2.000	0.500	0.500	2.000	0.500	0.500
Balanced	8	0.302	0.111	0.014	4.000	0.500	0.500	4.000	0.500	0.500
Balanced	16	0.199	0.059	0.004	8.000	0.500	0.500	8.000	0.500	0.500
PR1-heavy	2	0.348	0.217	0.130	1.348	0.696	0.652	0.652	0.304	0.348
PR1-heavy	4	0.303	0.127	0.043	2.748	0.730	0.660	1.252	0.270	0.340
PR1-heavy	8	0.229	0.069	0.013	5.648	0.765	0.683	2.352	0.235	0.317
PR1-heavy	16	0.156	0.035	0.004	11.668	0.798	0.713	4.332	0.202	0.287
PR2-heavy	2	0.348	0.217	0.130	0.652	0.304	0.348	1.348	0.696	0.652
PR2-heavy	4	0.303	0.127	0.043	1.252	0.270	0.340	2.748	0.730	0.660
PR2-heavy	8	0.229	0.069	0.013	2.352	0.235	0.317	5.648	0.765	0.683
PR2-heavy	16	0.156	0.035	0.004	4.332	0.202	0.287	11.668	0.798	0.713
Opposite	2	0.462	0.231	0.231	1.000	0.385	0.615	1.000	0.615	0.385
Opposite	4	0.343	0.091	0.091	2.000	0.272	0.728	2.000	0.728	0.272
Opposite	8	0.214	0.030	0.030	4.000	0.166	0.834	4.000	0.834	0.166
Opposite	16	0.117	0.008	0.008	8.000	0.082	0.918	8.000	0.918	0.082

the benefits of applying permanents to computational techniques for product-form queueing networks. We have shown in particular that, using permanents, the normalizing constant of the state probabilities can be expressed in a similar fashion for both the considered family of models and for BCMP networks, requiring only simple adjustments to the matrices on which the permanents are computed.

Future work may investigate multiple directions. First, it is an open question whether the addition of specialized mechanisms to the considered class of models retains a product-form solution. These may include, among others, open classes, priorities, class switching, and blocking mechanisms. Another possible research direction relates to the novel concept of reciprocal model that may help in studying two-station queueing network models. Lastly, we point out that the present paper has only considered exact theory, but approximations may help numerical evaluations of product-form LCFS networks with large populations. Approximations that may be considered include samplers and bounds for the permanent of non-negative matrices, for instance [45, 46], which would be directly applicable to estimate the normalizing constants studied in this paper.

References

[1] Baskett, F., Chandy, K.M., Muntz, R.R., Palacios, F.G.: Open, closed, and mixed networks of queues with different classes of customers. JACM **22**, 248–260 (1975)

[2] Kelly, F.P.: Networks of queues with customers of different types. Journal of Applied Probability **12**(3), 542–554 (1975)

[3] Lavenberg, S.S.: A perspective on queueing models of computer performance. Performance evaluation **10**(1), 53–76 (1989)

[4] Balsamo, S., Persone, V.D.N., Inverardi, P.: A review on queueing network models with finite capacity queues for software architectures performance prediction. Performance Evaluation **51**(2-4), 269–288 (2003)

[5] Palunčić, F., Alfa, A.S., Maharaj, B.T., Tsimba, H.M.: Queueing models for cognitive radio networks: A survey. IEEE Access **6**, 50801–50823 (2018)

[6] Amjath, M., Kerbache, L., Elomri, A., Smith, J.M.: Queueing network models for the analysis and optimisation of material handling systems: a systematic literature review. Flexible Services and Manufacturing Journal **36**(2), 668–709 (2024)

[7] Kelly, F.P.: Reversibility and Stochastic Networks. John Wiley & Sons, New York (1979)

[8] Valiant, L.G.: The complexity of computing the permanent. Theoretical computer science **8**(2), 189–201 (1979)

[9] Reiser, M., Kobayashi, H.: Queueing networks with multiple closed chains: Theory and computational algorithms. IBM J. Res. Dev. **19**(3), 283–294 (1975)

[10] Marin, A., Rossi, S.: Aggregation and truncation of reversible markov chains modulo state renaming. In: International Conference on Analytical and Stochastic Modeling Techniques and Applications, pp. 152–165 (2017). Springer

[11] Berger, A.W., Bregman, L.M., Kogan, Y.: Bottleneck analysis in multiclass closed queueing networks and its application. Queueing Syst **31**(3-4), 217–237 (1999)

[12] McKenna, J., Mitra, D.: Asymptotic expansions and integral representations of moments of queue lengths in closed markovian networks. JACM **31**(2), 346–360 (1984)

[13] Spirn, J.R.: Queuing networks with random selection for service. IEEE Trans. Softw. Eng. **5**(3), 287–289 (1979)

[14] Noetzel, A.S.: A generalized queueing discipline for product form network solutions. Journal of the ACM (JACM) **26**(4), 779–793 (1979)

[15] Le Boudec, J.-Y.: A BCMP extension to multiserver stations with concurrent classes of customers. In: Proceedings of the 1986 ACM SIGMETRICS Joint International Conference on Computer Performance Modelling, Measurement and Evaluation, pp. 78–91 (1986)

[16] Boucherie, R.J., Van Dijk, N.M.: On the arrivai theorem for product form queueing networks with blocking. Performance Evaluation **29**(3), 155–176 (1997)

[17] Gardner, K., Righter, R.: Product forms for FCFS queueing models with arbitrary server-job compatibilities: an overview. Queueing Systems **96**(1), 3–51 (2020)

[18] Gelenbe, E., Fourneau, J.-M.: G-networks with resets. Performance Evaluation **49**(1-4), 179–191 (2002)

[19] Comte, C., Dorsman, J.-P.: Pass-and-swap queues. Queueing Systems **98**(3), 275–331 (2021)

[20] Gelenbe, E.: G-networks: a unifying model for neural and queueing networks. Annals of Operations Research **48**(5), 433–461 (1994)

[21] Nelson, R.D.: The mathematics of product form queuing networks. ACM Computing Surveys **25**(3), 339–369 (1993)

[22] Harrison, P.G.: Reversed processes, product forms and a non-product form. Linear Algebra and Its Applications **386**, 359–381 (2004)

[23] Harrison, P.G.: Turning back time in Markovian process algebra. Theoretical Computer Science **290**(3), 1947–1986 (2003)

[24] Serfozo, R.: Introduction to Stochastic Networks vol. 44. Springer, New York (2012)

[25] Kobayashi, H., Mark, B.L.: System Modeling and Analysis: Foundations of System Performance Evaluation. Pearson Prentice Hall, New Jersey (2009)

[26] Yeung, R.W., Sengupta, B.: Matrix product-form solutions for Markov chains with a tree structure. Advances in Applied Probability **26**(4), 965–987 (1994)

[27] Takine, T.: Matrix product-form solution for an LCFS-PR single-server queue with multiple arrival streams governed by a Markov chain. Queueing systems **42**(2), 131–151 (2002)

[28] Ivnitski, V.A.: Network of single-server queues with dependent service times. Queueing systems **37**(4), 363–377 (2001)

[29] Miyoshi, N.: On the Stationary LCFS-PR Single-server Queue: A Characterization Via Stochastic Intensity

[30] Shah, D., Xie, Q.: Centralized congestion control and scheduling in a datacenter. arXiv preprint arXiv:1710.02548 (2017)

[31] Balsamo, S., Marin, A.: Representing LCFSPR BCMP service center with Coxian service time by GSPN. (2010)

[32] Zahorjan, J.: The distribution of network states during residence times in product form queueing networks. Perform. Eval. **4**(2), 99–104 (1984)

[33] Reiser, M., Lavenberg, S.S.: Mean-value analysis of closed multichain queueing networks. JACM **27**(2), 312–322 (1980)

[34] Bapat, R.: Permanents in probability and statistics. Linear Algebra and its Applications **127**, 3–25 (1990)

[35] Minc, H.: Permanents. Encyclopedia of Mathematics and its Applications. Cambridge University Press, UK (1984)

[36] Buzen, J.P.: Computational algorithms for closed queueing networks with exponential servers. Comm. of the ACM **16**(9), 527–531 (1973)

[37] Savage, C.: A survey of combinatorial gray codes. SIAM review **39**(4), 605–629 (1997)

[38] Jerrum, M., Sinclair, A., Vigoda, E.: A polynomial-time approximation algorithm for the permanent of a matrix with nonnegative entries. Journal of the ACM (JACM) **51**(4), 671–697 (2004)

[39] Held, M., Karp, R.M.: A dynamic programming approach to sequencing problems. Journal of the Society for Industrial and Applied mathematics **10**(1), 196–210 (1962)

[40] Casale, G.: A note on stable flow-equivalent aggregation in closed networks. QUESTA **60**(3-4), 193–202 (2008)

[41] Little, J.D.C.: A proof of the queueing formula $L = \lambda W$. Operations Research **9**, 383–387 (1961)

[42] Silva, E., Muntz, R.R.: Simple relationships among moments of queue lengths in product form queueing networks. IEEE Trans. Comp **37**, 1125–1129 (1988)

[43] Lundow, P.H., Markström, K.: Efficient computation of permanents, with applications to boson sampling and random matrices. Journal of Computational Physics **455**, 110990 (2022)

[44] Lazowska, E.D., Zahorjan, J., Graham, G.S., Sevcik, K.C.: Quantitative System Performance. Prentice-Hall, Englewood Cliffs, NJ (1984)

[45] Huber, M., Law, J.: Fast approximation of the permanent for very dense problems. In: Proceedings of the Nineteenth Annual ACM-SIAM Symposium on Discrete Algorithms, pp. 681–689 (2008)

[46] Vontobel, P.O.: The Bethe permanent of a nonnegative matrix. IEEE Transactions on Information Theory **59**(3), 1866–1901 (2012)

A Running case solution for $N = 3$ jobs

Table 6 gives the product-form solution (8) for the running case shown in Figure 4 with $N = 3$ jobs. State vectors are read left-to-right in scheduling order, with the executing job on the left, if present. State probabilities are computed dividing the unnormalized probabilities in the table by the normalizing constant S, which is obtained by summing all the values in the last column.

B Characterization of back probabilities

B.1 Proof of Theorem 4

We first prove (49). Using that $n = N - p$ and fixing the job in the last position at the PR station in $\boldsymbol{A}_n$, we rewrite the summation on the state probabilities as

$$B_{\mathrm{PR},k}(\boldsymbol{N}) = \frac{1}{S}\sum_{n=0}^{N-1} \operatorname{perm}\begin{bmatrix} \alpha_1 & \cdots & \alpha_1^n & \alpha_1^n\beta_1 & \cdots & \alpha_1^{N-2}\beta_1 & 0 \\ \alpha_2 & \cdots & \alpha_2^n & \alpha_2^{n+1}\beta_2 & \cdots & \alpha_2^{N-2}\beta_2 & 0 \\ \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \alpha_{k-1} & \cdots & \alpha_{k-1}^n & \alpha_{k-1}^n\beta_{k-1} & \cdots & \alpha_{k-1}^{N-2}\beta_{k-1} & 0 \\ 0 & \cdots & 0 & 0 & \cdots & 0 & \alpha_k^{N-1}\beta_k \\ \alpha_{k+1} & \cdots & \alpha_{k+1}^n & \alpha_{k+1}^n\beta_{k+1} & \cdots & \alpha_{k+1}^{N-1}\beta_{k+1} & 0 \\ \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \alpha_N & \cdots & \alpha_N^n & \alpha_N^n\beta_N & \cdots & \alpha_N^{N-1}\beta_N & 0 \end{bmatrix}$$

Table 6 Running case solution with $N = 3$ for all classes r.

Node	NP State $\boldsymbol{x}$	PR State $\boldsymbol{y}$	Unnormalized Probability
1	$\emptyset$	$3, 1, 2$	$\alpha_1\alpha_2^2\beta_1\beta_2\beta_3$
2	$\emptyset$	$2, 1, 3$	$\alpha_1\alpha_3^2\beta_1\beta_2\beta_3$
3	$\emptyset$	$3, 2, 1$	$\alpha_1^2\alpha_2\beta_1\beta_2\beta_3$
4	$\emptyset$	$1, 2, 3$	$\alpha_2\alpha_3^2\beta_1\beta_2\beta_3$
5	$\emptyset$	$2, 3, 1$	$\alpha_1^2\alpha_3\beta_1\beta_2\beta_3$
6	$\emptyset$	$1, 3, 2$	$\alpha_2^2\alpha_3\beta_1\beta_2\beta_3$
7	3	$2, 1$	$\alpha_1^2\alpha_2\alpha_3\beta_1\beta_2$
8	3	$1, 2$	$\alpha_1\alpha_2^2\alpha_3\beta_1\beta_2$
9	2	$3, 1$	$\alpha_1^2\alpha_2\alpha_3\beta_1\beta_3$
10	2	$1, 3$	$\alpha_1\alpha_2\alpha_3^2\beta_1\beta_3$
11	1	$3, 2$	$\alpha_1\alpha_2^2\alpha_3\beta_2\beta_3$
12	1	$2, 3$	$\alpha_1\alpha_2\alpha_3^2\beta_2\beta_3$
13	$3, 1$	2	$\alpha_1\alpha_2^2\alpha_3^2\beta_2$
14	$2, 1$	3	$\alpha_1\alpha_2^2\alpha_3^2\beta_3$
15	$3, 2$	1	$\alpha_1^2\alpha_2\alpha_3^2\beta_1$
16	$1, 2$	3	$\alpha_1^2\alpha_2\alpha_3^2\beta_3$
17	$2, 3$	1	$\alpha_1^2\alpha_2^2\alpha_3\beta_1$
18	$1, 3$	2	$\alpha_1^2\alpha_2^2\alpha_3\beta_2$
19	$3, 1, 2$	$\emptyset$	$\alpha_1^2\alpha_2\alpha_3^3$
20	$2, 1, 3$	$\emptyset$	$\alpha_1^2\alpha_2^3\alpha_3$
21	$3, 2, 1$	$\emptyset$	$\alpha_1\alpha_2^2\alpha_3^3$
22	$1, 2, 3$	$\emptyset$	$\alpha_1^3\alpha_2^2\alpha_3$
23	$2, 3, 1$	$\emptyset$	$\alpha_1\alpha_2^3\alpha_3^2$
24	$1, 3, 2$	$\emptyset$	$\alpha_1^3\alpha_2\alpha_3^2$

We now apply expansion by minors to the last column, obtaining the derivation

$$
\begin{aligned}
B_{\mathrm{PR},k}(\boldsymbol{N}) &= \frac{\alpha_k^{N-1}\beta_k}{S}\sum_{n=0}^{N-1}\operatorname{perm}\begin{bmatrix}
\alpha_1 & \alpha_1^2 & \cdots & \alpha_1^n & \alpha_1^n\beta_1 & \cdots & \alpha_1^{N-2}\beta_1 \\
\alpha_2 & \alpha_2^2 & \cdots & \alpha_2^n & \alpha_2^n\beta_2 & \cdots & \alpha_2^{N-2}\beta_2 \\
\vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\
\alpha_{k-1} & \alpha_{k-1}^2 & \cdots & \alpha_{k-1}^n & \alpha_{k-1}^n\beta_{k-1} & \cdots & \alpha_{k-1}^{N-2}\beta_{k-1} \\
\alpha_{k+1} & \alpha_{k+1}^2 & \cdots & \alpha_{k+1}^n & \alpha_{k+1}^n\beta_{k+1} & \cdots & \alpha_{k+1}^{N-1}\beta_{k+1} \\
\vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\
\alpha_N & \alpha_N^2 & \cdots & \alpha_N^n & \alpha_N^n\beta_N & \cdots & \alpha_N^{N-1}\beta_N
\end{bmatrix} \\
&= \frac{\alpha_k^{N-1}\beta_k}{S}\sum_{n=0}^{N-1}\operatorname{perm}\boldsymbol{A}_{n,k} \\
&= \alpha_k^{N-1}\beta_k\frac{S_k}{S}
\end{aligned}
$$

To prove the expression for $B_{\text{NP},k}(\boldsymbol{N})$ we use Corollary 4. Applying (48) to a model with reciprocal parameters $\bar{\alpha}_k$ and $\bar{\beta}_k$, and using (27), we get

$$\begin{aligned}
B_{\text{NP},k}(\boldsymbol{N}) &= \overline{\alpha}_k^{N-1}\bar{\beta}_k \frac{\bar{S}_k}{\bar{S}} \\
&= \overline{\alpha}_k^{N-1}\bar{\beta}_k \frac{\prod_{\substack{j=1\\j\neq k}}^{N} \bar{\beta}_j \bar{\alpha}_j^{(N-1)}}{\prod_{s=1}^{N} \bar{\beta}_s \bar{\alpha}_s^{N}} \frac{S_k}{S} \\
&= \left(\prod_{\substack{j=1\\j\neq k}}^{N} \alpha_j\right) \bar{\alpha}_k^{N-1}\bar{\beta}_k \frac{\prod_{\substack{j=1\\j\neq k}}^{N} \bar{\beta}_j \bar{\alpha}_j^{N}}{\prod_{s=1}^{N} \varpi_s \bar{\beta}_s \bar{\alpha}_s^{N}} \frac{S_k}{S} \\
&= \left(\prod_{j=1}^{N} \alpha_j\right) \frac{S_k}{S}
\end{aligned}$$

B.2 Identities

Corollary 5 *Let $B_{\text{NP},k}(\boldsymbol{N}-1_r)$ and $B_{\text{PR},k}(\boldsymbol{N}-1_r)$ refer to a model with a class-r job less, with $r \neq k$. Then the following identities hold*

$$\frac{B_{\text{PR},k}(\boldsymbol{N})}{B_{\text{NP},k}(\boldsymbol{N})} = \frac{\alpha_k^{N-1}\beta_k}{\prod_{j=1}^{N}\alpha_j} \tag{72}$$

$$\frac{B_{\text{NP},k}(\boldsymbol{N})}{B_{\text{NP},r}(\boldsymbol{N})} = \frac{\alpha_r}{\alpha_k}\frac{B_{\text{NP},k}(\boldsymbol{N}-1_r)}{B_{\text{NP},r}(\boldsymbol{N}-1_k)} \qquad k \neq r \tag{73}$$

$$\frac{B_{\text{PR},k}(\boldsymbol{N})}{B_{\text{PR},r}(\boldsymbol{N})} = \frac{\alpha_k}{\alpha_r}\frac{B_{\text{PR},k}(\boldsymbol{N}-1_r)}{B_{\text{PR},r}(\boldsymbol{N}-1_k)} \qquad k \neq r \tag{74}$$

Proof The identity in (72) follows directly from (48) and (49), as they share the common factor S_k/S. Conversely, identity (73) follows from

$$\frac{B_{\text{NP},k}(\boldsymbol{N})}{B_{\text{NP},r}(\boldsymbol{N})} = \frac{\prod_{i=1}^{N}\alpha_i}{\prod_{j=1}^{N}\alpha_j}\frac{S_k}{S_r} = \frac{\alpha_r}{\alpha_k}\frac{\prod_{\substack{i=1\\i\neq r}}^{N}\alpha_i}{\prod_{\substack{j=1\\j\neq k}}^{N}\alpha_j}\frac{S_k}{S_{kr}}\frac{S_{kr}}{S_r} = \frac{\alpha_r}{\alpha_k}\frac{B_{\text{NP},k}(\boldsymbol{N}-1_r)}{B_{\text{NP},r}(\boldsymbol{N}-1_k)}$$

Lastly, identity (74) follows similarly to the previous case as

$$\frac{B_{\text{PR},k}(\boldsymbol{N})}{B_{\text{PR},r}(\boldsymbol{N})} = \frac{\alpha_k^{N-1}\beta_k}{\alpha_r^{N-1}\beta_r}\frac{S_k}{S_r} = \frac{\alpha_k}{\alpha_r}\frac{\alpha_k^{N-2}\beta_k}{\alpha_r^{N-2}\beta_r}\frac{S_k}{S_{kr}}\frac{S_{kr}}{S_r} = \frac{\alpha_k}{\alpha_r}\frac{B_{\text{PR},k}(\boldsymbol{N}-1_r)}{B_{\text{PR},r}(\boldsymbol{N}-1_k)}$$

□

The following statement is obtained immediately from Corollary 4 since $\pi(\boldsymbol{x};\boldsymbol{y}) = \bar{\pi}(\boldsymbol{y};\boldsymbol{x})$ and (75)-(76) are marginal probabilities.

Corollary 6 *In the reciprocal model defined in Corollary 4 with demands $\bar{\alpha}_r$ and $\bar{\beta}_r$, the probabilities of a class-k job holding the last position at the NP and PR stations are given by*

$$\bar{B}_{\text{NP},k}(\boldsymbol{N}) = B_{\text{PR},k}(\boldsymbol{N}) \tag{75}$$

$$\bar{B}_{\mathrm{PR},k}(\boldsymbol{N}) = B_{\mathrm{NP},k}(\boldsymbol{N}) \tag{76}$$

where $B_{\mathrm{NP},k}(\boldsymbol{N})$ and $B_{\mathrm{PR},k}(\boldsymbol{N})$ are the corresponding terms in the original model with mean service times α_r and β_r.

B.3 An analogy in BCMP networks

We can also show that a formula similar to (52) may also be developed for BCMP networks. Let $U_{i,r}(\boldsymbol{N})$ be the mean utilization of class-r at station i in a BCMP model with population $\boldsymbol{N}$. Using an argument similar to the proof of Theorem 5, which leverages in the BCMP case that the class-r throughput is $T_r = G_r/G = N_r S_r/S$, it is not difficult to show that the following identities hold in a BCMP network

$$\frac{U_{i,k}(\boldsymbol{N})}{U_{j,k}(\boldsymbol{N})} = \frac{\theta_{i,k}}{\theta_{j,k}} \tag{77}$$

$$\frac{U_{j,k}(\boldsymbol{N})}{U_{i,r}(\boldsymbol{N})} = \frac{\theta_{j,k}}{\theta_{i,r}}\frac{T_k(\boldsymbol{N})}{T_r(\boldsymbol{N})} = \frac{\theta_{j,k}}{\theta_{i,r}}\frac{T_k(\boldsymbol{N}-1_r)}{T_r(\boldsymbol{N}-1_k)} = \frac{U_{j,k}(\boldsymbol{N}-1_r)}{U_{i,r}(\boldsymbol{N}-1_k)} \tag{78}$$

If classes are segregated ($R = N$), using the last identity with $i = j$, we also have

$$\frac{U_{i,k}(\boldsymbol{N})}{U_{i,r}(\boldsymbol{N})} = \frac{U_{i,k}(\boldsymbol{N}-1_r)}{U_{i,r}(\boldsymbol{N}-1_k)}\frac{(1+Q_i(\boldsymbol{N}-1_r-1_k))}{(1+Q_i(\boldsymbol{N}-1_k-1_r))} = \frac{Q_{i,k}(\boldsymbol{N}-1_r)}{Q_{i,r}(\boldsymbol{N}-1_k)} \tag{79}$$

Note first that, if i is a queueing station with a scheduling policy other than IS and classes are segregated ($R = N$), then the MVA recurrence relation may be written as

$$Q_{j,k}(\boldsymbol{N}) = U_{j,k}(\boldsymbol{N}) + \sum_{\substack{r=1\\r\neq k}}^{R} U_{j,k}(\boldsymbol{N})Q_{j,r}(\boldsymbol{N}-1_k) = U_{i,k}(\boldsymbol{N})\frac{\theta_{j,k}}{\theta_{j,k}}\left(1+\sum_{\substack{r=1\\r\neq k}}^{R} Q_{j,r}(\boldsymbol{N}-1_k)\right) \tag{80}$$

where we have used that $Q_{j,k}(\boldsymbol{N}-1_k) = 0$ since $N_k = 1$. Using $\sum_j Q_{j,k}(\boldsymbol{N}) = N_k$ Lastly, we plug (79) and (78) obtaining the recurrence relation

$$T_k(\boldsymbol{N}) = \left[\sum_{j=1}^{M}\theta_{j,k}\left(1+\sum_{\substack{r=1\\r\neq k}}^{R}\frac{U_{j,r}(\boldsymbol{N}-1_k)}{U_{j,k}(\boldsymbol{N}-1_r)}Q_{j,k}(\boldsymbol{N}-1_r)\right)\right]^{-1} \tag{81}$$

from which we readily get $U_{i,k}(\boldsymbol{N}) = \theta_{i,k}X_k(\boldsymbol{N})$. After computing $U_{i,k}(\boldsymbol{N})$, the remaining mean queue-lengths $Q_{j,k}(\boldsymbol{N})$ are immediately obtained using (80). The procedure can also be repeated for other classes $r \neq k$, optionally using (78). For a cyclic network with $M = 2$ stations, the last expression shows structural similarities with (52).

C Marginal queue-length probabilities

Following a similar logic as in Section 6, we can now give a set of recurrence relations for the marginal queue-length probabilities. Let $\pi_{\text{NP}}(n|\boldsymbol{N})$ be the probability that the NP station has a queue length of n jobs in the model with population $\boldsymbol{N}$. Similarly, let $\pi_{\text{PR}}(p|\boldsymbol{N})$ be the probability that the PR station has a queue length of $p = N - n$ jobs.

Theorem 7

$$\pi_{\text{NP}}(n|\boldsymbol{N}) = \sum_{k=1}^{N} B_{\text{NP},k}(\boldsymbol{N})\pi_{\text{NP}}(n-1|\boldsymbol{N}-1_k) \qquad n \geq 1 \tag{82}$$

$$\pi_{\text{PR}}(p|\boldsymbol{N}) = \sum_{k=1}^{N} B_{\text{PR},k}(\boldsymbol{N})\pi_{\text{PR}}(p-1|\boldsymbol{N}-1_k) \qquad p \geq 1 \tag{83}$$

$$\pi_{\text{NP}}(0|\boldsymbol{N}) = \sum_{k=1}^{N} \frac{\beta_k}{\alpha_k} B_{\text{NP},k}(\boldsymbol{N})\pi_{\text{NP}}(0|\boldsymbol{N}-1_k) \tag{84}$$

$$\pi_{\text{PR}}(0|\boldsymbol{N}) = \sum_{k=1}^{N} \frac{\alpha_k}{\beta_k} B_{\text{PR},k}(\boldsymbol{N})\pi_{\text{PR}}(0|\boldsymbol{N}-1_k) \tag{85}$$

Proof

$$\text{perm}\, \boldsymbol{A}_n = \sum_{k=1}^{N} \alpha_k \,\text{perm} \begin{bmatrix} \alpha_1^2 & \cdots & \alpha_1^{n-1} & \alpha_1^n & \alpha_1^n\beta_1 & \alpha_1^{n+1}\beta_1 & \cdots & \alpha_1^{N-1}\beta_1 \\ \alpha_2^2 & \cdots & \alpha_2^{n-1} & \alpha_2^n & \alpha_2^n\beta_2 & \alpha_2^{n+1}\beta_2 & \cdots & \alpha_2^{N-1}\beta_2 \\ \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \alpha_{k-1}^2 & \cdots & \alpha_{k-1}^{n-1} & \alpha_{k-1}^n & \alpha_{k-1}^n\beta_{k-1} & \alpha_{k-1}^{n+1}\beta_{k-1} & \cdots & \alpha_{k-1}^{N-1}\beta_{k-1} \\ \alpha_{k+1}^2 & \cdots & \alpha_{k+1}^{n-1} & \alpha_{k+1}^n & \alpha_{k+1}^n\beta_{k+1} & \alpha_{k+1}^{n+1}\beta_{k+1} & \cdots & \alpha_{k+1}^{N-1}\beta_{k+1} \\ \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \alpha_N^2 & \cdots & \alpha_N^{n-1} & \alpha_N^n & \alpha_N^n\beta_N & \alpha_N^{n+1}\beta_N & \cdots & \alpha_N^{N-1}\beta_N \end{bmatrix}$$

$$= \sum_{k=1}^{N} \left(\prod_{j=1}^{N} \alpha_j\right) \text{perm} \begin{bmatrix} \alpha_1 & \cdots & \alpha_1^{n-2} & \alpha_1^{n-1} & \alpha_1^{n-1}\beta_1 & \alpha_1^n\beta_1 & \cdots & \alpha_1^{N-2}\beta_1 \\ \alpha_2 & \cdots & \alpha_2^{n-2} & \alpha_2^{n-1} & \alpha_2^{n-1}\beta_2 & \alpha_2^n\beta_2 & \cdots & \alpha_2^{N-2}\beta_2 \\ \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \alpha_{k-1} & \cdots & \alpha_{k-1}^{n-2} & \alpha_{k-1}^{n-1} & \alpha_{k-1}^{n-1}\beta_{k-1} & \alpha_{k-1}^n\beta_{k-1} & \cdots & \alpha_{k-1}^{N-2}\beta_{k-1} \\ \alpha_{k+1} & \cdots & \alpha_{k+1}^{n-2} & \alpha_{k+1}^{n-1} & \alpha_{k+1}^{n-1}\beta_{k+1} & \alpha_{k+1}^n\beta_{k+1} & \cdots & \alpha_{k+1}^{N-2}\beta_{k+1} \\ \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \alpha_N & \cdots & \alpha_N^{n-2} & \alpha_N^{n-1} & \alpha_N^{n-1}\beta_N & \alpha_N^n\beta_N & \cdots & \alpha_N^{N-2}\beta_N \end{bmatrix}$$

$$= \left(\prod_{j=1}^{N} \alpha_j\right) \sum_{k=1}^{N} \text{perm}\, \boldsymbol{A}_{n-1,k}$$

Normalizing by S proves (82), that is

$$\pi_{\text{NP}}(n|\boldsymbol{N}) = \frac{\text{perm}\, \boldsymbol{A}_n}{S} = \left(\prod_{j=1}^{N} \alpha_j\right) \sum_{k=1}^{N} \frac{S_k}{S} \frac{\text{perm}\, \boldsymbol{A}_{n-1,k}}{S_k} = \sum_{k=1}^{N} B_{\text{NP},k}(\boldsymbol{N})\pi_{\text{NP}}(n-1|\boldsymbol{N}-1_k)$$

for $n \geq 1$. To prove (85), observe that expanding by minors on the first column of $\boldsymbol{A}_0$, we get

$$\operatorname{perm} \boldsymbol{A}_0 = \sum_{k=1}^{N} \beta_k \operatorname{perm} \begin{bmatrix} \alpha_1\beta_1 & \cdots & \alpha_1^{N-1}\beta_1 \\ \alpha_2\beta_2 & \cdots & \alpha_2^{N-1}\beta_2 \\ \vdots & \ddots & \vdots \\ \alpha_{k-1}\beta_{k-1} & \cdots & \alpha_{k-1}^{N-1}\beta_{k-1} \\ \alpha_{k+1}\beta_{k+1} & \cdots & \alpha_{k+1}^{N-1}\beta_{k+1} \\ \vdots & \ddots & \vdots \\ \alpha_N\beta_N & \cdots & \alpha_N^{N-1}\beta_N \end{bmatrix} = \sum_{k=1}^{N} \frac{\beta_k}{\alpha_k} \left(\prod_{j=1}^{N} \alpha_j \right) \operatorname{perm} \boldsymbol{A}_{0,k}$$

Therefore

$$\pi_{\text{NP}}(0|\boldsymbol{N}) = \frac{\operatorname{perm} \boldsymbol{A}_0}{S} = \sum_{k=1}^{N} \frac{\beta_k}{\alpha_k} \left(\prod_{j=1}^{N} \alpha_j \right) \frac{S_k}{S} \frac{\operatorname{perm} \boldsymbol{A}_{0,k}}{S} = \sum_{k=1}^{N} \frac{\beta_k}{\alpha_k} B_{\text{NP},k}(\boldsymbol{N}) \pi_{\text{NP}}(0|\boldsymbol{N} - 1_k)$$

The relations for $\pi_{\text{PR}}(n|\boldsymbol{N})$ and $\pi_{\text{PR}}(0|\boldsymbol{N})$ now follow from the reciprocal model using Corollary 5, Corollary 6, and noting that $\bar{\pi}_{\text{NP}}(n|\boldsymbol{N}) = \pi_{\text{PR}}(n|\boldsymbol{N})$ and $\bar{\pi}_{\text{NP}}(0|\boldsymbol{N}) = \pi_{\text{PR}}(0|\boldsymbol{N})$. $\square$

The probabilities obtained in the last theorem can now be used to compute the mean queue-lengths at each station, using which (52) can be applied again to recursively obtain $B_{\text{NP},k}(\boldsymbol{N})$ and the probabilities in (72)-(74). This allows a complete evaluation of the marginal probabilities incurring a computational cost, in both time and space, a factor N larger than for the recurrence relation for mean queue-lengths.