Perfect sampling for multiclass closed queueing networks

Anne Bouillard, Ana Bušić, and Christelle Rovetta

June 4, 2015

Abstract

In this paper we present an exact sampling method for multiclass closed queuing networks. We consider networks for which stationary distribution does not necessarily have a product form. The proposed method uses a compact representation of sets of states, that is used to derive a bounding chain with significantly lower complexity of one-step transition in the coupling from the past scheme. The coupling time of this bounding chain can be larger than the coupling time of the exact chain, but it is finite in expectation. Numerical experiments show that coupling time is close to that of the exact chain. Moreover, the running time of the proposed algorithm outperforms the classical algorithm.

Keywords: multiclass queueing networks, simulation, coupling from the past

1 Introduction

Closed queueing networks are largely used in various application domains due to their modeling simplicity and product-form stationary distribution [11]. In the multiclass case, the product-form structure remains valid only under restrictive conditions on the service policy [3].

When the stationary distribution does not have a product-form, the exact analysis may not be computationally tractable, and we may turn to approximations [2, 16, 4, 18], bounds [20, 8], or simulation [1].

This paper concerns simulation, with a focus on stopping criteria. The asymptotic variance appearing in the Central Limit Theorem has been the most common metric to devise stopping rules, while mixing times have become a standard alternative [1, 15]. Unfortunately, there are no generic and tractable techniques to compute or bound either the asymptotic variance or the mixing time for non-reversible Markov chains.

Propp and Wilson introduced a method for sampling a random variable according to the stationary distribution of a finite ergodic Markov chain [17]: the coupling from the past (CFTP) algorithm. The CFTP algorithm automatically detects and stops when the sample has the correct distribution. In this way it is possible to generate i.i.d. samples from the chain, and the asymptotic variance of the resulting simulator is the standard variance of the random variable whose mean we wish to estimate.

The main drawback of the original CFTP is that it considers a coupling from all initial conditions. In the case of closed queueing networks the cardinality of the state space is exponential in the number of queues, which is intractable.

Different techniques can be used to efficiently compute one step of the CFTP algorithm: the simplest solution, for monotone Markov chains, is to compute the minimal and maximal trajectories only [17]. For Markov chains with no monotone representations, new techniques have been developed to approximate each step of the computation, at the cost of slightly increasing the number of iterations of the algorithm. Bounding chains have been constructed to detect coalescence for state spaces with lattice structure [13, 10], or for models with short range local interactions, such as interacting particle systems [12]. For applications to queueing networks, see for instance [19, 9].

The main difficulty with closed queueing networks is that the customer population is constant. This imposes a global constraint on the model, so the approach of [13] cannot be applied directly. Without monotonicity, the complexity of one iteration of the original CFTP algorithm by Propp and Wilson [17] depends on the cardinality of the state space, which is exponential in the number of queues.

For the single class closed queueing networks, Kijima and Matsui [14] proposed a perfect sampling algorithm with overall complexity $O(K^3 \ln(KM))$, where K is the number of queues and M the total number of customers. However, their method strongly relies on the product form representation of the stationary distribution and it cannot be applied to the general case of multiclass networks. In Bouillard and al. [7], a new representation of the sets of states has been proposed. This representation is used to derive a bounding chain for the CFTP algorithm for closed queueing networks, that enables exact sampling from the stationary distribution without considering all initial conditions in the CFTP. This method is far more general, as it does not rely on the product-form property.

In this paper, we propose a generalization of the compact state space representation in [7] for the multiclass closed queueing networks. Each state is represented by a path in a multidimensional diagram. This diagram is a directed graph with nodes in $\{0,\dots,K\} \times \prod_{z=1}^{Z} \{0,\dots,M_z\}$, where K denotes the number of queues, Z the number of classes, and M_z the total number of customers of class z (the detailed description of diagrams is given in Section 3). The diagram transition function used in our MDCFTP (multiclass diagram CFTP) algorithm needs to read only once this multidimensional diagram, so the complexity of one step of our algorithm is in $O(K \prod_{z=1}^{Z} M_z)$. On the other hand, multiclass diagrams are an over-representation of the states (a set of paths that represents a set of states may represent more states than just the desired subset). Thus the coupling time of the MDCFTP algorithm is in general larger than the coupling time of the classical CFTP. Numerical experiments (Section 4) suggest that the coupling times of the two algorithms are very close. Overall, the proposed MDCFTP algorithm significantly outperforms the classical one.

A major difficulty in the multiclass case is the fact that the class to be served depends on the current state of the system, so the coupling construction needs a very careful design of the event representation of the dynamic of the system. We provide such an event description under the assumption that the class to be served in queue i depends only on the current state of queue i.

The paper is organized as follows. In Section 2 we present the model and an event representation of the system that ensures the convergence of the CFTP scheme in finite expected time. In Section 3, we describe the multiclass diagram representation of the state space, propose the MDCFTP algorithm and discuss its complexity. Numerical experiments are given in Section 4. Final remarks and conclusions are contained in Section 5.

2 Model

2.1 Description

We consider a multiclass closed queueing network of K queues and Z classes of customers. Each queue $k \in \{1, \ldots, K\}$ has an infinite buffer capacity, a single exponential server and a service discipline that will be detailed in Section 2.3. For simplicity of exposition, we assume a class-independent service rate μ_k. The generalization to class-dependent service times is straightforward, by adding fake transitions that do not modify the state for the classes that have smaller service rates. The set of classes that can be in queue k is denoted by $Z(k) \subseteq \{1, \ldots, Z\}$.

Customers are not allowed to change class, thus the number of customers in each class remains constant. For each class $z \in \{1, \ldots, Z\}$ of customer, the total number of customers in this class is denoted by M_z, the set of queues visited by customers of class z by $K(z) \subseteq \{1, \ldots, K\}$ and $P^z \in \mathbb{R}^{K,K}$ is the routing matrix for class z: $P^z_{i,j}$ is the probability that a customer of class z leaving queue i is routed to queue j, with the convention that if $i \notin K(z)$, then $P^z_{i,i} = 1$ and $P^z_{i,j} = 0$ if $j \neq i$. Matrix P^z is stochastic, so for all $i, j \in \{1, \ldots, K\}$, $P^z_{i,j} \geq 0$ and $\sum_{j=1}^{K} P^z_{ij} = 1$.

We assume that the directed graph $G_z = (K(z), R_z)$ where $R_z = \{(i, j) \in K(z)^2$ such that $P^z_{ij} > 0\}$ is strongly connected. The total number of customers in the system is denoted by $M = \sum_{z=1}^{Z} M_z$.

2.2 State space

A state of the network is a matrix $\mathbf{x} = (x_{z,k}) \in \mathbb{N}^{Z \times K}$ where $x_{z,k}$ represents the number of customers of class z in queue k. A state $\mathbf{x}$ must satisfy the following constraints:

$$\sum_{k=1}^{K} x_{z,k} = M_z \quad \text{and } \forall k \notin K(z),\ x_{z,k} = 0. \tag{1}$$

Throughout the paper, we will use the following notations.

For $k \in \{1, \ldots, K\}$ and $z \in \{1, \ldots, Z\}$, we denote by:

- $\mathbf{x}_{z,*} = (x_{z,k}, \ldots, x_{z,K}) \in \mathbb{N}^K$ is the queue repartition (row)-vector of class z;
- $\mathbf{x}_{*,k} = (x_{1,k}, \ldots, x_{Z,k})^t \in \mathbb{N}^Z$ is the class repartition (row)-vector in queue k;
- $|\mathbf{x}_{*,k}| = \sum_{z=1}^{Z} x_{z,k}$ is the total number of customers in queue k.

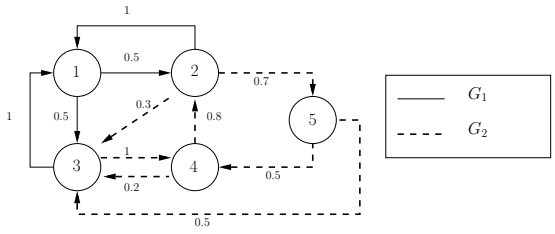

Figure 1: The multiclass network described in Example 1.

The set of all possible state matrices is denoted by $\mathcal{S} = \{\mathbf{x} \in \mathbb{N}^{Z\times K} \text{ satisfying (1)}\}$. Its cardinality is

$$|\mathcal{S}| = \prod_{z=1}^{Z} \binom{M_z}{M_z + |K(z)| - 1}.$$

If for each class z, $K(z) \ll M_z$ then $|\mathcal{S}| = O(\prod_{z=1}^{Z} M_z^{|K(z)|}) = O(M_\times^K)$, where $M_\times = \prod_{z=1}^{Z} M_z$.

Example 1 *Consider the multiclass queueing network of Figure 1 having* 5 *queues,* 2 *classes, with* $M_1 = 2$ *and* $M_2 = 3$*, and routing matrices*

$$P^1 = \begin{pmatrix} 0 & 0.5 & 0.5 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \quad \textit{and} \quad P^2 = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.3 & 0 & 0.7 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0.8 & 0.2 & 0 & 0 \\ 0 & 0 & 0.5 & 0.5 & 0 \end{pmatrix}.$$

The total number of customers is $M = 5$ *and the queues visited by each class are* $K(1) = \{1, 2, 3\}$ *and* $K(2) = \{2, 3, 4, 5\}$*. The cardinality of the state space is* $|\mathcal{S}| = 120$ *and an example of such a state is*

$$\mathbf{x} = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 1 \end{pmatrix} \in \mathcal{S}.$$

2.3 Service discipline and transitions

In order to perform the perfect sampling algorithm for multiclass queueing networks, we need to define transitions on sets of states. These transitions depend on the service discipline of the queues.

The function $\widetilde{t}_{i,j,z} : \mathcal{S} \to \mathcal{S}$ describes the routing of a class z customer from queue i to queue j:

$$\widetilde{t}_{i,j,z}(\mathbf{x}) = \mathbf{x} - \mathbb{1}_{\{x_{z,i}>0\}}\mathbf{e}_{zi} + \mathbb{1}_{\{x_{z,i}>0\}}\mathbf{e}_{zj},$$

where $\mathbf{e}_{zk} \in \mathbb{N}^{Z\times K}$ is the matrix having all its coefficients equal to 0 except $(\mathbf{e}_{zk})_{z,k} = 1$.

When there are several classes of customers in a queue, then the queue discipline determines the class of customers to serve. For each queue $i \in \{1, \ldots, K\}$, $f_i : \mathbb{N}^Z \times [0, 1] \to Z(i) \cup \{0\}$ is the function that describes the discipline in queue i. We assume that f_i has the following properties:

Assumption 1 *For a state* $\mathbf{x}$ *and a parameter* $\theta \in [0,1]$*:*

1. *The service discipline is Markovian and* f_i *only depends on* θ *and* $\mathbf{x}_{*,i}$*.*
2. $|\mathbf{x}_{*,i}| = 0$ *if and only if* $f_i(\mathbf{x}_{*,i}, \theta) = 0$ *(the service is greedy).*
3. *If* $|\mathbf{x}_{*,i}| > 0$ *then* $f_i(\mathbf{x}_{*,i}, \theta) \in \{z$ *such that* $x_{z,i} > 0\}$ *(there is a customer of the chosen class in queue* i*).*

We give two examples of disciplines that satisfy Assumption 1:

- PRIORITY gives the preemptive priority to the class with the smallest index. It can be defined to any total order $\preceq$ on the classes.

$$f_i(\mathbf{x}_{*,i}, \theta) = \min\{z \mid x_{z,i} > 0\}\mathbb{1}_{\{|\mathbf{x}_{*,i}|>0\}}.$$

- LONGEST (serve-the-longest-queue)

$$f_i(\mathbf{x}_{*,i}, \theta) \in \arg\max\{x_{i,z} \mid z \in Z(i)\} \cup \{0\}.$$

PRIORITY does not depend on θ, but this parameter is used to break ties in LONGEST between the classes where the number of customers is maximal.

For $i \in \{1, \dots, K\}$, $J = (j_1, \dots, j_Z) \in \{1, \dots, K\}^Z$ a vector of queues, $\mathbf{x}$ a state, and $\theta \in [0,1]$, we define a transition on state $\mathbf{x}$ by the function:

$$t_{i,J,\theta}(\mathbf{x}) = \widetilde{t}_{i,j_z,z}(\mathbf{x}),$$

where $z = f_i(\mathbf{x}_{*,i}, \theta)$ is the class chosen by queue i. This transition describes the routing of a customer from queue i and J gives the destination queue according to the class that is served in queue i. If the transition does not depend on θ (as in PRIORITY), we will write $t_{i,J,\theta}(\mathbf{x}) = t_{i,J}(\mathbf{x})$ to alleviate the notations.

Example 2 *Consider state* $\mathbf{x} = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 1 \end{pmatrix}$ *in Example 1. Queue* 2 *contains* 3 *customers (*$\mathbf{x}_{*,2} = (1,2)$*) and function* $t_{2,(1,5),\theta}$ *describes a possible routing for a customer in queue* 2*: if a customer of class 1 is served, then it is routed to queue* 1*; a customer of class 2 would be routed to queue* 5*. For* $\theta = 0$*, the class of the served customer is given by* $z = f_2(\mathbf{x}_{*,2}, 0)$*.*

- *If queue* 2 *has the PRIORITY discipline then* $z = 1$ *and*

$$t_{2,(1,5)}(\mathbf{x}) = \widetilde{t}_{2,1,1}(\mathbf{x}) = \begin{pmatrix} 2 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 1 \end{pmatrix}.$$

- *If queue* 2 *has the LONGEST discipline then* $z = 2$ *and*

$$t_{2,(1,5),0}(\mathbf{x}) = \widetilde{t}_{2,5,2}(\mathbf{x}) = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 2 \end{pmatrix}.$$

This definition of a transition enables to define it for sets of states: for all $S \subseteq \mathcal{S}$, $\theta \in [0,1]$, $i \in \{1, \dots, K\}$ and $J = (j_1, \dots, j_Z) \in \{1, \dots, K\}^Z$, $t_{i,J,\theta}(S) := \underset{\mathbf{x} \in S}{\cup} t_{i,J,\theta}(\mathbf{x})$ and $t_{i,J}(S) := \underset{\mathbf{x} \in S}{\cup} t_{i,J}(\mathbf{x})$ if the transition do not depend on θ.

2.4 Markov chain and perfect sampling

Denote by $(W_n)_{n\in\mathbb{N}}$ an i.i.d. sequence of random variables with distribution

$$\mathbb{P}(W_n = (i, J)) = \frac{\mu_i}{\sum_{k=1}^{K} \mu_k} \prod_{z=1}^{Z} P^z_{i,j_z},$$

where $i \in \{1, \ldots, K\}$ and $J = (j_1, \ldots, j_Z) \in \{1, \ldots, K\}^Z$. It can easily be checked that the probability that the first component of W_n is i equals $\frac{\mu_i}{\sum_{k=1}^{K} \mu_k}$, the probability that the next service occurs at queue i. Moreover, denoting $W_n = (k, J)$, for each class $z \in \{1, \ldots, K\}$, $\mathbb{P}(j_z = j \mid k = i) = P^z_{i,j}$, which is the probability of routing a customer of class z from queue i to queue j.

Let $(\Theta_n)_{n\in\mathbb{N}}$ be an i.i.d sequence of random variables, uniformly distributed on $[0, 1]$ and independent of $(W_n)_{n\in\mathbb{N}}$. We set $(U_n)_{n\in\mathbb{N}} = (W_n, \Theta_n)_{n\in\mathbb{N}}$.

Let $(X_n)_n$ a random sequence such that $X_0 \in \mathcal{S}$ and

$$X_{n+1} = t_{U_n}(X_n).$$

This equation describes the Markov chain of our model of multiclass network. This Markov chain is ergodic, due to the assumption that the routing graph of each class is strongly connected and Assumption 1. Our objective is to sample the stationary distribution of $(X_n)_{n\in\mathbb{N}}$ with the perfect sampling technique [17][1]. The following theorem concerns the termination and correctness of Algorithm

Algorithm 1: CFTP using sets of states

Data: $(U_{-n} = (i_{-n}, J_{-n}, \Theta_{-n}))_{n\in\mathbb{N}}$ an i.i.d sequence of r.v
Result: $\mathbf{x} \in \mathcal{S}$

```
begin
    n ← 1;
    t ← t_{U_{-1}};
    while |t(S)| ≠ 1 do
        n ← 2n;
        t ← t_{U_{-1}} ∘ ··· ∘ t_{U_{-n}};
    return x, the unique element of t(S)
```

1.

Theorem 1 *If for each class z the routing graph G_z is strongly connected, and the service policies satisfy Assumption 1, then Algorithm 1 terminates in finite time with probability* 1 *and the termination time has a finite expectation. The state s_0 is distributed according to the stationary distribution of $\{X_n\}$.*

The proof is a straightforward corollary of the result by Propp and Wilson [17] and Theorem 2, that gives the existence of a coupling sequence, i.e. a sequence that leads to coalescence of the trajectories started in all the initial conditions.

[1] Algorithm 1 is the variant of the CFTP algorithm from [17] that doubles the value of n at each iteration. In the case when all the states are considered, the basic CFTP algorithm can be used, but this choice has been made to simplify comparison with Algorithm 3 in Section 3.

Theorem 2 *If the service discipline of each queue satisfies Assumptions 1 then there exists a finite sequence of transitions* $t = t_{U_1} \circ \cdots \circ t_{U_n}$ *such that* $|t(\mathcal{S})| = 1$.

The proof of Theorem 2 can be found in the extended version of this paper [5]. Note that the complexity of this algorithm is at least linear in $|\mathcal{S}|$.

3 Diagram representation

The cardinality of the state space is exponential in K and Z, so it is not possible to perform the perfect sampling algorithm directly, as one must first enumerate all the state space in order to compute a transition. In this section, we present multiclass *diagrams*, a more compact way to describe sets of states. In most of the cases, a diagram representing a given set of states will in fact represent more states, but we show that this representation ensures the termination of the perfect sampling algorithm in finite expected time, and the complexity of the transition becomes linear in K (while still exponential in Z).

3.1 Definition

Let $D = (N, A)$ be a directed graph where $N \subseteq \{0, \ldots, K\} \times \mathbb{N}^Z$ is the set of nodes and A the set of arcs. Let $g : \mathcal{S} \to \mathcal{P}(N^2)$ denote the function which associates a set of arcs to a state $\mathbf{x} \in \mathcal{S}$:

$$g(\mathbf{x}) = \bigcup_{i=1}^{K} \Big\{ \big([i-1, (\sum_{k=1}^{i-1} x_{1,k}, \ldots, \sum_{k=1}^{i-1} x_{Z,k})], [i, (\sum_{k=1}^{i} x_{1,k}, \ldots, \sum_{k=1}^{i} x_{Z,k})]\big) \Big\}.$$

Graphically, $g(\mathbf{x})$ can be seen in D as a path from node $[0, (0, \ldots, 0)]$ to node $[K, (M_1, \ldots, M_Z)]$ (Figure 2). Moreover, consider an arc $a = \big([k-1, \mathbf{s}], [k, \mathbf{d}]\big) \in g(\mathbf{x})$ where $\mathbf{s} = (s_1, \ldots, s_Z) \in \mathbb{N}^Z$ and $\mathbf{d} = (d_1, \ldots, d_Z) \in \mathbb{N}^Z$ are two row-vectors. The *slope* of a on its second component can be considered as a class-repartition vector in queue k. Indeed,

$$\mathbf{d} - \mathbf{s} = (\sum_{i=1}^{k} x_{1,i} - \sum_{i=1}^{k-1} x_{1,i}, \ldots, \sum_{i=1}^{k} x_{Z,i} - \sum_{i=1}^{k-1} x_{Z,i}) = (x_{1,k}, \ldots, x_{Z,k}) = \mathbf{x}_{*,k}.$$

Definition 1 *A directed graph* $D = (N, A)$ *is called a* ***diagram*** *if there exists* $S \subseteq \mathcal{S}$ *such that*

$$A = g(S) := \bigcup_{\mathbf{x} \in S} g(\mathbf{x}).$$

A diagram is said to be ***complete*** *if* $A = g(\mathcal{S})$. *It is denoted* $\mathcal{D} = (N, \mathcal{A})$.

For an arc $a = \big([k-1, \mathbf{s}], [k, \mathbf{d}]\big) \in A$, vector $v(a) = \mathbf{d} - \mathbf{s} \in \mathbb{N}^Z$ is called the *value* of a. It represents the class-repartition vector in queue k. Subset $A_k = \{\big([k-1, \mathbf{s}], [k, \mathbf{d}]\big) \in A\}$ denotes the set of all arcs in column k.

Example 3 *Consider* $\mathcal{S}$ *the state space of Example 1 and state* $\mathbf{x} = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 1 \end{pmatrix}$. *Diagram* $D = (N, g(\{\mathbf{x}\}))$ *is given in Figure 2 and has* 5 *arcs:*

$$\big([0, (0,0)], [1, (1,0)]\big), \big([1, (1,0)], [2, (2,2)]\big), \big([2, (2,2)], [3, (2,2)]\big),$$
$$\big([3, (2,2)], [4, (2,2)]\big), \big([4, (2,2)], [5, (2,3)]\big).$$

The value of a_2 is $v(a_2) = (1, 2) = \mathbf{x}_{*,2}$. The complete diagram $D = (N, g(\mathcal{S}))$ is depicted on Figure 3 and has $|g(\mathcal{S})| = 71$ arcs.

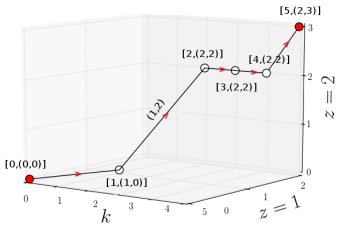

Figure 2: Diagram $D = (N, g(\{\mathbf{x}\}))$.

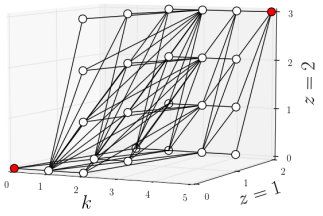

Figure 3: Complete diagram $D = (N, g(\mathcal{S}))$.

Lemma 1 *Let $D = (N, A)$ be a diagram. If $K \geq 2$, then*

$$|A| \leq 2 \prod_{z=1}^{Z} (M_z + 1) + (K - 2) \prod_{z=1}^{Z} \frac{(M_z + 1)(M_z + 2)}{2}.$$

Proof 1 (Sketch of proof) *The first term bounds the number of arcs in the first and last columns ($\leq M_z + 1$ each), the second term bounds the number of arcs in the other $K - 2$ columns ($\leq \prod_{z=1}^{Z} \sum_{m=0}^{M_z} m$).*

The equality in Lemma 1 holds for the complete diagram when each class can visit every queue.

A consequence of Lemma 1 is that the space needed for the representation of a diagram is $|A| = O(KM_\times^2)$.

Diagrams and sets of states. In order to perform the transition on the diagrams, we first need to define functions that transform a set of states into a diagram and the reverse. For $S \subseteq \mathcal{S}$, ϕ associates to a set of states $S \in \mathcal{S}$ the diagram $\phi(S) = (N, g(S))$. For $D = (N, A)$, ψ transforms diagram $D = (N, A)$ into the largest set of states $S \subseteq \mathcal{S}$ such that $g(S) = A$:

$$\psi(D) = \bigcup_{S \subseteq \mathcal{S},\ A = g(S)} S.$$

The followings properties are straightforward from the definitions of ϕ and ψ.

Lemma 2 *For $S \subseteq \mathcal{S}$ a set of states and $D = (N, A)$ a diagram:*

1. *D is complete if and only if $\psi(D) = \mathcal{S}$.*
2. *If D contains only one path (i.e. $|A| = K$) then $|\psi(D)| = 1$.*
3. *If $|S| = 1$ then $\phi(S)$ contains only one path.*
4. *If $S \subseteq \mathcal{S}$ such that $A = g(S)$ then $S \subseteq \psi(D)$.*

3.2 Transition algorithm

We now extend the transitions to diagrams. Let $(i, J) \in \{1, \ldots, K\}^{Z+1}$ and $\theta \in [0, 1]$. Function $T_{i,J,\theta}$ is defined for each diagram D as

$$T_{i,J,\theta}(D) = \phi \circ t_{i,J,\theta} \circ \psi(D).$$

Lemma 3 *Let $S \subseteq \mathcal{S}$ be a set of states and D be a diagram. For all $(i, J) \in \{1, \ldots, K\}^{Z+1}$ and $\theta \in [0, 1]$,*

1. *if $S \subseteq \psi(D)$ then $t_{i,J,\theta}(S) \subseteq \psi(T_{i,J,\theta}(D))$;*
2. *if $|\psi(D)| = 1$ then $|\psi(T_{i,J,\theta}(D))| = 1$.*

We now present the algorithm to compute $T_{i,J,\theta}(D)$ directly without having to use $t_{i,J,\theta}(\phi(D))$, which would be too costly. The intuition is that the transformation will be similar for sets of paths, and then it can be done simultaneously for them all. The algorithm that computes $T_{i,j,\theta}(D)$ is given as Algorithm 2.

Before describing it, we adapt the definition of the class to be served to diagrams. As the service disciplines we consider satisfy Assumption 1, the way a path is transformed according to (i, J, θ) depends only on the value of the arc in column i. For $a \in A_i$ and $\theta \in [0, 1]$, we can define $F_i(a, \theta) = f_i(v(a), \theta)$, which selects the class to be served for arc a. This class will be the same for every path going through that arc.

For a diagram $D = (N, A)$ and $b \in A$, we denote by $\mathcal{P}\text{aths}(b, A) \subseteq A$ the subset of arcs on paths through arc b:

$$\mathcal{P}\text{aths}(b, A) = \{a \in A \mid \exists \mathbf{x} \in \mathcal{S} \text{ s.t. } a \in g(\mathbf{x}) \text{ AND } b \in g(\mathbf{x})\},$$

and for $B \subseteq A$, $\mathcal{P}\text{aths}(B, A)$ denotes the subset of arcs on paths through an arc $b \in B$:

$$\mathcal{P}\text{aths}(B, A) = \bigcup_{b \in B} \mathcal{P}\text{aths}(b, A).$$

Example 4 *Consider $D = (N, g(\mathcal{S}))$ the complete diagram of Example 3 and $a = \big([2, (2, 0)], [3, (2, 0)]\big) \in A$, $b = \big([2, (1, 0)], [3, (2, 0)]\big) \in A$ two arcs in column 3. For $B = \{a, b\} \subseteq A_k$. Subset $\mathcal{P}\text{aths}(b, A)$ is given in Figure 4 and subset $\mathcal{P}\text{aths}(B, A)$ in Figure 5.*

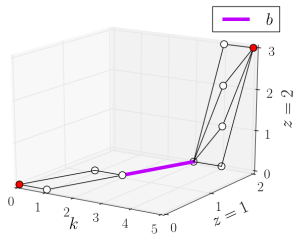

Figure 4: Subset $\mathcal{P}\text{aths}(b, A)$

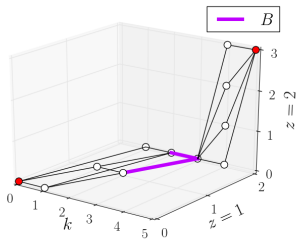

Figure 5: Subset $\mathcal{P}\text{aths}(B, A)$

For each arc $a \in A_i$, function F_i gives the class for which the transition will be performed. As there are $|Z(i)|+1$ different possible values for $F_i(a,\theta)$, we will compute $|Z(i)|+1$ different types of transition. For all $z \in Z(i) \cup \{0\}$, set

$$P[z] = \{a \in A_i \mid F_i(a,\theta) = z\} \quad \text{and} \quad \mathcal{S}\text{erve}[z] = \mathcal{P}\text{aths}(P[z], A).$$

$\mathcal{S}$erve[0] corresponds to the sub-diagram of the states where queue i is empty. It will remain a sub-diagram of $T_{i,J,\theta}(D)$. For all $z \in Z(i)$, $\mathcal{S}$erve[z] is the sub-diagram corresponding to states where a customer of class z is served, and routed to queue $j = J[z]$. This sub-diagram is transformed into $\mathcal{S}$erve$'$[z]. We focus on the case $i < j$ (the case $i > j$ is similar). Each arc $b = \big([k-1, \mathbf{s}], [k, \mathbf{d}]\big) \in \mathcal{S}\text{erve}[z]$ is transformed into c the following way:

- If $k < i$ or $k > j$, b is not affected by the transformation, so $c = b$.
- If $k = i$ then b corresponds to a class-repartition vector of queue i, where one customer of class z is served. Then $c = \big([k-1, \mathbf{s}], [k, \mathbf{d} - \mathbf{e}_z]\big)$.
- If $i < k < j$ then b corresponds a class-repartition vector that is not affected by the service. However, the origin of the arcs it was connected to has changed from $\mathbf{s}$ to $\mathbf{s} - \mathbf{e}_z$, so its destination must change similarly. So $c = \big([k-1, \mathbf{s} - \mathbf{e}_z], [k, \mathbf{d} - \mathbf{e}_z]\big)$.
- If $k = j$, b corresponds to a class-repartition vector of queue j, where one customer of class z arrives. As the origin of the arc it was connected to has changed to $\mathbf{s} - \mathbf{e}_z$, we have $c = \big([k-1, \mathbf{s} - \mathbf{e}_z], [k, \mathbf{d}]\big)$.

Diagram $T_{i,J,\theta}(D) = (N, A')$ with $A' = \bigcup_{z \in Z(i) \cup \{0\}} \mathcal{S}\text{erve}'[z]$. Indeed, this construction ensures that for all $\mathbf{x} \in \psi(D)$, $t_{iJ,\theta}(\mathbf{x}) \in \psi(T_{i,J,\theta}(D))$.

Suppose that the complexity of computing $F_i(a,\theta)$ is C. Then the complexity of Algorithm 2 is $C|A_i| + Z|A| = O((C + KZ)M_\times^2)$. For PRIORITY and LONGEST service discipline, $C = O(Z)$, so the overall complexity is $O(KZM^{2Z})$.

Example 5 *Consider the complete diagram of Example 3 and transition* $T_{2,(1,5)}$. *Suppose that queue* 2 *has a PRIORITY discipline (class 1 has the priority). Figure 6 illustrates the partition of arcs in column 2 into* $P[0]$, $P[1]$ *and* $P[2]$.

- $P[0] = \{a \in A_2 \mid v(a) = (0,0)\}$ *and* $\mathcal{S}\text{erve}[0] = \mathcal{P}\text{aths}(P[0], A)$
- $P[1] = \{a \in A_2 \mid v(a) = (v1, v2),\ v1 > 0\}$ *and* $\mathcal{S}\text{erve}[1] = \mathcal{P}\text{aths}(P[1], A)$
- $P[2] = \{a \in A_2 \mid v(a) = (0, v2),\ v2 > 0\}$ *and* $\mathcal{S}\text{erve}[2] = \mathcal{P}\text{aths}(P[2], A)$

Notice that sets $\mathcal{S}$erve[0], $\mathcal{S}$erve[1] *and* $\mathcal{S}$erve[2] *are not necessary disjoint. For example, arc* $b = \big([4,2,3], [5,2,3]\big) \in \mathcal{S}\text{erve}[1] \cap \mathcal{S}\text{erve}[2]$ *(see Figures 7 and 8). Then for all* $z \in \{1, \ldots, Z\}$ *we will compute* $\mathcal{S}$erve$'$[z] *from* $\mathcal{S}$erve[z] *according to Algorithm 2. For class 1, the transition is performed for arcs in* $\mathcal{S}$erve[1] *from queue 2 to queue 1. For class 2, the transition is performed for arcs in* $\mathcal{S}$erve[2] *from queue 2 to queue 5.*

Finaly, $T_{i,J}(D, \theta) = (N, A')$, *with* $A' = \mathcal{S}\text{erve}[0] \cup \mathcal{S}\text{erve}'[1] \cup \mathcal{S}\text{erve}'[2]$.

Algorithm 2: Algorithm $T_{i,J,\theta}$

Data: $D = (N, A)$, $i \in \{1, \ldots, K\}$, $J \in \mathbb{N}^Z$, $\theta \in [0, 1]$
Result: $T_{i,J,\theta}(D)$

begin
  for $z = 0$ *to* Z **do** $P[z] \leftarrow \{a \in A_i \mid F_i(a, \theta) = z\}$;
  $\mathcal{S}\text{erve}'[0] \leftarrow \mathcal{P}\text{aths}(P[0], A)$;
  for $z = 1$ *to* Z **do**
    $\mathcal{S}\text{erve}[z] \leftarrow \mathcal{P}\text{aths}(P[z], A)$;
    $\mathcal{S}\text{erve}'[z] \leftarrow \emptyset$;
    $j \leftarrow J[z]$;
    if $i < j$ **then**
      foreach $b = \big([k-1, \boldsymbol{s}], [k, \boldsymbol{d}]\big) \in \mathcal{S}\text{erve}[z]$ **do**
        $c = \big([k-1, \mathbf{s} - \mathbb{1}_{\{i<k\leq j\}}\mathbf{e}_z], [k, \mathbf{d} - \mathbb{1}_{\{i\leq k<j\}}\mathbf{e}_z]\big)$;
        $\mathcal{S}\text{erve}'[z] \leftarrow \mathcal{S}\text{erve}'[z] \cup \{c\}$;
    if $i > j$ **then**
      foreach $b = \big([k-1, \boldsymbol{s}], [k, \boldsymbol{d}]\big) \in \mathcal{S}\text{erve}[z]$ **do**
        $c = \big([k-1, \mathbf{s} + \mathbb{1}_{\{j<k\leq i\}}\mathbf{e}_z], [k, \mathbf{d} + \mathbb{1}_{\{j\leq k<i\}}\mathbf{e}_z]\big)$;
        $\mathcal{S}\text{erve}'[z] \leftarrow \mathcal{S}\text{erve}'[z] \cup \{c\}$;
  $A' \leftarrow \bigcup_{z=0}^{Z} \mathcal{S}\text{erve}'[z]$;
  return (N, A')

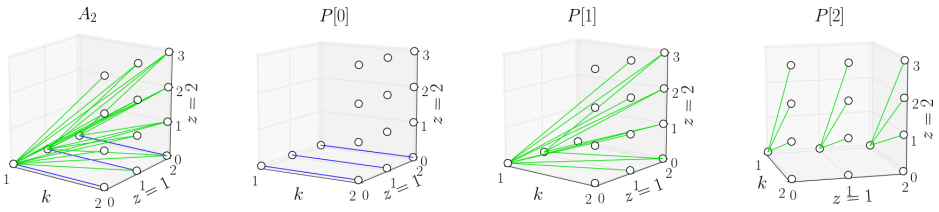

Figure 6: Arc repartition in column 2 with PRIORITY discipline.

3.3 Perfect sampling with diagrams

In this section, we show that the diagram representation can be used in the perfect sampling algorithm. First, as in Section 2.4, we need to ensure that the complete diagram can be reduced to a diagram containing only one state by a finite sequence of transitions.

Theorem 3 *If each queue discipline satisfies Assumptions 1 then there exists a finite sequence of transitions T such that $|\psi(T(\mathcal{D})| = 1$.*

The proof can be found in the extended version of this paper [5].

Theorem 4 *Algorithm 3 samples a state according to the stationary distribution on the states and terminates in finite expected time.*

Proof 2 *Theorem 3 implies that Algorithm 3 ends in finite expected time. Let $N < \infty$ be the value of n when Algorithm 3 ends. Consider $t = t_{U_{-1}} \circ \cdots \circ$*

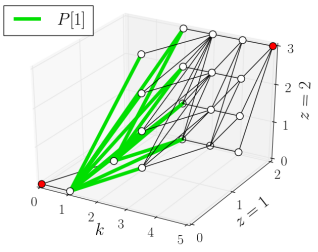

Figure 7: $\mathcal{S}\text{erve}[1] = \mathcal{P}\text{aths}(P[1], A)$.

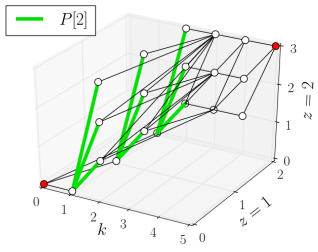

Figure 8: $\mathcal{S}\text{erve}[2] = \mathcal{P}\text{aths}(P[2], A)$.

Algorithm 3: Multiclass Diagram CFTP

Data: $(U_{-n} = (i_{-n}, J_{-n}), \Theta_{-n})_{n\in\mathbb{N}}$ an i.i.d sequence of r.v
Result: $\mathbf{x} \in \mathcal{S}$

```
begin
    n ← 1;
    T ← T_{U_{-1}};
    while |ψ(T(D))| ≠ 1 do
        n ← 2n;
        T ← T_{U_{-1}} ∘ ··· ∘ T_{U_{-n}};
    return x, the unique state of ψ(T(D))
```

$t_{U_{-N}}$ with the same random sequence $(U_{-n})_{n\in\mathbb{N}}$. Lemma 3 implies that $t(\mathcal{S}) \subseteq \psi(T(\mathcal{D})) \subseteq \mathcal{S}$. But $|\psi(T(\mathcal{D}))| = 1$, which means, by Lemma 2, that $|t(\mathcal{S})| = 1$. Let $\mathbf{x}$ be the unique state of $\psi(T(\mathcal{D})) = t(\mathcal{S})$, the state returned by Algorithm 3. State $\mathbf{x}$ is also the result returned by Algorithm 1. So it samples the stationary distribution.

4 Numerical experiments

In this section, we compare our MDCFTP (Algorithm 3) with the classical CFTP algorithm (Algorithm 1).

We compare the size of the state representation, the coupling times and the running times. The coupling time is the value n when the algorithm stops. Throughout the experiments, we use the PRIORITY discipline where class 1 is given the priority over class 2. The algorithms have been implemented in Python and performed on a laptop.

4.1 Network of Example 1

We first consider the network of Example 1 with m customers in each class ($M = 2m$), with $m \in [1, 20]$. In Figure 9 we compare the performance of Algorithms 1 and 3 (using the same sequences (U_{-n})). Each value is the mean of 100 random samples.

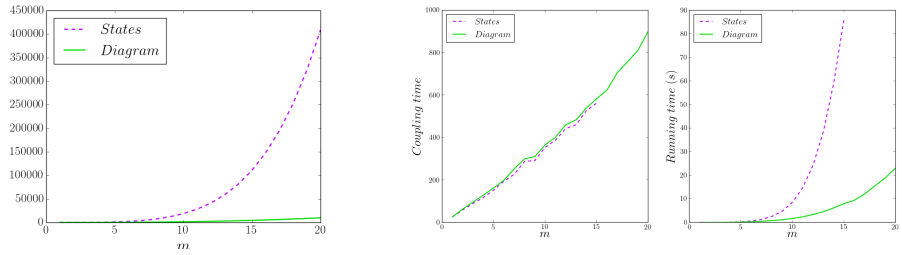

Figure 9: Comparisons for the network of Example 1. Left: cardinality of the state space $|\mathcal{S}|$ vs. number of arcs in the diagram $|g(\mathcal{S})|$; center: comparison of the coupling times; right: comparison of the running times.

As expected, the cardinality of the state space grows exponentially with m, while $|g(\mathcal{S})|$ only grows polynomially. The coupling times are very close, which indicates that our representation is precise enough to ensure a reasonable coupling time. Algorithm 3 significantly outperforms Algorithm 1 in terms of the running times. For Algorithm 1, the sampling could not be computed in less than six hours for $m > 15$.

4.2 Bidirectional ring network

We consider a bidirectional ring network with K queues and 2 classes, and matrices transitions $P^1_{i,i \mod (K)+1} = 0.9$, $P^1_{i \mod (K)+1,i} = 0.1$, $P^2_{i,i \mod (K)+1} = 0.1$ and $P^2_{i \mod (K)+1,i} = 0.9$. The number of customers in each class is 5.

We perform exactly the same experiments as in the previous example, but make the number of queues vary between 2 and 20 (Figure 10).

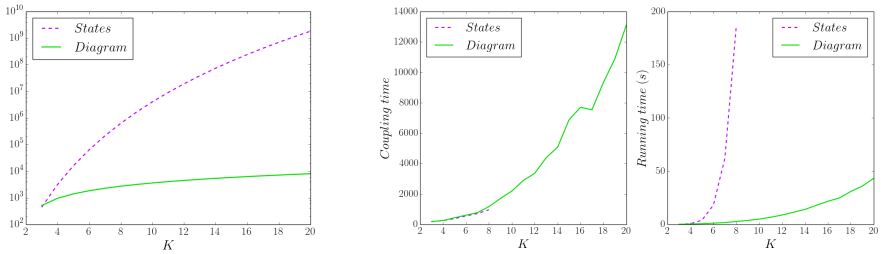

Figure 10: Bidirectional network: left: cardinality of the state space $|\mathcal{S}|$ vs. number of arcs in the diagram $|g(\mathcal{S})|$; center: comparison of the coupling times; right: comparison of the running times.

Not surprisingly, the cardinality of the state space grows exponentially with K. We were able to perform Algorithm 1 only for very small values ($K \leq 7$). Indeed, the number of states is in $O(m^{2K})$ whereas the state representation with diagrams is in $O(Km^4)$. In those cases, the coupling times of the two algorithms are very close and the running time correlated with the cardinality of the representation.

4.3 Comparisons of the number of states $|\mathcal{S}|$ and the size of the diagram representation $|g(\mathcal{S})|$

As our representation is still exponential in Z, we have restricted ourselves to experiments with two classes of customers. We compare the size of the representation of the state space for Algorithms 1 and 3. The size of the representation only depends on the sets of queues visited by each class, and not on the exact topology of the network. For the sake of simplicity, we assume that each class has m customers and visits every queue. We make m vary for several values of Z and K. The results are depicted in Figure 11.

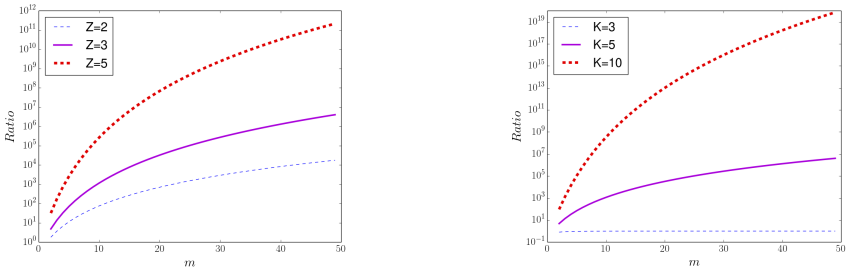

Figure 11: Ratio $\frac{|\mathcal{S}|}{|g(\mathcal{S})|}$. Left: $K = 5$, $Z \in \{2, 3, 5\}$; right: $Z = 3$, $K \in \{3, 5, 10\}$.

5 Conclusion

The main contribution of the paper is the derivation of the CFTP algorithm for multiclass closed queueing networks under various service policies. The only assumption made is that each station i choses the class to serve using only local information - the current state of queue i.

Our CFTP algorithm uses multiclass diagrams, a more compact representation of the state space. As in the monoclass case, using diagrams allows reduction of the complexity of the one-step transition function in the CFTP scheme from exponential to linear, in terms of the number of queues in the system. Unfortunately, diagram CFTP is still exponential in terms of the number of classes, so it is efficient only when the number of classes stays relatively small (less than 5). The main open question is the existence of a compact representation that is also polynomial in the number of classes, while keeping the coupling time of the bounding chain close to that of the original one.

We plan to investigate more closely the implementation of the transition function for the multiclass diagrams and the possible generalizations of the gap free diagrams in [6] that allowed a complexity reduction from $O(KM^2)$ to $O(KM)$ in the monoclass case.

For simplicity of exposition, we focused here on infinite capacity case. Extension to buffers with finite capacity does not represent any major difficulty, but it is not straightforward.

The major theoretical challenge is the study of the coupling times.

Acknowledgments

The work presented in this paper has been carried out at LINCS (www.lincs.fr) and has been founded by the French National Research Agency grant ANR-12-MONU-0019.

References

[1] S. Asmussen and P. W. Glynn. *Stochastic simulation: algorithms and analysis*. Springer, New York, 2007.

[2] S. Balsamo. Queueing networks with blocking: Analysis, solution algorithms and properties. In D. D. Kouvatsos, editor, *Network Performance Engineering*, volume 5233 of *LNCS*, pages 233–257. Springer Berlin Heidelberg, 2011.

[3] F. Baskett, K. M. Chandy, R. R. Muntz, and F. G. Palacios. Open, closed, and mixed networks of queues with different classes of customers. *J. Assoc. Comput. Mach.*, 22:248–260, 1975.

[4] B. Baynat and Y. Dallery. A product-form approximation method for general closed queueing networks with several classes of customers. *Performance Evaluation*, 24(3):165–188, 1996.

[5] A. Bouillard, A. Busic, and C. Rovetta. Perfect sampling for multiclass closed queueing networks. hal-01159962. June 2015.

[6] A. Bouillard, A. Bušić, and C. Rovetta. Clones: Closed queueing networks exact sampling. In *8th International Conference on Performance Evaluation Methodologies and Tools, VALUETOOLS 2014*. ICST, 2014.

[7] A. Bouillard, A. Bušić, and C. Rovetta. Perfect sampling for closed queueing networks. *Performance Evaluation*, 79(0):146–159, 2014.

[8] A. Bušić, I. Vliegen, and A. Scheller-Wolf. Comparing Markov chains: aggregation and precedence relations applied to sets of states, with applications to assemble-to-order systems. *Math. Oper. Res.*, 37(2):259–287, 2012.

[9] A. Bušić, B. Gaujal, and F. Perronnin. Perfect sampling of networks with finite and infinite capacity queues. In *19th International Conference on Analytical and Stochastic Modeling Techniques and Applications, ASMTA 2012*, volume 7314 of *LNCS*, pages 136–149. Springer, 2012.

[10] A. Bušić, B. Gaujal, and F. Pin. Perfect sampling of Markov chains with piecewise homogeneous events. *Performance Evaluation*, 69(6):247–266, 2012.

[11] W. Gordon and G. Newel. Closed queueing systems with exponential servers. *Oper. Res.*, 15,2:254–265, 1967.

[12] M. Huber. Perfect sampling using bounding chains. *Ann Appl Probab*, 14(2):734–753, 2004.

[13] W. Kendall and J. Møller. Perfect simulation using dominating processes on ordered spaces, with application to locally stable point processes. *Adv. in Appl. Probab.*, 32(3):844–865, 2000.

[14] S. Kijima and T. Matsui. Randomized approximation scheme and perfect sampler for closed Jackson networks with multiple servers. *Annals of Operations Research*, 162(1):35–55, 2008.

[15] D. Levin, Y. Peres, and E. Wilmer. *Markov Chains and Mixing Times*. American Mathematical Society, 2009.

[16] R. Marie. An approximate analytical method for general queueing networks. *IEEE Transactions on Software Engineering*, 5(5):530–538, 1979.

[17] J. G. Propp and D. B. Wilson. Exact sampling with coupled Markov chains and applications to statistical mechanics. *Random Struct. Algorithms*, 9(1-2):223–252, 1996.

[18] K. Satyam, A. Krishnamurthy, and M. Kamath. Solving general multiclass closed queuing networks using parametric decomposition. *Computers & Operations Research*, 40:1777–1789, 2013.

[19] K. Sigman. Exact simulation of the stationary distribution of the FIFO M/G/c queue: the general case for $\rho < c$. *Queueing Systems*, 70(1):37–43, 2012.

[20] N. M. van Dijk. Bounds and error bounds for queueing networks. *Ann. Oper. Res.*, 79:295–319, 1998.

A Additional proofs

A.1 Proof of Theorem 2

If each queue discipline satisfies Assumptions 1 then there exists a finite sequence of transitions t on the network such that $|t(\mathcal{S})| = 1$.

We first remove the dependence on the parameters θ by performing the same transition M times. The following lemma is the key of the proof.

Lemma 1 *Let $(i, J) \in \{1, \dots, K\}^{Z+1}$ such that for all $z \in Z[i], J(z) \neq z$, and $(\theta_1, \dots, \theta_{|M|}) \in [0, 1]^M$. Then $\mathbf{y} = t_{i,J,\theta_M} \circ \dots \circ t_{i,J,\theta_1}(\mathbf{x})$ satisfies the following properties:*

1. *$|\mathbf{y}_{*,i}| = 0$ (the departure queue is empty);*
2. *For all $z \in \{1, \dots, Z\}$, $y_{z,J[z]} = x_{z,J[z]} + x_{z,i}$ (customers have moved destination queues);*
3. *$y_{z,k} = x_{z,k}$ otherwise.*

Proof 3 *Each time a transition $t_{i,J,\theta}$ is performed, the number of customers in queue k decreases by one if it is non null. As $|\mathbf{x}_{*,i}| \leq M$, $|\mathbf{y}_{*,i}| = 0$. Every class z customer in queue i moves to queue $J[z]$, hence the result.*

□

Remark that this composition of transitions $t_{i,J,\theta_M} \circ \dots \circ t_{i,J.\theta_1}$ does not depend on $(\theta_1, \dots \theta_M)$, so, to simplify, we write $t^M_{i,J}$.

We now focus on class z and show that there exists a sequence of transitions that couples for this class. As the network G_z is strongly connected, there exists a path $w = (w_1, w_2, \dots, w_{N_z}) \in K(z)^{N_z}$ of length N_z that crosses every queue in $K(z)$.

Proposition 1 *For $\mathbf{x} \in \mathcal{S}$ and $J_1, \dots, J_{N_z} \in \{1, \dots, K\}^Z$ such that $\forall n \leq N_z - 1$, $J_n[z] = w_{n+1}$. Then $\mathbf{y} = t^M_{w_{N_z-1}J_{N_z-1}} \circ \dots \circ t^M_{w_1 J_1}(\mathbf{x})$ satisfies $y_{z,w_{N_z}} = M_z$ and $y_{z,k} = 0$ for all $k \neq w_{N_z}$.*

Proof 4 *We prove this result by induction. Set $\mathbf{y}^{(n)} = t^{|M|}_{w_n J_n} \circ \dots \circ t^{|M|}_{w_1 J_1}(\mathbf{x})$. We will prove that for all $n \leq N_z$,*

$$y^{(n)}_{z,k} = \begin{cases} 0 & \text{if } k \in \{w_1, w_2, \dots, w_n\} \setminus \{w_{n+1}\} \\ \sum_{k \in \{w_1, \dots, w_{n+1}\}} x_{z,k} & \text{if } k = w_{n+1} \\ x_{z,k} & \text{otherwise.} \end{cases}$$

The case $n = 0$ is straightforward as $\mathbf{y}^{(0)} = \mathbf{x}$. Suppose that $\mathbf{y}^{(n-1)}$ satisfy the properties above. We apply Lemma 1 to $\mathbf{y}^{(n-1)}$ and (w_n, J_n). We then obtain that $y^{(n)}_{z,w_{n+1}} = y^{(n-1)}_{z,w_n} + y^{(n-1)}_{z,w_{n+1}} = \sum_{i=1}^{n} x_{z,w_i} + x_{z,w_{n+1}} = \sum_{i=1}^{n+1} x_{z,w_i}$, and that $y^{(n)}_{z,w_n} = y^{(n-1)}_{z,w_n} = 0$. The other queues are not modified regarding class z, so $\mathbf{y}^{(n)}$ satisfy the properties above.

To conclude, is is sufficient to notice that $\mathbf{y} = \mathbf{y}^{(N_z-1)}$ with $\{w_i,\ i \in \{1, \dots, N_z\}\} = K(z)$.

□

We now have all the ingredients to conclude the proof of Theorem 2. Denote by t_z the sequence $t^M_{w_{N_z-1}J_{N_z-1}} \circ \cdots \circ t^M_{w_1 J_1}$ constructed for each $z \in \{1, \ldots, Z\}$. Then $t = t_Z \circ \ldots \circ t_1$ is a coupling sequence. Indeed, Proposition 1 ensures that for each class there is some step such that all the customers of this class are in the same queue. But Lemma 1 also ensures that all the customers once in the same queue, will remain in a same queue after performing M times the same transition (and our sequence is constructed this way).

A.2 Proof of Theorem 3

If each queue discipline satisfies Assumptions 1 then there exists a finite sequence of transitions T such that $|\psi(T(\mathcal{D})| = 1$.

The proof is similar to that of Theorem 2. In fact, replacing $t_{i,J}$ by $T_{i,J}$ in t leads to the sequence T, and we will show that T is a coupling sequence.

The key element is a sort of equivalent of Lemma 1, but we cannot be so precise. We focus on diagrams such that class z either has no costumers in queue k in all the states represented by the diagram, or has all its customers in queue k. More precisely, we say that a diagram $D = (N, A)$ satisfy $E_{z,k}$ if $\forall a \in A_k$, $v(a)_z = 0$ (all the states represented by D have no customer of class z in queue k); and we say that D satisfy $F_{z,k}$ if $\forall a \in A_k$, $v(a)_z = M_z$ (all customers of class z are in queue k)

Lemma 2 *Let $(i, J) \in \{1, \ldots, K\}^{Z+1}$ and $(\theta_1, \ldots, \theta_M) \in [0, 1]^M$, set $D' = T_{i,J,\theta_M} \circ \cdots \circ T_{i,J,\theta_1}(D)$. Then*

1. *D' satisfies $E_{z,i}$ for all $z \in \{1, \ldots, Z\}$ (queue i is empty);*
2. *If D satisfies $E_{z,j}$ and $j \neq J[z]$, then D' also satisfies $E_{z,j}$;*
3. *If D satisfies $F_{z,i}$, then D' satisfies $F_{z,J[z]}$;*
4. *If D satisfies $F_{z,j}$, $j \neq i$ then D' also satisfies $F_{z,j}$.*

Proof 5 *First consider column i, each time a transition $T_{i,J}(., \theta_\ell)$ is applied, $\forall a \in A_i$, the quantity $|v(a)| = \sum_{z=1}^{Z} v(a)$ decreases by 1 if it is positive. Then after performing M times this transition, $v(a) = (0, \ldots, 0)$ for all $a \in A_i$.*

Second, consider a class z and a queue $j \neq J[z]$ such that D satisfy $E_{z,j}$. By construction, for all $a \in A_j$, $v(a)_z$ is not modified by $T_{i,J}$. So $E_{z,j}$ still hold in D'.

If D satisfy $F_{z,j}$, then it satisfy $E_{z,k}$ for every $k \neq j$ (there are exactly M_z customers of class z on every path.

□

Now consider the sequence T_z for a class of customers. Set $D^{(n)} = T^M_{w_n J_n} \circ \ldots \circ T^M_{w_1 J_1}(\mathbf{x})$. From Lemma 2, one can deduce that $D^{(n)}$ satisfy E_{z,w_n}, and $E_{z,k}$ for all $k \in \{w_1, \ldots, w_{n-1}\} \setminus \{w_{n+1}\}$. Then $D^{(N_z-1)}$ satisfy $E_{z,k}$ for all k except w_{N_z}. As a consequence, one must have for all $a \in A_{N_z}$, $v(a)_z = M_z$ and $D^{(N_z-1)}$ satisfies $F_{z,w_{N_z}}$.

Applying Lemma 2 leads to the desired result: $T(D)$ satisfy either $E_{z,k}$ or $F_{z,k}$ for all $(z, k) \in \{1, \ldots, Z\} \times \{1, \ldots K\}$, which means that there is only one path in the diagram.