Perfect sampling for closed queueing networks

Anne Bouillard[a], Ana Bušić[b], Christelle Rovetta[b]

[a]*ENS Paris, 45 rue d'Ulm 75005 Paris, France*
[b]*Inria, 23 avenue d'Italie, 75013 Paris, France*

Abstract

In this paper we investigate coupling from the past (CFTP) algorithms for closed queueing networks. The stationary distribution has a product form only in a very limited number of particular cases when queue capacity is finite, and numerical algorithms are intractable due to the cardinality of the state space. Moreover, closed networks do not exhibit any monotonic property enabling efficient CFTP. We derive a bounding chain for the CFTP algorithm for closed queueing networks. This bounding chain is based on a compact representation of sets of states that enables exact sampling from the stationary distribution without considering all initial conditions in the CFTP. The coupling time of the bounding chain is almost surely finite, and numerical experiments show that it is close to the coupling time of the exact chain.

Keywords: queueing networks, simulation, coupling from the past

1. Introduction

One reason for the popularity of Markovian representations of queueing networks is that they admit a product-form stationary distribution under general conditions. This structure is no longer guaranteed when the queues have a finite capacity, so that an exact analysis may not be computationally tractable. When the stationary distribution cannot be computed then we may turn to approximations or simulation.

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, 2]. 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.

In the 1990's, Propp and Wilson introduced a method for sampling a random variable according to the stationary distribution of a finite ergodic Markov chain [3]: 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 number of steps required in the CFTP algorithm is proportional to the coupling time of the chain, but the time complexity strongly depends on the complexity of the one-step transition of the chain. In particular, for closed queueing networks, no efficient CFTP method has been proposed previously, with the exception of networks with a product form distribution [4].

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 [3]. 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 [5?],

Email addresses: anne.bouillard@ens.fr (Anne Bouillard), Ana.Busic@inria.fr (Ana Bušić), christelle.rovetta@inria.fr (Christelle Rovetta) or for models with short range local interactions, such as interacting particle systems [6]. For applications of [5] to queueing networks, see for instance [7, 8].

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 [5] cannot be applied directly. Without monotonicity, the complexity of one iteration of the original CFTP algorithm by Propp and Wilson [3] depends on the cardinality of the state space, which is exponential in the number of queues.

In this paper, we develop an effective CFTP algorithm for closed queueing networks. Our main contribution is a new technique for constructing bounding chains, which is adapted to a large class of Markovian closed queueing networks. It is based on a compact representation for sets of states, using diagrams that we introduce in Section 3. We perform the CFTP algorithm on the space of the diagrams, for which the one-step transition is simpler to compute than using the original state space. For the diagrams, one iteration in the CFTP algorithm can be computed in $O(KM^2)$ time, as we discuss in Section 3.2.3.

The paper is organized as follows. In Section 2 we present the queueing model and discuss the solutions that have been proposed in the literature for special cases. The analysis is based on a *diagram representation* introduced in Section 3. This is used in Section 4 to prove that the CFTP algorithm terminates in finite time, almost surely. Finally, Section 5 contains results from numerical, comparing our algorithm with the classical CFTP in terms of the number of iterations. Note that classical CFTP can be used only for very small models, as its one-step transition depends on the cardinality of the state space. Final remarks and conclusions are contained in Section 6.

2. Model and background

2.1. Presentation of the model

We denote by $\mathbb{N}$ the set of non-negative integers and by $e_i \in \mathbb{N}^k$ the vector such that $(e_i)_j = 1$ if $i = j$ and 0 otherwise.

Consider a closed network of $\cdot/M/1/C$ queues with M customers. We denote by $Q = \{1, \dots, K\}$ the set of queues. For $i \in Q$, μ_i is the service rate and C_i the capacity of queue i. After a service in queue i, a customer is routed to queue j with probability $p_{i,j}$, independently of the current state and past evolution of the network. If queue j is full, the customer is blocked at queue i for another service time. Let $P = (p_{i,j})_{i,j \in Q}$ denote the routing probability matrix; for all $i, j \in Q$, $p_{i,j} \geq 0$ and for all $i \in Q$, $\sum_{j \in Q} p_{i,j} = 1$.

The evolution of this network follows a continuous time Markov chain on the state space

$$\mathcal{S} = \left\{ x = (x_1, x_2, \dots, x_K) \in \mathbb{N}^K \ : \ \sum_{i=1}^{K} x_i = M \quad \text{and} \quad 0 \leq x_i \leq C_i, \ \forall i \in Q \right\}.$$

The upper bound for $|\mathcal{S}|$ is given by $\binom{K+M-1}{K-1}$; this is the exact value for $|\mathcal{S}|$ if all queues have infinite capacity.

The topology of the network can be represented by a directed graph $G = (Q, R)$ where $R = \{(i, j) \ : \ p_{i,j} > 0\}$. As we consider closed networks, without loss of generality, we can assume that G is strongly connected.

For $(i, j) \in R$, we denote by $t_{i,j} \ : \ \mathcal{S} \to \mathcal{S}$ the function that describes routing of a customer from queue i to queue j:

$$t_{i,j}(x) = \begin{cases} x - e_i + e_j & \text{if } x_i > 0 \text{ and } x_j < C_j, \\ x & \text{otherwise.} \end{cases}$$

This transition does not affect the state if queue i is empty or if queue j is full.

We consider a discrete time Markov chain, obtained using uniformization with constant $\sum_{i=1}^{K} \mu_i$. This chain has the same state space as the continuous time chain.

A functional representation of this discrete time Markov chain can be given using functions $t_{i,j}$ and routing matrix P. Denote by $(U_n)_{n \geq 1}$ an i.i.d. sequence of random variables with distribution

$$P(U_1 = (i, j)) = \frac{\mu_i}{\sum_{j \in Q} \mu_j} p_{i,j}.$$

Define $F : \mathcal{S} \times R$ as

$$F(x,(i,j)) = t_{i,j}(x).$$

Let X_0 be the initial state, independent of $(U_n)_{n\geq 1}$, and $X_{n+1} = F(X_n, U_{n+1}),\ n \in \mathbb{N}$.

2.2. Coupling from the past algorithm

The coupling from the past (CFTP) algorithm has been first introduced in 1996 by Propp and Wilson [3]. The key idea is to sample a value at time 0 of a trajectory that starts arbitrary far in the past.

Let $\{X_n\}_{n\in\mathbb{N}}$ be an ergodic Markov chain on a finite state space $\mathcal{S}$, and F be a functional representation of $\{X_n\}$: $\forall n \in \mathbb{N}$, $X_{n+1} = F(X_n, U_{n+1})$, where $(U_n)_{n\geq 1}$ is an i.i.d. sequence of random variables independent of X_0. Consider the evolution of this Markov chain starting from some time in the past. Let $(U_{-n})_{n\in\mathbb{N}}$ be an i.i.d. sequence of r.v. For $x \in \mathcal{S}$, denote by

$$F^{(n)}(x) = F(F(\cdots F(x, U_{-n+1}), \ldots, U_{-1}), U_0)$$

the state of the chain at time 0, given that it has started from state x at time $-n$. Similarly, for a subset $S \subset \mathcal{S}$, let $F^{(n)}(S) = \{F^{(n)}(x)\ :\ x \in S\}$ (note that the same sequence $(U_{-n})_{n\in\mathbb{N}}$ is used for all the states in S). CFTP algorithm is given in Algorithm 1.

Algorithm 1: Coupling from the past algorithm [3]

```
  Data: $(U_{-n})_{n\in\mathbb{N}}$ an i.i.d. sequence of r.v., $F$ a functional representation of an ergodic Markov chain
        on a finite set $\mathcal{S}$.
  Result: A sample $s_0 \in \mathcal{S}$ according to the stationary distribution of the Markov chain
1 begin
2     $n \leftarrow 1$;
3     $S \leftarrow F^{(1)}(\mathcal{S})$;
4     while $|S| \neq 1$ do
5         $n \leftarrow 2n$;
6         $S \leftarrow F^{(n)}(\mathcal{S})$;
7     end
8     return $s_0$ the element of $S$
9 end
```

The following theorem concerns the termination and correctness of the algorithm:

Theorem 1. *Depending on the choice of F, Algorithm 1 terminates in finite time with probability 0 or 1. The state s_0 is distributed according to the stationary distribution of $\{X_n\}$.*

With our choice of F, the CFTP algorithm for closed queueing network model terminates in finite time with probability 1 and the termination time has a finite expectation.

The second statement is a consequence of the existence of a sequence $(i_n, j_n)_{1\leq n\leq N} \in R$ such that $|t_{i_N,j_N} \circ \ldots \circ t_{i_1,j_1}(\mathcal{S})| = 1$. We do not prove this (intuitive) result here, as we will prove a more general result in Section 4.

2.3. State of the art

When all the queues in the network have unlimited capacity, the stationary distribution of the process has a product form [9]. Furthermore, in the single server case, the computation of the normalizing constant can be done in $O(K\,M)$ time using Buzen's algorithm [10].

Besides the fact that the stationary distribution can be computed efficiently in the unlimited capacity case, specific methods have been also developed to perform perfect sampling. In [4], the authors first derive a new Markov chain that is reversible and monotonous, and that has the same stationary distribution as the original chain. Additionally, this new Markov chain has a sufficiently simple structure so that the coupling time can be computed. Overall complexity of their algorithm is in $O(K^3 \ln(KM))$. However, the method strongly relies on the product form representation of the stationary distribution and it cannot be applied to the general case of queues with finite capacity.

When the queues have finite capacity, the system is said *with blocking*. Different types of blocking exist, depending on the system that is modeled, as described and compared in [11]. According to the terminology introduced in that survey, our blocking model is RS-RD (*repetitive service – random destination*). It is proved that the stationary distribution of closed queueing networks with such a blocking policy has a product form if the routing is reversible. Approximation results are derived in [12] in the other cases.

In [13], the perfect sampling of bounded free choice Petri nets were studied. This particular class of Petri nets exhibits another kind of monotonic property: there is a set of *extremal states*, such that when the trajectories issued from these states couple, this implies that trajectories issued from all the states have coupled. In terms of queueing networks, this class corresponds to ring networks (networks composed of a single cycle) with $\cdot/M/1/C$ queues. For any more general network, where at least one queue has more than one successors, this monotonicity property is broken.

Finally, the bounding chain approaches that have been used to reduce the algorithmic complexity of the CFPT algorithm for systems that do not exhibit monotonicity properties rely either on a lattice property for the state space or on a simple local dynamics of the chain. For example, in the graph coloring problem considered in [6], each vertex of the graph is assigned a subset of potential colors that is defined only by the colors of its neighbors. This technique cannot be directly adapted to our model, because of the global constraint – the constant total number of customers in the network.

3. Diagram representation

The main limitation of Algorithm 1 is its need to enumerate the entire state space in order to compute the transition function. In this section, we present a more compact way to represent sets of states in order to avoid this enumeration. A state has two characteristics: the number of customers in each queue, which lays between zero and its capacity, and the total number of customers, which is constant and equal to M. The main idea is use these constraints to represent states as paths in a directed graph where each arc represents a possible number of customers in a queue. Such a directed graph is called a *diagram*. We will first present more formally the concept of diagrams and their relation with sets of states, and then describe how the transitions can be performed directly on diagrams without having to consider sets of states.

3.1. Diagram as a super-representative of a set of states

The idea is to construct a diagram in which any element of $\mathcal{S}$ is represented by a path. We begin by defining the complete diagram, that represents every state of $\mathcal{S}$, and then we will define diagrams more generally as some sub-graphs of this complete diagram.

3.1.1. Complete diagram

We define a directed graph such that there is a bijection between the state space $\mathcal{S}$ and the maximal paths of this graph. We call this graph the *complete diagram* of $\mathcal{S}$, denoted $\mathcal{D} = (N, \mathcal{A})$, where N is the set of nodes and $\mathcal{A}$ the set of arcs.

Nodes of the complete diagram, $N \subseteq \{0, \dots, K\} \times \{0, \dots, M\}$. Intuitively, $(c, \ell) \in N$ means "the number of customers in the c first queues is ℓ". As a consequence, if c is fixed, there are some constraints for a node (c, ℓ) to be in N: there exist a state where there are exactly ℓ customers in the first c queues. For $c = 0$, the only possibility is $\ell = 0$ and for $c = K$, necessarily $\ell = M$. In general, ℓ must satisfy

$$\ell^c_{\min} = \max(0, M - \sum_{i=c+1}^{K} C_i) \leq \ell \leq \min(M, \sum_{i=1}^{c} C_i) = \ell^c_{\max}.$$

Indeed, the number of customers cannot exceed the total capacity of the first c queues (hence $\ell^c_{\max}$) and cannot be less than the total number of customers minus the maximum number of customers that can be present in the other queues. Then,

$$N = \{(c, \ell) \ : \ c \in \{1, \dots, K-1\} \text{ and } \ell \in \{\ell^c_{\min}, \dots, \ell^c_{\max}\}\};$$

Arcs of the complete diagram, $\mathcal{A} \subseteq N^2$. Intuitively, $((c-1,\ell'),(c,\ell)) \in \mathcal{A}$ means "there are $\ell - \ell'$ customers in queue c". So $((c',\ell'),(c,\ell))$ is an arc only if $c' = c-1$ and $0 \le \ell - \ell' \le C_c$. Then

$$\mathcal{A} = \{((c-1,\ell'),(c,\ell)) \ : \ 1 \le c \le K \text{ and } 0 \le \ell - \ell' \le C_c\}.$$

For an arc $a = ((c-1,\ell'),(c,\ell)) \in \mathcal{A}$, we set $v(a) = \ell - \ell'$ the value of a.

Graphically, the nodes can be placed on a grid, (c,ℓ) being placed on the c-th column and the ℓ-th line, as shown on Figure 1. The value of an arc (corresponding to the number of customers in a queue) is then represented by the slope of this arc in the diagram. Although diagrams are directed graphs by definition, we omit the arrows in the graphical representation. This choice has been done to increase readability of the figures. As all the arcs are directed from left to right, there is no possible confusion.

Now we establish the bijection between the state space and the set of maximal paths in $\mathcal{D}$. Indeed, by construction, every maximal path p starts from $(0,0)$, ends at (K,M) and has length K. It can be written $p = ((0,0),(1,\ell_1),\dots,(K,M))$ and corresponds to the state

$$(\ell_1, \ell_2 - \ell_1, \dots, \ell_c - \ell_{c-1}, \dots, M - \ell_{K-1}) \in \mathcal{S}.$$

Similarly, for each state $(x_1,\dots,x_K) \in \mathcal{S}$, there is an associated the path

$$p = ((0,0),(1,x_1),(2,x_1+x_2),\dots,(c,\sum_{i=1}^{c} x_i),\dots,(K,M)).$$

We denote by f this bijection: $f : \mathcal{S} \to \mathcal{D}$, $\forall x = (x_1,\dots,x_K) \in \mathcal{S}$,

$$f(x) = ((0,0),(1,x_1),(2,x_1+x_2),\dots,(c,\sum_{i=1}^{c} x_i),\dots,(K,M)).$$

Example 1. *Consider a network of $K = 5$ queues with capacities $C = (1,2,3,2,1)$, and $M = 4$ customers. The cardinality of the state space is then 30. Each state can be represented by a path from node $(0,0)$ to node $(5,4)$ in the complete diagram given in Figure 1.*

3.1.2. Diagrams

Subsets of paths in the complete diagram represent subsets of the state space.

Definition 1. *$D = (N, A)$ is a* diagram *if $A \subseteq \mathcal{A}$ is such that $(0,0)$ is the only source node (*i.e. *node with in-degree 0) and (K,M) is the only sink node (*i.e *node with out-degree 0) in D apart from isolated nodes.*

In a diagram, *the maximal paths* are from $(0,0)$ to (K,M) and from now on we will only consider maximal paths. This also means that every arc of this diagram belongs to a path representing a state of the Markov chain.

Following the same lines as for the complete diagram, we define $\Pi(D)$ as the set of all paths from node $(0,0)$ to node (K,M) in $D = (N,A)$. For the set of states, we consider the usual inclusion order $\subseteq$. We can also define a natural order on the diagrams, that follows the arc inclusion. We say that $D = (N,A)$ is a sub-diagram of $D' = (N,A')$ and write $D \preceq D'$ if $A \subseteq A'$.

We now make a connection between sets of states and diagrams. First define the diagram from a set of paths from $(0,0)$ to (K,M). Let Π be a set of such paths. We denote by $A_\Pi \subseteq \mathcal{A}$ the set of arcs appearing in at least one of those paths.

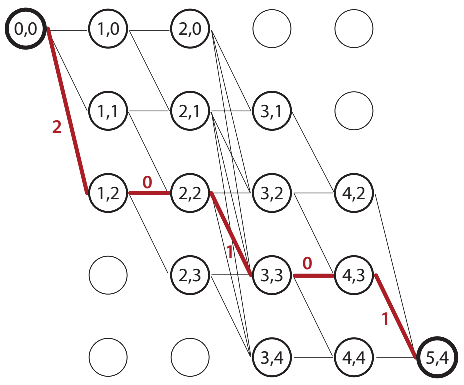


Figure 1: Complete diagram with $K = 5$ queues and $M = 4$ customers with capacity $C = (2, 1, 3, 1, 2)$. The bold (and red) path corresponds to state $(2, 0, 1, 0, 1)$. Node $(0, 0)$ is the start and $(5, 4)$ the end of each path.

From a diagram $D = (N, A)$, we can define the set of states $f^{-1}(\Pi(D))$. Conversely, from a set of states S, we can define $f(S) \subseteq \Pi(\mathcal{D})$ the set of paths of the diagram, and then the diagram $D = (N, A_{f(S)})$. For all $S \subseteq \mathcal{S}$ and all $D \preceq \mathcal{D}$, we define

$$\phi(S) = (N, A_{f(S)}) \qquad \text{and} \qquad \psi(D) = f^{-1}(\Pi(D)).$$

Lemma 1. *The following propositions hold:*

1. *ϕ and ψ are monotone;*
2. *For all $S \subseteq \mathcal{S}$, $S \subseteq \psi \circ \phi(S)$;*
3. *For all $D \preceq \mathcal{D}$, $D = \phi \circ \psi(D)$.*

Proof. 1. This is true by construction.

2. Let x be an element of S. The arcs of $f(x)$ are $A_{f(x)} \subseteq A_{f(S)}$. As a consequence, $x \in \psi(N, A_{f(x)}) \subseteq \psi(\phi(S))$ and $S \subseteq \psi(\phi(S))$;
3. $\phi(\psi(D)) = (N, A_{f(f^{-1}(\Pi(D)))}) = (N, A_{\Pi(D)}) = D$.

□

It is worth to notice that ϕ and ψ form an isotone Galois connection [14].

Definition 2. *Let $S \subseteq \mathcal{S}$ and $D = (N, A)$ be a diagram. If $S = \psi(D)$, we say that D is a representative of S. If $S \subseteq \psi(D)$, we say that D is a super-representative of S.*

Lemma 2.
- *$\mathcal{D}$ is a representative of $\mathcal{S}$;*
- *$\phi(\{x\}) = (N, A_{f(x)})$ is a representative of $\{x\}$. In other words, $\psi(\phi(\{x\}) = \{x\}$ for all $x \in \mathcal{S}$;*
- *If (N, A) is a representative of S, then $A = A_{f(S)}$;*
- *$\phi(S)$ is the smallest super-representative of S: if D is a super-representative of S, then $\phi(S) \preceq D$.*

Proof.
- $\Pi(\mathcal{D}) = f(\mathcal{S})$, so $\psi(\mathcal{D}) = \mathcal{S}$.
- $(N, A_{f(x)})$ has exactly K arcs, so it can contain only have one path of length K, that is $f(x)$. So $\psi(\phi(\{x\})) = \{x\}$.

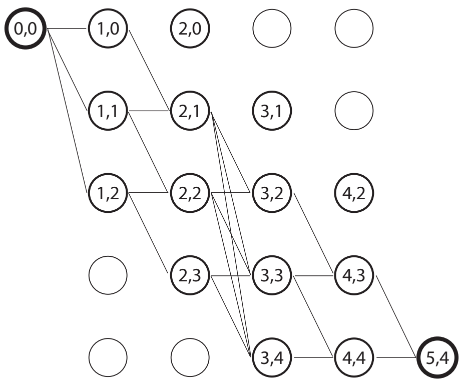


Figure 2: A diagram with $|\Pi(D)| = 19$.

- If $\psi(N, A) = S$, then $(N, A) = \phi(\psi(N, A)) = \phi(S)$. Then $A = A_{f(S)}$.
- This is a direct consequence of the fact that ϕ and ψ are an isotone Galois connection.

$\square$

Example 2. *Consider D the diagram of Figure 2. It is a sub-diagram of the complete diagram of Figure 1. With $S = \{(01111), (10201), (11011), (01300), (11200), (20110), (21001), (21100)\}$, it can easily be seen that $\phi(S) = D$, but $|S| = 8$ and $|\psi(D)| = 19$. So $\psi(\phi(S)) \subsetneq S$: diagram D is a super-representative of S.*

3.2. Transitions on diagrams

We now define the transition function on a diagram. We first describe it formally and show that the property of being a super-representative of a set of states is preserved by this transition. Then we present an effective computation of the transition of a diagram which can be implemented in time $O(KM^2)$.

3.2.1. Definitions and basic properties

Let $(i, j) \in R$. We note $T_{i,j}$ the transition function defined by:

$$T_{i,j}(D) = \phi \circ t_{i,j} \circ \psi(D),\ D \subset \mathfrak{D}.$$

Proposition 1. *(i) If D is a super-representative for S then $T_{i,j}(D)$ is a super-representative for $t_{i,j}(S)$.*

(ii) If $|\psi(D)| = 1$ then $|\psi \circ T_{i,j}(D)| = 1$.

Proof. (i) $S \subseteq \psi(D) \Rightarrow t_{i,j}(S) \subseteq t_{i,j} \circ \psi(D) \subseteq \psi \circ \phi \circ t_{i,j} \circ \psi(D) = \psi(T_{i,j}(D))$. So $T_{i,j}(D)$ is a super-representative of $t_{i,j}(S)$.

(ii) $|\psi(D)| = 1 \Rightarrow |t_{i,j} \circ \psi(D)| = 1 \Rightarrow t_{i,j} \circ \psi(D) = \psi \circ \phi \circ t_{i,j} \circ \psi(D)$ and $|\psi \circ T_{i,j}(D)| = 1$.

$\square$

3.2.2. Algorithm

We present an algorithm that performs transformation $T_{i,j}$ on a diagram $D = (N, A)$. Let us first focus on the case where $i < j$, and then describe the necessary adaptation to deal with $j < i$.

We divide the set of edges of A in three subsets (not necessarily disjoint).

- $\mathcal{E}mpty$ is the set of arcs that belong to paths with a value 0 at column i: $\mathcal{E}mpty = \{a \in A \mid a \in p \in \Pi(D) \text{ and } f(p)_i = 0\}$. These paths represent the states of $\phi(A)$ for which queue i is empty;

- $\mathcal{Full}$ is the set of arcs that belong to paths with value C_j at column j: $\mathcal{Full} = \{a \in A \mid a \in p \in \Pi(D) \text{ and } f(p)_j = C_j\}$. These paths represent the states of $\phi(A)$ for which queue j is full;
- $\mathcal{Transit}$ is the set of arcs that belong to the remaining paths: $\mathcal{Transit} = \{a \in A \mid a \in p \in \Pi(D) \text{ and } f(p)_i > 0 \text{ and } f(p)_j < C_j\}$. These paths correspond to states for which one customer can be transmitted from queue i to queue j.

Basically the algorithm performs the following operations: leave the arcs in $\mathcal{Empty}$ and $\mathcal{Full}$ unchanged, as these correspond to states where the transmission of one customer from i to j is not possible, and transform the arcs in $\mathcal{Transit}$ according to $t_{i,j}$. The new set of arcs is

$$\mathcal{Transit}' = \begin{cases} & \{((c-1,\ell),(c,\ell')) \mid ((c-1,\ell),(c,\ell')) \in \mathcal{Transit} \text{ and } c < i \text{ or } c > j\} \\ \cup & \{((i-1,\ell),(i,\ell'-1)) \mid ((i-1,\ell),(i,\ell')) \in \mathcal{Transit}\} \\ \cup & \{((c-1,\ell-1),(c,\ell'-1)) \mid ((c-1,\ell),(c,\ell')) \in \mathcal{Transit} \text{ and } i < c < j\} \\ \cup & \{((j-1,\ell-1),(i,\ell')) \mid ((j-1,\ell),(i,\ell')) \in \mathcal{Transit}\}. \end{cases}$$

Theorem 2. *Let D be a diagram and $\mathcal{Empty}$, $\mathcal{Full}$, $\mathcal{Transit}$ and $\mathcal{Transit}'$ defined as above. Set $A' = \mathcal{Empty} \cup \mathcal{Full} \cup \mathcal{Transit}'$. Then $T_{i,j}(D) = (N, A')$.*

Proof. Note that every path in D has all its arcs in either $\mathcal{Empty}$, $\mathcal{Full}$ or $\mathcal{Transit}$. Consider a path in $\Pi(D)$.

- If this path has all its arcs in $\mathcal{Empty}$ or all its arcs in $\mathcal{Full}$, then this corresponds to a state of $\psi(D)$ that is not affected by $t_{i,j}$. Hence this must be a path in (N, A').
- If this path is in $\mathcal{Transit}$, then it corresponds to a state of $\psi(D)$ in which queue i is not empty and queue j is not full, so one customer will be transferred from queue i to queue j. This affects only the columns $i-1$ to j and the image of these paths by $t_{i,j}$ is given by $\mathcal{Transit}'$.

□

Example 3. *Figure 3 shows the successive steps to compute $T_{1,3}(D)$, where D is the diagram of Figure 2. First, we determine the subsets $\mathcal{Empty}$, $\mathcal{Full}$ and $\mathcal{Transit}$. Note that we only need to focus on arcs between columns 0 to 3, as the arcs involving the other columns are left unchanged. Second, we determine $\mathcal{Transit}'$. Graphically, the value of every arc of $\mathcal{Transit}$ starting in column 0 decreases by 1 (its destination is modified accordingly); the value of every arc ending in column 3 increases by 1 (its origin is modified accordingly); every other arc of $\mathcal{Transit}$ is shifted upwards by one unit. Finally, $T_{1,3}(D) = (N, A')$ with $A' = \mathcal{Empty} \cup \mathcal{Full} \cup \mathcal{Transit}'$.*

The case $j < i$ is similar. Consider a new diagram $\overline{D}$, obtained from D by changing the orientation of all the arcs in A and renaming the nodes (c,ℓ) to $(K-c, M-\ell)$. Diagram $\overline{D}$ then represents exactly the same set of states when the order of the queues is reversed. Then the above algorithm applied to $\overline{D}$ gives a new diagram $\overline{D}'$. By changing once again the orientation of all the arcs and renaming the nodes in to their original labels, we get $T_{i,j}(D)$.

3.2.3. Implementation and complexity

Let $D = (N, A)$ be a diagram. Each column c of D can be represented by an incidence (Boolean) matrix X_c of size $(M+1) \times (M+1)$, *i.e.*, $X_c(\ell,\ell') = 1$ if $((c-1,\ell),(c,\ell')) \in A$ and 0 otherwise. Those matrices are upper-triangular as $X_c(\ell,\ell') = 1 \Rightarrow \ell \leq \ell'$. A diagram is then represented by K matrices $X_1, \ldots, X_K$. A transition $T_{i,j}$ only affects matrices X_i, $X_{i+1}, \ldots,$ X_j but in the worst case transition $T_{1,K}$ affects each matrix. Identifying the sets $\mathcal{Empty}$, $\mathcal{Full}$ and $\mathcal{Transit}$ can be done using K vector by matrix multiplications, in time $O(KM^2)$. The subset $\mathcal{Transit}'$ can be computed by using shift operations on binary matrices in time $O(KM^2)$, and the recombination of $\mathcal{Empty}$, $\mathcal{Full}$ and $\mathcal{Transit}'$ is just the sum of at most K matrices, which can also be done in $O(KM^2)$). As a consequence, the overall time complexity of computing $T_{i,j}(D)$ is $O(KM^2)$.

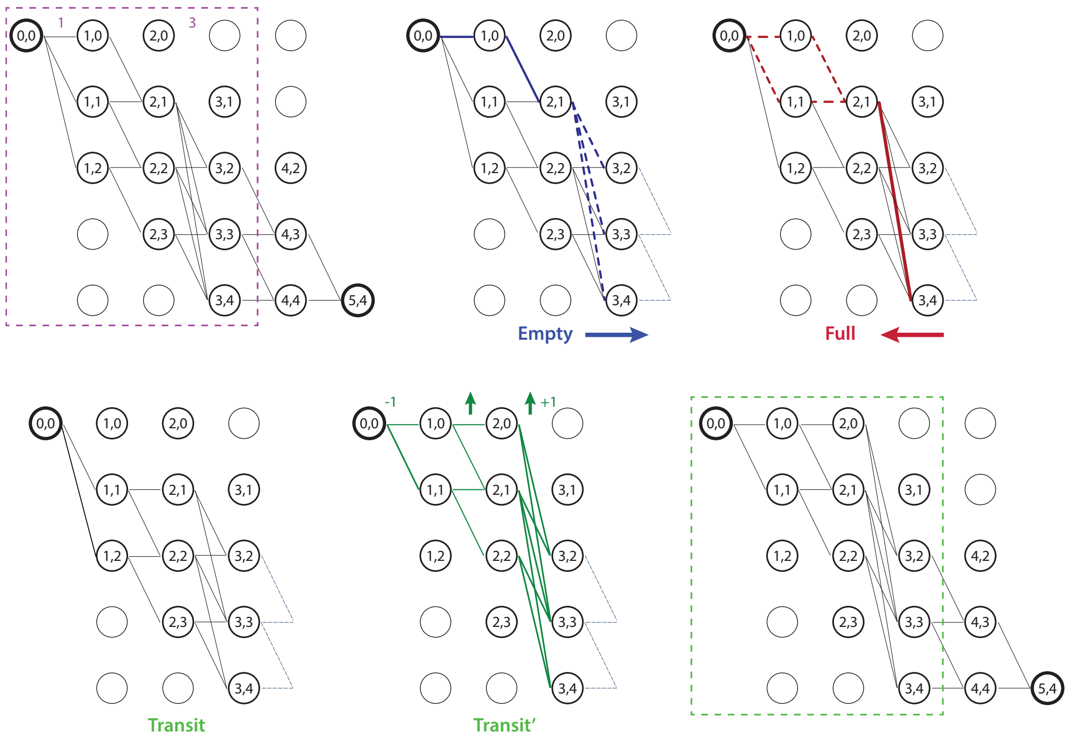


Figure 3: Transition $T_{1,3}(D)$.

4. Diagram CFTP

4.1. Correctness of the algorithm

In Algorithm 2 we present a CFTP algorithm that uses diagrams instead of sets of states. There are few changes compared to Algorithm 1: $\mathcal{S}$ is replaced by $\mathcal{D}$, $t_{i,j}$ by $T_{i,j}$ and $F^{(n)}$ by $T^{(n)} = T_{U_0} \circ T_{U_{-1}} \circ \cdots \circ T_{U_{-n+1}}$. The stopping criterion then becomes $|\Pi(T^{(n)}(\mathcal{D}))| = 1$.

If the algorithm almost surely terminates in finite time, then s_0 is distributed according to the stationary distribution of the queueing network. This is a direct consequence of Proposition 1. Indeed, if $|\Pi(F^{(n)}(\mathcal{D}))| = 1$, then diagram $F^{(n)}(\mathcal{D})$ is the representative of the unique state that would be obtained with the exact CFTP.

4.2. Termination of the algorithm

We now want to show that the diagram is a representation of states that is refined enough to ensure the termination of the CFTP algorithm with probability 1. We distinguish two cases: when every queue has capacity at least M (then they behave like $./M/1/\infty$ queues), and the general case.

4.2.1. Every queue has infinite capacity

Consider (Q, R) the topological representation of the network. Consider a spanning tree of this graph, where all the arcs of this tree are directed to their ancestor and a sequence of the arcs of this tree $(i_1, j_1), (i_2, j_2), \ldots, (i_{K-1}, j_{K-1})$ where the nodes $i_1, \ldots, i_{K-1}$ follow a topological ordering (increasing from the leaves to the root). In other words, every queue appears exactly once in $\{i_1, \ldots, i_{K-1}\}$, except the root of the tree and $\forall z \in \{1, \ldots, K-1\}$, $i_z \notin \{j_{z+1}, \ldots, j_{K-1}\}$.

The high-level idea is to construct a *coupling sequence of transitions* for which the image of the complete diagram is a single-path diagram. Informally, such sequence can be obtained by "sending the customers" from the leaves to the root of such spanning tree.

Algorithm 2: CFTP algorithm with diagram

```
Data: $(U_{-n} = (i_{-n}, j_{-n}))_{n\in\mathbb{N}}$ an i.i.d. sequence of r.v., $F$ a functional representation of an ergodic
      Markov chain on a finite set $\mathcal{S}$.
Result: A sample $s_0 \in \mathcal{S}$ according to the stationary distribution of the Markov chain.
1  begin
2      $n \leftarrow 1$;
3      $D \leftarrow T^{(1)}(\mathcal{D})$;
4      while $|\Pi(D)| \neq 1$ do
5          $n \leftarrow 2n$;
6          $D \leftarrow T^{(n)}(\mathcal{D})$;
7      end
8      $p_0 \leftarrow$ the unique element of set $\Pi(D)$ ;
9      return $f^{-1}(p_0)$
10 end
```

Theorem 3. *If every queue has capacity at least M, then $|T^M_{(i_{K-1},j_{K-1})} \circ \ldots \circ T^M_{(i_1,j_1)}(\mathcal{D})| = 1$.*

Proof. The proof has three steps.
(i) Action of $T^M_{(i,j)}$ on $D = (N, A)$. Assume without loss of generality that $i < j$. Let us denote A_i the set of arcs in D from column $i-1$ to column i and set $q_{\max} = \max(v(a),\ a \in A_i)$. For each path π in $\Pi(D)$, $(t_{i,j}(\psi(\pi)))_i = \max(\psi(\pi)_i - 1, 0)$. Then in $T_{i,j}(D)$, there is no arc in column i with value more than $\max(q_{\max} - 1, 0)$. But then, applying this argument M times, in $T^M_{i,j}(D)$, there is no arc with value more than $\max(q_{\max} - M, 0) = 0$ as $q_{\max} \leq M$.
(ii) On columns with every arc having value 0. Consider a diagram D such that every outgoing arc in column $i-1$ has value 0, and that no transition of type (j, i), for some $j \neq i$ is performed. Then the slopes of column i are unchanged and remain 0. Indeed, for any state $x \in \psi(D)$, $x_i = 0$ and $t_{j,j'}(x)_i = 0$ whenever $j' \neq i$. As a consequence, in $T_{j,j'}(D)$, every outgoing arc from column $i-1$ has value 0.
(iii) Conclusion. Compute $T^M_{(i_1,j_1)}(\mathcal{D})$. All the arcs in column i_1 have value 0, and as $i_1 \notin \{j_2, \ldots, j_{K-1}\}$, the value of the arcs at this columns will remain 0. Consider a diagram D where all the arcs in the columns $i_1, \ldots, i_{z-1}$ have value 0. When applying $T^M_{i_z,j_z}$ to D, the values of the arcs in those columns will remain 0. Moreover all the arcs in column i_z will become 0. So, by induction, $T^M_{(i_{K-1},j_{K-1})} \circ \ldots \circ T^M_{(i_1,j_1)}(\mathcal{D})$ is a diagram where all the arcs in all the columns have value 0, except the arcs of column j_{K-1}, that must then have value M. There is only one such arc and consequently only one path with this property, corresponding to the state that has all the customers in queue j_{K-1}. □

4.2.2. General case

The proof in the general case is much more complex. We will use the network of Figure 4 to explain the main steps of the proof. Every queue has finite capacity 2 and the total number of customers is 4.

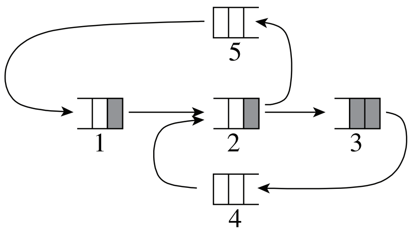


Figure 4: Example of a network of queues.

In the general case the spanning tree transition sequence of Theorem 3, $\widehat{T} = T^M_{(i_{K-1},j_{K-1})} \circ \ldots \circ T^M_{(i_1,j_1)}$, is no longer a coupling sequence for the original system. Consider the network in Figure 4 (with $C = (2, 2, 2, 2, 2)$ and $M = 4$) and the following spanning tree, rooted in queue 2: $i = (i_1, \ldots, i_4) = (5, 1, 3, 4)$ and $j = (j_1, \ldots, j_4) = (1, 2, 4, 2)$. Then $\widehat{T} = T^M_{(i_4,j_4)} \circ \ldots \circ T^M_{(i_1,j_1)} = T^M_{(4,2)} \circ T^M_{(3,4)} \circ T^M_{(1,2)} \circ T^M_{(5,1)}$. We get $\widehat{T}(2, 2, 0, 0, 0) = (2, 2, 0, 0, 0) \neq (0, 2, 0, 2, 0) = \widehat{T}(0, 2, 2, 0, 0)$. Intuitively, the customers remain "blocked" in different branches of the spanning tree due to the finite capacity of their common ancestors (queue 2 in our example).

To avoid this, the idea is to "connect" different branches of the tree. Instead of trying to send all the customers to the root of a spanning tree, we will send them across a path that connects all the queues.

As we assume that the network is strongly connected, it is possible to find a path that visits all the queues at least once. Consider a shortest path (for path inclusion) with this property. In particular, this is not a cycle and the first and last queues are visited only once. We also assume that the queues are numbered in their order of appearance in the path and denote by $w = 1, 2, 3, i_4, \ldots, i_{n-1}, K$ this path (we know that the first queue is queue 1, the second 2 and the third 3, and the last queue is K). There might exist p and q such that $i_p = i_q$ and $p \neq q$. In the network of Figure 4, we can choose $w = 1, 2, 3, 4, 2, 5$.

The high level idea for the general case is to send customers one by one across this path and they will end up first filling queue K, the last queue of the path, then queue i_{n-1} etc. More formally, if we consider the original Markov chain, the sequence

$$(t^{C_K}_{(i_{n-1},K)} \circ t^{C_{i_{n-1}}}_{(i_{n-2},i_{n-1})} \ldots \circ t^{C_{i_4}}_{(3,i_4)} \circ t^{C_3}_{(2,3)} \circ t^{C_2}_{(1,2)})^M$$

is a coupling sequence of transitions. However, finding a coupling sequence for diagrams is more complex. As diagrams are super-representatives of the subsets of states, we will first need to make sure that diagrams become exact representatives of subsets of states for some initial prefix of this path in order to map the coupling sequence on the states to a coupling sequence on the diagrams. We first formalize this idea of exact representatives on a prefix of the path.

Saturated states. The path consisting of the p first nodes of w is denoted w_p. *Saturated states* for the prefix w_p are the states where the customers in the queues appearing in w_p are all concentrated at the end of the path w_p. We now give a formal definition of these states.

First recall that when considering any state of $\mathcal{S}$, the nodes (c, ℓ) that can be reached by a path from $(0, 0)$ satisfy

$$\ell_{\min} = \max(0, M - \sum_{i=c+1}^{K} C_i) \leq \ell \leq \min(M, \sum_{i=1}^{c} C_i) = \ell_{\max}.$$

Let $j_1 = i_p, \ldots, j_c = 1$ be the order of appearance of the queues in w_p read from right to left. For each $\ell \in [\ell_{\min}, \ell_{\max}]$, a saturated state x_ℓ satisfies

$$x_{j_q} = \max(0, \min(C_{j_q}, \ell - \sum_{i=1}^{q-1} C_i)). \tag{1}$$

For each $\ell \in [\ell_{\min}, \ell_{\max}]$, the number of customers of a saturated state is fixed in each of the c first columns and there are no constraints for the other queues.

The saturated diagram for w_p is the diagram of the saturated states for w_p and is denoted D_p.

For the network of Figure 4, the saturated states for $w_2 = 1, 2$ are the paths represented by the diagram of Figure 5 (left) and the saturated states for $w_5 = 1, 2, 3, 4, 2$ are the paths represented by the diagram of Figure 5 (right). Note that even if those two paths end with the same queue, the saturated states are completely different. Indeed, in the first case, the constraints for the customers only concern queues 1 and 2, whereas in the second case, the constraints concern all queues except the last one.

Figure 5: Saturated diagrams for the network of Figure 4.

Tree property. We now define a property of the diagrams that will be maintained by applying a special sequence of $T_{i,j}$. Saturated diagrams satisfy this property.

A diagram $D = (N, A)$ is said to satisfy the *tree property* $(\mathcal{T}_c)$ if for all $c' \leq c$ such that (c', ℓ) is reachable,

1. The set $I(c', \ell) = \{s \mid ((c'-1, \ell), (c', \ell + s)) \in A\}$ is an interval denoted $I(c', \ell) = [\underline{s}(c', \ell), \overline{s}(c', \ell)]$;
2. If $(c'-1, \ell+1)$ is reachable, then $\overline{s}(c', \ell) = \underline{s}(c', \ell+1)$.

A diagram that satisfies the tree property $(\mathcal{T}_c)$ has the following properties:

- The structure of the diagram of the c first columns is a tree of root $(0,0)$ (each node on a path from $(0,0)$ to any node of column c has in-degree 1);
- There is a one-to-one correspondence between the paths of the diagrams from $(0,0)$ to column c and the restrictions of the states to the c first queues;
- A diagram with the tree property $(\mathcal{T}_K)$ contains only one path: it is the representative of a single state.

Obviously, if a diagram D satisfies $(\mathcal{T}_c)$, then it also satisfies $(\mathcal{T}_{c'})$ for all $c' \leq c$.

On Figure 5, D_2 and D_5 respectively satisfy $(\mathcal{T}_2)$ and $(\mathcal{T}_4)$. More generally, we have the following property:

Lemma 3. *The diagram D_p satisfies $(\mathcal{T}_c)$, where $c = \max(i_k \ : \ 1 \leq k \leq p)$.*

Proof. The proof follows from (1) and the fact that the queues are numbered according to their order of appearance in the path. □

We now state the two key lemmas of the proof.

Lemma 4. *If D satisfies property $(\mathcal{T}_c)$ and $i, j \leq c$, then,*

$$\psi(T_{i,j}(D)) = t_{i,j}(\psi(D))$$

and $T_{i,j}(D)$ also satisfies property $(\mathcal{T}_c)$.

Proof. We consider the case $i < j \leq c$. The case $j < i \leq c$ is similar.

We denote by A_k the arcs between columns $k-1$ and k: $A_k = A \cap \{((k-1,\ell),(k,\ell')) \in \mathcal{A}\}$ and $\mathcal{E}mpty_k$, $\mathcal{F}ull_k$ and $\mathcal{T}ransit_k$ the arcs of A_k in respectively $\mathcal{E}mpty$, $\mathcal{F}ull$ and $\mathcal{T}ransit$. First notice that if $\mathcal{T}ransit_j = \emptyset$, $T_{i,j}(D) = D$, so only the case where $\mathcal{T}ransit_j \neq \emptyset$ needs to be investigated. In that case, the following properties are satisfied:

- $\forall k \in \{i, \ldots, j\}$, $\mathcal{E}mpty_k \cap (\mathcal{T}ransit_k \cup \mathcal{F}ull_k) = \emptyset$;
- $\mathcal{T}ransit_j \cap \mathcal{F}ull_j = \emptyset$ and $\forall k \in \{i, \ldots, j-1\}$, $|\mathcal{T}ransit_k \cap \mathcal{F}ull_k| = 1$.

Indeed, in column $i-1$, from property $(\mathcal{T}_c)$, there exists ℓ_i^e such that $\forall \ell < \ell_i^e$, $I(i,\ell) = \{0\}$, $\forall \ell > \ell_i^e$, $\{0\} \notin I(i,\ell)$ and $\{0\} \subsetneq I(i, \ell_i^e)$. Then the arcs in $\mathcal{E}mpty_k$ are the ones that also belong to a path containing $((i-1,\ell),(i,\ell))$, with $\ell \leq \ell_i^e$ and the arcs in $\mathcal{T}ransit_k \cup \mathcal{F}ull_k$ are the ones on the paths not containing those arcs. Then because of property $(\mathcal{T}_c)$, $\mathcal{E}mpty_k \cap (\mathcal{T}ransit_k \cup \mathcal{F}ull_k) = \emptyset$ for all $k \in \{i, \ldots, j\}$. For each $k \in \{i, j\}$, we set $\ell_k^e = max\{\ell \mid \exists \ell' \text{ s.t. } ((k-1,\ell),(k,\ell')) \in \mathcal{E}mpty_k\}$.

Next, by construction $\mathcal{T}ransit_j \cap \mathcal{F}ull_j = \emptyset$ also holds for any diagram. Moreover, in any other column $k \in \{i, \ldots, j-1\}$, $|\mathcal{T}ransit_j \cap \mathcal{F}ull_j| = 1$. Indeed, there exists a unique ℓ_j^f such that $((j-1, \ell_j^f), (j, \ell_j^f + C_j - 1)) \in \mathcal{T}ransit_j$ and $((j-1, \ell_j^f), (j, \ell + C_j)) \in \mathcal{F}ull_j$. As the in-degree of node $(j-1, \ell_j^f)$ is one (tree property), there exists a unique arc $((j-2, \ell_{j-1}^f), (j-1, \ell_j^f)) \in \mathcal{F}ull_{j-1} \cap \mathcal{T}ransit_{j-1}$, and similarly, $\forall k \geq i$, there exists a unique ℓ_k^f such that $((k-1, \ell_k^f), (k, \ell_{k+1}^f)) \in \mathcal{F}ull_k \cap \mathcal{T}ransit_k$.

Now, following Theorem 2 and denoting $I'(c',\ell) = \{s \mid ((c'-1,\ell),(c',\ell+s)) \in A'\}$, we are ready to compute $D' = (N, A') = T_{i,j}(D)$, which is given by the following table:

	$I'(i,\ell)$	$I'(k,\ell)$	$I'(j,\ell)$
$\ell < \ell_k^e$	$I(i,\ell) = \{0\}$	$I(k,\ell)$	$I(j,\ell)$
$\ell = \ell_k^e$	$[\underline{s}(i,\ell), \overline{s}(i,\ell) - 1]$	$I(k,\ell) \cup I(k,\ell+1)$	$I(k,\ell) \cup [\underline{s}(j,\ell+1)+1, \overline{s}(j,\ell+1)+1]$
$\ell_k^f - 1 > \ell > \ell_k^e$	$[\underline{s}(i,\ell) - 1, \overline{s}(i,\ell) - 1]$	$I(k,\ell+1)$	$[\underline{s}(j,\ell+1)+1, \overline{s}(j,\ell+1)+1]$
$\ell = \ell_k^f - 1$	$[\underline{s}(i,\ell) - 1, \overline{s}(i,\ell) - 1]$	$I(k,\ell+1) \cap [0, \ell_{k+1}^f - \ell_k^f]$	$[\underline{s}(j,\ell+1)+1, C_j]$
$\ell = \ell_k^f$	$[\underline{s}(i,\ell) - 1, \overline{s}(i,\ell)]$	$I(k,\ell+1) \cap [\ell_{k+1}^f - \ell_k^f, M]$	$\{C_j\}$
$\ell > \ell_k^f$	$I(i,\ell)$	$I(k,\ell)$	$I(j,\ell) = \{C_j\}$

It can be easily checked that D' then satisfies property $(\mathcal{T}_c)$.

□

A direct consequence of this lemma is that if D is a representative of S, $\psi(T_{i,j}(D)) = \psi(T_{i,j}(\phi(S))) = t_{i,j}(\psi(\phi(S))) = t_{i,j}(S)$ and $T_{i,j}(D)$ is a representative of $t_{i,j}(S)$.

Figure 6 illustrates Lemma 4: we consider the saturated diagram D_4 and compute $T_{4,2}(D_4)$. We have $w_4 = 1,2,3,4$ and $c = 4$. Lemma 3 implies property $(\mathcal{T}_4)$. Lemma 4 implies in particular that this property is preserved by $T_{4,2}$.

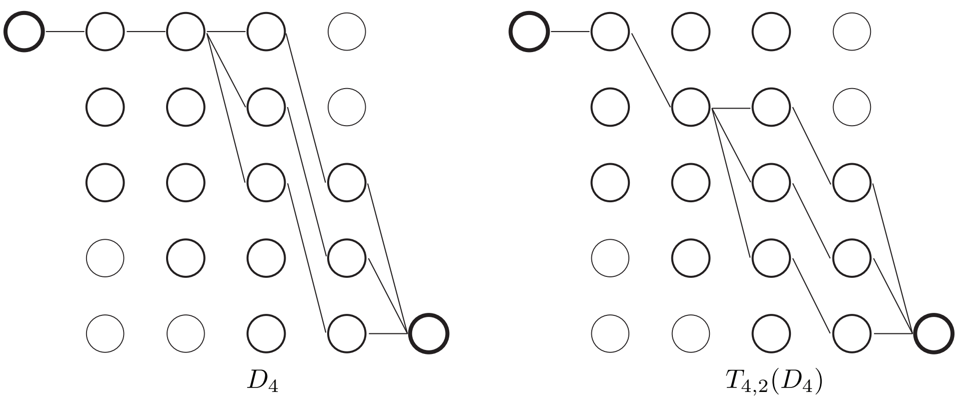


Figure 6: Computation of $T_{4,2}(D_4)$.

It remains to study the behavior of a transition of the diagram satisfying $(\mathcal{T}_c)$ when queue $c+1$ is involved. We only need to consider the transitions $T_{i,c+1}$ with $i \le c$ as we will only use such transitions. The idea is to define a non-negative quantity $RC(D)$ that decreases each time $T_{i,c+1}$ is performed. In the proof of the main theorem, this will be used to show that when this quantity becomes stationary, then a saturated diagram is obtained for a longer prefix.

Fix c and define the quantity

$$RC(D) = \sum_{a \in A_{c+1}} C_{c+1} - v(a),$$

that is the sum on all the arcs of column $c+1$ of the remaining capacity of queue $c+1$. Note that for all arcs in A_{c+1}, $C_{c+1} - v(a) \ge 0$, so $RC(D) \ge 0$.

Lemma 5. *Suppose $D \preceq D_p$ and satisfies $(\mathcal{T}_c)$ with $c = \max(i_k : 1 \le k \le p)$. Then $RC(T_{i_p,c+1}(D)) \le RC(D)$, with equality if and only if D satisfies property $(\mathcal{T}_{c+1})$, $D \preceq D_{p+1}$ and $T_{i_p,c+1}(D) = D$.*

Proof. Let us first focus on $RC(T_{i,c+1}(D)) \le RC(D)$. Set $D' = (N, A') = T_{i_p,c+1}(D)$ and assume that $D' \ne D$. We show that there is a surjective relation h between A_{c+1} and A'_{c+1} such that $\forall a \in A_{c+1}$, $v(a) \le v(h(a))$. Then, $C_{c+1} - v(a) \ge C_{c+1} - v(h(a))$, hence the inequality. Similarly to the proof of

Lemma 4, $\mathcal{E}mpty_{c+1} \cap (\mathcal{F}ull_{c+1} \cup \mathcal{T}ransit_{c+1}) = \emptyset$. For $a \in \mathcal{E}mpty_{c+1}$, we set $h(a) = a$. By definition, $\mathcal{F}ull_{c+1} \cap \mathcal{T}ransit_{c+1} = \emptyset$. We set $h(a) = a$ if $a \in \mathcal{F}ull_{c+1}$ and if $a = ((c, \ell), (c+1, \ell')) \in \mathcal{T}ransit_{c+1}$, we set $h(a) = ((c, \ell - 1), (c+1, \ell')) \in A'_{c+1}$. Because $\mathcal{E}mpty_{c+1}$, $\mathcal{F}ull_{c+1}$ and $\mathcal{T}ransit_{c+1}$ partition A_{c+1}, $h(A_{c+1}) = A'_{c+1}$. So h is a surjective relation and it can be easily checked that $\forall a \in A_{c+1}$, $v(a) \le v(h(a))$.

Now suppose that $RC(T_{i_p,c+1}(D)) = RC(D)$. This means that $\forall a \in A_{c+1}$, $h(a) = a$ and $T_{i_p,c+1}(D) = D$. So every path of $\Pi(D)$, it is in $\mathcal{F}ull$ or $\mathcal{E}mpty$. If it is in $\mathcal{F}ull$, then the state is saturated (because $D \preceq D_p$). If it is in $\mathcal{E}mpty$, then every other queue $< c+1$ is also empty (we know that i_p is empty and $D \preceq D_p$), which means that this state is also saturated: $D \preceq D_{p+1}$.

□

We are now ready to show the following theorem:

Theorem 4. *There is a function* $T = T_{k_{s-1},k_s} \circ \cdots \circ T_{k_1,k_2}$ *such that* $|\Pi(T(\mathcal{D}))| = 1$.

Proof. We proceed by induction on the prefixes w_p of w and show that there exists a function such that $T_p(\mathcal{D}) = D_p$.

Initialization: Any diagram satisfies property $(\mathcal{T}_1)$ and any diagram is saturated for w_1.

Induction: Suppose there exists a function T_p that reaches D_p. Let t_p be the corresponding function that transforms the state space. Two cases need to be considered:

a) i_{p+1} *appears in* w_p*:* According to Lemma 3, D_p satisfies property $(\mathcal{T}_c)$, where $c = \max(i_k \ : \ 1 \le k \le p)$. Then, according to Lemma 4, by applying function $T_{i,j}$, with $i, j \le c$ to D_p, exact representatives are computed. Let $q = \max\{r \le p \mid i_r = i_{p+1}\}$.

$$\tilde{T}(D_p) = (T_{i_p,i_p+1} \circ T_{i_{p-1},i_p} \circ \cdots \circ T_{i_{q+1},i_{q+2}})^{C_{i_{p+1}}}(D_p) = D_{p+1}$$

is a saturated diagram for w_{p+1}. It is sufficient to show that for every $x \in \psi(D_p)$, $\tilde{t} \in \psi(D_{p+1})$ and that every saturated state of $\psi(D_{p+1})$ can be obtained. Note that $(t_{i_p,i_p+1} \circ t_{i_{p-1},i_p} \circ \cdots \circ t_{i_{q+1},i_{q+2}})$ moves one customer from queue i_{q+1} if any to the last queue of $i_{q+1}, \ldots, i_{p+1}$ that is not full, hence the result. Indeed, either a state is already saturated (for w_{p+1}) and queues $i_{q+1}, \ldots, i_{p+1}$ are full, or it is not saturated, and all the queues in $\{i_1, \ldots, i_{q-1}\} \backslash \{i_{q+1}, \ldots, i_{p+1}\}$ are empty.

b) i_{p+1} *does not appear in* w_p*:* Let $D^n = (T_{i_p,i_{p+1}} \circ T_p)^n(\mathcal{D})$ and $D'^n = T_p \circ (T_{i_p,i_{p+1}} \circ T_p)^{n-1}(\mathcal{D})$. $D'^n \preceq D_p$. According to Lemma 5, the sequence $RC(D_n)$ is decreasing (T_p does not modify the arcs in A_{c+1}), bounded above by 0, so it is ultimately stationary. Moreover, when it is stationary, diagram D_{p+1} is reached. Let n_0 be the rank from which $RC(D^n)$ is stationary. We can take

$$T_{p+1} = (T_{i_p,i_{p+1}} \circ T_p)^{n_0}.$$

Conclusion: There exists a sequence such that $T(\mathcal{D})$ is saturated for w. As w is a path that visits each queue at least once, this implies property $(\mathcal{T}_K)$ (Lemma 3). This diagram then contains exactly one path. □

5. Numerical experiments

We investigate the coupling time of diagram CFTP (Algorithm 2) on two small examples. We call the *coupling time* (CT) the value of n when the algorithm stops.[1]

The coupling time of Algorithm 2 is greater than that of Algorithm 1. This is because the diagrams computed in Algorithm 2 are super-representatives of the sets of states that are obtained in Algorithm 1, provided each uses the same sequence of variables (U_n).

[1]In Algorithms 1 and 2, at each iteration of the while loop the value of n is doubled. While it is not necessary to double n in the classical (non-monotone) CFTP (Algorithm 1), this adds at most a multiplicative factor 2 to the exact coupling time. In the new diagram CFTP it is mandatory to double n since this avoids quadratic dependence on the coupling time (for similar reasons as in the monotone case, see [3]).

We will see that in our experiments the algorithmic cost induced by this is negligible compared to the algorithmic gain of one-step computation of the algorithm when using diagrams.

As our main objective is to compare the coupling times and the running times of diagram CFTP (Algorithm 2) and the classical Propp and Wilson CFTP [3] (Algorithm 1), we consider only small examples for which Algorithm 1 can be performed in reasonable time.

The first example is a network of 8 queues with customer population limited to 35. The second example is a ring network, a special case for which there is another effective CFTP algorithm that is based on identifying a set of extremal trajectories [13]. This method cannot be applied if there is a queue that has more than one successor. In the closed queueing networks setting, this constraint implies a ring network topology.

Our experiments are performed on a desktop machine using Matlab.

5.1. Small network

Consider the network in Figure 7 with $K = 8$ queues. This is a model of a store-and-forward packet switching network from [12].

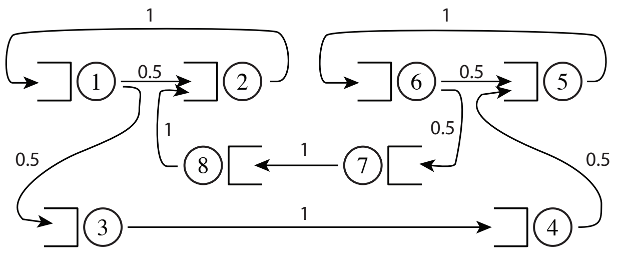


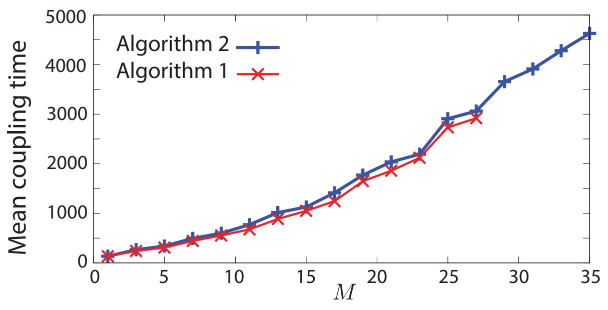
Figure 7: Queueing network with $K = 8$ queues.

The mean coupling times and the running times (in seconds) for Algorithm 1 and Algorithm 2 are reported in Figure 8 when the number M of customers in the network takes values 1,3,5,...,35. For a given M, the capacity of each queue is $\lceil \frac{M}{2} \rceil$, and each point of the graph is the average of 100 independent runs.

The coupling times are very close — their ratio is between 1 and 1.2. This confirms that while the diagrams are not exact representatives of sets of states, their coupling time is close to that of the original system.

The solidarity of coupling times in the two models depends on the choice of ordering of queues. Intuitively, transitions in the diagram are more "informative" if the queues in the diagram are close together. In particular, this is why we chose to put queues that have routing probabilities 1 next to one another.

Although the coupling times are close, Algorithm 1 does not stop in reasonable time (less than one day) in our experiment when $M > 27$. Algorithm 2 is not so sensitive to M for this network: the running time of the algorithm is less than 2.3 sec. when $M = 35$.



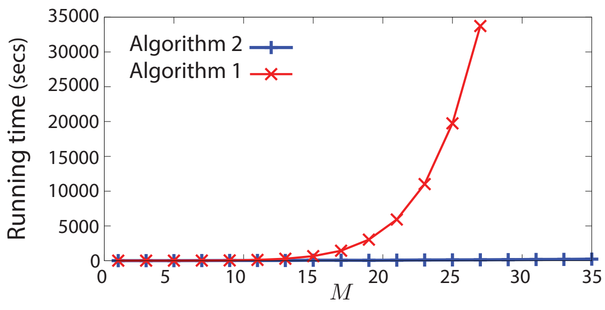


Figure 8: Comparison of Algorithm 1 and 2 for the network from Figure 7. Left: mean coupling time; right: running time (for 100 runs).

5.2. Ring network

Consider the *ring network* in which $p_{i,j} = 1$ iff $j = (i+1) \pmod{K}$. In this particular network there exists a set of *extremal states* of cardinality M, that are the states where all the customers are in the same queue. The coupling time of trajectories issued from this set of states is the same as the exact coupling time of all the trajectories [13]. Thus, in this particular network we can compare the exact coupling time with the coupling time of Algorithm 1.

In these experiments, K takes values 1,3,5,...,25. Two cases are studied: $M = 2K$ and $M = 5K$, and the capacity of the queues are $2\frac{K}{M}$, that is, respectively 4 and 10. Each point on the graph is the mean of 100 simulations.

The ratio of the mean coupling times of Algorithms 1 and 2, and the mean of the coupling times in Algorithms 1 and 2 are reported in Figure 9.

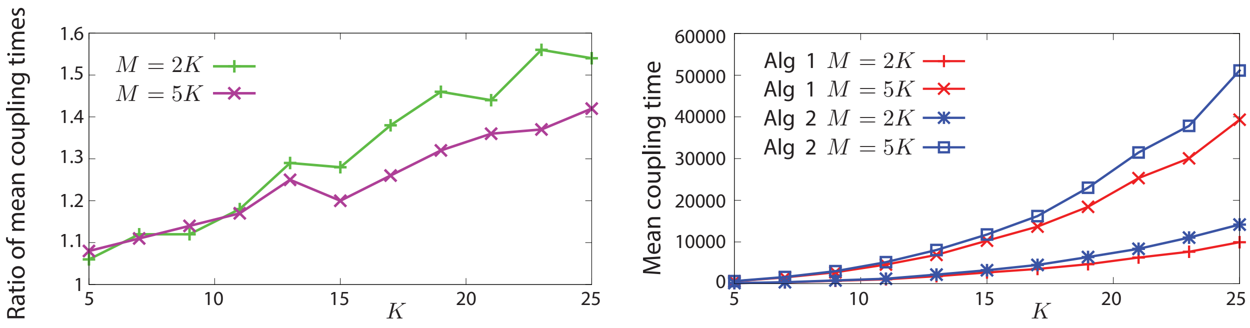


Figure 9: Comparison of Algorithms 1 and 2 for the ring network. Left: ratio of mean coupling times; right: mean coupling times.

The ratios are close to 1, grow with K and M, and are always between 1 and 2. Remarkably, this ratio seems to decrease with M/K. Moreover, the coupling time of both algorithms grows exponentially with K and M.

6. Conclusion

The main contribution of this paper is deriving an effective CFTP algorithm for closed queueing networks. Our approach does not assume a product form or some specific topology of the network, unlike all the previous results in the literature.

For simplicity of exposition, we focused here on single server queues. Extension to multiserver case is straightforward, at least for the following two results: 1) almost sure termination of the algorithm and 2) polynomial computational complexity of the transition function. The main difference is the definition of events. In the single server case, there is at most one routing event per couple of queues. If there are n servers, one can define n events per couple of queues so that all the events have state-independent probabilities. This will however potentially add a factor n to the complexity of the one-step transition function.

The other possible extension is to consider closed queueing networks with other types of routing, or other models where each event can modify only a very limited number of components and only by some bounded amount.

Another direction for future research is to investigate more closely the implementation of the transition function for the diagrams. We conjecture that, under reasonable assumptions, there is an algorithm for the transition function that is in $O(KM)$.

The major theoretical challenge is the analytical analysis of the coupling times.

Acknowledgments

This research is supported by the French National Research Agency grant ANR-12-MONU-0019.

[1] S. Asmussen, P. W. Glynn, Stochastic Simulation: Algorithms and Analysis, Vol. 57 of Stochastic Modelling and Applied Probability, Springer-Verlag, New York, 2007.

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

[3] J. Propp, D. Wilson, Exact sampling with coupled Markov chains and applications to statistical mechanics, Random Structures and Algorithms, Dordrecht/Boston/London, 1996.

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

[5] W. Kendall, J. Mller, Perfect simulation using dominating processes on ordered spaces, with application to locally stable point processes, Advances in Applied Probability 32 (3) (2000) 844–865.

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

[7] K. Sigman, Exact simulation of the stationary distribution of the fifo m/g/c queue: the general case for ¡c, Queueing Systems 70 (1) (2012) 37–43.

[8] J. Blanchet, A. Wallwater, Exact Sampling of Stationary and Time-Reversed Queues, ArXiv e-printsarXiv:1403.8117.

[9] W. Gordon, G. Newel, Closed queueing systems with exponential servers, Oper. Res. 15,2 (1967) 254–265.

[10] J. Buzen, Computational algorithms for closed queueing networks with exponential servers, Comm. ACM 16 (1973) 527–531.

[11] R. O. Onvural, Survey of closed queueing networks with blocking, ACM Comput. Surv. 22 (2) (1990) 83–121.

[12] S. Balsamo, Queueing networks with blocking: Analysis, solution algorithms and properties, in: D. D. Kouvatsos (Ed.), Network Performance Engineering, Vol. 5233 of Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2011, pp. 233–257.

[13] A. Bouillard, B. Gaujal, Backward coupling in bounded free-choice nets under markovian and non-markovian assumptions, Discrete Event Dynamic Systems 18 (4) (2008) 473–498.

[14] G. Birkhoff, Lattice theory, 3rd Edition, Vol. 25 of American Mathematical Society Colloquium Publications, American Mathematical Society, Providence, R.I., 1979.