Omnithermal perfect simulation for multi-server queues

Stephen Connor
stephen.connor@york.ac.uk

University of York

LMS-EPSRC Durham Symposium
July-August 2017

Dominated CFTP in a nutshell

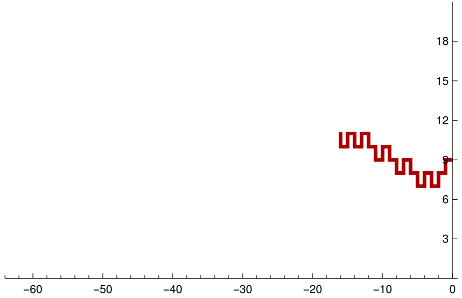
Suppose that we're interested in simulating from the equilibrium distribution of some ergodic Markov chain X.

Think of a (hypothetical) version of the chain, $\tilde{X}$, which was started by your (presumably distant) ancestor from some state x at time $-\infty$:

- at time zero this chain is in equilibrium: $\tilde{X}_0 \sim \pi$;
- dominated CFTP (domCFTP) tries to determine the value of $\tilde{X}_0$ by looking into the past only a *finite* number of steps;
- do this by identifying a time in the past such that *all earlier starts from x lead to the same result at time zero*.

domCFTP: basic ingredients

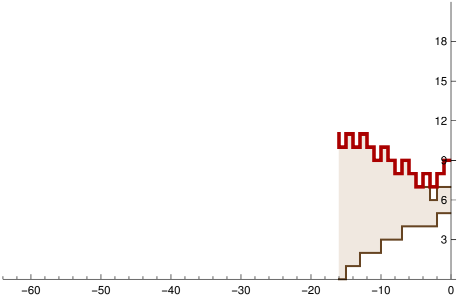
- *dominating process Y*
  - draw from equilibrium π_Y
  - simulate backwards in time



domCFTP: basic ingredients

- *dominating process Y*
  - draw from equilibrium π_Y
  - simulate backwards in time
- *sandwiching*

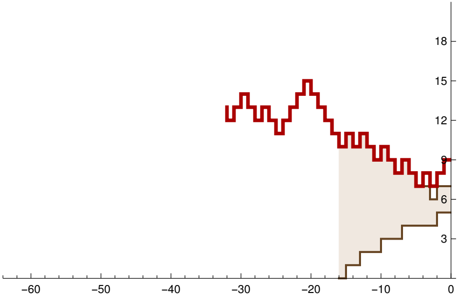
$$\text{Lower}_{\text{late}} \preccurlyeq \text{Lower}_{\text{early}} \preccurlyeq \ldots \preccurlyeq \text{Target} \preccurlyeq \ldots \preccurlyeq \text{Upper}_{\text{early}} \preccurlyeq \text{Upper}_{\text{late}}$$



domCFTP: basic ingredients

- *dominating process Y*
  - draw from equilibrium π_Y
  - simulate backwards in time
- *sandwiching*

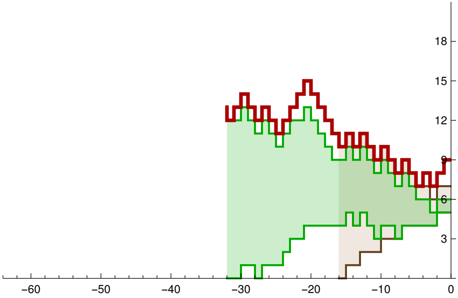
$$\text{Lower}_{\text{late}} \preccurlyeq \text{Lower}_{\text{early}} \preccurlyeq \ldots \preccurlyeq \text{Target} \preccurlyeq \ldots \preccurlyeq \text{Upper}_{\text{early}} \preccurlyeq \text{Upper}_{\text{late}}$$



domCFTP: basic ingredients

- *dominating process Y*
  - draw from equilibrium π_Y
  - simulate backwards in time
- *sandwiching*

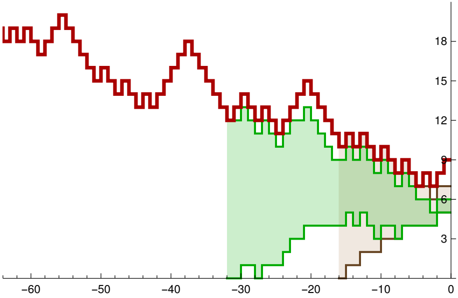
$$\text{Lower}_{\text{late}} \preccurlyeq \text{Lower}_{\text{early}} \preccurlyeq \ldots \preccurlyeq \text{Target} \preccurlyeq \ldots \preccurlyeq \text{Upper}_{\text{early}} \preccurlyeq \text{Upper}_{\text{late}}$$



domCFTP: basic ingredients

- *dominating process Y*
  - draw from equilibrium π_Y
  - simulate backwards in time
- *sandwiching*

$$\text{Lower}_{\text{late}} \preccurlyeq \text{Lower}_{\text{early}} \preccurlyeq \ldots \preccurlyeq \text{Target} \preccurlyeq \ldots \preccurlyeq \text{Upper}_{\text{early}} \preccurlyeq \text{Upper}_{\text{late}}$$



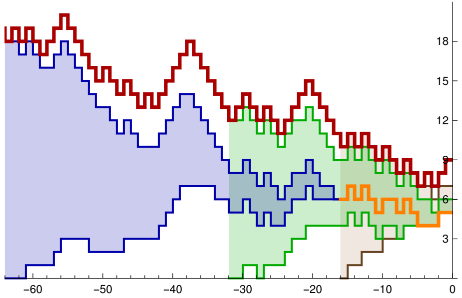
domCFTP: basic ingredients

- *dominating process Y*
  - draw from equilibrium π_Y
  - simulate backwards in time
- *sandwiching*

$$\text{Lower}_{\text{late}} \preccurlyeq \text{Lower}_{\text{early}} \preccurlyeq \ldots \preccurlyeq \text{Target} \preccurlyeq \ldots \preccurlyeq \text{Upper}_{\text{early}} \preccurlyeq \text{Upper}_{\text{late}}$$

- *coalescence*

  eventually a Lower and an Upper process must coalesce



$M/G/c$ queue

- Customers arrive at times of a Poisson process: interarrival times $T_n \sim \mathrm{Exp}(\lambda)$
- Service durations S_n are i.i.d. with $\mathbb{E}[S] = 1/\mu$ (and $\mathbb{E}\left[S^2\right] < \infty$)
- Customers are served by c servers, on a First Come First Served (FCFS) basis

Queue is *stable* iff $\rho := \dfrac{\lambda}{\mu c} < 1$.

The (ordered) workload vector just before the arrival of the n^{th} customer satisfies the *Kiefer-Wolfowitz* recursion:

$$\boxed{\mathbf{W}_{n+1} = R(\mathbf{W}_n + S_n\delta_1 - T_n\mathbf{1})^+ \quad \text{for } n \geq 0}$$

- add workload S_n to first coordinate of $\mathbf{W}_n$ (server currently with least work)
- subtract T_n from every coordinate (work done between arrivals)
- reorder the coordinates in increasing order
- replace negative values by zeros.

Aim

Sample from the equilibrium distribution of this workload vector

DomCFTP for queues

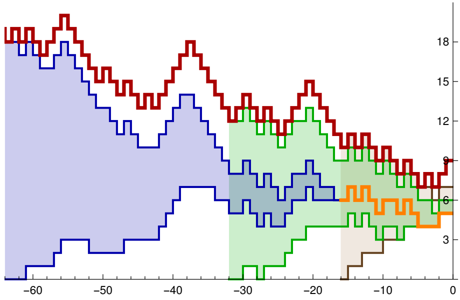
We need to find a dominating process for our $M_\lambda/G/c$ $[FCFS]$ queue X.

C & Kendall (2015): dominate with $M/G/c$ $[RA]$

- RA = **random assignment**, so c independent copies of $M_{\lambda/c}/G/1$
- Evidently stable iff $M/G/c$ is stable
- Easy to simulate in equilibrium, and in reverse
- Care needed with domination arguments: service durations must be assigned **in order of initiation of service**

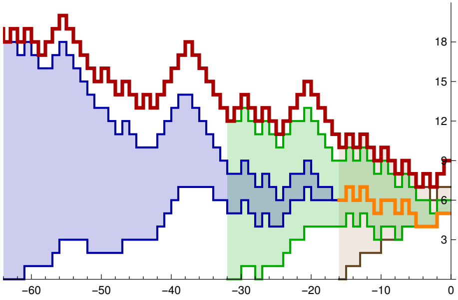
domCFTP algorithm

- Dominating process Y is stationary $M/G/c$ **[RA]** queue



domCFTP algorithm

- Dominating process Y is stationary $M/G/c$ **[RA]** queue
- Check for coalescence of **sandwiching processes**, U^c and L^c:
  - these are workload vectors of $M/G/c\,[FCFS]$ queues
  - L^c starts from empty
  - U^c is instantiated using **residual workloads** from Y



Omnithermal simulation

Back to the general perfect simulation setting...

Suppose that the target process X has a distribution π_β that depends on some underlying parameter β.

In some situations it is possible to modify a perfect simulation algorithm so as to sample *simultaneously* from π_β for all β in some given range: call this **omnithermal simulation**.

E.g.

- Random Cluster model (Propp & Wilson, 1996)
- Area Interaction Process (Shah, 2004)

Omnithermal simulation

Back to the general perfect simulation setting...

Suppose that the target process X has a distribution π_β that depends on some underlying parameter β.

In some situations it is possible to modify a perfect simulation algorithm so as to sample *simultaneously* from π_β for all β in some given range: call this **omnithermal simulation**.

E.g.

- Random Cluster model (Propp & Wilson, 1996)
- Area Interaction Process (Shah, 2004)

Question

Can we perform omnithermal simulation for $M/G/c$ queues with varying numbers of servers?

Comparing queues with different numbers of servers

Consider a natural partial order between vectors of different lengths:

for $V^c \in \mathbb{R}^c$ and $V^{c+m} \in \mathbb{R}^{c+m}$, write $V^{c+m} \preceq V^c$ if and only if

$$V^{c+m}(k+m) \leq V^c(k)\,, \quad k = 1, \ldots, c\,.$$

("Busiest c servers in V^{c+m} each no busier than corresponding server in V^c".)

Comparing queues with different numbers of servers

Consider a natural partial order between vectors of different lengths:

for $V^c \in \mathbb{R}^c$ and $V^{c+m} \in \mathbb{R}^{c+m}$, write $V^{c+m} \preceq V^c$ if and only if

$$V^{c+m}(k+m) \leq V^c(k)\,, \quad k = 1, \ldots, c\,.$$

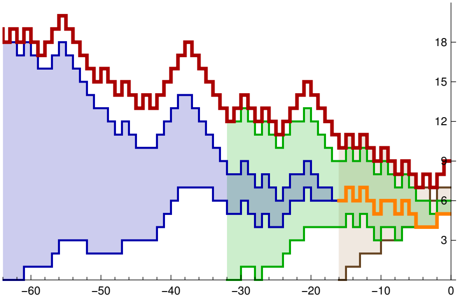
("Busiest c servers in V^{c+m} each no busier than corresponding server in V^c".)

Observation

Dynamics for workload vectors with different numbers of servers are monotonic w.r.t. this partial order

So we can produce processes U^{c+m} and L^{c+m} over $[T, 0]$, coupled to our c-server dominating process Y, such that:

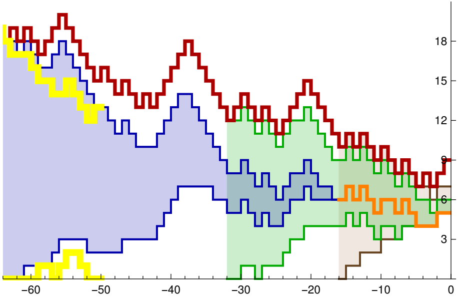
- U^{c+m} and L^{c+m} sandwich our $M/G/(c+m)$ FCFS process of interest
- $U_t^{c+m} \preceq U_t^c$ and $L_t^{c+m} \preceq L_t^c$



So we can produce processes U^{c+m} and L^{c+m} over $[T, 0]$, coupled to our c-server dominating process Y, such that:

- U^{c+m} and L^{c+m} sandwich our $M/G/(c+m)$ FCFS process of interest
- $U_t^{c+m} \preceq U_t^c$ and $L_t^{c+m} \preceq L_t^c$

But U^{c+m} and L^{c+m} **won't** necessarily coalesce before time 0!



Establishing coalescence

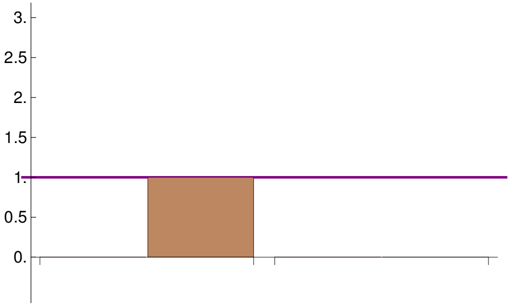
Write C_t^c for the remaining time (at time t) until coalescence of U_t^c and L_t^c **under the assumption of no more arrivals**:

$$C_t^c = U_t^c(n_t^c), \text{ where } n_t^c = \max\{1 \leq k \leq c \;:\; U_t^c(k) \neq L_t^c(k)\} \;.$$

Establishing coalescence

Write C_t^c for the remaining time (at time t) until coalescence of U_t^c and L_t^c **under the assumption of no more arrivals**:

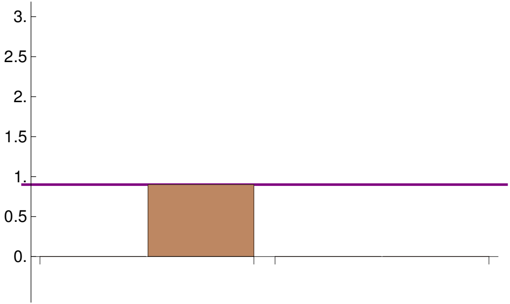
$$C_t^c = U_t^c(n_t^c), \text{ where } n_t^c = \max\left\{1 \leq k \leq c \,:\, U_t^c(k) \neq L_t^c(k)\right\} .$$



Establishing coalescence

Write C_t^c for the remaining time (at time t) until coalescence of U_t^c and L_t^c **under the assumption of no more arrivals**:

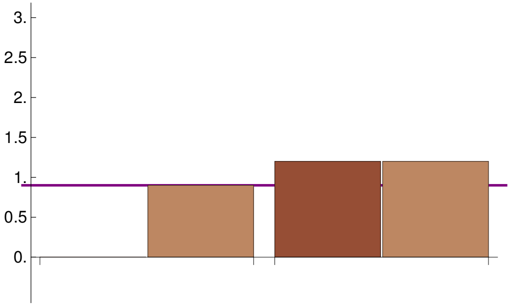
$$C_t^c = U_t^c(n_t^c), \text{ where } n_t^c = \max\{1 \le k \le c \,:\, U_t^c(k) \neq L_t^c(k)\} \ .$$



Establishing coalescence

Write C_t^c for the remaining time (at time t) until coalescence of U_t^c and L_t^c **under the assumption of no more arrivals**:

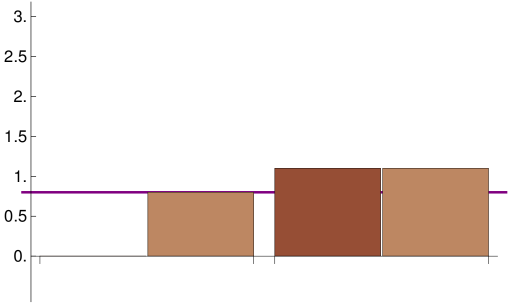
$$C_t^c = U_t^c(n_t^c), \text{ where } n_t^c = \max\left\{1 \leq k \leq c \;:\; U_t^c(k) \neq L_t^c(k)\right\} .$$



Establishing coalescence

Write C_t^c for the remaining time (at time t) until coalescence of U_t^c and L_t^c **under the assumption of no more arrivals**:

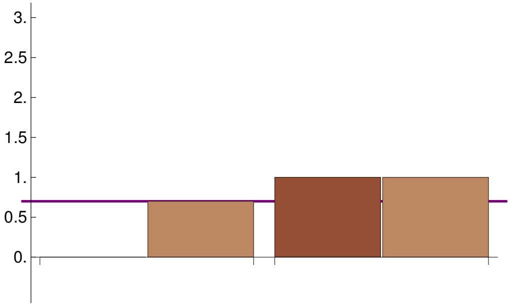
$$C_t^c = U_t^c(n_t^c), \text{ where } n_t^c = \max\{1 \le k \le c \,:\, U_t^c(k) \neq L_t^c(k)\} \;.$$



Establishing coalescence

Write C_t^c for the remaining time (at time t) until coalescence of U_t^c and L_t^c **under the assumption of no more arrivals**:

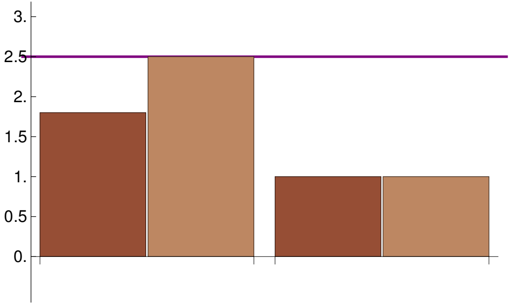
$$C_t^c = U_t^c(n_t^c), \text{ where } n_t^c = \max\left\{1 \leq k \leq c \,:\, U_t^c(k) \neq L_t^c(k)\right\} .$$



Establishing coalescence

Write C_t^c for the remaining time (at time t) until coalescence of U_t^c and L_t^c **under the assumption of no more arrivals**:

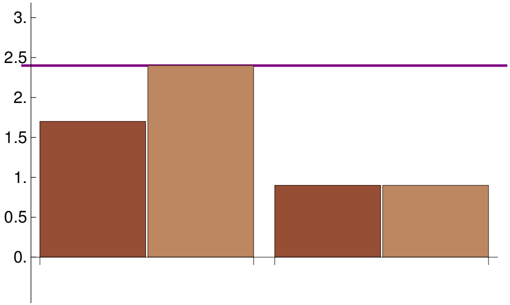
$$C_t^c = U_t^c(n_t^c), \text{ where } n_t^c = \max\left\{1 \leq k \leq c \,:\, U_t^c(k) \neq L_t^c(k)\right\} .$$



Establishing coalescence

Write C_t^c for the remaining time (at time t) until coalescence of U_t^c and L_t^c **under the assumption of no more arrivals**:

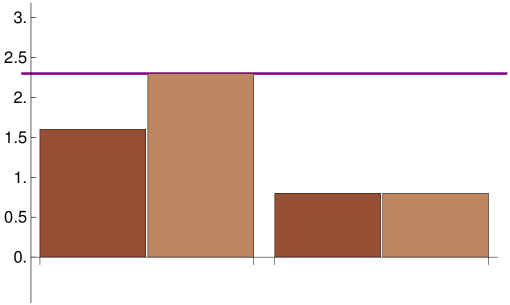
$$C_t^c = U_t^c(n_t^c), \text{ where } n_t^c = \max\{1 \le k \le c \,:\, U_t^c(k) \ne L_t^c(k)\} \,.$$



Establishing coalescence

Write C_t^c for the remaining time (at time t) until coalescence of U_t^c and L_t^c **under the assumption of no more arrivals**:

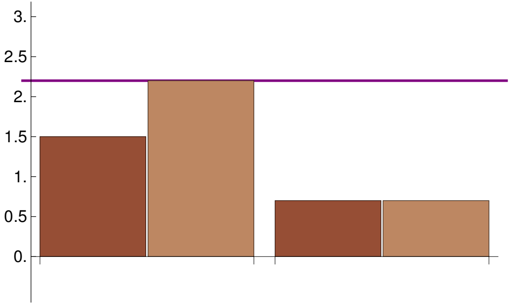
$$C_t^c = U_t^c(n_t^c), \text{ where } n_t^c = \max\{1 \leq k \leq c \,:\, U_t^c(k) \neq L_t^c(k)\} \,.$$



Establishing coalescence

Write C_t^c for the remaining time (at time t) until coalescence of U_t^c and L_t^c **under the assumption of no more arrivals**:

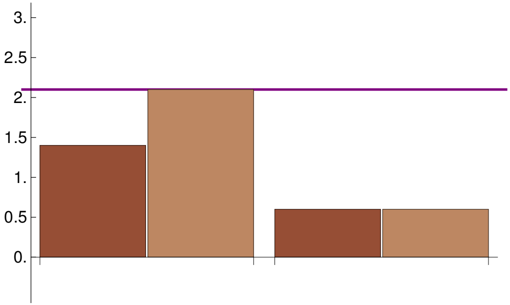
$$C_t^c = U_t^c(n_t^c), \text{ where } n_t^c = \max\left\{1 \le k \le c \,:\, U_t^c(k) \neq L_t^c(k)\right\}\,.$$



Establishing coalescence

Write C_t^c for the remaining time (at time t) until coalescence of U_t^c and L_t^c **under the assumption of no more arrivals**:

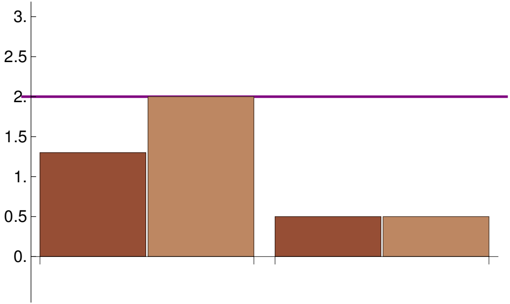
$$C_t^c = U_t^c(n_t^c), \text{ where } n_t^c = \max\{1 \leq k \leq c \,:\, U_t^c(k) \neq L_t^c(k)\} \,.$$



Establishing coalescence

Write C_t^c for the remaining time (at time t) until coalescence of U_t^c and L_t^c **under the assumption of no more arrivals**:

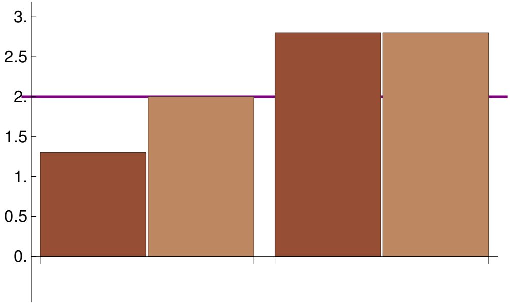
$$C_t^c = U_t^c(n_t^c), \text{ where } n_t^c = \max\{1 \leq k \leq c \,:\, U_t^c(k) \neq L_t^c(k)\} \,.$$



Establishing coalescence

Write C_t^c for the remaining time (at time t) until coalescence of U_t^c and L_t^c **under the assumption of no more arrivals**:

$$C_t^c = U_t^c(n_t^c), \text{ where } n_t^c = \max\{1 \leq k \leq c \,:\, U_t^c(k) \neq L_t^c(k)\} \; .$$



Evolution of C_t^c

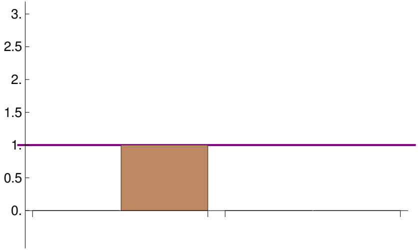
Suppose that we see an arrival (in both U^c and L^c) at time t, with associated workload S.

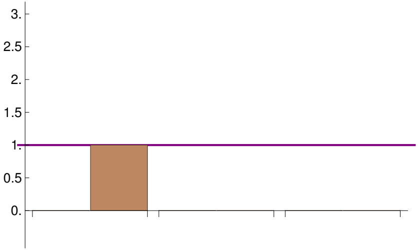
- if $U_{t-}^c(1) = L_{t-}^c(1)$ then arrival does not affect the time to coalescencence;
- if not, then we will see an increase in the time to coalescence iff the new service is placed at some coordinate $k \geq n_{t-}^c$ in U^c.

$$\boxed{C_t^c = \max\left\{ C_{t-}^c\,,\ (U_{t-}^c(1) + S)\mathbf{1}_{\left[U_{t-}^c(1) \neq L_{t-}^c(1)\right]} \right\}}$$

Problem

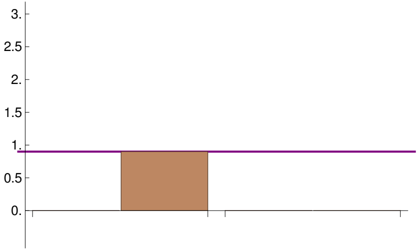
It is **not** true in general that $C_0^{c+m} \leq C_0^c \implies C_t^{c+m} \leq C_t^c$.

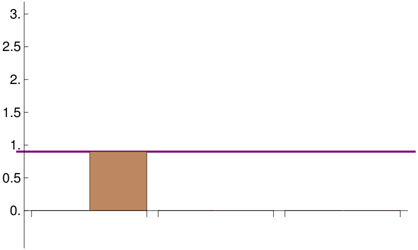




Problem

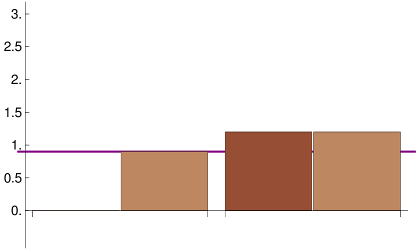
It is **not** true in general that $C_0^{c+m} \leq C_0^c \implies C_t^{c+m} \leq C_t^c$.

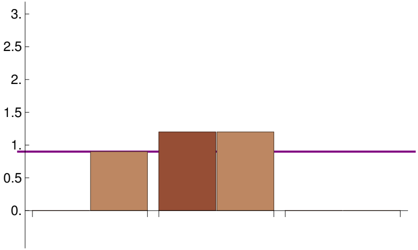




Problem

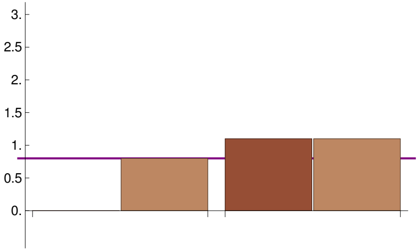
It is **not** true in general that $C_0^{c+m} \leq C_0^c \implies C_t^{c+m} \leq C_t^c$.

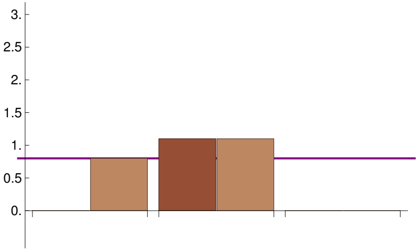




Problem

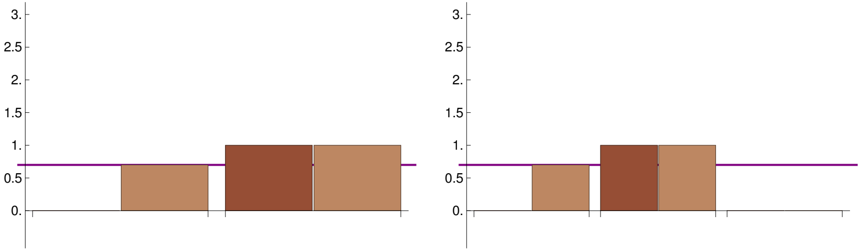
It is **not** true in general that $C_0^{c+m} \leq C_0^c \implies C_t^{c+m} \leq C_t^c$.





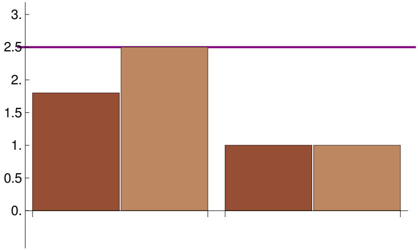
Problem

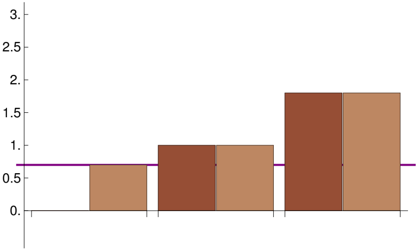
It is **not** true in general that $C_0^{c+m} \leq C_0^c \implies C_t^{c+m} \leq C_t^c$.



Problem

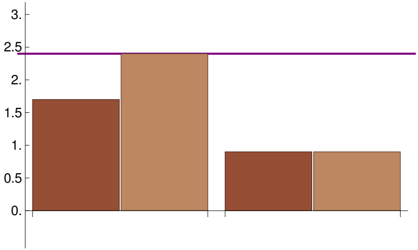
It is **not** true in general that $C_0^{c+m} \leq C_0^c \implies C_t^{c+m} \leq C_t^c$.

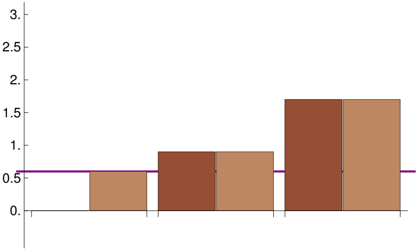




Problem

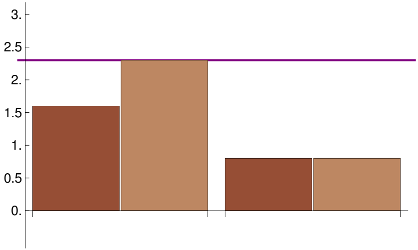
It is **not** true in general that $C_0^{c+m} \leq C_0^c \implies C_t^{c+m} \leq C_t^c$.

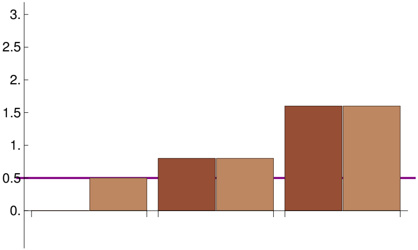




Problem

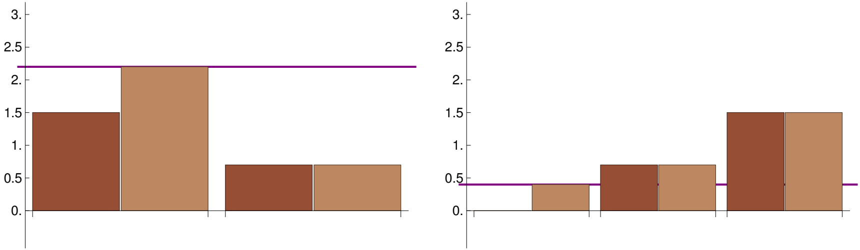
It is **not** true in general that $C_0^{c+m} \leq C_0^c \implies C_t^{c+m} \leq C_t^c$.





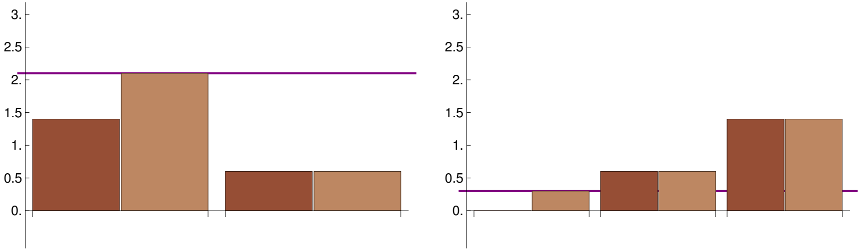
Problem

It is **not** true in general that $C_0^{c+m} \leq C_0^c \implies C_t^{c+m} \leq C_t^c$.



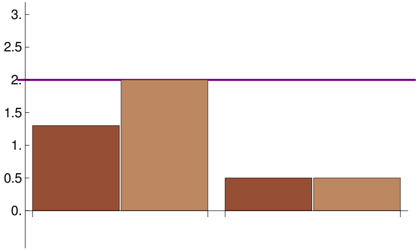
Problem

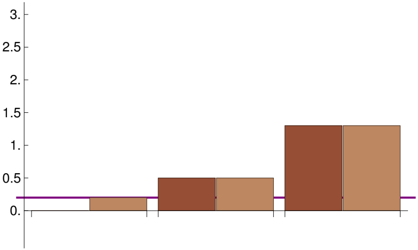
It is **not** true in general that $C_0^{c+m} \leq C_0^c \implies C_t^{c+m} \leq C_t^c$.



Problem

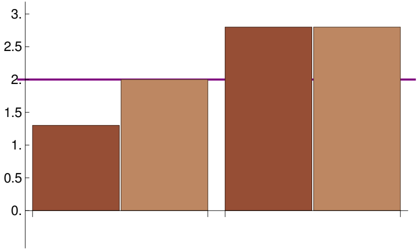
It is **not** true in general that $C_0^{c+m} \leq C_0^c \implies C_t^{c+m} \leq C_t^c$.

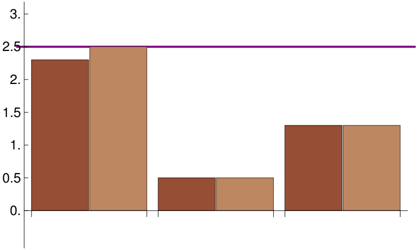




Problem

It is **not** true in general that $C_0^{c+m} \leq C_0^c \implies C_t^{c+m} \leq C_t^c$.





Solution

Write $T^c \leq 0$ for the coalescence time of U^c and L^c.

Condition A

At NO arrival time $\tau \in [T, T^c]$ do we find $L^c_{\tau-}(1) = U^c_{\tau-}(1) > 0$

Solution

Write $T^c \leq 0$ for the coalescence time of U^c and L^c.

Condition A

At NO arrival time $\tau \in [T, T^c]$ do we find $L^c_{\tau-}(1) = U^c_{\tau-}(1) > 0$

Theorem

If Condition A holds then $T^{c+m} \leq T^c$ for any $m \in \mathbb{N}$.

Solution

Write $T^c \leq 0$ for the coalescence time of U^c and L^c.

Condition A

At NO arrival time $\tau \in [T, T^c]$ do we find $L^c_{\tau-}(1) = U^c_{\tau-}(1) > 0$

Theorem

If Condition A holds then $T^{c+m} \leq T^c$ for any $m \in \mathbb{N}$.

This gives us a method for performing omnithermal domCFTP:

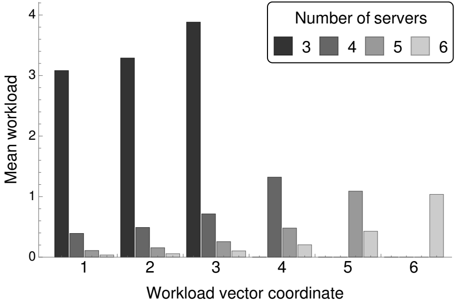
1. for a given run of the c-server domCFTP algorithm, check to see whether Condition A holds. If not, repeatedly backoff ($T \leftarrow 2T$) until Condition A is satisfied;
2. run L^{c+m} (for any $m \in \mathbb{N}$) over $[T, 0]$, and return $L^{c+m}(0)$.

Example output

Simulation results from 5,000 runs for $M/M/c$ with $\lambda = 2.85$, $\mu = 1$ and $c = 3$ ($\rho = 0.95$)

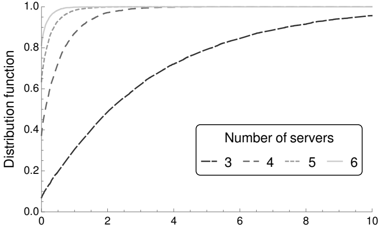
- 333 (7%) runs needed extending
- only 2 runs needed more than 2 additional backoffs

Mean workload at each server, for $c = 3$ and $m \in \{0, 1, 2, 3\}$:

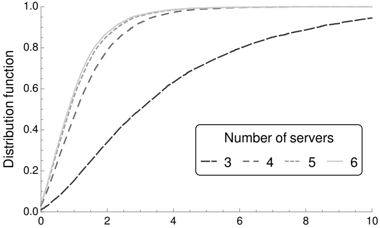


Example output

Distribution functions for workload at (a) first and (b) last coordinates of the workload vector:



(a) First coordinate workload



(b) Last coordinate workload

How expensive is this in practice?

Not very!

- Simulations indicate that Condition A is satisfied (with no need for further backoffs) $> 90\%$ of the time when $\rho \leq 0.75$, and in $> 70\%$ of cases when $\rho = 0.85$
- In addition, runs in which Condition A initially fails typically don't require significant extension
- Theoretical analysis of run-time would be nice, but hard!

Extensions

This idea can be applied in other settings.

1. Consider keeping c fixed, but increasing the rate at which servers work; same analysis as above holds.

2. Moreover, there's no need to restrict attention to Poisson arrivals! Blanchet, Pei & Sigman (2015) show how to implement domCFTP for $GI/GI/c$ queues, again using a random assignment dominating process.

Conclusions

- It is highly feasible to produce **perfect** simulations of stable $GI/GI/c$ queues using domCFTP
- Furthermore, with minimal additional effort this can be accomplished in an **omnithermal** way, allowing us to simultaneously sample from the equilibrium distribution when
  - using $c + m$ servers, for any $m \in \mathbb{N}$
  - increasing the service rate
  - or both.
- There are other perfect simulation algorithms out there, e.g. gradient simulation for fork-join networks (Chen & Shi, 2016), for which it may be possible to produce omnithermal variants.

References

- Chen, X & Shi, X. (2016). Perfect Sampling and Gradient Simulation for Fork-Join Networks. [arXiv]
- Connor, S.B. (2016). Omnithermal perfect simulation for multi-server queues. *Submitted.*
- Connor, S.B., & Kendall, W.S. (2015). Perfect simulation of $M/G/c$ queues. *Advances in Applied Probability*, **47**(4), 1039–1063.
- Kendall, W.S. (1998). Perfect simulation for the area-interaction point process. In *Probability Towards 2000* eds L. Accardi and C. C. Heyde. Springer, New York, pp. 218–234.
- Propp, J.G., & Wilson, D.B. (1996). Exact sampling with coupled Markov chains and applications to statistical mechanics. *Random Structures and Algorithms*, **9**, 223–252.
- Shah, S.R. (2004). *Perfect simulation of conditional & weighted models*. Ph.D. thesis, Department of Statistics, University of Warwick.