Some trajectories in $\mathbb{R}^n$ arising in statistics, scheduling and queues

Yoni Nazarathy

The University of Queensland

Presented at the Maths Seminar Series,
QUT, Sep 11, 2015.

Outline

1. Moment matching of truncated distributions
2. Cyclic queueing systems
3. Scheduling with linear slowdown
4. Overflow fluid buffer networks

Moment matching of truncated distributions

Joint work with Benoit Liquet

Problem: Matching moments of truncated distributions

Moment matching

$$\int x^i g(x\,;\,\theta)\,dx = m_i^*, \qquad i = 1, \ldots, n$$

Problem: Matching moments of truncated distributions

Moment matching

$$\int x^i g(x\,;\,\theta)\,dx = m_i^*, \qquad i = 1, \ldots, n$$

Truncated distributions

$$g_{a,b}(x\,;\,\theta) = \frac{f(x\,;\,\theta)}{\int_a^b f(u\,;\,\theta)\,du}$$

Problem: Matching moments of truncated distributions

Moment matching

$$\int x^i g(x\,;\,\theta)\,dx = m_i^*, \qquad i = 1, \ldots, n$$

Truncated distributions

$$g_{a,b}(x\,;\,\theta) = \frac{f(x\,;\,\theta)}{\int_a^b f(u\,;\,\theta)\,du}$$

Equations for θ (given $[a, b]$ and m^*) are "hard"

Exponential:

$$m_1^* = \theta^{-1} \frac{(b\theta+1)e^{a\theta} - (a\theta+1)e^{b\theta}}{e^{a\theta} - e^{b\theta}}$$

Normal:

$$m_1^* = \theta_1 - \theta_2 \frac{\phi\left(\frac{b-\theta_1}{\theta_2}\right) - \phi\left(\frac{a-\theta_1}{\theta_2}\right)}{\Phi\left(\frac{b-\theta_1}{\theta_2}\right) - \Phi\left(\frac{a-\theta_1}{\theta_2}\right)}$$

$$m_2^* = \theta_1^2 + \theta_2^2 - \theta_2 \frac{(\theta_1+b)\phi\left(\frac{b-\theta_1}{\theta_2}\right) - (\theta_1+a)\phi\left(\frac{a-\theta_1}{\theta_2}\right)}{\Phi\left(\frac{b-\theta_1}{\theta_2}\right) - \Phi\left(\frac{a-\theta_1}{\theta_2}\right)}$$

Idea

- Start with support $(-\infty, \infty)$ and truncate "bit by bit"
- $z \in (0, 1]$ is level of truncation, e.g.

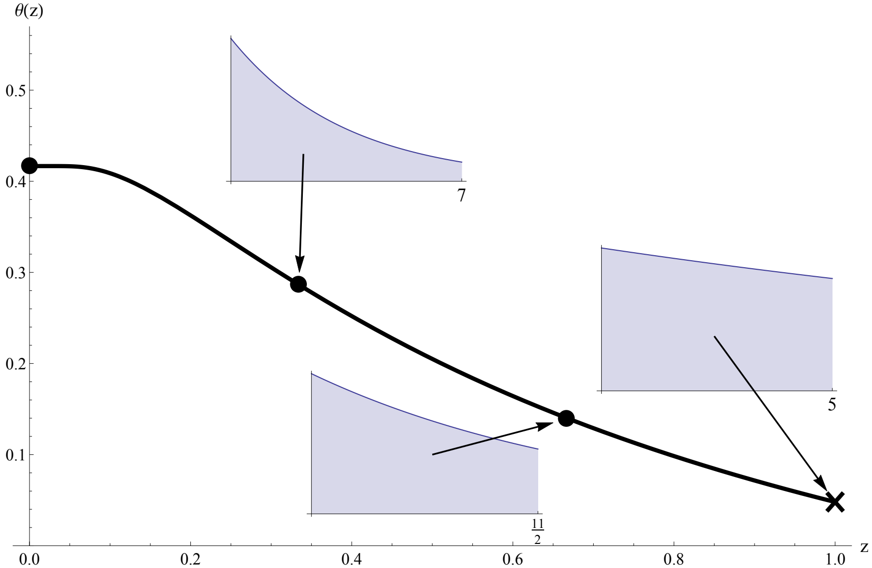
$$\left(a - \frac{1-z}{z},\ b + \frac{1-z}{z}\right)$$

- $\theta(z)$ is the solution for each z
- Derive expression for $F(\cdot, \cdot)$ in the ODE:

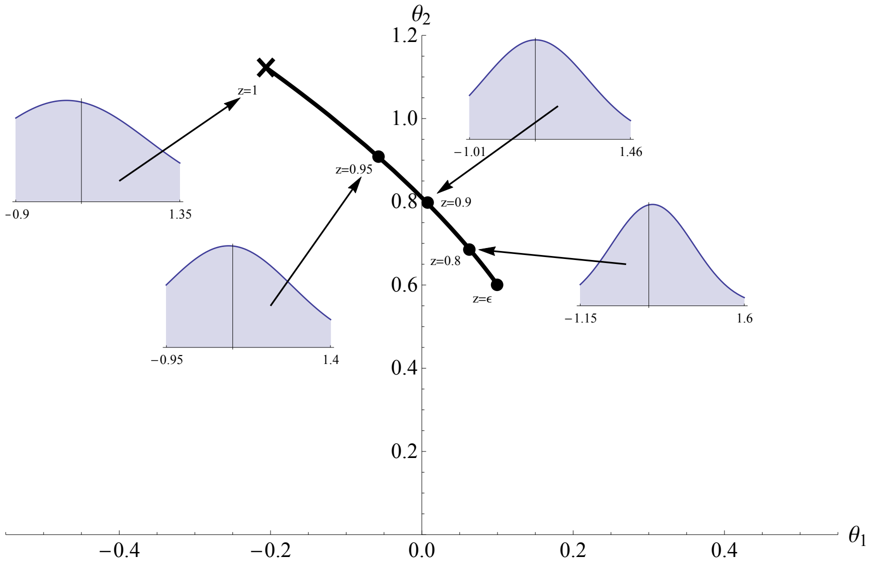
$$\frac{d}{dz}\theta(z) = F(\theta(z),\ z)$$

- In most cases $\theta(0^+)$ has a simple closed form
- Find the trajectory $\theta(z)$ numerically

$f(x\,;\,\theta) = \theta \exp(-\theta x)$, $m_1^* = 2.4$, $[a, b] = [0, 5]$



$f(x\,;\,\theta) = \frac{1}{\theta_2\sqrt{2\pi}} \exp\left(-\frac{(x-\theta_1)^2}{2\theta_2^2}\right),\ m_1^* = 0.1,\ \sqrt{m_2^* - (m_1^*)^2} = 0.6,\ [a,b] = [-0.9, 1.35]$



The ODE

$$\frac{d}{dz}\theta(z) = \frac{1}{z^2} B\big(z, \theta(z)\big)^{-1} c\big(z, \theta(z)\big)$$

$$\begin{aligned} c_i(z,\theta(z)) = \quad & \big((b + \frac{1-z}{z})^i - m_i^*\big) f\big(b + \frac{1-z}{z}\,;\,\theta(z)\big) \\ & + \big((a - \frac{1-z}{z})^i - m_i^*\big) f\big(a - \frac{1-z}{z}\,;\,\theta(z)\big) \end{aligned}$$

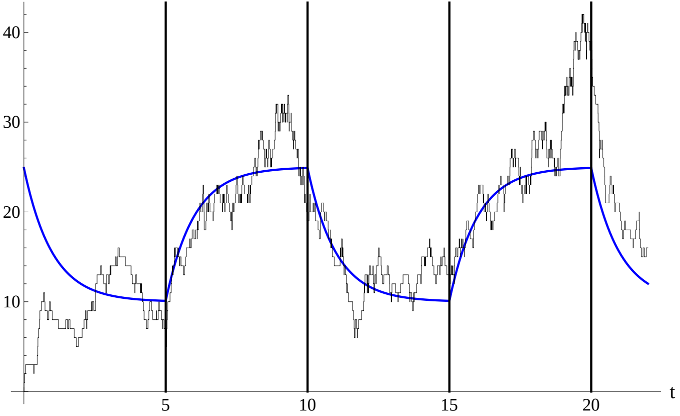
$$B_{i,j}(z,\theta(z)) = \int_{a-\frac{1-z}{z}}^{b+\frac{1-z}{z}} (x^i - m_i^*) h_j\big(x, \theta(z)\big)\, dx$$

$$h_j\Big(x, (\tilde{\theta}_1, \ldots, \tilde{\theta}_n)\Big) = \frac{d}{d\,\tilde{\theta}_j} f(x\,;\,\tilde{\theta}_1, \ldots, \tilde{\theta}_n)$$

Cyclic queueing systems

Joint work with Matthieu Jonckheere and Leonardo Rojas-Nandayapa

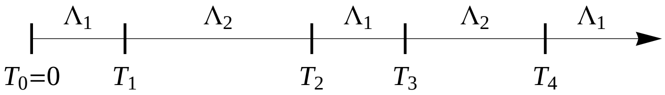
Problem: Performance of queues in cyclic environments



Want to evaluate:

$$F(y) = \lim_{t \to \infty} \frac{1}{t} \int_0^t \mathbb{1}\{X(u) \le y\}\, du$$

Two variations



Hysteresis control

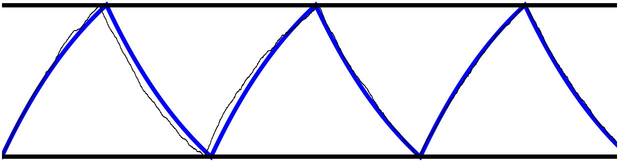
$$T_n = \inf \left\{ t > T_{n-1} : X(t) = \begin{cases} \ell_2 & \text{for } n \text{ odd,} \\ \ell_1 & \text{for } n \text{ even.} \end{cases} \right\}$$

Fixed cycles

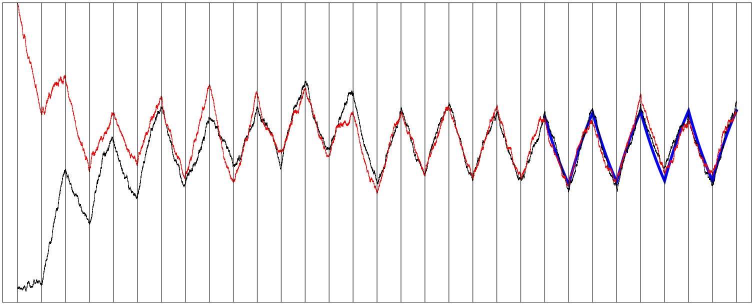
$$T_n - T_{n-1} = \begin{cases} \tau_1 & \text{for } n \text{ odd,} \\ \tau_2 & \text{for } n \text{ even.} \end{cases}$$

Basic idea: Approximate the random trajectory with switched ODE

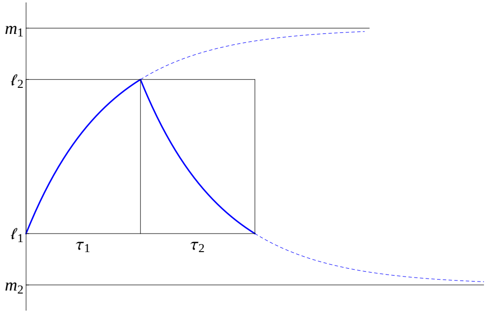
Hysteresis control



Fixed cycles



Use the ODE to construct an approximation for $F(\cdot)$



$$F(y) = \frac{1}{\tau_1 + \tau_2}\big(\tau_1(y) + (\tau_2 - \tau_2(y)\big)$$

- Hysteresis control: ℓ_1, ℓ_2 given $\Rightarrow$ find τ_1, τ_2
- Fixed Cycles: τ_1, τ_2 given $\Rightarrow$ find ℓ_1, ℓ_2

$$x_1\Big|_{x_1(0)=\ell_1}^{(\tau_1)} = \ell_2, \qquad x_2\Big|_{x_2(0)=\ell_2}^{(\tau_2)} = \ell_1$$

Example: Infinite server case, $\frac{d}{dt}x_i(t) = \lambda_i - \mu_i\, x_i(t)$

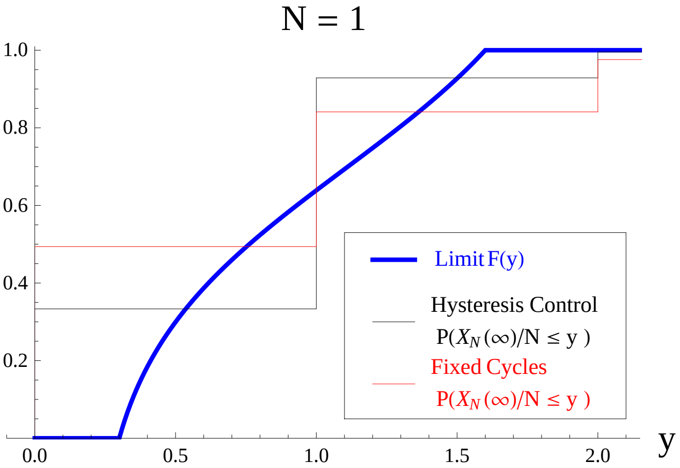
ODE based approximation for $F(\cdot)$

$$F(y) = \int_{-\infty}^{y} f(u)du, \qquad f(u) = \frac{\frac{(\mu_1-\mu_2)u+(\lambda_2-\lambda_1)}{(\mu_1 u-\lambda_1)(\mu_2 u-\lambda_2)}}{\log\left(\frac{\mu_1\ell_1-\lambda_1}{\mu_1\ell_2-\lambda_1}\right)^{\frac{1}{\mu_1}}\left(\frac{\mu_2\ell_2-\lambda_2}{\mu_2\ell_1-\lambda_2}\right)^{\frac{1}{\mu_2}}}\mathbb{1}\{\ell_1 \le u \le \ell_2\}$$

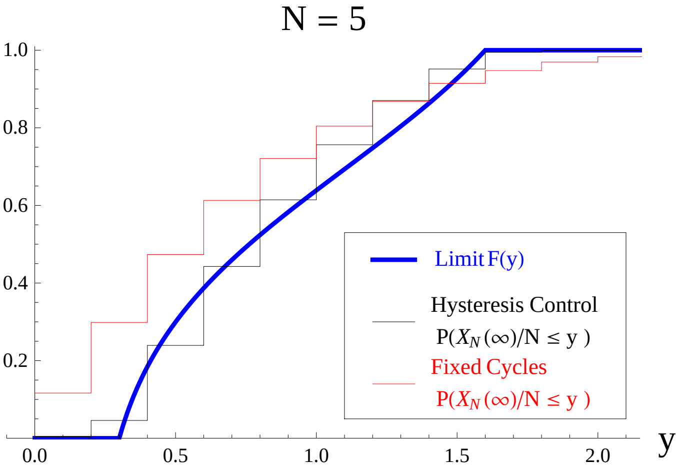
For fixed cycles set: $\ell_i = \frac{(e^{\tau_i\mu_i}-1)\frac{\lambda_i}{\mu_i}+(e^{\tau_{\bar{i}}\mu_{\bar{i}}}-1)\frac{\lambda_{\bar{i}}}{\mu_{\bar{i}}}e^{\tau_i\mu_i}}{e^{\tau_i\mu_i+\tau_{\bar{i}}\mu_{\bar{i}}}-1}$

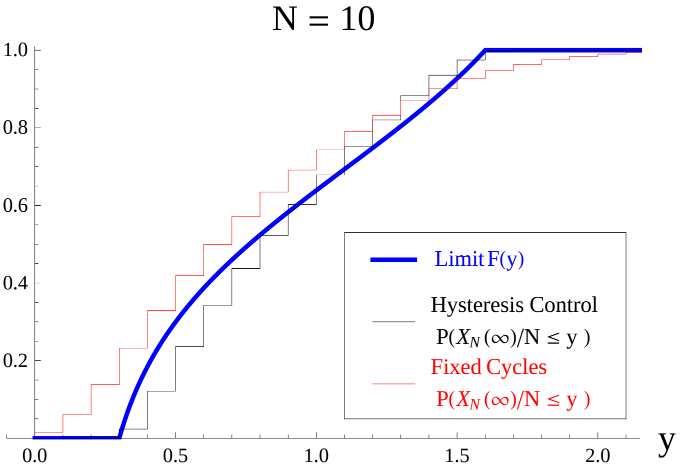
Approximation becomes exact when accelerating the arrival rates:

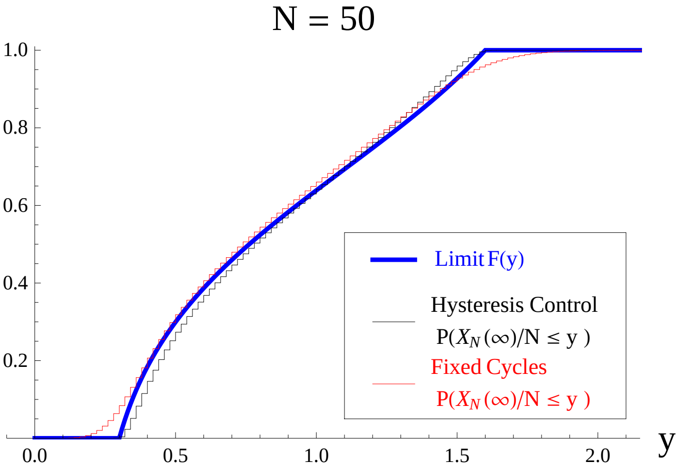
$$\lambda_i^{(N)} = N\lambda_i \quad \text{with} \quad N \to \infty$$

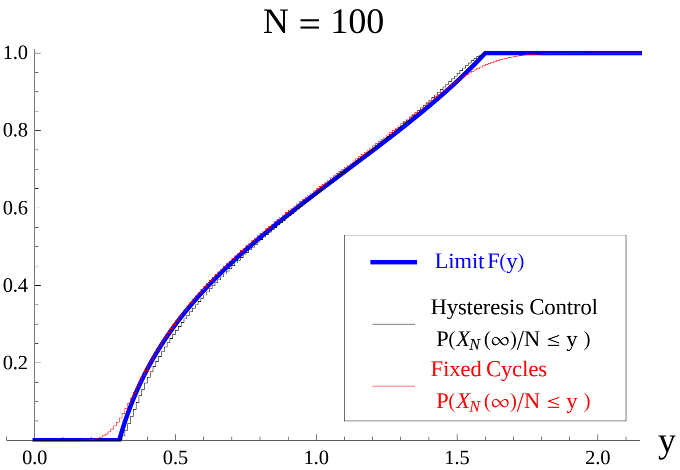


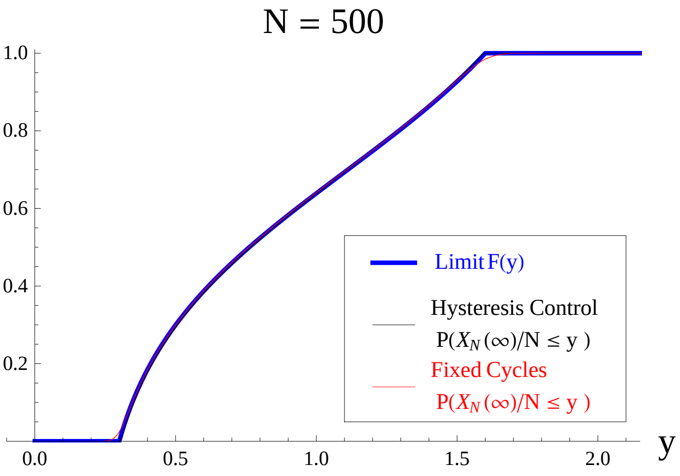
N = 1
1.0
0.8
0.6
0.4
0.2
0.0
0.5
1.0
1.5
2.0
y
Limit F(y)
Hysteresis Control
P(X_N(∞)/N ≤ y)
Fixed Cycles
P(X_N(∞)/N ≤ y)


N = 5
1.0
0.8
0.6
0.4
0.2
0.0
0.5
1.0
1.5
2.0
y
Limit F(y)
Hysteresis Control
P(X_N(∞)/N ≤ y)
Fixed Cycles
P(X_N(∞)/N ≤ y)


N = 10
1.0
0.8
0.6
0.4
0.2
0.0
0.5
1.0
1.5
2.0
y
Limit F(y)
Hysteresis Control
P($X_N(\infty)$/N ≤ y)
Fixed Cycles
P($X_N(\infty)$/N ≤ y)


N = 50
1.0
0.8
0.6
0.4
0.2
0.0
0.5
1.0
1.5
2.0
y
Limit F(y)
Hysteresis Control
P($X_N(\infty)$/N ≤ y)
Fixed Cycles
P($X_N(\infty)$/N ≤ y)


N = 100
1.0
0.8
0.6
0.4
0.2
0.0
0.5
1.0
1.5
2.0
y
Limit F(y)
Hysteresis Control
P(X_N(∞)/N ≤ y)
Fixed Cycles
P(X_N(∞)/N ≤ y)


N = 500
1.0
0.8
0.6
0.4
0.2
0.0
0.5
1.0
1.5
2.0
y
Limit F(y)
Hysteresis Control
P(X_N(∞)/N ≤ y)
Fixed Cycles
P(X_N(∞)/N ≤ y)

Scheduling with linear slowdown

Joint work with Liron Ravner

Problem: Optimisation with rush hour traffic

Processor sharing scheduling for n users

$$1 = \int_{a_i}^{d_i} v\big(q(t)\big)\,dt \qquad q(t) = \sum_{j=1}^{n} \mathbb{1}\{t \in [a_j, d_j]\}$$

Problem: Optimisation with rush hour traffic

Processor sharing scheduling for n users

$$1 = \int_{a_i}^{d_i} v\big(q(t)\big)\,dt \qquad q(t) = \sum_{j=1}^{n} \mathbb{1}\{t \in [a_j, d_j]\}$$

Linear slowdown

$$v\big(q(t)\big) = \beta - \alpha\big(q(t) - 1\big)$$

Problem: Optimisation with rush hour traffic

Processor sharing scheduling for n users

$$1 = \int_{a_i}^{d_i} v\big(q(t)\big)\,dt \qquad q(t) = \sum_{j=1}^{n} \mathbb{1}\{t \in [a_j, d_j]\}$$

Linear slowdown

$$v\big(q(t)\big) = \beta - \alpha\big(q(t) - 1\big)$$

(assume $n < \beta/\alpha + 1$)

Problem: Optimisation with rush hour traffic

Processor sharing scheduling for n users

$$1 = \int_{a_i}^{d_i} v\big(q(t)\big)\,dt \qquad q(t) = \sum_{j=1}^{n} \mathbb{1}\{t \in [a_j, d_j]\}$$

Linear slowdown

$$v\big(q(t)\big) = \beta - \alpha\big(q(t) - 1\big)$$

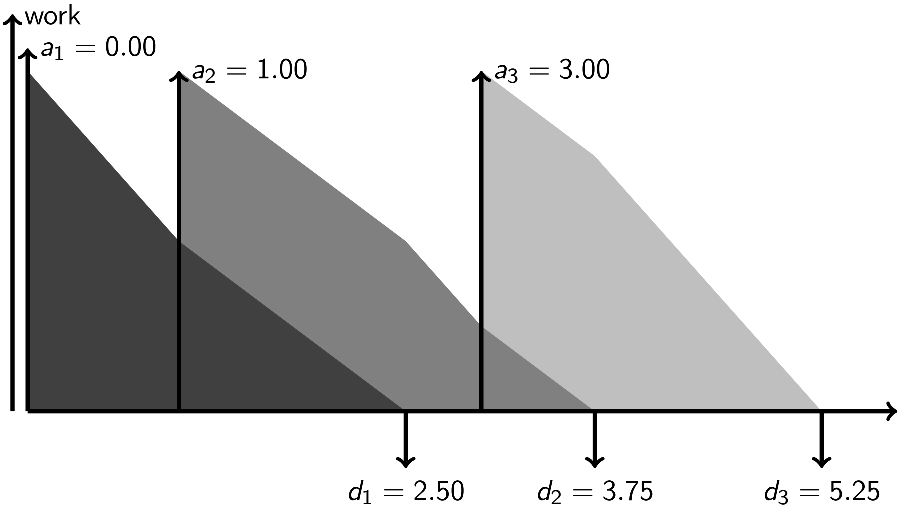
$$(\text{assume } n < \beta/\alpha + 1)$$

Objective

$$\min_{\mathbf{a}\in\mathbb{R}^n} c(\mathbf{a}) = \sum_{i=1}^{n} c_i\big(a_i, d_i(\mathbf{a})\big), \qquad c_i(a_i, d_i) = (d_i - d_i^*)^2 + \gamma\,(d_i - a_i)$$

Piecewise affine relationship between **a** and **d**

E.g. with $n = 3$, $\beta = 1/2$, $\alpha = 1/6$:



Key attributes and algorithms

An exponential number of convex quadratic programs

The objective function is piecewise quadratic with number of regions equal to,

$$\frac{\binom{2n}{n}}{n+1} \sim \frac{4^n}{n^{3/2}\sqrt{\pi}},$$

and with explicit expressions for describing each of the QPs.

Algorithms:

Key attributes and algorithms

An exponential number of convex quadratic programs

The objective function is piecewise quadratic with number of regions equal to,

$$\frac{\binom{2n}{n}}{n+1} \sim \frac{4^n}{n^{3/2}\sqrt{\pi}},$$

and with explicit expressions for describing each of the QPs.

Algorithms:

1. Calculate **d** based on **a** or vice versa

Key attributes and algorithms

An exponential number of convex quadratic programs

The objective function is piecewise quadratic with number of regions equal to,

$$\frac{\binom{2n}{n}}{n+1} \sim \frac{4^n}{n^{3/2}\sqrt{\pi}},$$

and with explicit expressions for describing each of the QPs.

Algorithms:

1. Calculate **d** based on **a** or vice versa
2. Exhaustive search

Key attributes and algorithms

An exponential number of convex quadratic programs

The objective function is piecewise quadratic with number of regions equal to,

$$\frac{\binom{2n}{n}}{n+1} \sim \frac{4^n}{n^{3/2}\sqrt{\pi}},$$

and with explicit expressions for describing each of the QPs.

Algorithms:

1. Calculate **d** based on **a** or vice versa
2. Exhaustive search
3. Neighbour search

Key attributes and algorithms

An exponential number of convex quadratic programs

The objective function is piecewise quadratic with number of regions equal to,

$$\frac{\binom{2n}{n}}{n+1} \sim \frac{4^n}{n^{3/2}\sqrt{\pi}},$$

and with explicit expressions for describing each of the QPs.

Algorithms:

1. Calculate **d** based on **a** or vice versa
2. Exhaustive search
3. Neighbour search
4. Efficient trajectory calculation for $\mathbf{d}(a_i)$

Key attributes and algorithms

An exponential number of convex quadratic programs

The objective function is piecewise quadratic with number of regions equal to,

$$\frac{\binom{2n}{n}}{n+1} \sim \frac{4^n}{n^{3/2}\sqrt{\pi}},$$

and with explicit expressions for describing each of the QPs.

Algorithms:

1. Calculate **d** based on **a** or vice versa
2. Exhaustive search
3. Neighbour search
4. Efficient trajectory calculation for $\mathbf{d}(a_i)$
5. Global search by means of CPI

Key attributes and algorithms

An exponential number of convex quadratic programs

The objective function is piecewise quadratic with number of regions equal to,

$$\frac{\binom{2n}{n}}{n+1} \sim \frac{4^n}{n^{3/2}\sqrt{\pi}},$$

and with explicit expressions for describing each of the QPs.

Algorithms:

1. Calculate **d** based on **a** or vice versa
2. Exhaustive search
3. Neighbour search
4. Efficient trajectory calculation for $\mathbf{d}(a_i)$
5. Global search by means of CPI
6. Combined global-local search heuristic

Key attributes and algorithms

An exponential number of convex quadratic programs

The objective function is piecewise quadratic with number of regions equal to,

$$\frac{\binom{2n}{n}}{n+1} \sim \frac{4^n}{n^{3/2}\sqrt{\pi}},$$

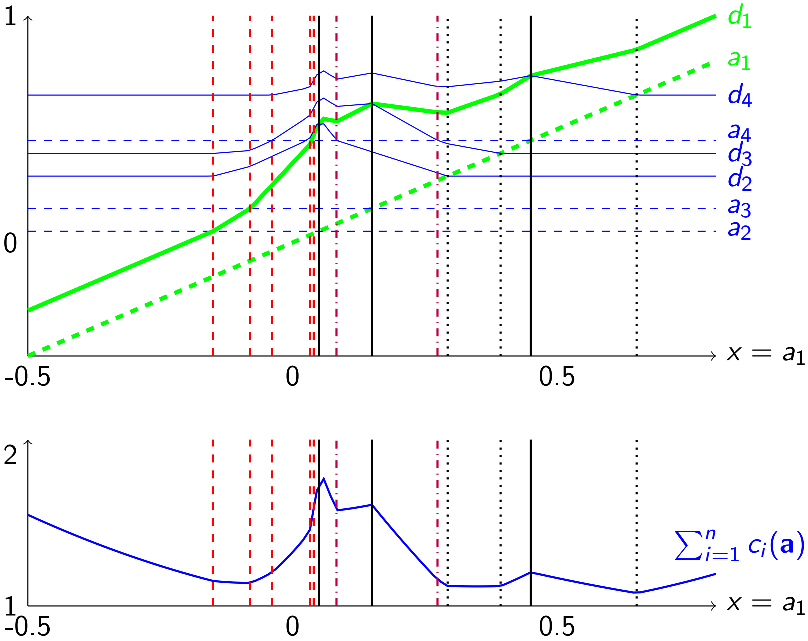
and with explicit expressions for describing each of the QPs.

Algorithms:

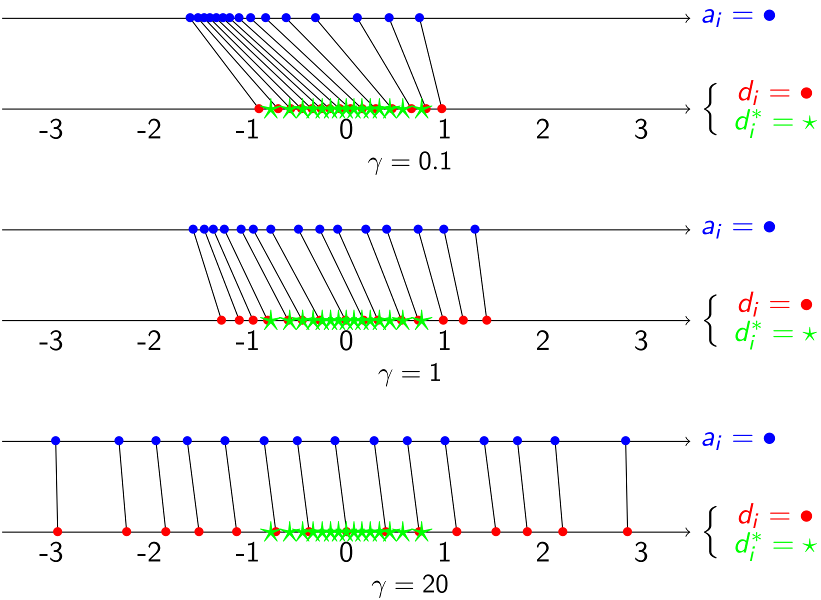
1. Calculate **d** based on **a** or vice versa
2. Exhaustive search
3. Neighbour search
4. Efficient trajectory calculation for $\mathbf{d}(a_i)$
5. Global search by means of CPI
6. Combined global-local search heuristic

Not known if problem is NP–complete

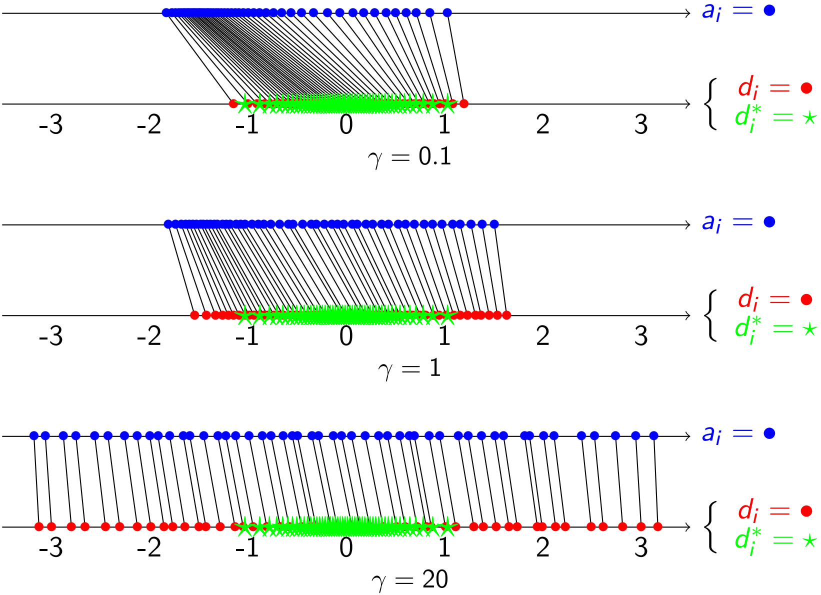
Trajectory of $\mathbf{d}$ when changing a_1 ($\alpha = 1.5$, $\beta = 5$ and $\mathbf{d}^* = \mathbf{0}$)



Optimal dynamics $n = 15$ ($\alpha = \frac{0.8}{n}$, $\beta = 1$ and $\mathbf{d}^*$ quantiles of Normal$(0, \frac{1}{4})$)



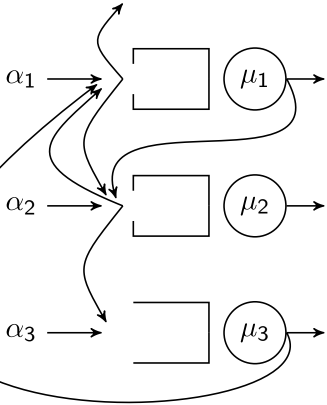
Heuristic dynamics $n = 50$ ($\alpha = \frac{0.8}{n}$, $\beta = 1$ and $\mathbf{d}^*$ quantiles of Normal$(0, \frac{1}{4})$)



Overflow fluid buffer networks

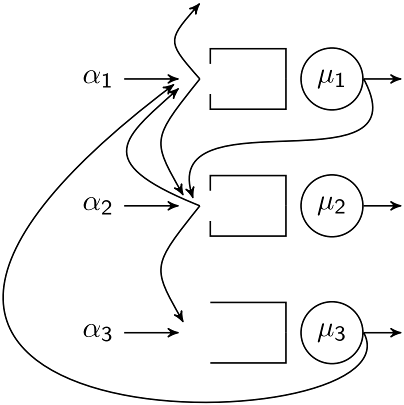
Joint work with Stijn Fleuren and Erjen Lefeber

Problem: Modelling complex manufacturing systems



An overflow fluid buffer model

$\alpha_1, \ldots, \alpha_n$	–	exogenous arrival rates
$\mu_1, \ldots, \mu_n$	–	service rates
$P = (p_{i,j})$	–	routing matrix (sub-stochastic or stochastic)
$K_1, \ldots, K_n$	–	buffer sizes (can also be ∞)
$Q = (q_{i,j})$	–	overflow matrix (strictly sub-stochastic)



λ_i: The effective arrival rate to buffer i

Basic Jackson:

$$\lambda_i = \alpha_i + \sum_{j=1}^{n} \lambda_j p_{j,i}$$

λ_i: The effective arrival rate to buffer i

Basic Jackson:

$$\lambda_i = \alpha_i + \sum_{j=1}^{n} \lambda_j p_{j,i}$$

Evolution of equations:

Basic Jackson, 1950's: $\lambda = \alpha + \lambda P$

λ_i: The effective arrival rate to buffer i

Basic Jackson:

$$\lambda_i = \alpha_i + \sum_{j=1}^{n} \lambda_j p_{j,i}$$

Evolution of equations:

Basic Jackson, 1950's:	$\lambda = \alpha + \lambda P$
Goodman and Massey, 1984:	$\lambda = \alpha + (\lambda \wedge \mu) P$

λ_i: The effective arrival rate to buffer i

Basic Jackson:

$$\lambda_i = \alpha_i + \sum_{j=1}^{n} \lambda_j p_{j,i}$$

Evolution of equations:

Basic Jackson, 1950's:	$\lambda = \alpha + \lambda P$
Goodman and Massey, 1984:	$\lambda = \alpha + (\lambda \wedge \mu) P$
Our steady state flow equations:	$\lambda = \alpha + (\lambda \wedge \mu) P + (\lambda - \mu)^+ Q$

$$\lambda_i = \alpha_i + \sum_{j=1}^{n} (\lambda_j \wedge \mu_j) p_{j,i} + \sum_{j=1}^{n} (\lambda_j - \mu_j)^+ q_{j,i}$$

Buffer trajectories, $X(t) \in \mathbb{R}^n$

Modes

$$\mathcal{E}(t) = \{i : X_i(t) = 0\}, \quad \mathcal{F}(t) := \{i : X_i(t) = K_i\}$$

Equations based on mode

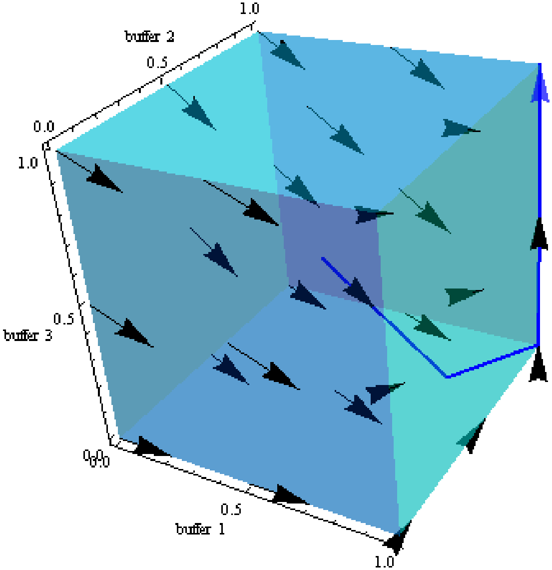
$$\lambda = \alpha + (\lambda \wedge \mu)P^{\mathcal{E}} + \mu P^{\overline{\mathcal{E}}} + (\lambda - \mu)^+ Q^{\mathcal{F}}$$

Trajectories

$$X(t) = X(0) + \int_0^t \Delta\Big(\mathcal{E}(u), \mathcal{F}(u)\Big)\, du$$

with,

$$\Delta_i(\mathcal{E}, \mathcal{F}) = \begin{cases} \lambda_i(\mathcal{E}, \mathcal{F}) - \lambda_i(\mathcal{E}, \mathcal{F}) \wedge \mu_i, & i \in \mathcal{E}, \\ \lambda_i(\mathcal{E}, \mathcal{F}) - \mu_i, & i \notin \mathcal{E}, i \notin \mathcal{F}, \\ \lambda_i(\mathcal{E}, \mathcal{F}) - \lambda_i(\mathcal{E}, \mathcal{F}) \vee \mu_i, & i \in \mathcal{F}. \end{cases}$$



buffer 1
buffer 2
buffer 3
0.0
0.5
1.0

Properties and results

- Conditions for uniqueness and existence
- Efficient algorithm for solving the equations
- Trajectories are non-cycling
- Example of an exponential number of mode changes

References

- B. Liquet and Y. Nazarathy, "*A dynamic view to moment matching of truncated distributions*", Statistics and Probability Letters, 2015.
- M. Jonckheere, Y. Nazarathy and L. Rojas-Nandayapa, "*Scaling Approximations for Cyclic Queues*", in preparation (draft available upon demand).
- L. Ravner and Y. Nazarathy, "*Scheduling for a Processor Sharing System with Linear Slowdown*", arXiv preprint arXiv:1508.03136, 2015.
- S. Fleuren, E. Lefeber and Y. Nazarathy, "*Single Class Queueing Networks with Overflows: The Deterministic Fluid Case*", in preparation (draft available upon demand).