Scheduling Arrivals to a Processor Sharing Congestion System with Linear Slowdown

Yoni Nazarathy

The University of Queensland

Liron Ravner

The Hebrew University of Jerusalem

Presented Sep 29 2015, at the AustMS Conference, Flinders University, Adelaide.

General Problem: Getting in time through rush hour traffic

General Problem: Getting in time through rush hour traffic

- Users have ideal **arrival** and **departure** times and want to minimise travel times as well

General Problem: Getting in time through rush hour traffic

- Users have ideal **arrival** and **departure** times and want to minimise travel times as well
- Users interact and **slow** each other down

General Problem: Getting in time through rush hour traffic

- Users have ideal **arrival** and **departure** times and want to minimise travel times as well
- Users interact and **slow** each other down
- When to enter traffic?

General Problem: Getting in time through rush hour traffic

- Users have ideal **arrival** and **departure** times and want to minimise travel times as well
- Users interact and **slow** each other down
- When to enter traffic?
- In this work: a basic global scheduling model and problem for a finite number of homogenous users

General Problem: Getting in time through rush hour traffic

- Users have ideal **arrival** and **departure** times and want to minimise travel times as well
- Users interact and **slow** each other down
- When to enter traffic?
- In this work: a basic global scheduling model and problem for a finite number of homogenous users

Processor Sharing

n users interact

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

Processor Sharing

n users interact

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

Linear slowdown

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

Processor Sharing

n users interact

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

Linear slowdown

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

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

Processor Sharing

n users interact

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

Linear slowdown

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

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

Objective

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

Objective (simple version)

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

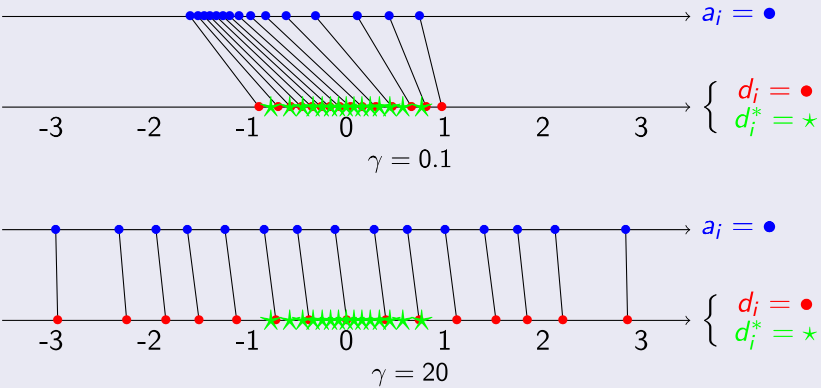
Objective (complex version)

$$\begin{aligned} c_i(a_i, d_i) = {} & g_i^{(1)}\Big((d_i - d_i^*)^+\Big) + g_i^{(2)}\Big((d_i^* - d_i)^+\Big) \\ & + g_i^{(3)}\Big((a_i - a_i^*)^+\Big) + g_i^{(4)}\Big((a_i^* - a_i)^+\Big) \\ & + g_i^{(5)}\Big(d_i - a_i\Big) \end{aligned}$$

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

$$v\big(q(t)\big) = \beta - \alpha\big(q(t) - 1\big)$$
$$c_i(a_i, d_i) = (d_i - d_i^*)^2 + \gamma\,(d_i - a_i)$$

Optimal Schedules



Basic Equations

Observation

If we order d_i^* then a_i and d_i follow the same order

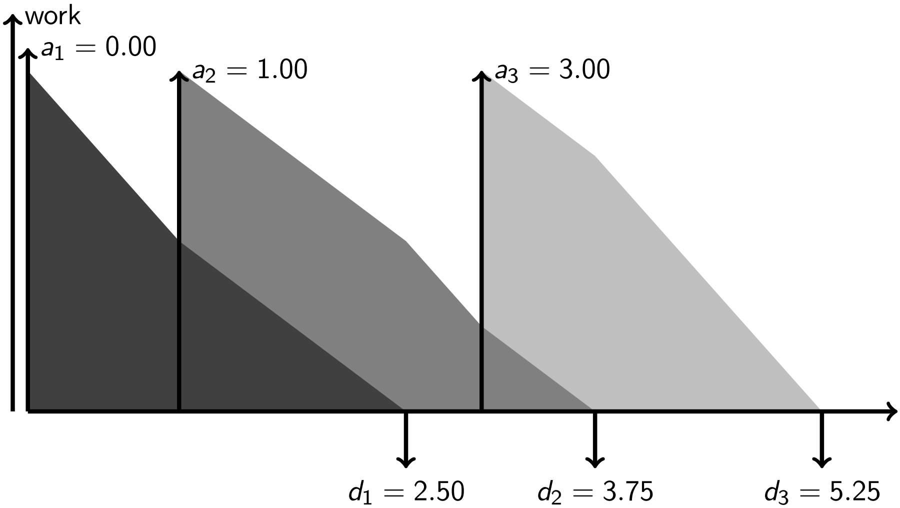
$$
\begin{aligned}
1 &= \int_{a_i}^{d_i} \beta - \alpha\Big(\sum_{j=1}^{n} \mathbb{1}\{t \in [a_j, d_j]\} - 1\Big) dt \\
&= (\beta + \alpha)(d_i - a_i) - \alpha \sum_{j=1}^{n} (d_i \wedge d_j - a_i \vee a_j)^+ \\
&= \dots \\
&= -(\beta - \alpha(i - h_i))a_i + (\beta - \alpha(k_i - i))d_i - \alpha\Big(\sum_{j=h_i}^{i-1} d_j - \sum_{j=i+1}^{k_i} a_j\Big)
\end{aligned}
$$

with "encodings" of the interleaving of $\mathbf{d}$ and $\mathbf{a}$:

$$
k_i := \max\big\{k \;:\; a_k \le d_i\big\}, \qquad h_i := \min\big\{h \;:\; d_h \ge a_i\big\}
$$

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 $\mathbf{d}$ based on $\mathbf{a}$ 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. Polynomial time optimal solutions when $\gamma = 0$ or $\gamma \to \infty$

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. Polynomial time optimal solutions when $\gamma = 0$ or $\gamma \to \infty$
6. Global search by means of CPI (with polynomial time iterations)

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. Polynomial time optimal solutions when $\gamma = 0$ or $\gamma \to \infty$
6. Global search by means of CPI (with polynomial time iterations)
7. 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,

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

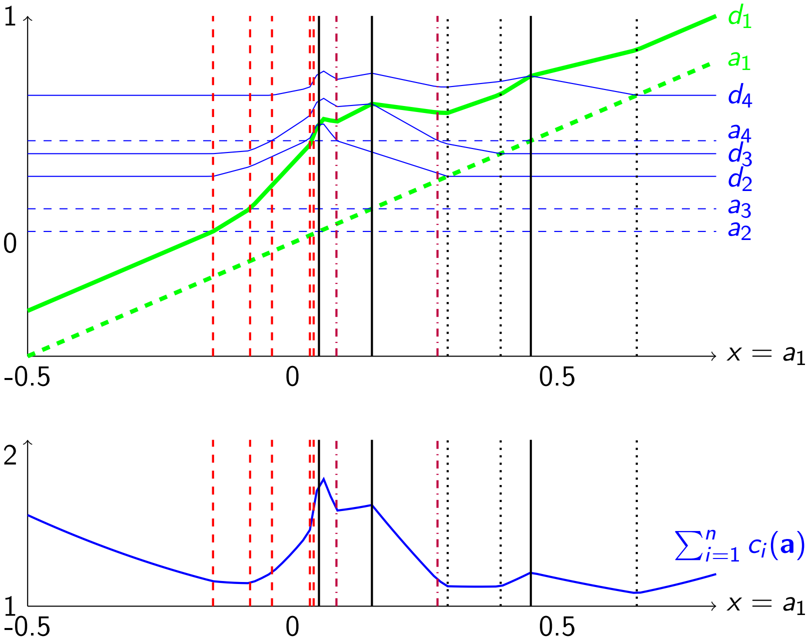
and with explicit expressions for describing each of the QPs.

Algorithms:

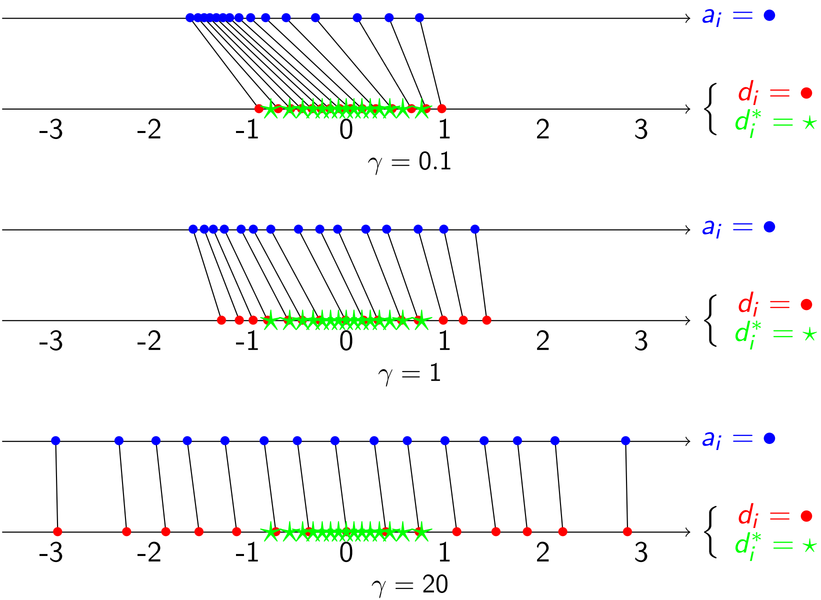
1. Calculate **d** based on **a** or vice versa
2. Exhaustive search
3. Neighbour search
4. Efficient trajectory calculation for $\mathbf{d}(a_i)$
5. Polynomial time optimal solutions when $\gamma = 0$ or $\gamma \to \infty$
6. Global search by means of CPI (with polynomial time iterations)
7. Combined global-local search heuristic

Not known if problem is NP–complete

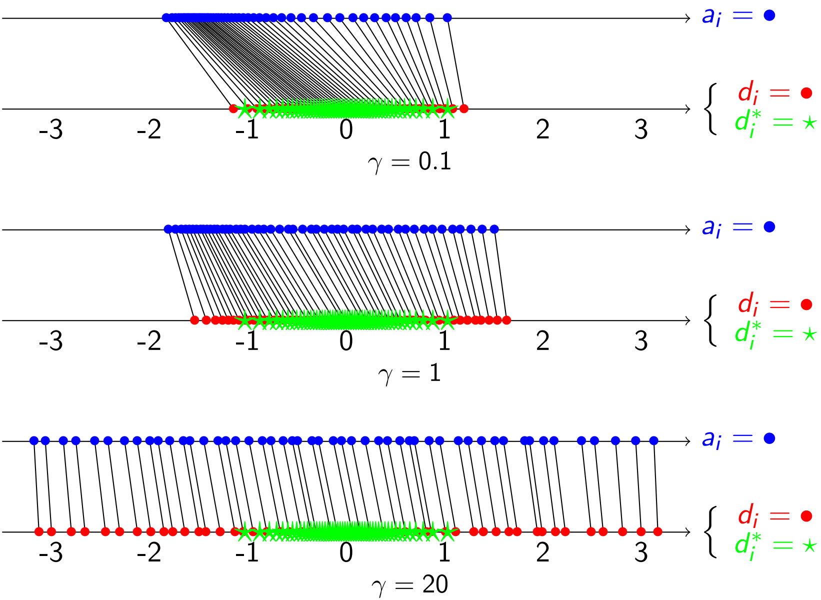
Trajectory of $\mathbf{d}$ when changing a_1 ($\alpha = 1.5$, $\beta = 5$ and $\mathbf{d}^* = \mathbf{0}$)



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$(0, \frac{1}{4})$)



More General Dynamics... (not yet handled but appears possible)

A tandem sequence of K processors

$$\ell_i^k = \int_{a_i^{k-1}}^{a_i^k} v_i^k\big(q^k(t)\big)\, dt, \quad k = 1, \ldots, K$$

$$q^k(t) = \sum_{j=1}^{n} \mathbb{1}\{t \in [a_j^{k-1}, a_j^k]\}$$

$$v_i^k(q) = \beta_i^k - \alpha_i^k(q-1) = \text{or even arbitrary}$$

Look at start and end of the sequence: $a_i = a_i^0$, $d_i = a_i^K$

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

Belief: Similar problem structure - maybe proving NP completeness is easier in this general case...

L. Ravner and Y. Nazarathy, "*Scheduling for a Processor Sharing System with Linear Slowdown*", preprint, arXiv:1508.03136, 2015.

L. Ravner, M. Haviv, H. Vu, "*A strategic timing of arrivals to a linear slowdown processor sharing system*", preprint, arXiv:1508.03420, 2015.