Dynamic Matching Models

Ana Bušić

Inria Paris - Rocquencourt
CS Department of École normale supérieure

joint work with Varun Gupta, Jean Mairesse and Sean Meyn

3rd Workshop on Cognition and Control
January 16 & 17, 2015, University of Florida
Dynamic Bipartite Matching Model

Static model – long history in economics **Finding Stable Matches** 2012 Nobel Prize awarded to L. S. Shapley.

Dynamic model introduced by Caldentey, Kaplan, and Weiss (2009)

*Multiclass queueing model – Supply/Demand play symmetric roles*
Dynamic Bipartite Matching Model

Static model – long history in economics Finding Stable Matches 2012 Nobel Prize awarded to L. S. Shapley.

Dynamic model introduced by Caldentey, Kaplan, and Weiss (2009) Multiclass queueing model – Supply/Demand play symmetric roles

- Discrete time queueing model with two types of arrival: “supply” and “demand”.
- Arrival of Supply/Demand is i.i.d., with
  \[ |A^D(t)| = |A^S(t)| \quad \text{for all } t \]
- Instantaneous matchings according to a bipartite matching graph.
- Supply/Demand that cannot be matched are stored in a buffer.
Dynamic Bipartite Matching Model

**Static model** – long history in economics **Finding Stable Matches** 2012 Nobel Prize awarded to L. S. Shapley.

**Dynamic model introduced by** Caldentey, Kaplan, and Weiss (2009)

*Multiclass queueing model – Supply/Demand play symmetric roles*

- Discrete time queueing model with two types of arrival: “supply” and “demand”.
- Arrival of Supply/Demand is i.i.d., with

\[ |A^D(t)| = |A^S(t)| \quad \text{for all } t \]

- **Instantaneous matchings** according to a bipartite matching graph.
- Supply/Demand that cannot be matched are stored in a buffer.
Dynamic Bipartite Matching Model

Static model – long history in economics Finding Stable Matches 2012 Nobel Prize awarded to L. S. Shapley.

Dynamic model introduced by Caldentey, Kaplan, and Weiss (2009)
Multiclass queueing model – Supply/Demand play symmetric roles

- Discrete time queueing model with two types of arrival: “supply” and “demand”.
- Arrival of Supply/Demand is i.i.d., with

\[ |A^D(t)| = |A^S(t)| \quad \text{for all } t \]

- Instantaneous matchings according to a bipartite matching graph.
- Supply/Demand that cannot be matched are stored in a buffer.
Dynamic Bipartite Matching Model

Static model – long history in economics **Finding Stable Matches** 2012 Nobel Prize awarded to L. S. Shapley.

Dynamic model introduced by Caldentey, Kaplan, and Weiss (2009)

*Multiclass queueing model – Supply/Demand play symmetric roles*

- Discrete time queueing model with two types of arrival: “supply” and “demand”.
- Arrival of Supply/Demand is i.i.d., with
  \[ |A^D(t)| = |A^S(t)| \quad \text{for all } t \]
- **Instantaneous matchings** according to a bipartite matching graph.
- Supply/Demand that cannot be matched are stored in a buffer.
**Dynamic Bipartite Matching Model**

**Static model** – long history in economics **Finding Stable Matches** 2012 Nobel Prize awarded to L. S. Shapley.

**Dynamic model introduced by** Caldentey, Kaplan, and Weiss (2009)

*Multiclass queueing model – Supply/Demand play symmetric roles*

- Discrete time queueing model with two types of arrival: “supply” and “demand”.
- Arrival of Supply/Demand is i.i.d., with
  \[ |A^D(t)| = |A^S(t)| \quad \text{for all } t \]
- **Instantaneous matchings** according to a bipartite matching graph.
- Supply/Demand that cannot be matched are stored in a buffer.
Dynamic Bipartite Matching Model

Static model – long history in economics Finding Stable Matches 2012 Nobel Prize awarded to L. S. Shapley.

Dynamic model introduced by Caldentey, Kaplan, and Weiss (2009)

Multiclass queueing model – Supply/Demand play symmetric roles

- Discrete time queueing model with two types of arrival: “supply” and “demand”.
- Arrival of Supply/Demand is i.i.d., with

\[ |A^D(t)| = |A^S(t)| \quad \text{for all } t \]

- Instantaneous matchings according to a bipartite matching graph.
- Supply/Demand that cannot be matched are stored in a buffer.
Matching in Health-care

Matching Kidneys and Donors

Who can join this program?

For recipients: If you are eligible for a kidney transplant and are receiving care at a transplant center in the United States, you can join ... You must have a living donor who is willing and medically able to donate his or her kidney ...

For donors: You must also be willing to take part ...
Matching Policies

Model specified by 1) Matching graph, 2) Joint probability measure $\mu$ for arrivals of Supply/Demand, and
Matching Policies

Model specified by 1) Matching graph, 2) Joint probability measure $\mu$ for arrivals of Supply/Demand, and 3) A matching policy.
Matching Policies

Model specified by 1) Matching graph, 2) Joint probability measure $\mu$ for arrivals of Supply/Demand, and 3) a matching policy.

Admissible policies

- State feedback: Decision $U(t)$ depends only on buffers $Q(t)$ and immediate arrivals $A(t)$,

$$Q(t + 1) = Q(t) - U(t) + A(t)$$
Matching Policies

Model specified by 1) Matching graph, 2) Joint probability measure $\mu$ for arrivals of Supply/Demand, and 3) A matching policy.

Admissible policies

- State feedback: Decision $U(t)$ depends only on buffers $Q(t)$ and immediate arrivals $A(t)$,

$$Q(t + 1) = Q(t) - U(t) + A(t)$$

- Match according to graph, maintaining

$$|U^D(t)| = |U^S(t)|$$
$$|Q^D(t)| = |Q^S(t)| \quad \text{for all } t$$
Matching Policies

Model specified by 1) Matching graph, 2) Joint probability measure $\mu$ for arrivals of Supply/Demand, and 3) A matching policy.

Admissible policies

- State feedback: Decision $U(t)$ depends only on buffers $Q(t)$ and immediate arrivals $A(t)$,

$$Q(t + 1) = Q(t) - U(t) + A(t)$$

- Match according to graph, maintaining

$$|U^D(t)| = |U^S(t)|$$
$$|Q^D(t)| = |Q^S(t)|$$

for all $t$

$Q$ is a discrete time Markov chain

Stability = positive recurrence of $Q$. 


Necessary stability conditions

For a matching graph \((\mathcal{D}, \mathcal{S}, E)\) we denote:

\[
\mathcal{D}(s) = \{d \in \mathcal{D} : (d, s) \in E\}, \quad \mathcal{S}(d) = \{s \in \mathcal{S} : (d, s) \in E\}.
\]

**Necessary conditions:** If the model is stable then the marginals of \(\mu\) satisfy

\[
\text{NCond} : \begin{cases} 
\mu_{\mathcal{D}}(U) < \mu_{\mathcal{S}}(S(U)), & \forall U \subseteq \mathcal{D} \\
\mu_{\mathcal{S}}(V) < \mu_{\mathcal{D}}(D(V)), & \forall V \subseteq \mathcal{S}
\end{cases}
\]
Necessary stability conditions

For a matching graph \((\mathcal{D}, S, E)\) we denote:

\[
\mathcal{D}(s) = \{ d \in \mathcal{D} : (d, s) \in E \}, \quad S(d) = \{ s \in S : (d, s) \in E \}.
\]

**Necessary conditions:** If the model is stable then the marginals of \(\mu\) satisfy

\[
\text{NCond} : \quad \left\{ \begin{array}{l}
\mu_\mathcal{D}(U) < \mu_S(S(U)), \quad \forall U \subsetneq \mathcal{D} \\
\mu_S(V) < \mu_\mathcal{D}(D(V)), \quad \forall V \subsetneq S
\end{array} \right.
\]

**Prop.** Given \([(\mathcal{D}, S, E), \mu]\), there exists an algorithm of time complexity \(O((|\mathcal{D}| + |S|)^3)\) to decide if \(\text{NCond}\) is satisfied.
Proof

Proof using network flow arguments:
\[ \mathcal{N} = (\mathcal{D} \cup \mathcal{S} \cup \{i, f\}, E \cup \{(i, d), d \in \mathcal{D}\} \cup \{(s, f), s \in \mathcal{S}\}) . \]
Capacities: \( \mu_D(d) \) for \((i, d)\), \( \mu_S(s) \) for \((s, f)\), \( \infty \) for \((d, s)\).

Lemma.

1. There exists a flow of value 1 in \( \mathcal{N} \) iff \( \mu \) satisfies \( \text{NCond}_\leq \) (< replaced by \( \leq \) in \( \text{NCond} \)).
2. There exists a flow \( T \) of value 1 such that \( T(d, s) > 0 \) for all \((d, s) \in E\) iff \( \mu \) satisfies \( \text{NCond} \).
Proof of Lemma 2

⇒ Follows easily from connectivity of the matching graph.
⇐ Fix \( \eta \) such that \( 0 < \eta < 1/|E| \). A strictly positive flow of value \( |E|\eta \):

\[
T_\eta(x, y) = \begin{cases} 
\eta & \text{for } (x, y) = (d, s) \in E \\
|S(d)| \, \eta & \text{for } (x, y) = (i, d) \\
|D(s)| \, \eta & \text{for } (x, y) = (s, f) 
\end{cases}
\]

Define: \( \tilde{\mu}_D(d) = \frac{\mu_D(d) - |S(d)| \eta}{1 - |E| \eta} \), \( \tilde{\mu}_S(s) = \frac{\mu_S(s) - |D(s)| \eta}{1 - |E| \eta} \).

For \( \eta \) small enough, \( \tilde{\mu}_D, \tilde{\mu}_S \) are probability measures satisfying NCond.
For \( \tilde{\mu}_D, \tilde{\mu}_S \) there exists a flow \( \tilde{T} \) of value 1.
A strictly positive flow of value 1: \( T = T_\eta + (1 - |E| \eta) \tilde{T} \).
Verification algorithm

The pair \((\mu_D, \mu_S)\) satisfies \text{NCond} iff the pair \((\tilde{\mu}_D, \tilde{\mu}_S)\) satisfies \text{NCond} for \(\eta\) strictly positive and small enough.

Run \text{MaxFlow} on the input \((\mathcal{N}, \tilde{\mu}_D, \tilde{\mu}_S)\) by considering \(\eta\) as a formal parameter “as small as needed”.

Quantities of type: 
\[x + y \eta\]
for \(x, y \in \mathbb{R}\).

Addition: 
\[(x_1 + y_1 \eta) + (x_2 + y_2 \eta) = (x_1 + x_2) + (y_1 + y_2) \eta\]

Comparisons:
\[x_1 + y_1 \eta = x_2 + y_2 \eta\]
\[\iff\]
\[x_1 = x_2,\ y_1 = y_2\]

\[x_1 + y_1 \eta < x_2 + y_2 \eta\]
\[\iff\]
\[x_1 < x_2 \text{ or } (x_1 = x_2, y_1 < y_2)\]
Verification algorithm

The pair \((\mu_D, \mu_S)\) satisfies NCOND iff the pair \((\tilde{\mu}_D, \tilde{\mu}_S)\) satisfies NCOND for \(\eta\) strictly positive and small enough.

Run \textsc{MaxFlow} on the input \((\mathcal{N}, \tilde{\mu}_D, \tilde{\mu}_S)\) by considering \(\eta\) as a formal parameter “as small as needed”.

Quantities of type: \(x + y\eta\) for \(x, y \in \mathbb{R}\).

Addition: \((x_1 + y_1\eta) + (x_2 + y_2\eta) = (x_1 + x_2) + (y_1 + y_2)\eta\).

Comparisons:

\[
\begin{align*}
[x_1 + y_1\eta = x_2 + y_2\eta] & \iff [x_1 = x_2, \ y_1 = y_2] \\
[x_1 + y_1\eta < x_2 + y_2\eta] & \iff [(x_1 < x_2) \text{ or } (x_1 = x_2, \ y_1 < y_2)]
\end{align*}
\]
Verification algorithm

The pair \((\mu_D, \mu_S)\) satisfies \(\text{NCond}\) iff the pair \((\tilde{\mu}_D, \tilde{\mu}_S)\) satisfies \(\text{NCond}\) for \(\eta\) strictly positive and small enough.

Run \(\text{MaxFlow}\) on the input \((\mathcal{N}, \tilde{\mu}_D, \tilde{\mu}_S)\) by considering \(\eta\) as a formal parameter “as small as needed”.

Quantities of type: \(x + y\eta\) for \(x, y \in \mathbb{R}\).

Addition: \((x_1 + y_1\eta) + (x_2 + y_2\eta) = (x_1 + x_2) + (y_1 + y_2)\eta\).

Comparisons:

\[
\begin{align*}
[x_1 + y_1\eta = x_2 + y_2\eta] & \iff [x_1 = x_2, y_1 = y_2] \\
[x_1 + y_1\eta < x_2 + y_2\eta] & \iff [(x_1 < x_2) \text{ or } (x_1 = x_2, y_1 < y_2)]
\end{align*}
\]

On any given input, \(\text{MaxFlow}\) stops in finite time.

A posteriori, assign to \(\eta\) a value which is small enough not to reverse any strict inequality.
Consider any $\mu$ with $\text{supp}(\mu) = F$. We have

$$\mu_S(\{1', 2'\}) = \mu(3, 1') + \mu(4, 2') \leq \mu_D(\{3, 4\}),$$

which contradicts $\text{NCond}$ for $U = \{3, 4\}$.
Connectivity properties

Consider a bipartite matching structure \((\mathcal{D}, S, E, F)\). Associated \textbf{directed} graph: the nodes are \(\mathcal{D} \cup S\) and the arcs are

\[
d \rightarrow s, \quad \text{if } (d, s) \in E, \quad s \rightarrow d, \quad \text{if } (d, s) \in F.
\]

![Graphs](image.png)
Connectivity properties

**Thm.** For a bipartite matching structure \((\mathcal{D}, \mathcal{S}, \mathcal{E}, \mathcal{F})\) the following properties are equivalent:

1. There exists \(\mu\) such that \(\text{supp}(\mu) = \mathcal{F}\), \(\text{supp}(\mu_\mathcal{D}) = \mathcal{D}\), \(\text{supp}(\mu_\mathcal{S}) = \mathcal{S}\) and \(\mu\) satisfies \(\text{NCond}\).
2. The associated directed graph is strongly connected.
**Connectivity properties**

**Thm.** For a bipartite matching structure \((D, S, E, F)\) the following properties are equivalent:

1. There exists \(\mu\) such that \(\text{supp}(\mu) = F\), \(\text{supp}(\mu_D) = D\), \(\text{supp}(\mu_S) = S\) and \(\mu\) satisfies \(\mathsf{NCOND}\).

2. The associated directed graph is strongly connected.

**Thm.** If the associated directed graph of \((D, S, E, F)\) is strongly connected, then any bipartite matching model \([ (D, S, E, F), \mu, \mathsf{POL} ]\) has a unique strictly connected component with all states leading to it.
State space decomposition

The state space can be decomposed into facets, defined only by the non-empty classes.

Def. A facet is an ordered pair \((U, V)\) such that: \(U \subset \mathcal{D}, V \subset \mathcal{S}\) and \(U \times V \subset (\mathcal{D} \times \mathcal{S} - E)\).

\[
\begin{aligned}
& \mathcal{D}_\bullet(\mathcal{F}) = U, \quad \mathcal{D}_\circ(\mathcal{F}) = \mathcal{D}(V), \quad \mathcal{D}_\circ(\mathcal{F}) = \mathcal{D} - (\mathcal{D}_\bullet(\mathcal{F}) \cup \mathcal{D}_\circ(\mathcal{F})) \\
& \mathcal{S}_\bullet(\mathcal{F}) = V, \quad \mathcal{S}_\circ(\mathcal{F}) = \mathcal{S}(U), \quad \mathcal{S}_\circ(\mathcal{F}) = \mathcal{S} - (\mathcal{S}_\bullet(\mathcal{F}) \cup \mathcal{S}_\circ(\mathcal{F})).
\end{aligned}
\]
Sufficient conditions

Conditions $S\text{COND}$:

$$\mu_D(D_\circ(F)) + \mu_S(S_\circ(F)) > 1 - \mu(E \cap D_\circ(F) \times S_\circ(F)), \quad \forall F \neq (\emptyset, \emptyset)$$

Prop. (Sufficient conditions) A bipartite model with probability $\mu$ satisfying $S\text{COND}$ is stable under any admissible matching policy.

Proof. Variation of the linear Lyapunov function (number of unmatched customers):

<table>
<thead>
<tr>
<th></th>
<th>$D_\circ$</th>
<th>$D_\circ$</th>
<th>$D_\bullet$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S_\circ$</td>
<td>$-1$</td>
<td>$0$</td>
<td>$0$</td>
</tr>
<tr>
<td>$S_\circ$</td>
<td>$0$</td>
<td>$0$ or $1$</td>
<td>$1$</td>
</tr>
<tr>
<td>$S_\bullet$</td>
<td>$0$</td>
<td>$1$</td>
<td>$1$</td>
</tr>
</tbody>
</table>
Sufficient conditions

Def. A facet $F$ is called saturated if $D_o(F) = \emptyset$ or $S_o(F) = \emptyset$.

$SCond \implies NCond$ (considering only the saturated facets).

For the NN graph:
$SCond = \{ NCond \cap (\mu_D(1) + \mu_S(1') > 1 - \mu(2,2')) \}.$

For $\mu = \mu_D \times \mu_S$ and $\mu_D = \mu_S = (x, y, 1 - x - y)$:

$NCond : \begin{cases} x < 0.5 \\ 2x + y > 1 \end{cases}$

$SCond : \begin{cases} NCond \\ 2x + y^2 > 1 \end{cases}$
Match the Longest has maximal stability region

**Match the Longest** (ML) policy: a newly arriving customer of class $c$ is matched to a server in $S(c)$ with the largest buffer (similarly for newly arriving server).

**Thm.** For any bipartite graph, ML has a maximal stability region.
Match the Longest has maximal stability region

Match the Longest (ML) policy: a newly arriving customer of class $c$ is matched to a server in $S(c)$ with the largest buffer (similarly for newly arriving server).

**Thm.** For any bipartite graph, ML has a maximal stability region.

**Proof:**

- Quadratic Lyapunov function: $L(x, y) = \sum_{d \in D} x_d^2 + \sum_{s \in S} y_s^2$.
- ML minimizes the value of this Lyapunov function at each step.
Match the Longest has maximal stability region

Match the Longest (ML) policy: a newly arriving customer of class $c$ is matched to a server in $S(c)$ with the largest buffer (similarly for newly arriving server).

**Thm.** For any bipartite graph, ML has a maximal stability region.

**Proof:**

- **Quadratic Lyapunov function:** $L(x, y) = \sum_{d \in D} x_d^2 + \sum_{s \in S} y_s^2$.
- ML minimizes the value of this Lyapunov function at each step.
- **Facet-dependent randomized policy.** In a non-zero facet $F$: the server $s \in S_\bullet(F)$ is matched to $d \in D_\bullet(F) \cap D(s)$ with probability $P^F_{sd}$. These probabilities can be chosen such that:

\[
\forall d \in D_\bullet, \quad \sum_{s \in S(d)} \mu_S(s) P^F_{sd} > \mu_D(d).
\]

(symmetrically for customers)

- For this randomized policy stability can be shown using Foster-Lyapunov criterion.
Priorities and Match the Shortest are not always stable

Prop. NN model with either the MS policy or the PR (priority) policy such that customers of class 1 (resp. servers of class 1′) give priority to servers of class 2′ (resp. to customers of class 2):

For both policies, the stability region is not maximal.

Consider \( \mu_D = (1/3, 2/5, 4/15) \), \( \mu_S = \mu_D \), and \( \mu = \mu_D \times \mu_S \). \( \text{NCOND} \) are satisfied, but the Markov chain is transient (for MS or PR as above).
Stability region for Match the shortest
Optimization

Cost function $c$ on buffer levels.

Average-cost:

$$\eta = \limsup_{N \to \infty} \frac{1}{N} \sum_{t=0}^{N-1} \mathbb{E}[c(Q(t))]$$
Optimization

Cost function $c$ on buffer levels.

Average-cost:

$$\eta = \limsup_{N \to \infty} \frac{1}{N} \sum_{t=0}^{N-1} \mathbb{E}[c(Q(t))]$$

Queue dynamics: $Q(t + 1) = Q(t) - U(t) + A(t), \quad t \geq 0$
Optimization

Cost function $c$ on buffer levels.

Average-cost:

$$\eta = \limsup_{N \to \infty} \frac{1}{N} \sum_{t=0}^{N-1} \mathbb{E}[c(Q(t))]$$

Queue dynamics: $Q(t + 1) = Q(t) - U(t) + A(t), \quad t \geq 0$

Input process $U$ represents the sequence of matching activities. Input space:

$$U_\diamond = \left\{ \sum_{e \in E} n_e u^e : n_e \in \mathbb{Z}_+ \right\}$$

with $u^e = 1^i + 1^j$ for $e = (i, j) \in \mathcal{E}$. 
**Optimization**

Cost function \( c \) on buffer levels.

Average-cost:

\[
\eta = \limsup_{N \to \infty} \frac{1}{N} \sum_{t=0}^{N-1} \mathbb{E}[c(Q(t))]
\]

Queue dynamics: \( Q(t+1) = Q(t) - U(t) + A(t), \quad t \geq 0 \)

Input process \( U \) represents the sequence of matching activities. Input space:

\[
U_\diamond = \left\{ \sum_{e \in E} n_e u^e : n_e \in \mathbb{Z}_+ \right\}
\]

with \( u^e = 1^i + 1^j \) for \( e = (i, j) \in \mathcal{E} \).

\( X(t) = Q(t) + A(t) \) the state process of the MDP model.

\[
X(t+1) = X(t) - U(t) + A(t+1)
\]

The state space \( X_\diamond = \{ x \in \mathbb{Z}^\ell_+ : \xi^0 \cdot x = 0 \} \) with \( \xi^0 = (1, \ldots, 1, -1, \ldots, -1) \).
Workload

For any $D \subset \mathcal{D}$, corresponding workload vector $\xi^D$ defined so that

$$\xi^D \cdot x = \sum_{i \in D} x_i^D - \sum_{j \in S(D)} x_j^S$$
Workload

For any $D \subset \mathcal{D}$, corresponding workload vector $\xi^D$ defined so that

$$\xi^D \cdot x = \sum_{i \in D} x^D_i - \sum_{j \in S(D)} x^S_j$$

Necessary and sufficient condition for a stabilizing policy:

$\text{NCond: } \delta_D := -\xi^D \cdot \alpha > 0$ for each $D$

$\alpha = E[A(t)]$ arrival rate vector.
Workload

For any $D \subset \mathcal{D}$, corresponding workload vector $\xi^D$ defined so that

$$\xi^D \cdot x = \sum_{i \in D} x^D_i - \sum_{j \in S(D)} x^S_j$$

Necessary and sufficient condition for a stabilizing policy:

$\text{NCond: } \delta_D := -\xi^D \cdot \alpha > 0$ for each $D$

$$\alpha = \mathbb{E}[A(t)]$$ arrival rate vector.

Why is this workload? Consistent with routing/scheduling models:

Fluid model,

$$\frac{d}{dt} x(t) = -u(t) + \alpha$$

The minimal time to reach the origin from $x(0) = x$: $T^*(x) = \max_D \frac{\xi^D \cdot x}{\delta_D}$
Workload

For any $D \subset \mathcal{D}$, corresponding workload vector $\xi^D$ defined so that

$$\xi^D \cdot x = \sum_{i \in D} x^D_i - \sum_{j \in S(D)} x^S_j$$

Necessary and sufficient condition for a stabilizing policy:

$\text{NCond}: \delta_D := -\xi^D \cdot \alpha > 0$ for each $D$

$\alpha = \mathbb{E}[A(t)]$ arrival rate vector.

Why is this workload? Consistent with routing/scheduling models:

Fluid model,

$$\frac{d}{dt} x(t) = -u(t) + \alpha$$

The minimal time to reach the origin from $x(0) = x$: $T^*(x) = \max_D \frac{\xi^D \cdot x}{\delta_D}$

Heavy-traffic: $\delta_D \sim 0$ for one or more $D$
Workload Dynamics

Fix one workload vector $\xi^D$; denote $(\xi, \delta)$ for $(\xi^D, \delta_D)$.

Workload $W(t) = \xi \cdot X(t)$
Workload Dynamics

Fix one workload vector $\xi^D$; denote $(\xi, \delta)$ for $(\xi^D, \delta_D)$.

Workload $W(t) = \xi \cdot X(t)$ can be positive or negative.
Workload Dynamics

Fix one workload vector \( \xi^D \); denote \((\xi, \delta)\) for \((\xi^D, \delta^D)\).

Workload \( W(t) = \xi \cdot X(t) \) can be positive or negative. Dynamics as in other queueing models,

\[
E[W(t+1) - W(t) | X(t), U(t)] \geq -\delta
\]
Workload Dynamics

Fix one workload vector $\xi^D$; denote $(\xi, \delta)$ for $(\xi^D, \delta_D)$.

Workload $W(t) = \xi \cdot X(t)$ can be positive or negative. Dynamics as in other queueing models,

$$E[W(t+1) - W(t) \mid X(t), U(t)] \geq -\delta$$

Achieved $\iff S(D)$ matches with $D$ only.
Fix one workload vector $\xi^D$; denote $(\xi, \delta)$ for $(\xi^D, \delta_D)$.

Workload $W(t) = \xi \cdot X(t)$ can be positive or negative.
Dynamics as in other queueing models,

$$E[W(t+1) - W(t) \mid X(t), U(t)] \geq -\delta$$

Achieved $\iff S(D)$ matches with $D$ only.

Workload relaxation: take this as the model for control.
Relaxations

A workload relaxation takes this as the model for control:

One Dimensional Workload relaxation,

\[
\hat{W}(t + 1) = \hat{W}(t) - \delta + I(t) + N(t + 1)
\]

\(\hat{W}(t + 1)\) = \(\hat{W}(t)\) \(-\delta\) \(+ I(t)\) \(+ N(t + 1)\)

Idleness \(\geq 0\) \(\text{Zero mean}\)
Relaxations

A workload relaxation takes this as the model for control:
One Dimensional Workload relaxation,

\[
\hat{W}(t+1) = \hat{W}(t) - \delta + I(t) + N(t+1)
\]

Idleness \( \geq 0 \)  
Zero mean

Effective cost \( \bar{c} : \mathbb{R} \rightarrow \mathbb{R}_+ \): Given a cost function \( c \) for \( Q \),

\[
\bar{c}(w) = \min \{ c(x) : \xi \cdot x = w \}
\]
Relaxations

A workload relaxation takes this as the model for control:

One Dimensional Workload relaxation,

\[ \hat{W}(t + 1) = \hat{W}(t) - \delta + I(t) + N(t + 1) \]

(Idleness \( \geq 0 \))

Zero mean

Effective cost \( \bar{c} : \mathbb{R} \rightarrow \mathbb{R}_+ \): Given a cost function \( c \) for \( Q \),

\[ \bar{c}(w) = \min\{c(x) : \xi \cdot x = w\} \]

piecewise linear if \( c \) is linear
Relaxations

A workload relaxation takes this as the model for control:

One Dimensional Workload relaxation,

\[ \hat{W}(t+1) = \hat{W}(t) - \delta + I(t) + N(t+1) \]

where \( \hat{W}(t) \) represents the workload at time \( t \), \( \delta \) is a constant, \( I(t) \) is the idle time at time \( t \), and \( N(t+1) \) is a zero mean random variable.

Effective cost \( \bar{c} : \mathbb{R} \rightarrow \mathbb{R}_+ \): Given a cost function \( c \) for \( Q \),

\[ \bar{c}(w) = \min\{c(x) : \xi \cdot x = w\} \]

is piecewise linear if \( c \) is linear.

Conclusions

Control of the relaxation = inventory model of Clark & Scarf

Hedging policy, with threshold \( \bar{r} \): *Idling is not permitted unless* \( \hat{W}(t) < -\bar{r} \)
Relaxations

A workload relaxation takes this as the model for control:

One Dimensional Workload relaxation,

\[
\hat{W}(t+1) = \hat{W}(t) - \delta + \underbrace{I(t)}_{\text{Idleness } \geq 0} + \underbrace{N(t+1)}_{\text{Zero mean}}
\]

Effective cost \( \bar{c} : \mathbb{R} \rightarrow \mathbb{R}_+ \): Given a cost function \( c \) for \( Q \),

\[
\bar{c}(w) = \min\{c(x) : \xi \cdot x = w\}
\]

piecewise linear if \( c \) is linear

Conclusions

Control of the relaxation = inventory model of Clark & Scarf

Hedging policy, with threshold \( \bar{r} \): \textit{Idling is not permitted unless} \( \hat{W}(t) < -\bar{r} \)

Heavy-traffic: For average-cost optimal control, \( \bar{r} \sim \frac{1}{2} \frac{\sigma^2}{\delta} \log(1 + c_+/c_-) \)
Tracking the Relaxation

\[ \begin{align*}
\phi^D_1 & = (0, 0, 1, -1, 0, 0) \\
W(t) & = Q^D_3(t) - Q^S_1(t)
\end{align*} \]

Relaxation: Matching of Supply 1 and Demand 2 allowed only if \( W(t) < -\bar{r} \)

Example 1
Cost:
\[ c(x) = x^D_1 + 2x^D_2 + 3x^D_3 + 3x^S_1 + 2x^S_2 + x^S_3 \]

Effective Cost:
\[ \bar{c}(w) = 4|w| \]

Example 2
Cost:
\[ c(x) = 3x^D_1 + 2x^D_2 + x^D_3 + 3x^S_1 + 2x^S_2 + x^S_3 \]

Effective Cost:
\[ \bar{c}(w) = \max(2w, -5w) \]
Tracking the Relaxation

\[ \xi^1 = (0, 0, 1, -1, 0, 0)^T \]
Tracking the Relaxation

$\xi^1 = (0, 0, 1, -1, 0, 0)^T$

$W(t) = Q_3^D(t) - Q_1^S(t)$

Relaxation:
Matching of Supply 1 and Demand 2 allowed only if $W(t) < -\bar{r}$
Tracking the Relaxation

Example 1
Cost: \( c(x) = x_1^D + 2x_2^D + 3x_3^D + 3x_1^S + 2x_2^S + x_3^S \)

\[ \longrightarrow \text{Effective Cost: } \bar{c}(w) = 4|w| \]
Tracking the Relaxation

$\xi^1 = (0, 0, 1, -1, 0, 0)^T$

$W(t) = Q^D_3(t) - Q^S_1(t)$

Relaxation:
Matching of Supply 1 and Demand 2 allowed only if $W(t) < -\bar{r}$

Example 2
Cost: $c(x) = 3x_1^D + 2x_2^D + x_3^D + 3x_1^S + 2x_2^S + x_3^S$

$\implies$ Effective Cost: $\bar{c}(w) = \max(2w, -5w)$
Tracking the Relaxation

\[ W(t) = Q_3^D(t) - Q_1^S(t) \]

**Example 1**

Cost: \( c(x) = x_1^D + 2x_2^D + 3x_3^D + 3x_1^S + 2x_2^S + x_3^S \)

\[ \Rightarrow \text{Effective Cost: } \bar{c}(w) = 4|w| \]

Matching of Supply 1 and Demand 2 allowed only if \( W(t) < -\bar{r} \)
Tracking the Relaxation \[ W(t) = Q^D_3(t) - Q^S_1(t) \]

**Example 1**

Cost: \[ c(x) = x^D_1 + 2x^D_2 + 3x^D_3 + 3x^S_1 + 2x^S_2 + x^S_3 \]

\[ \Rightarrow \text{Effective Cost: } \bar{c}(w) = 4|w| \]

Matching of Supply 1 and Demand 2 allowed only if \( W(t) < -\bar{r} \)
Tracking the Relaxation

\[ W(t) = Q_3^D(t) - Q_1^S(t) \]

**Example 1**

Cost: \( c(x) = x_1^D + 2x_2^D + 3x_3^D + 3x_1^S + 2x_2^S + x_3^S \)

\[ \implies \text{Effective Cost: } \bar{c}(w) = 4|w| \]

Matching of Supply 1 and Demand 2 allowed only if \( W(t) < -\bar{r} \)
Tracking the Relaxation

\[ W(t) = Q^D_3(t) - Q^S_1(t) \]

**Example 1**

Cost: \( c(x) = x^D_1 + 2x^D_2 + 3x^D_3 + 3x^S_1 + 2x^S_2 + x^S_3 \)

\[ \implies \text{Effective Cost: } \overline{c}(w) = 4|w| \]

Matching of Supply 1 and Demand 2 allowed only if \( W(t) < -\overline{r} \)

Workload Relaxation:

\[ Q^S_1(t) = Q^S_2(t) = 0 \quad \text{if } W(t) > 0 \]

\[ Q^D_2(t) = Q^D_3(t) = 0 \quad \text{if } W(t) < 0 \]
Tracking the Relaxation

\[ W(t) = Q_3^D(t) - Q_1^S(t) \]

**Example 1**

Cost: \[ c(x) = x_1^D + 2x_2^D + 3x_3^D + 3x_1^S + 2x_2^S + x_3^S \]

\[ \implies \text{Effective Cost: } \bar{c}(w) = 4|w| \]

Matching of Supply 1 and Demand 2 allowed only if \( W(t) < -\bar{r} \)

Workload Relaxation:

\[ Q_1^S(t) = Q_2^S(t) = 0 \quad \text{if } W(t) > 0 \]
\[ Q_2^D(t) = Q_3^D(t) = 0 \quad \text{if } W(t) < 0 \]
Tracking the Relaxation

\[ W(t) = Q_3^D(t) - Q_1^S(t) \]

**Example 1**

Cost: \( c(x) = x_1^D + 2x_2^D + 3x_3^D + 3x_1^S + 2x_2^S + x_3^S \)

\[ \implies \text{Effective Cost: } \bar{c}(w) = 4|w| \]

Average Cost Estimated in Simulation:

\[ \bar{r}^* = 14.9 \]

Matching of Supply 1 and Demand 2 allowed only if \( W(t) < -\bar{r} \)

Workload Relaxation:

\[ Q_1^S(t) = Q_2^S(t) = 0 \quad \text{if } W(t) > 0 \]

\[ Q_2^D(t) = Q_3^D(t) = 0 \quad \text{if } W(t) < 0 \]

Simulation with \( \bar{r}^* = 14.9 \)
Tracking the Relaxation

\[ W(t) = Q_3^D(t) - Q_1^S(t) \]

**Example 1**

Cost: \( c(x) = x_1^D + 2x_2^D + 3x_3^D + 3x_1^S + 2x_2^S + x_3^S \)

\( \implies \) Effective Cost: \( \bar{c}(w) = 4|w| \)

- Average Cost Comparisons:

  - Priority
  - MaxWeight
  - Threshold (15)

Matching of Supply 1 and Demand 2 allowed only if \( W(t) < -\bar{r} \)

Workload Relaxation:

\[ Q_1^S(t) = Q_2^S(t) = 0 \quad \text{if} \quad W(t) > 0 \]
\[ Q_2^D(t) = Q_3^D(t) = 0 \quad \text{if} \quad W(t) < 0 \]

Simulation with \( \bar{r}^* = 14.9 \)
Tracking the Relaxation

\[ W(t) = Q_3^D(t) - Q_1^S(t) \]

**Example 2**

Cost: \( c(x) = 3x_1^D + 2x_2^D + x_3^D + 3x_1^S + 2x_2^S + x_3^S \)

\( \Rightarrow \) Effective Cost: \( \bar{c}(w) = \max(2w, -5w) \)

Matching of Supply 1 and Demand 2 allowed only if \( W(t) < -\bar{r} \)
Tracking the Relaxation

\[ W(t) = Q_3^D(t) - Q_1^S(t) \]

Example 2

Cost: \( c(x) = 3x_1^D + 2x_2^D + x_3^D + 3x_1^S + 2x_2^S + x_3^S \)

\[ \implies \text{Effective Cost: } \bar{c}(w) = \max(2w, -5w) \]

Matching of Supply 1 and Demand 2 allowed only if \( W(t) < -\bar{r} \)

Workload Relaxation:

\( Q_1^S(t) = Q_2^S(t) = 0 \) if \( W(t) > 0 \)

\( Q_1^D(t) = Q_3^D(t) = 0 \) if \( W(t) < 0 \)
Tracking the Relaxation

\[ W(t) = Q_3^D(t) - Q_1^S(t) \]

**Example 2**

Cost: \( c(x) = 3x_1^D + 2x_2^D + x_3^D + 3x_1^S + 2x_2^S + x_3^S \)

\[ \implies \text{Effective Cost: } \bar{c}(w) = \max(2w, -5w) \]

Matching of Supply 1 and Demand 2 allowed only if \( W(t) < -\bar{r} \)

Workload Relaxation:

\[ Q_1^S(t) = Q_2^S(t) = 0 \quad \text{if } W(t) > 0 \]

\[ Q_1^D(t) = Q_3^D(t) = 0 \quad \text{if } W(t) < 0 \]
Tracking the Relaxation \[ W(t) = Q_3^D(t) - Q_1^S(t) \]

Example 2

Cost: \[ c(x) = 3x_1^D + 2x_2^D + x_3^D + 3x_1^S + 2x_2^S + x_3^S \]

\[ \implies \text{Effective Cost: } \bar{c}(w) = \max(2w, -5w) \]

Average Cost Estimated in Simulation:

Matching of Supply 1 and Demand 2 allowed only if \( W(t) < -\bar{r} \)

Workload Relaxation:

\[ Q_1^S(t) = Q_2^S(t) = 0 \quad \text{if } W(t) > 0 \]
\[ Q_1^D(t) = Q_3^D(t) = 0 \quad \text{if } W(t) < 0 \]

Simulation with \( \bar{r}^* = 7.2 \)
Tracking the Relaxation

\[ W(t) = Q_3^D(t) - Q_1^S(t) \]

**Example 2**

Cost: \( c(x) = 3x_1^D + 2x_2^D + x_3^D + 3x_1^S + 2x_2^S + x_3^S \)

\[ \implies \text{Effective Cost: } \bar{c}(w) = \max(2w, -5w) \]

Matching of Supply 1 and Demand 2 allowed only if \( W(t) < -\bar{r} \)

Workload Relaxation:

\[ Q_1^S(t) = Q_2^S(t) = 0 \quad \text{if } W(t) > 0 \]

\[ Q_1^D(t) = Q_3^D(t) = 0 \quad \text{if } W(t) < 0 \]

Simulation with \( \bar{r}^* = 7.2 \)
**h-MaxWeight**

**h-MaxWeight Policy:**  \( U(t) = \phi^{MW}(Q(t)) \)

\[
\phi^{MW}(x) = \arg \min_u E[\nabla h(x) \cdot \Delta(t + 1) \mid X(t) = x, U(t) = u]
\]

where \( \Delta(t + 1) = X(t + 1) - X(t) = -U(t) + A(t + 1) \)

**Average drift:**  
\[-\phi^{MW}(x) + \alpha =
\]

\[
E[\Delta(t + 1) \mid X(t) = x] = E[-U(t) + A(t + 1) \mid X(t) = x]
\]
\( h \)-MaxWeight

**h-MaxWeight Policy:** \( U(t) = \phi^{MW}(Q(t)) \)

\[ \phi^{MW}(x) = \arg \min_u E[\nabla h(x) \cdot \Delta(t + 1) \mid Q(t) = x, U(t) = u] \]

where \( \Delta(t + 1) = X(t + 1) - X(t) = -U(t) + A(t + 1) \)

*Hope that \( h \) approximates solution to a dynamic programming equation*

Average drift: \(-\phi^{MW}(x) + \alpha =\)

\[ E[\Delta(t + 1) \mid X(t) = x] = E[-U(t) + A(t + 1) \mid X(t) = x] \]
**h-MaxWeight**

**h-MaxWeight Policy:** \( U(t) = \phi^{MW}(Q(t)) \)

\[
\phi^{MW}(x) = \arg \min_u E[\nabla h(x) \cdot \Delta(t + 1) \mid Q(t) = x, U(t) = u]
\]

where \( \Delta(t + 1) = X(t + 1) - X(t) = -U(t) + A(t + 1) \)

Hope that \( h \) approximates solution to a dynamic programming equation

For average cost optimality, this means,

\[
E[h(Q(t + 1)) - h(Q(t)) \mid Q(t) = x] = \nabla h(x) \cdot [-\phi^{MW}(x) + \alpha] \approx -c(x)
\]

\[+ \frac{1}{2} E\left[\Delta(t + 1)^T \nabla^2 h(\bar{X}) \Delta(t + 1)\right] \quad \text{bounded}\]

Average drift: \(-\phi^{MW}(x) + \alpha = \)

\[
E[\Delta(t + 1) \mid X(t) = x] = E[-U(t) + A(t + 1) \mid X(t) = x]
\]
Asymptotic optimality

Family of arrival processes \( \{ A^\delta(t) \} \) parameterized by Additional assumptions:

(A1) For one set \( D \subsetneq \mathcal{D} \) we have \( \xi^D \cdot \alpha^\delta = -\delta \), where \( \alpha^\delta \) denotes the mean of \( A^\delta(t) \).
Moreover, there is a fixed constant \( \delta > 0 \) such that \( \xi^{D'} \cdot \alpha^\delta \leq -\delta \) for any \( D' \subsetneq \mathcal{D}, D' \neq D, \) and \( \delta \in [0, \bar{\delta}^\bullet] \).

(A2) The distributions are continuous at \( \delta = 0 \), with linear rate: For some constant \( b \),
\[
E[\|A^\delta(t) - A^0(t)\|] \leq b\delta. \tag{1}
\]

(A3) The sets \( E \) and \( F \) do not depend upon \( \delta \), and the graph associated with \( E \) is connected. Moreover, there exists \( i_0 \in S(D), j_0 \in D^c \), and \( \epsilon_1 > 0 \) such that
\[
P\{A^\delta_{i_0}(t) \geq 1 \text{ and } A^\delta_{j_0}(t) \geq 1 \} \geq \epsilon_1, \quad 0 \leq \delta \leq \bar{\delta}^\bullet. \tag{2}
\]
Asymptotic optimality

There is a function $h$ such that, under Assumptions (A1)–(A3), for sufficiently large $\kappa > 0$, $\beta > 0$, and sufficiently small $\delta_+ > 0$ (each independent of $\delta$), the average cost $\eta$ under the $h$-MaxWeight policy satisfies,

$$\hat{\eta}^* \leq \eta^* \leq \eta \leq \hat{\eta}^* + O(1)$$

where $\eta^*$ is the optimal average cost for the MDP model, $\hat{\eta}^*$ is the optimal average cost for the workload relaxation, and the constant $O(1)$ does not depend upon $\delta$.

The average cost for the relaxation satisfies the uniform bound,

$$\hat{\eta}^* = \hat{\eta}^{**} + O(1)$$

where $\hat{\eta}^{**}$ is the optimal cost for the diffusion approx. for the relaxation:

$$\hat{\eta}^{**} = \frac{1}{\Theta} \bar{c}_- \log \left( 1 + \frac{\bar{c}_+}{\bar{c}_-} \right), \quad \text{where} \quad \frac{1}{\Theta} = \frac{1}{2} \frac{\sigma^2}{\delta}.$$
Final remarks/related work

- Performance bounds?
- Approximate optimal control for relaxations in higher dimensions?
Final remarks/related work

- Performance bounds?
- Approximate optimal control for relaxations in higher dimensions?
- More general arrival assumptions. Admission control?
- Non-bipartite matching? Networks?
Final remarks/related work

- Performance bounds?
- Approximate optimal control for relaxations in higher dimensions?
- More general arrival assumptions. Admission control?
- Non-bipartite matching? Networks?
- Applications in energy systems and/or healthcare?
Related models

Bipartite matching model

Workload relaxations