# Dynamic Matching in Overloaded Waiting Lists

Jacob D. Leshno (2017)

Presented by Silvio Ravaioli

April 20, 2018

# Dynamic matching: An example

- Two agents, Duarte and Paul, applying for an apartment through Columbia Housing. Duarte prefers S (an apartment in the Southern part of the campus), Paul prefers N (an apartment in the Northern part of the campus), equal waiting cost
- S apartment arrives in period 1 and N apartment arrives in period 2
- 2 possible allocations:
  Duarte gets S immediately, Paul waits and gets N
  Paul gets S immediately, Duarte waits and gets N
- The first allocation is socially optimal, but Paul may prefer the second if he is impatient

# Overloaded waiting lists

- Overloaded system: we always have a list of agents waiting to be matched, whereas items are scarce
- Agents have different preferences, which are private information
- Impatient agents may misreport preferences to get assigned earlier
- Items arrive stochastically over time
- Welfare depends on waiting time and matching of items to agents

# This paper

How should we assign items dynamically to maximize welfare?

This is an interesting question!

- Several real-world examples: public housing, organs for transplant, nursing home spots, daycare centers,...
- The Chicago public housing authority runs approximately 20,000 apartments, spread throughout the city
- 60,000 applicants wait to be assigned on the capped waiting list
- Apartments become available stochastically over time as current tenants move out

# Literature

- Queueing
  - Congestion costs: Naor (1969), Hassin and Haviv (2006)
  - Organs: Zenios (1999), Su and Zenios (2004,2005), Alagoz, Mailllart, Schaefer, Roberts (2007)
  - Public housing: Kaplan (1986,8), Talreja and Whitt (2008), Caldentey, Kaplan, Weiss (2009)
- Dynamic market design: Unver (2010), Abdulkadiroglu and Loertscher (2007)
- Dynamic mechanism design: Bergermann and Said (2010), Gershkov and Moldovanu (2008,10), Lavi, Nisan (2005), Pavan, Segal, Toikka (2010)
- Rationing and Misallocation: Barzel (1974), Glaeser and Luttmer (2003)

# Roadmap

- **Model**
  Two types of agent and items
- **Benchmark policies**
  Random assignment
  First Come First Served (FCFS)
- **Optimal policy**
  Load Independent Expected Wait (LIEW) mechanism
- **Simplified robust policy**
  Service in Random Order (SIRO) buffer-queue mechanism
- **Extension**
  Multiple types of agents and items

# Model

- The agent arrival process makes the waiting list overloaded: at any time $t$ there are at least $|\mathcal{A}_t| \geq M \gg 0$ agents waiting
- We can abstract from the arrival process
- Two private types of agents: $\alpha$ with probability $p_\alpha$ and $\beta$ with probability $p_\beta = 1 - p_\alpha$
- Identical and constant per-period waiting cost $c$
- One item, $A$ or $B$, arriving each period: $A$ with probability $p_A$, or $B$ with probability $p_B = 1 - p_A$
- Items must be assigned in the period they arrive (discuss)
- For simplicity, no structural imbalance: $p_A = p_\alpha = p$ (discuss)
- Type $\alpha$ prefers item $A$, type $\beta$ prefers item $B$
- Item valuations: 1 (if preferred), v<1 (otherwise)

# Model

- The agent arrival process makes the waiting list overloaded: at any time $t$ there are at least $|\mathcal{A}_t| \geq M \gg 0$ agents waiting
- We can abstract from the arrival process
- Two private types of agents: $\alpha$ with probability $p_\alpha$ and $\beta$ with probability $p_\beta = 1 - p_\alpha$
- Identical and constant per-period waiting cost $c$
- One item, $A$ or $B$, arriving each period: $A$ with probability $p_A$, or $B$ with probability $p_B = 1 - p_A$
- Items must be assigned in the period they arrive (discuss)
- For simplicity, no structural imbalance: $p_A = p_\alpha = p$ (discuss)
- Type $\alpha$ prefers item $A$, type $\beta$ prefers item $B$
- Item valuations: 1 (if preferred), v<1 (otherwise)

## Welfare

- Our goal is to maximize expected welfare
- Welfare is defined as the sum of agent utility gains

**Lemma 1**

*Maximizing expected welfare is equivalent to minimizing the probability of misallocation.*

In an overloaded system the total waiting time is constant as every item is immediately matched. A policy can only shift the waiting time between agents, creating externalities but not modifying the total value.

# Benchmark policy I - Random assignment

- Agents do not express their preferences and they are automatically assigned
- Example: not allowing to decline apartments
- Long-run misallocation rate

$$\xi = lim\ sup_{T \to \infty} \frac{1}{T} \sum_t \xi_t$$

- Probability of mismatch under random assignment:

$$\xi^{Rand} = p_A p_\beta + p_B p_\alpha = 2p(1-p)$$

- Note: No misallocation under full information (and balanced setting)

## Benchmark policy II - FCFS

- Single waiting line for both goods
- Items are offered according to First Come First Served (FCFS)
- Given current information, the mechanism decides whether to approach an agent and ask her to report own type, or to assign the item automatically given current information
- **Agents know own position in the waiting list**

**Definition 4 - Lemma 2**

Agent a is an **approached agent** if the mechanism already asked $a$ to report own type but $a$ has not been assigned yet. Any mechanism with at most $K < M$ approached agent at any time has the same misallocation rate under any overloaded arrival process

# Benchmark policy II - FCFS

- Consider agent $\alpha$ being the first agent in the waiting line
- If she is offered item $A$, she will always accept
- When offered $B$, agent $\alpha$ can choose:
  - Take current mismatched $B$ item:
    $U_\alpha(b) = v$
  - Decline $B$, and take position $k$ in the waiting line for $A$:
    $E[U_\alpha(wait)] = 1 - c\frac{k}{p}$
  - Decline and avoid mismatch only if
    $k \leq K_\alpha = \left\lfloor p\frac{1-v}{c} \right\rfloor = \lfloor p\bar{w} \rfloor$

We can translate the waiting list with declines mechanism into the equivalent FCFS buffer-queue mechanism.

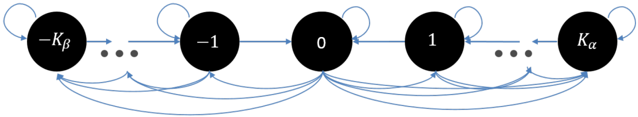
# FCFS - System dynamics

**Propositions 1 - 2**

There are at most $1 + max\{K_A, K_B\}$ approached agents at any time.
The dynamic behavior of the system can be captured by the state of the buffer-queue (number of agents who declined items).

- All the currently waiting agents who previously declined an offer must be of a single type
- The number of waiting agents of type $i$ must be $\leq K_i$
- The system is Markovian with state space

$$S = \{-K_\beta, ..., -2, -1, 0, 1, 2, ..., K_\alpha\}$$



# Welfare under FCFS

- Define $K_\alpha = \left\lfloor p\frac{1-v}{c} \right\rfloor$ and $K_\beta = \left\lfloor (1-p)\frac{1-v}{c} \right\rfloor$ as the maximum length of the buffers that satisfy IC
- EWL under FCFS in an unbalanced system $p_A \neq p_\alpha$

$$WFL^{FCFS} = (1-v)\xi^{FCFS} = (1-v)(p_A - p_\alpha)\frac{\left(\frac{p_\beta}{p_B}\right)^{K_\beta+1} + \left(\frac{p_\alpha}{p_A}\right)^{K_\alpha+1}}{\left(\frac{p_\beta}{p_B}\right)^{K_\beta+1} - \left(\frac{p_\alpha}{p_A}\right)^{K_\alpha+1}}$$

# Welfare under FCFS

**Theorem 1**

The expected welfare loss under FCFS in a balanced system $p_A = p_\alpha$ is

$$WFL^{FCFS} = (1 - v)\xi^{FCFS} = (1 - v)\frac{2p(1 - p)}{K_\alpha(1 - p) + K_\beta p + 1}$$

- Notice that the welfare loss only depends on $K_\alpha$ and $K_\beta$
- We can reduce the welfare loss by designing a buffer-queue policy that achieves higher $K_\alpha$ and $K_\beta$

# Welfare under FCFS

**Proposition 4**

Suppose the waiting list is overloaded and includes at least $M$ agents at any point in time. Under full information, in a balanced system, the mechanism can achieve a misallocation rate

$$\xi^{FB} \leq \frac{1}{2M}$$

and if the system is unbalanced

$$lim_{M \to \infty} \xi^{FB} = |p_A - p_\alpha|$$

**Corollary 1**

As the cost of waiting tends to zero the misallocation rate under FCFS buffer-queue policy tends to $\xi^{FB}$. In particular, if the system is balanced and the waiting list is overloaded, the misallocation rate is approximately zero.

# General buffer-queue policy

- In order to be IC, the expected waiting time $w_k$ for the agent at position $k \leq K_i$ must satisfy

$$w_k \leq \bar{w} = \frac{1 - v}{c}$$

- We can simplify the problem once more: we want to find the policy that minimizes the probability of *balking* (taking the mismatch item instead of waiting for the preferred one)
- We generalize FCFS: a queue policy is defined by a pair $\langle K, \varphi \rangle$
- Up to $K$ agents can be on the buffer-queue for an item
- Assign item with probability $\varphi(k, i)$ to agent in position $i$ when $k$ agents are on the buffer-queue

# General buffer-queue policy

**Definition 5**

A **buffer-queue policy** $\langle K, \varphi \rangle$ is defined by the maximal number of agents in the buffer-queue $K \in \mathbb{N}$ an nonnegative assignment probabilities $\varphi = \{\varphi(k,i)\}_{1 \le i \le k \le K}$ such that $\sum_i^k \varphi(k,i) = 1$ for all $1 \le k \le K$. If an item arrives when there are $k$ agents in the buffer queue, it will be allocated to the agent in position $i$ with probability $\varphi(k,i)$.

**Definition 6**

A **buffer-queue mechanism** $\mathcal{M} = (K_\alpha, \varphi_\alpha, K_\beta, \varphi_\beta)$ is defined by two buffer-queue policies: $\langle K_\alpha, \varphi_\alpha \rangle$ and $\langle K_\beta, \varphi_\beta \rangle$.

# General buffer-queue policy

- This extends the FCFS buffer-queue mechanism (waiting list with declines), which is equivalent to the buffer-queue mechanism $\mathcal{M}^{FCFS} = (K_\alpha^{FCFS}, \varphi_\alpha^{FCFS}, K_\beta^{FCFS}, \varphi_\beta^{FCFS})$
- Exception: $\mathcal{M}^{FCFS}$ leaves no choice when a buffer queue is full
- An agent is **truthful** if her strategy is to take an offered matching item, and decline a mismatched item iff the buffer queue for own preferred item is not full.
- A buffer-queue policy $\langle K, \varphi \rangle$ is **Incentive Compatible** (IC) if for any position $1 \leq k \leq K$ we have $w_k \leq \bar{w}$

# Optimal buffer-queue policy: LIEW

- The misallocation rate of a buffer-queue mechanism $\mathcal{M} = (K_\alpha, \varphi_\alpha, K_\beta, \varphi_\beta)$ is

$$\xi^{\mathcal{M}} = \frac{2p(1-p)}{(1-p)K_\alpha + pK_\beta + 1}$$

- How to achieve the upper bound for $K$?
- Optimal IC buffer-queue policy: every position has the same expected waiting time, that is also the expected waiting time of a random position
- Optimal buffer-queue policy: Load Independent Expected Wait (LIEW) queueing policy

# Optimal buffer-queue policy: LIEW

**Lemma 4**

The expected wait for a random position is independent of the assignment probabilities $E[w_{k\leq K}] = \frac{K+1}{2p}$

The expected waiting is the ratio between the expected number of agents waiting and the arrival rate $p$: $E[w_{k\leq K}] = \frac{1}{p}\sum_{k=1}^{K}\frac{k}{K} = \frac{K+1}{2p}$

**Proposition 6**

There is no IC policy with $K > K^* = \lfloor 2p\bar{w} \rfloor - 1$

For any IC policy we need $w_k \leq \bar{w}$ for every $k$, therefore IC also holds in expectation $E[w_{k\leq K}] \leq \bar{w}$

# Optimal buffer-queue policy: LIEW

**Definition 9**

A **Load Independent Expected Wait** ($LIEW_K$) buffer-queue policy is a $\langle K, \varphi \rangle$ policy such that, assuming all agents are truthful, the expected wait of an agent joining any position $1 \leq k \leq K$ is $w_k = \frac{K+1}{2p}$

**Theorem 2**

A LIEW buffer-queue mechanism achieves a weakly higher social welfare than any incentive-compatible Markovian mechanism.

The misallocation rate under $\mathcal{M}^*$ is

$$\xi^{OPT} = \frac{2p(1-p)}{K_\alpha(1-p)^* + K_\beta^* p + 1}$$

# FCFS vs LIEW

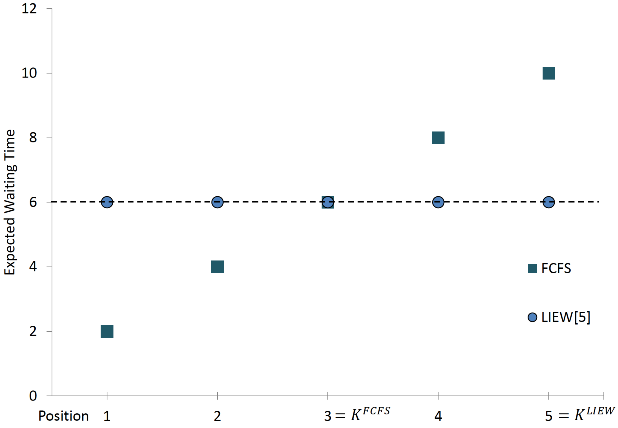



Figure 1: Expected waiting time per position $w_k$ when all agents are truthful under FCFS and LIEW(5), where $p_A = p = \frac{1}{2}$. The dotted line denotes the maximal acceptable wait $\bar{w} = \frac{1-v}{c} = 6$

# FCFS vs LIEW

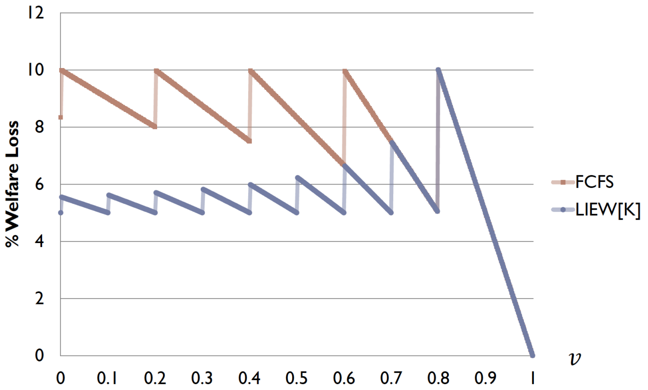



Figure 2: Welfare loss (WFL) for different policies and different values of $v$ (value of mismatched item), given as percentage of value lost due to misallocation. Parameters are $p = p_A = 1/2$ and $c = 0.1$.

# LIEW policy

- Generate LIEW as an intermediate policy between FCFS (incentive to stay) and LCFS (incentive to enter)
- Suppose you want to obtain LIEW(2): expected waiting time $\frac{3}{2p}$ in every position
- This policy achieves $K = 2$ when agents are willing to wait $\bar{w} = \frac{1-v}{c} = \frac{2}{p}$

$$\varphi^{LIEW(2)} = \begin{pmatrix} 1 & \\ \frac{1}{3-p} & \frac{2-p}{3-p} \end{pmatrix} \qquad \varphi^{LIEW(3)} = \begin{pmatrix} 1 & & \\ \frac{1}{2-p} & \frac{1-p}{2-p} & \\ 0 & \frac{1}{2} & \frac{1}{2} \end{pmatrix}$$

# Limits of LIEW policy

- Very complicated to explain, and somehow counterintuitive
- It strongly relies on agents' correct beliefs
  An agent may not join the queue if she is pessimistic and beliefs that "too many" agents will join as well
- It is heavily parameter dependent
  The designer needs to know $p$, $v$, and $c$
  If the parameters are wrong this mechanism performs poorly (even worse than FCFS)
- Truthful reporting is not a dominant strategy under $LIEW_K$

# Robust buffer-queue policy

- We want a mechanism that optimizes the buffer-queue while we maintain robustness
- Safe for agents: they do not regret joining if other agents join them after (regardless of their belief about the waiting list)
- Safe for the designer: robust to wrong parameters

# Robust buffer-queue policy

- A **scalable buffer-queue policy** $\langle \varphi, \mathcal{K} \rangle$ is given by the assignment weights $\nu$ such that $\varphi_\nu(k, i) = \frac{\nu_i}{\sum \nu_j}$ and maximal size selection function $K = \mathcal{K}(\bar{w})$
- A policy $\langle \nu, K \rangle$ is **belief free IC** (BF-IC) if for any belief $\sigma$ on following types $w_k^\sigma \leq \bar{w}$ if $k \leq K$
- A policy $\langle \nu, K \rangle$ is **weakly regret free** if for any belief $\sigma$ on following types the expected wait $W_{\sigma,k}(k, i) + 1 \leq \bar{w}$
- We need to strengthen the IC requirement to get as dominant strategy to report truthfully regardless of the agent's belief

# Service in Random Order (SIRO)

**Theorem 3**

The **Service in Random Order** (SIRO) BF-IC buffer-queue policy $\langle \nu^{SIRO}, \mathcal{K}_{SIRO} \rangle$ is the unique undominated BF-IC scalable buffer-queue policy.

- Equal probability to each waiting agent: $\varphi(k, i) = \frac{1}{k}$

$$\varphi^{SIRO} = \begin{pmatrix} 1 & & & & \\ \frac{1}{2} & \frac{1}{2} & & & \\ \frac{1}{3} & \frac{1}{3} & \frac{1}{3} & & \\ \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \\ \dots & \dots & \dots & \dots & \dots \end{pmatrix}$$

- Reminder: SIRO is less efficient than LIEW

# FCFS vs LIEW vs SIRO

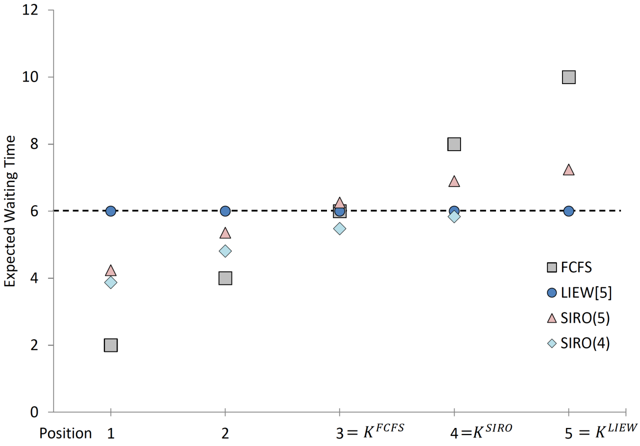



Figure 3: Expected waiting time per position $w_k$ when all agents are truthful under FCFS and LIEW(5) and SIRO, where $p_A = p = \frac{1}{2}$. The dotted line denotes the maximal acceptable wait $\bar{w} = \frac{1-v}{c} = 6$

# FCFS vs LIEW vs SIRO

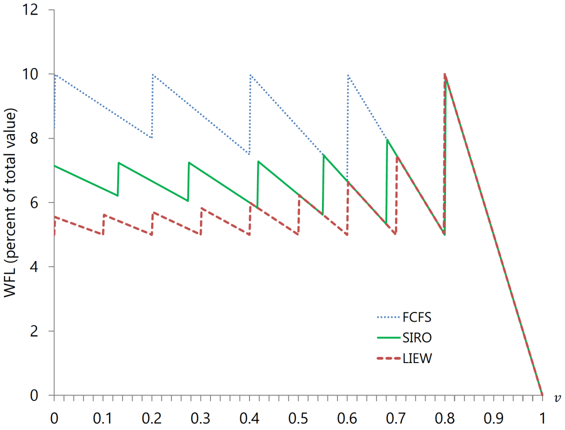



Figure 4: Welfare loss (WFL) for different policies and different values of $v$ (value of mismatched item), given as percentage of value lost due to misallocation. Parameters are $p = p_A = 1/2$ and $c = 0.1$. Red line: LIEW. Blue line: FCFS. Green: SIRO. All the mechanism have the maximal IC $K$.

# Properties of SIRO

- Simple to describe
- Parameter free
- Beliefs free
- Does not require to impose a restriction on the size of the buffer queue
- The position in the buffer queue is irrelevant
- SIRO achieves weakly higher welfare than FCFS under any parameter and belief
- SIRO captures more than half of the difference between LIEW and FCFS.

## Extension: Multiple items and types

- We can extend the previous results, but now the Markov chain representation is not tractable (multiple buffer-queues)
- We use simulations to compare SIRO and FCFS
- Finites set of items $G$ and finite sets of agent types $\Theta$
- Agents differ in their valuations over items but they have the same waiting cost

**Lemma 6**

Consider allocations $\mu, \mu'$ restricted up to period T. Under any realization, the difference between the total utility generated depends only on the difference in the number of mismatch $\Delta U(\mu, \mu') = \sum_t \sum_{g,\theta} v(g, \theta)(\xi_t(g, \theta; \mu) - \xi_t(g, \theta; \mu'))$

Total waiting costs are identical under any allocation $\mu$.

# Multi-item buffer-queue policy

- An agent is truthful if her strategy is to take an offered favorite item, and decline a mismatched item iff the buffer queue for her favorite item is not full.
- The truthful strategy is dominant if the buffer-queue policy is BF-IC for $p = \frac{1}{n}$
- Corollary 5: SIRO is the unique undominated BF-IC scalar buffer-queue policy for the separate buffer-queue mechanism in the symmetric n-item economy
- The SIRO mechanism can be explained with a simple verbal description
- When agents are relatively patient, SIRO strictly outperforms FCFS, and the gains increases with the number of types $n$

# Simulations

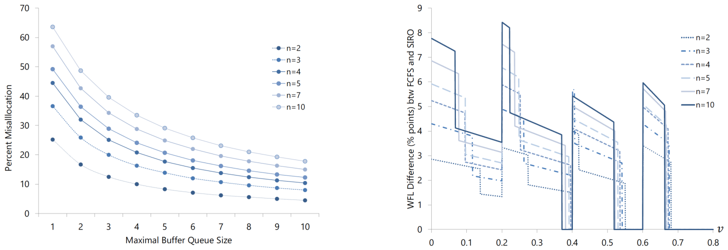



Figure 5: Simulation results for $n$ items. Agents have a value of 1 for a randomly chosen favorite item, other items have value $v$. Agents can decline non-favorite item and join the buffer queue for the item of their choice. Figure a (left) plots the misallocation rate given $n$ and maximal buffer queue size $K$. Figure b (right) plots the difference (in percentage points) between WFL under FCFS and under SIRO given $n$ and mismatch value $v$. Waiting cost is $c = 0.1 \cdot 2/n$.

# Discussion

1. The analysis focuses only on the class of buffer-queue mechanisms
2. Unfairness issue: agents are not assigned in order
   - But SIRO is fairer than FCFS as agents are offered a more equable expected wait
   - And the ordering of agents on waiting lists may be arbitrary
3. Note that LIEW and SIRO increase the variance of ex-post realized waiting time
4. If the waiting list is not overloaded, waiting cost enter the welfare analysis: tradeoff between misallocation and waiting
5. We ignore the Disjoint Queues mechanism
   - But agents' preferences may evolve over time
6. Agents who decline are a burden on the administration (delay)
   - But each agent is approached only one (items are standardized)