Performance paradox in stochastic dynamic matching models

Josu Doncel Vicente

mail: josu.doncel@ehu.eus
web: https://josudoncel.github.io/

joint work with J.M. Fourneau (U. Versailles), A. Busic and A. Cadas (Inria Paris)

Universidad del Pais Vasco / Euskal Herriko Unibertsitatea

December 19, 2023

Outline

Outline

Static Matching Models

Question:

Given a set of people with different likings, how to create couples so that the number of uncoupled people is minimized?

Problem formulation

Static Matching Models

Question:

Given a set of people with different likings, how to create couples so that the number of uncoupled people is minimized?

Problem formulation

- $G = (V, E)$, V is the set of people types and E represent the compatibilities (or likings)
- Initial population

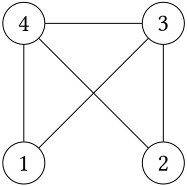
Static Matching Models

Question:

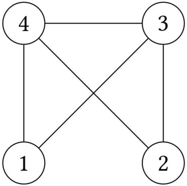
Given a set of people with different likings, how to create couples so that the number of uncoupled people is minimized?

Problem formulation

- $G = (V, E)$, V is the set of people types and E represent the compatibilities (or likings)
- Initial population



Static Matching Models



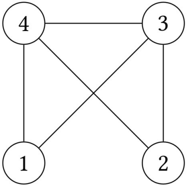
Known as the marriage problem (or assignment problem or the dancing problem...)

Widely studied in graph theory

Peterson 1890 and Konig 1937 $\Rightarrow$ Perfect matching

Questions:

Static Matching Models



Known as the marriage problem (or assignment problem or the dancing problem...)

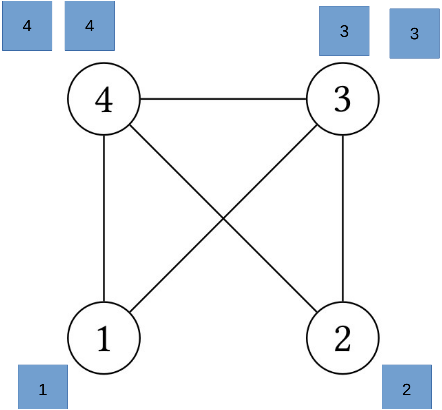
Widely studied in graph theory

Peterson 1890 and Konig 1937 $\Rightarrow$ Perfect matching

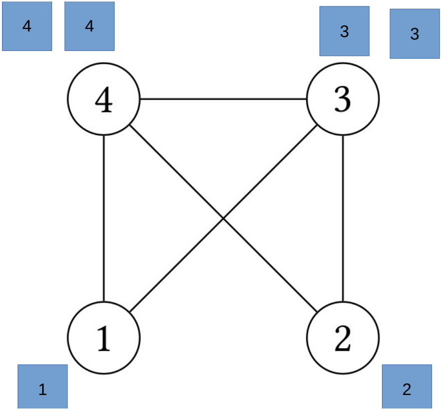
Questions:

Existence of the perfect matching? How to achieve it?

Example with 4 nodes



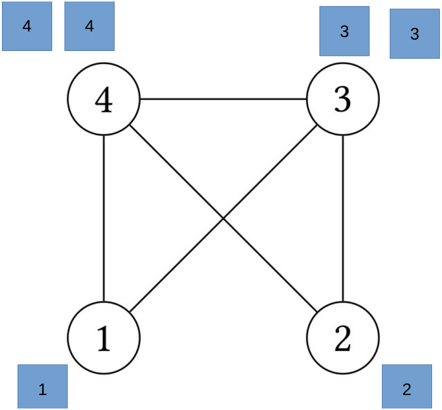
Example with 4 nodes



How to achieve the perfect matching?

- We prioritize matchings of 3 and 4
- We prioritize matchings of 1 and 2

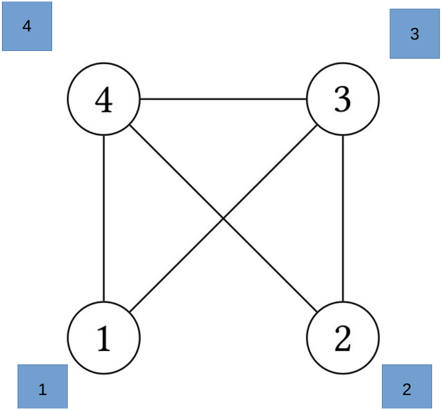
Example with 4 nodes



Policies to achieve the perfect matching?

- **We prioritize matchings of 3 and 4**
- We prioritize matchings of 1 and 2

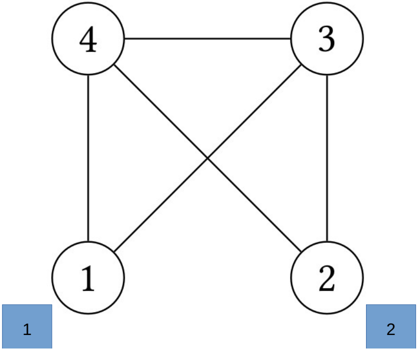
Example with 4 nodes



Policies to achieve the perfect matching?

- **We prioritize matchings of 3 and 4**
- We prioritize matchings of 1 and 2

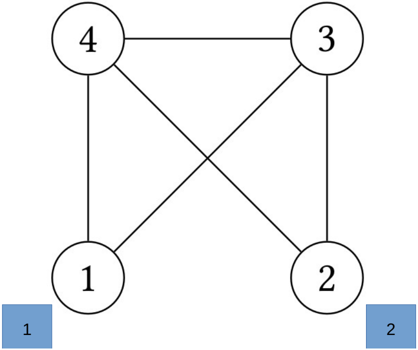
Example with 4 nodes



Policies to achieve the perfect matching?

- **We prioritize matchings of 3 and 4**
- We prioritize matchings of 1 and 2

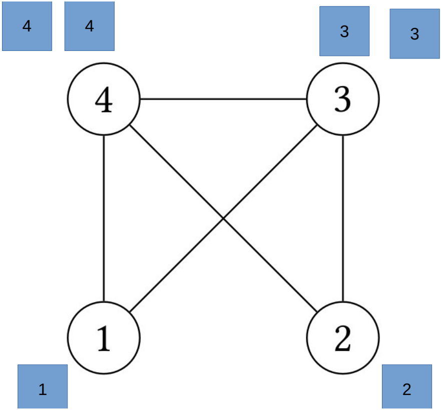
Example with 4 nodes



Policies to achieve the perfect matching?

- **We prioritize matchings of 3 and 4** ⇒ NOT Perfect Matching
- We prioritize matchings of 1 and 2

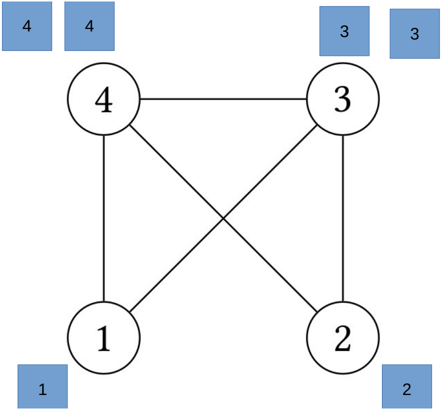
Example with 4 nodes



Policies to achieve the perfect matching?

- We prioritize matchings of 3 and 4
- **We prioritize matchings of 1 and 2**

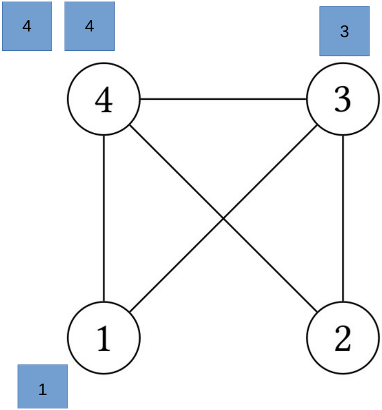
Example with 4 nodes



Policies to achieve the perfect matching?

- We prioritize matchings of 3 and 4
- **We prioritize matchings of 1 and 2**

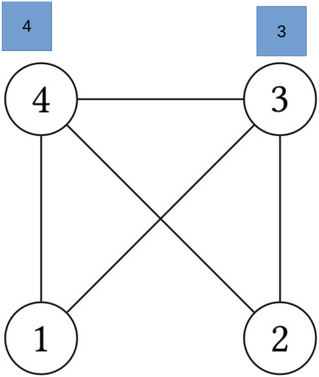
Example with 4 nodes



Policies to achieve the perfect matching?

- We prioritize matchings of 3 and 4
- **We prioritize matchings of 1 and 2**

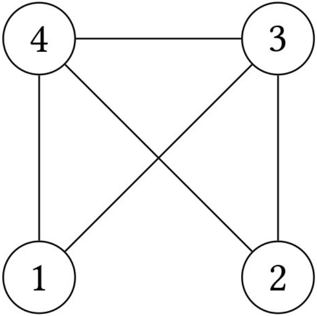
Example with 4 nodes



Policies to achieve the perfect matching?

- We prioritize matchings of 3 and 4
- **We prioritize matchings of 1 and 2**

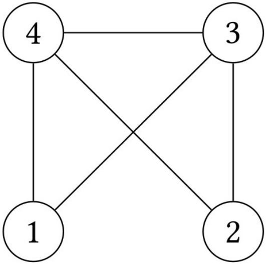
Example with 4 nodes



Policies to achieve the perfect matching?

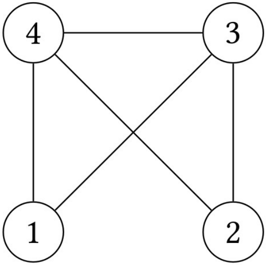
- We prioritize matchings of 3 and 4
- **We prioritize matchings of 1 and 2**$\Rightarrow$ Perfect Matching

Matching Models



Applications: Not only to marry people

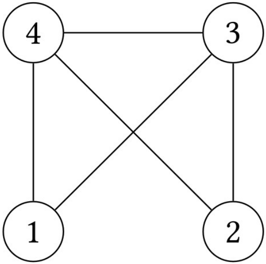
Matching Models



Applications: Not only to marry people

- Car sharing systems

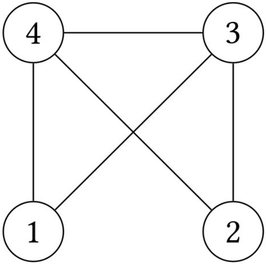
Matching Models



Applications: Not only to marry people

- Car sharing systems
- Online gaming

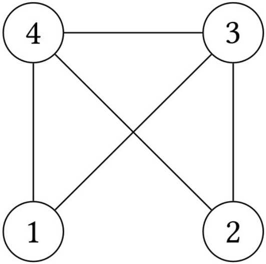
Matching Models



Applications: Not only to marry people

- Car sharing systems
- Online gaming
- Organ donations
- ....

Matching Models

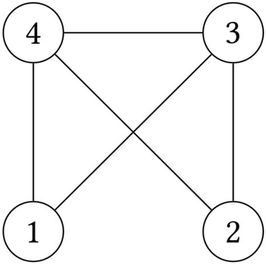


Applications: Not only to marry people

- Car sharing systems
- Online gaming
- Organ donations
- ....

In practice, elements arrive over time and ask for being matched

Matching Models



Applications: Not only to marry people

- Car sharing systems
- Online gaming
- Organ donations
- ....

In practice, elements arrive over time and ask for being matched ⇒ DYNAMIC

Dynamic Matching Model

The system is initially empty

Items arrive according to a random process

Dynamic Matching Model

The system is initially empty

Items arrive according to a random process to node *i*:

- Continuous time: Poisson process with rate λ_i
- Discrete time: with prob α_i in a given slot of time

Dynamic Matching Model

The system is initially empty

Items arrive according to a random process to node *i*:

- Continuous time: Poisson process with rate λ_i
- Discrete time: with prob α_i in a given slot of time

Unmatched items $\Rightarrow$ queue (infinite buffer)

Dynamic Matching Model

The system is initially empty

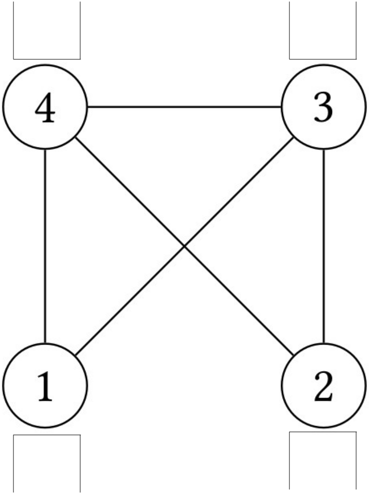
Items arrive according to a random process to node *i*:

- Continuous time: Poisson process with rate λ_i
- Discrete time: with prob α_i in a given slot of time

Unmatched items $\Rightarrow$ queue (infinite buffer)

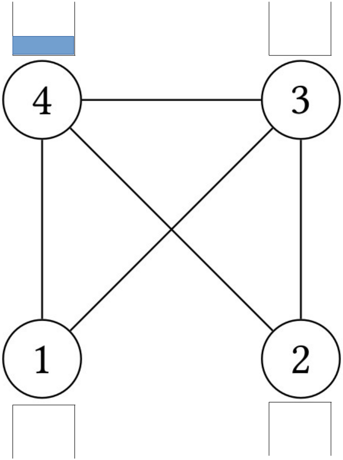
Introduced in Mairesse et al 2016

Dynamic Matching Model



Dynamic Matching Model

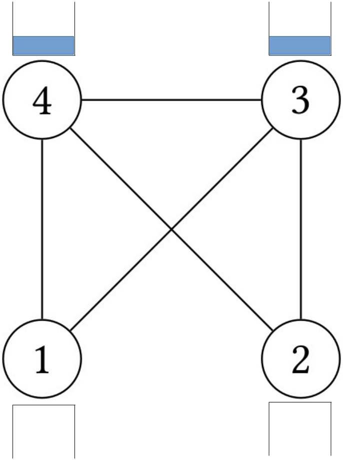
With probability α_4



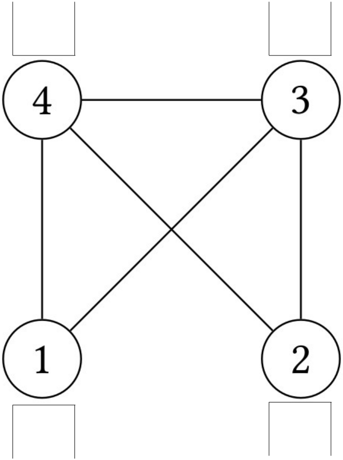
Dynamic Matching Model

With probability α_3

Compatibles $\Rightarrow$ Match and leave

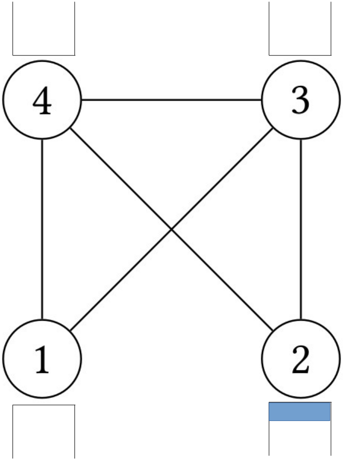


Dynamic Matching Model



Dynamic Matching Model

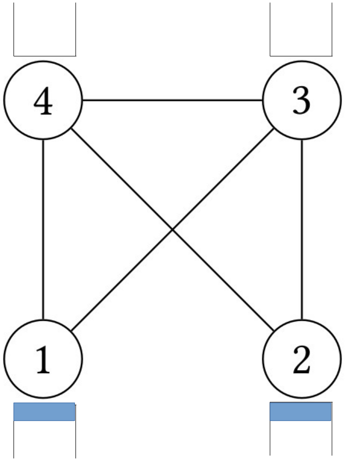
With probability α_2



Dynamic Matching Model

With probability α_1

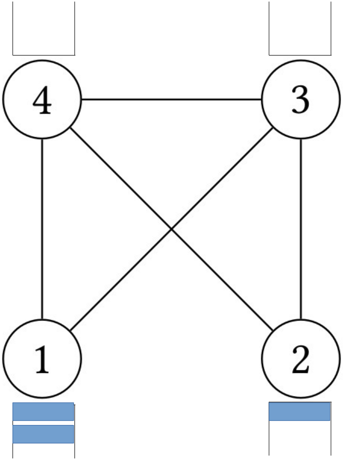
They are not compatible



Dynamic Matching Model

With probability α_1

They are not compatible



Dynamic Matching Model vs Queueing Models

Question: Is the dynamic matching model a queueing network?

Dynamic Matching Model vs Queueing Models

Question: Is the dynamic matching model a queueing network?

YES

Dynamic Matching Model vs Queueing Models

Question: Is the dynamic matching model a queueing network?

YES

Question: Arrivals from outside are random?

Dynamic Matching Model vs Queueing Models

Question: Is the dynamic matching model a queueing network?

YES

Question: Arrivals from outside are random?

YES

Dynamic Matching Model vs Queueing Models

Question: Is the dynamic matching model a queueing network?

YES

Question: Arrivals from outside are random?

YES

Question: What about service times?

Dynamic Matching Model vs Queueing Models

Question: Is the dynamic matching model a queueing network?

YES

Question: Arrivals from outside are random?

YES

Question: What about service times?

They are not i.i.d.
An item leaves only with a compatible item and it does it immediately (infinite service time)

Dynamic Matching Model vs Queueing Models

Question: Is the dynamic matching model a queueing network?

YES

Question: Arrivals from outside are random?

YES

Question: What about service times?

They are not i.i.d.
An item leaves only with a compatible item and it does it immediately (infinite service time)

QUEUEING THEORY CANNOT BE APPLIED DIRECTLY!

New research challenges!

Dynamic Matching Model

Matching Policy

When an incoming item is compatible with more than one item, the matching policy determines with which is it matched (and both leave the system)

Dynamic Matching Model

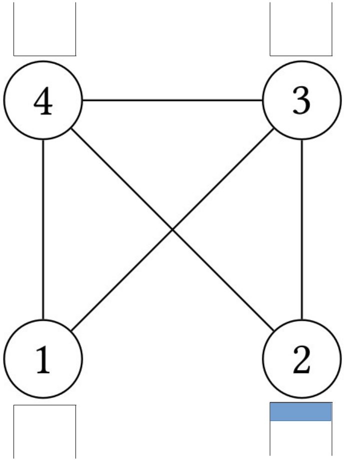
Matching Policy

When an incoming item is compatible with more than one item, the matching policy determines with which is it matched (and both leave the system)

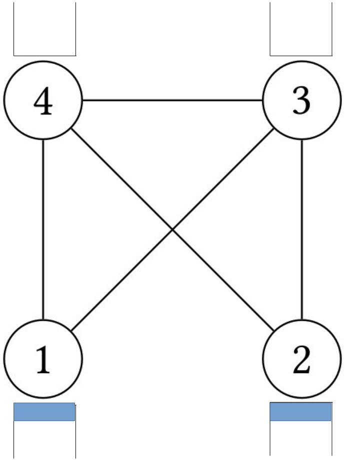
Examples:

- First-Come-First-Matched (FCFM)
- Match the Longest (ML)

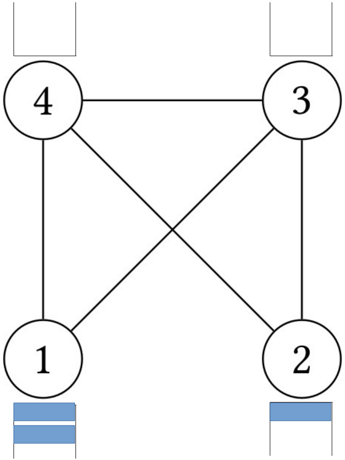
Dynamic Matching Model



Dynamic Matching Model

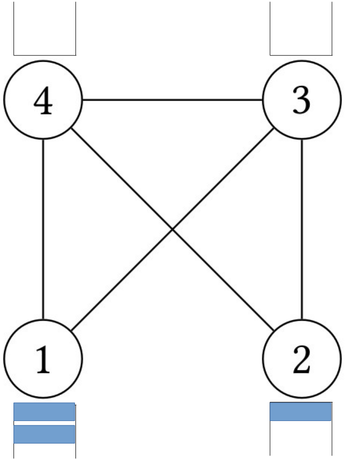


Dynamic Matching Model



Dynamic Matching Model

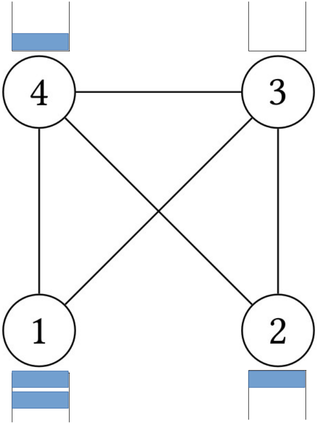
First-Come-First-Matched



Dynamic Matching Model

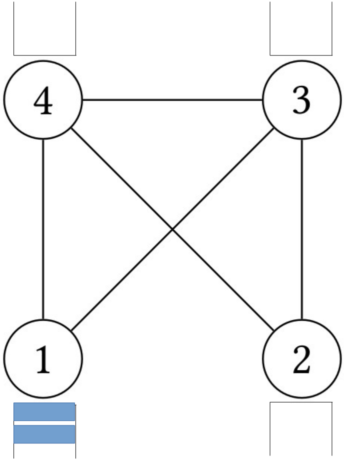
First-Come-First-Matched

When an item of type 4 arrives, it is matched with an item of type 2 (the oldest item)



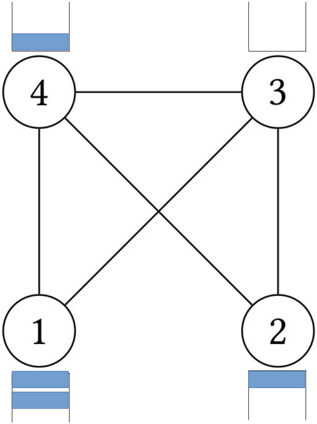
Dynamic Matching Model

First-Come-First-Matched

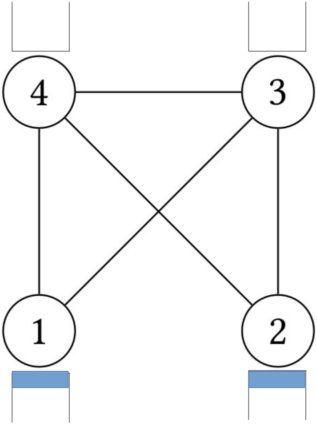


Match the Longest

When an item of type 4 arrives, it is matched with an item of type 1 (the longest queue)



Match the Longest



Dynamic Matching Model

Question: the number of unmatched items is a Markov chain?

Dynamic Matching Model

Question: the number of unmatched items is a Markov chain?

For a fixed matching policy: YES

The state is a word: $w = (w_1, w_2, \dots)$, where w_i is the i-th letter

Dynamic Matching Model

Question: the number of unmatched items is a Markov chain?

For a fixed matching policy: YES

The state is a word: $w = (w_1, w_2, \dots)$, where w_i is the i-th letter
The letters appear in order of arrival

Dynamic Matching Model

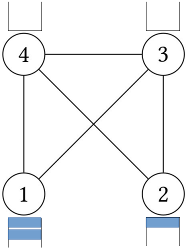
Question: the number of unmatched items is a Markov chain?

For a fixed matching policy: YES

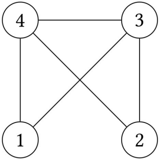
The state is a word: $w = (w_1, w_2, \dots)$, where w_i is the i-th letter

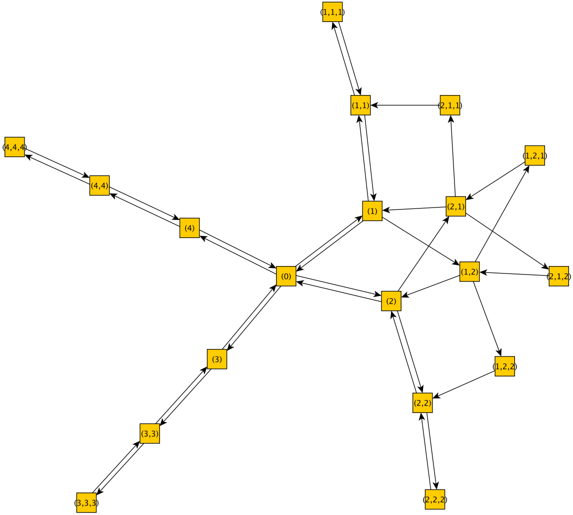
The letters appear in order of arrival

Example: $w = (2, 1, 1)$



Dynamic Matching Model





Outline

Dynamic Matching Models

We consider the discrete time version (derived from continuous time using uniformization)

$(G, \vec{\alpha}, \phi)$

Dynamic Matching Models

We consider the discrete time version (derived from continuous time using uniformization)

$(G, \vec{\alpha}, \phi)$

- G= (V,E): compatibility graph (item classes and compatibilities)

Dynamic Matching Models

We consider the discrete time version (derived from continuous time using uniformization)

$(G, \vec{\alpha}, \phi)$

- G= (V,E): compatibility graph (item classes and compatibilities)
- $\vec{\alpha} = (\alpha_1, \ldots, \alpha_n)$

Dynamic Matching Models

We consider the discrete time version (derived from continuous time using uniformization)

$(G, \vec{\alpha}, \phi)$

- G= (V,E): compatibility graph (item classes and compatibilities)
- $\vec{\alpha} = (\alpha_1, \ldots, \alpha_n)$
- ϕ: matching policy: FCFM, ML...

Dynamic Matching Models

We consider the discrete time version (derived from continuous time using uniformization)

$(G, \vec{\alpha}, \phi)$

- G= (V,E): compatibility graph (item classes and compatibilities)
- $\vec{\alpha} = (\alpha_1, \ldots, \alpha_n)$
- ϕ: matching policy: FCFM, ML...

Notation

- $\Gamma(i)$: the set of nodes that are compatible with node i
- $\alpha_{\Gamma(i)} = \sum_{j \in \Gamma(i)} \alpha_j$

Dynamic Matching Models

We consider the discrete time version (derived from continuous time using uniformization)

$(G, \vec{\alpha}, \phi)$

- G= (V,E): compatibility graph (item classes and compatibilities)
- $\vec{\alpha} = (\alpha_1, \ldots, \alpha_n)$
- ϕ: matching policy: FCFM, ML...

Notation

- $\Gamma(i)$: the set of nodes that are compatible with node i
- $\alpha_{\Gamma(i)} = \sum_{j \in \Gamma(i)} \alpha_j$

Definition: Independent set

The nodes that are not connected

$\mathcal{I}$: the set of all independent sets

Stability

Question:

How to characterize stability of $(G, \vec{\alpha}, \phi)$ (the Markov chain is recurrent)?

Stability

Question:

How to characterize stability of $(G, \vec{\alpha}, \phi)$ (the Markov chain is recurrent)?

Theorem (Mairesse et al, 2016)

A necessary condition for stability is

$$\alpha_I < \alpha_{\Gamma(I)}, \forall I \in \mathcal{I}$$

Stability

Question:

How to characterize stability of $(G, \vec{\alpha}, \phi)$ (the Markov chain is recurrent)?

Theorem (Mairesse et al, 2016)

A necessary condition for stability is

$$\alpha_I < \alpha_{\Gamma(I)}, \forall I \in \mathcal{I}$$

And for FCFM and ML, it is a necessary and sufficient condition.

Stability

Question:

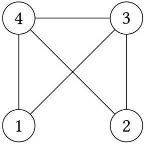
How to characterize stability of $(G, \vec{\alpha}, \phi)$ (the Markov chain is recurrent)?

Theorem (Mairesse et al, 2016)

A necessary condition for stability is

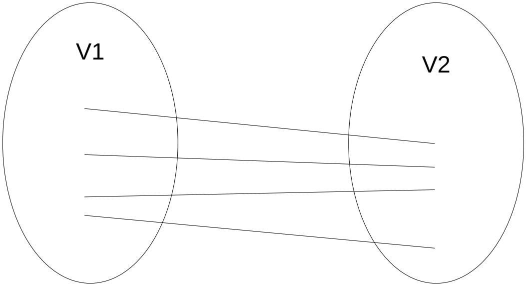
$$\alpha_I < \alpha_{\Gamma(I)}, \forall I \in \mathcal{I}$$

And for FCFM and ML, it is a necessary and sufficient condition.



$$\alpha_1 + \alpha_2 < \alpha_3 + \alpha_4$$
$$\alpha_3 < \alpha_1 + \alpha_2 + \alpha_4$$
$$\alpha_4 < \alpha_1 + \alpha_2 + \alpha_3$$

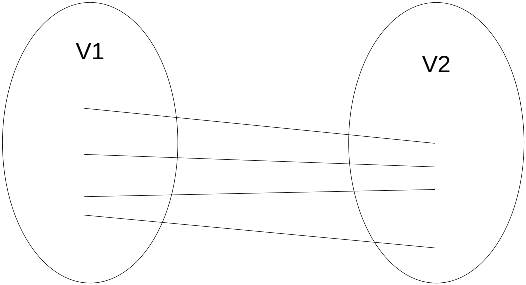
Stability



Theorem (Mairesse et al, 2016)

The compatibility graph cannot be bipartite

Stability

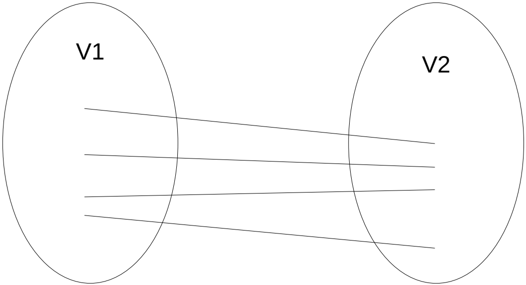


Theorem (Mairesse et al, 2016)

The compatibility graph cannot be bipartite

Proof: $\Gamma(V1) = V2$ and $\Gamma(V2) = V1$

Stability



Theorem (Mairesse et al, 2016)

The compatibility graph cannot be bipartite

Proof: $\Gamma(V1) = V2$ and $\Gamma(V2) = V1$
$\Rightarrow \alpha_{\Gamma(V1)} = \alpha_{V2} > \alpha_{V1}$ and $\alpha_{\Gamma(V2)} = \alpha_{V1} > \alpha_{V2}$ cannot be satisfied together!

Outline

Outline

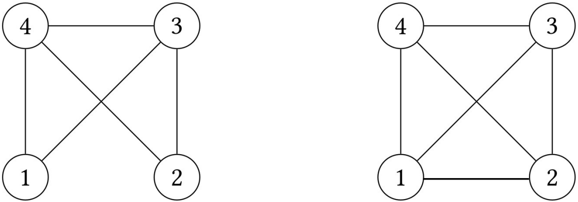
Performance Paradox in Dynamic Matching Models

Goal

We analyze the impact on the expected number of unmatched items when we add an edge in the compatibility graph

Goal

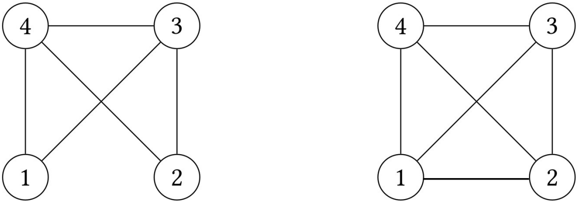
We analyze the impact on the expected number of unmatched items when we add an edge in the compatibility graph



Performance Paradox in Dynamic Matching Models

Goal

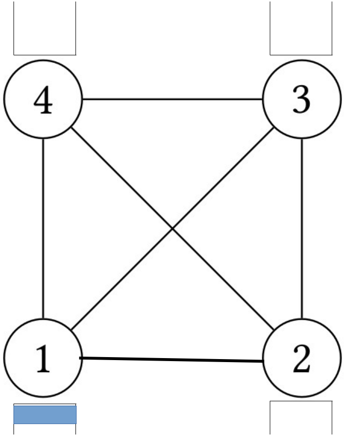
We analyze the impact on the expected number of unmatched items when we add an edge in the compatibility graph



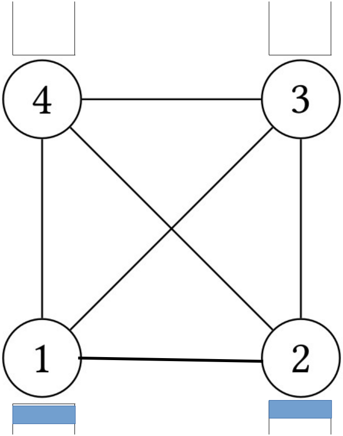
Intuitive Idea

The expected number of unmatched items cannot increase when we add an edge in the compatibility graph (more matchings can be done)

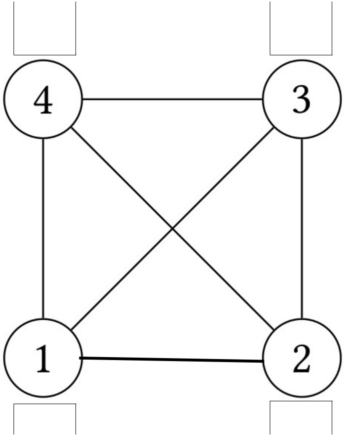
FCFM policy



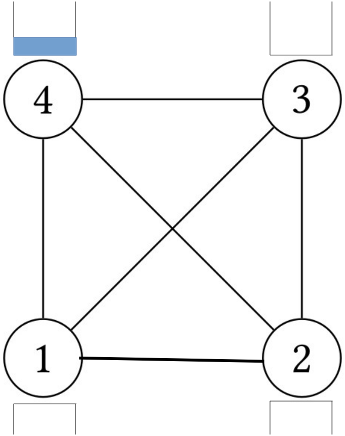
FCFM policy



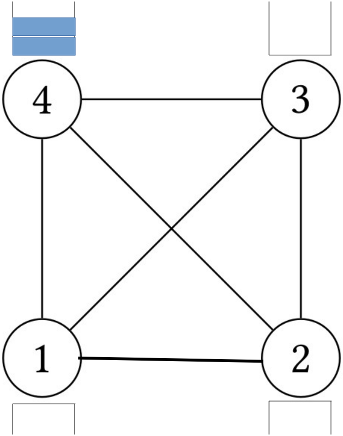
FCFM policy



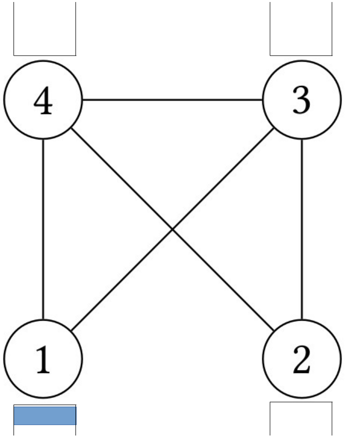
FCFM policy



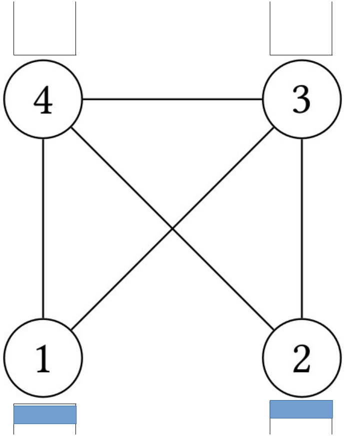
FCFM policy



FCFM policy

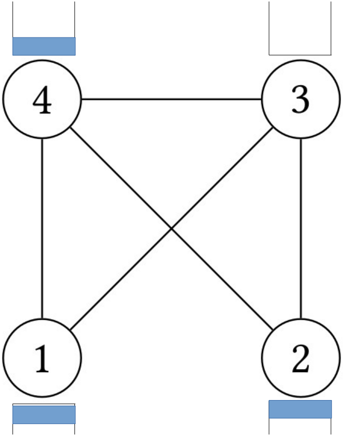


FCFM policy

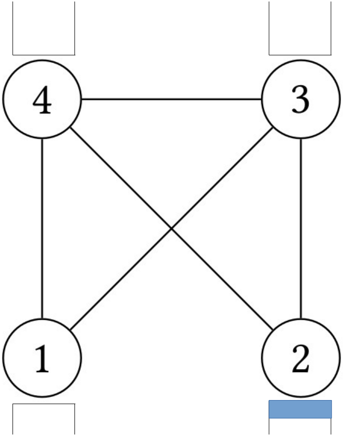


Performance Paradox in Dynamic Matching Models

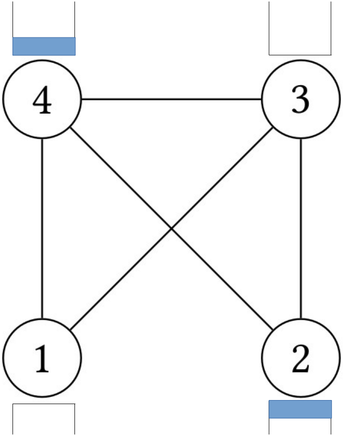
FCFM policy



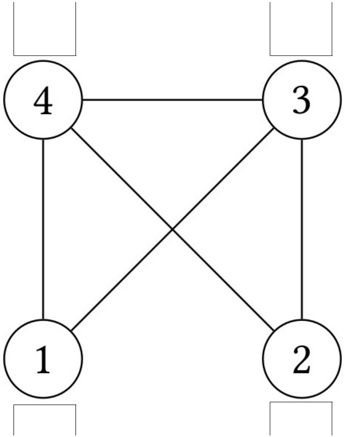
FCFM policy



FCFM policy

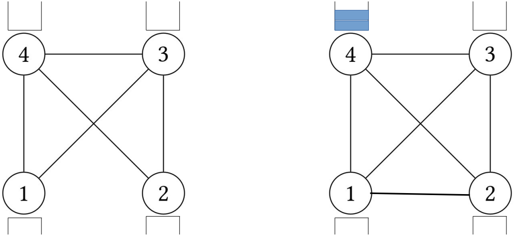


FCFM policy



Performance Paradox in Dynamic Matching Models

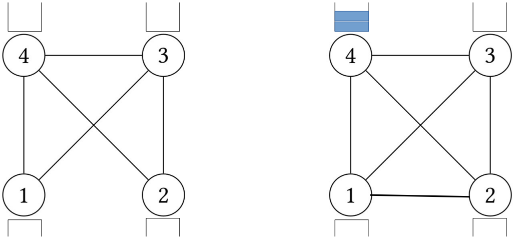
Our intuition is false!



Definition: Performance Paradox

Performance Paradox in Dynamic Matching Models

Our intuition is false!



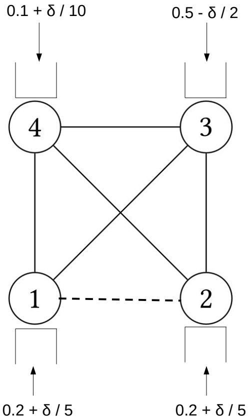
Definition: Performance Paradox

The performance paradox exists in dynamic matching model if, when we add an edge to the compatibility graph, then the expected number of unmatched items increases.

Analogous phenomenon to the Braess paradox

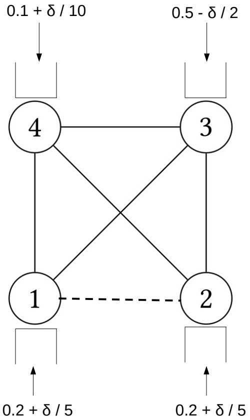
Our Results

Our Results



with $\delta \in (0, 0.1)$

Our Results



with $\delta \in (0, 0.1)$

Theorem

For the above compatibility graph and FCFM, the performance paradox exists if and only if $\delta < 0.0818$.

Our Results

Theorem

For the above compatibility graph and FCFM, the performance paradox exists if and only if $\delta < 0.0818$.

Sketch of the proof

The Markov chains of
- complete graph: single letter words
- quasicomplete graph: words with letters 1 and 2, words with a single letter (3 or 4)

Our Results

Theorem

For the above compatibility graph and FCFM, the performance paradox exists if and only if $\delta < 0.0818$.

Sketch of the proof

The Markov chains of
- complete graph: single letter words
- quasicomplete graph: words with letters 1 and 2, words with a single letter (3 or 4)

We compute the steady-state distribution of both Markov chains and compute the expected number of unmatched items
We simplify the obtained expressions

Our Results

Theorem

For the above compatibility graph and FCFM, the performance paradox exists if and only if $\delta < 0.0818$.

Sketch of the proof

The Markov chains of
- complete graph: single letter words
- quasicomplete graph: words with letters 1 and 2, words with a single letter (3 or 4)

We compute the steady-state distribution of both Markov chains and compute the expected number of unmatched items
We simplify the obtained expressions

Intuitive idea

When $\delta \to 0$, the stability condition in node 3 is nearly satisfied

Our Results: Arbitrary compatibility graph and FCFM

Our Results: Arbitrary compatibility graph and FCFM

Definition: Saturated nodes (or in heavy-traffic)

The node i is in saturation if, when $\delta \to 0$, then $\alpha_i - \alpha_{\Gamma(i)} \to 0$

Our Results: Arbitrary compatibility graph and FCFM

Definition: Saturated nodes (or in heavy-traffic)

The node i is in saturation if, when $\delta \to 0$, then $\alpha_i - \alpha_{\Gamma(i)} \to 0$
$\Rightarrow$ The stability condition is nearly satisfied in the saturated nodes (or independent sets)

Our Results: Arbitrary compatibility graph and FCFM

Definition: Saturated nodes (or in heavy-traffic)

The node i is in saturation if, when $\delta \to 0$, then $\alpha_i - \alpha_{\Gamma(i)} \to 0$
$\Rightarrow$ The stability condition is nearly satisfied in the saturated nodes (or independent sets)

We assume that there is a unique saturation node.

Theorem

When the node in saturation is connected to the nodes where the edge is added, there exists a performance paradox for δ small enough

Our Results: Arbitrary compatibility graph and FCFM

Definition: Saturated nodes (or in heavy-traffic)

The node i is in saturation if, when $\delta \to 0$, then $\alpha_i - \alpha_{\Gamma(i)} \to 0$
$\Rightarrow$ The stability condition is nearly satisfied in the saturated nodes (or independent sets)

We assume that there is a unique saturation node.

Theorem

When the node in saturation is connected to the nodes where the edge is added, there exists a performance paradox for δ small enough

Sketch of the proof

- We compute the expected number of unmatched items (Comte 2021)
- We study the obtained expressions when $\delta \to 0$ and check that the condition for the performance paradox is satisfied.

Outline

From FCFM to other matching policies

From FCFM to other matching policies

Definition: Greedy matching policy

A matching policy is greedy if when an arriving item is compatible with more than one item, it is matched with one of them

Definition: Greedy matching policy

A matching policy is greedy if when an arriving item is compatible with more than one item, it is matched with one of them

Examples: FCFM, ML, random, priorities...

From FCFM to other matching policies

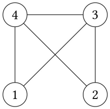
Definition: Greedy matching policy

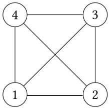
A matching policy is greedy if when an arriving item is compatible with more than one item, it is matched with one of them

Examples: FCFM, ML, random, priorities...

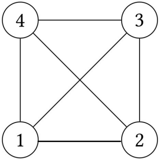
Lemma

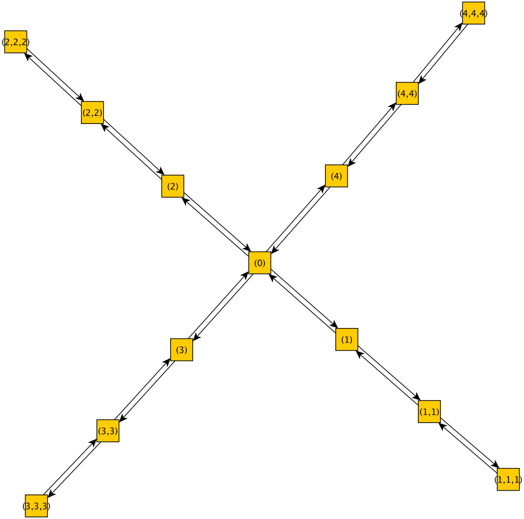
The Markov chains of the below models coincide for any greedy policy (same transitions and states).





(Truncated) Markov chain of the Complete graph

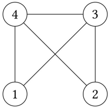


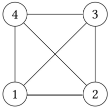


From FCFM to other matching policies

Lemma

The Markov chains of the below models coincide for any greedy policy (same transitions and states).





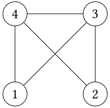
Theorem

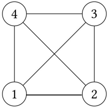
The performance paradox for the above graph exists for any greedy policy (because it exists for FCFM).

From FCFM to other matching policies

Lemma

The Markov chains of the below models coincide for any greedy policy (same transitions and states).



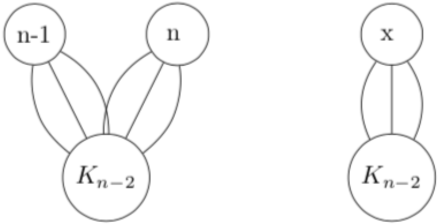


Theorem

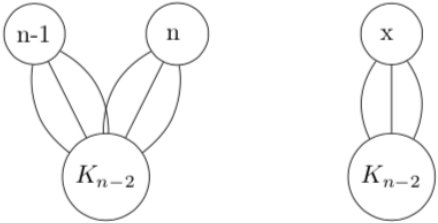
The performance paradox for the above graph exists for any greedy policy (because it exists for FCFM).

Large Networks using Aggregation

Large Networks using Aggregation



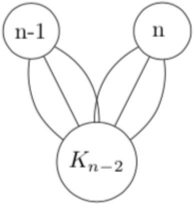
Large Networks using Aggregation

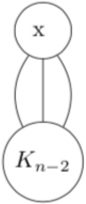


Lemma

Assuming that $\alpha_x = \alpha_n + \alpha_{n-1}$, the mean number of unmatched items in both models for any greedy policy is the same.

Large Networks using Aggregation





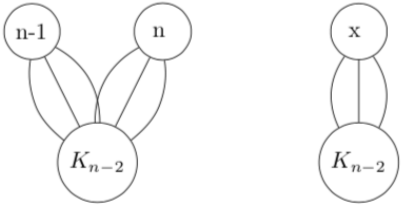
Lemma

Assuming that $\alpha_x = \alpha_n + \alpha_{n-1}$, the mean number of unmatched items in both models for any greedy policy is the same.

Sketch of the proof

The Markov chain of the left is lumpable in a way that all the items of n and $n-1$ are aggregated
The lumped Markov chain corresponds to that of the right model.

Large Networks using Aggregation

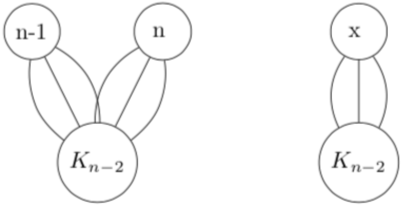


Lemma

Assuming that $\alpha_x = \alpha_n + \alpha_{n-1}$, the mean number of unmatched items in both models for any greedy policy is the same.

- The proof is valid if we replace K_{n-2} by any other compatibility graph

Large Networks using Aggregation

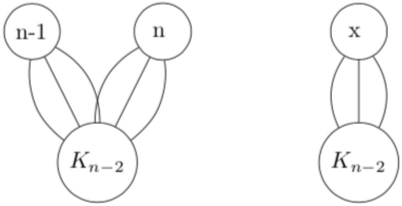


Lemma

Assuming that $\alpha_x = \alpha_n + \alpha_{n-1}$, the mean number of unmatched items in both models for any greedy policy is the same.

- The proof is valid if we replace K_{n-2} by any other compatibility graph
- The proof is valid if in the left model there are more than one nodes: $n, n-1, n-2, \ldots$

Large Networks using Aggregation



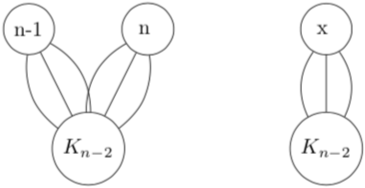
Lemma

Assuming that $\alpha_x = \alpha_n + \alpha_{n-1}$, the mean number of unmatched items in both models for any greedy policy is the same.

Theorem

If the model of the right has a performance paradox, so does the model of the left (with previous generalizations)

Large Networks using Aggregation



Lemma

Assuming that $\alpha_x = \alpha_n + \alpha_{n-1}$, the mean number of unmatched items in both models for any greedy policy is the same.

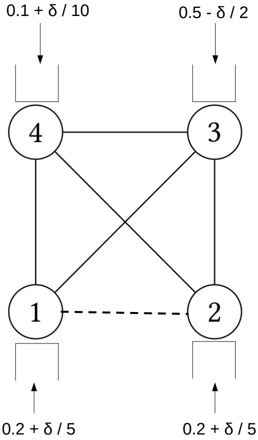
Theorem

If the model fo the right has a performance paradox, so does the model of the left (with previous generalizations)

$\Rightarrow$ Graphs with performance paradox can be as large as we want!

On the Assumption about the Unique Saturated Node

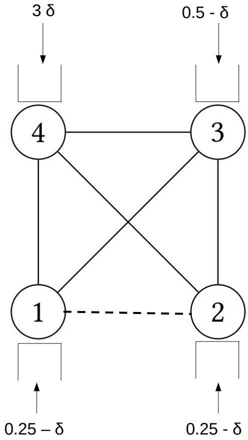
On the Assumption about the Unique Saturated Node



We assume there exists a unique node in saturation

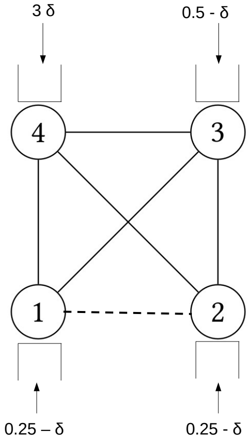
Example: Node 3

On the Assumption about the Unique Saturated Node



The independent sets $\{1, 2\}$ and $\{3\}$ are in saturation

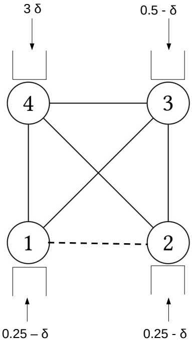
On the Assumption about the Unique Saturated Node



The independent sets $\{1, 2\}$ and $\{3\}$ are in saturation

There exists a performance paradox if and only if 0.0563.

On the Assumption about the Unique Saturated Node



The independent sets $\{1, 2\}$ and $\{3\}$ are in saturation

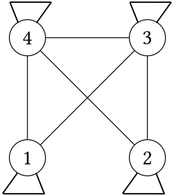
There exists a performance paradox if and only if $\delta < 0.0563$.

The uniqueness of the saturated node is a technical assumption for our main result about general graphs

Outline

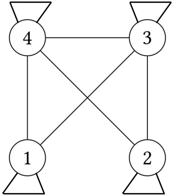
Dynamic Matching Model with Loops

We allow self-matchings in the compatibility graph



Dynamic Matching Model with Loops

We allow self-matchings in the compatibility graph



Our Result (Busic et al 2022)

For FCFM, the steady-state distribution of unmatched items has a product form expression

Dynamic Matching Model with Loops

Observation

In a matching model with loops, we have a finite-state Markov chain for any greedy policy and an arbitrary graph
$\Rightarrow$ Stability is not an issue!

Dynamic Matching Model with Loops

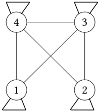
Observation

In a matching model with loops, we have a finite-state Markov chain for any greedy policy and an arbitrary graph
$\Rightarrow$ Stability is not an issue!

Current reseach in the following conjecture

For any policy and any compatibility graph, the performance paradox does not exist in dynamic matching models with loops

Work in progress: The performance paradox does not exist for FCFM and priorities and for the below graph.



Dynamic Matching Model with Loops

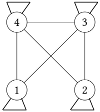
Observation

In a matching model with loops, we have a finite-state Markov chain for any greedy policy and an arbitrary graph
$\Rightarrow$ Stability is not an issue!

Current reseach in the following conjecture

For any policy and any compatibility graph, the performance paradox does not exist in dynamic matching models with loops

Work in progress: The performance paradox does not exist for FCFM and priorities and for the below graph.

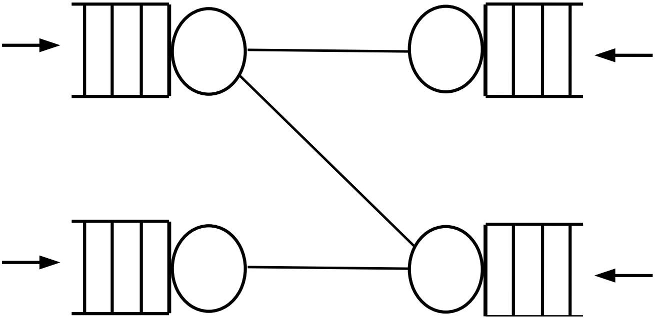


Bipartite Dynamic Matching Models

Assumption: random arrivals and in pairs

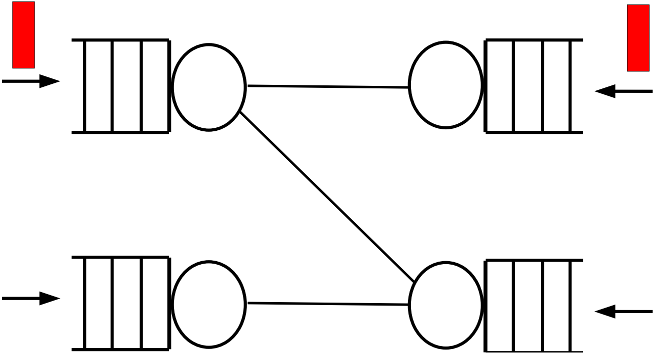
Bipartite Dynamic Matching Models

Assumption: random arrivals and in pairs



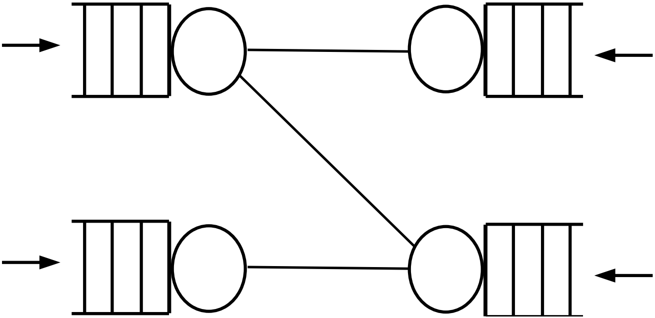
Bipartite Dynamic Matching Models

Assumption: random arrivals and in pairs

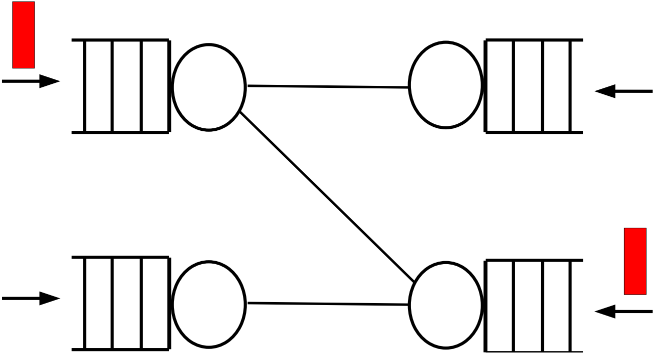


Bipartite Dynamic Matching Models

Assumption: random arrivals and in pairs

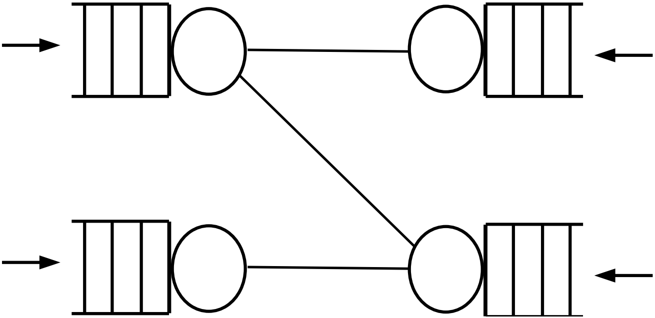


Assumption: random arrivals and in pairs



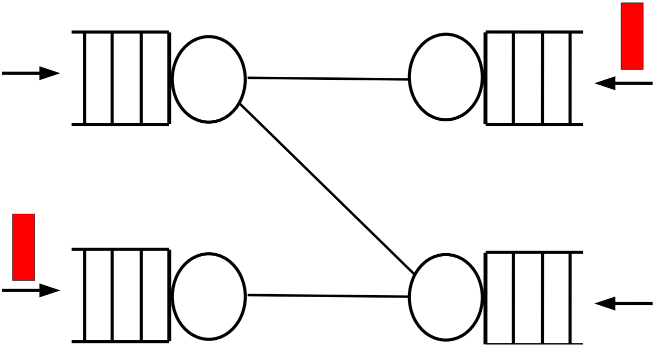
Bipartite Dynamic Matching Models

Assumption: random arrivals and in pairs



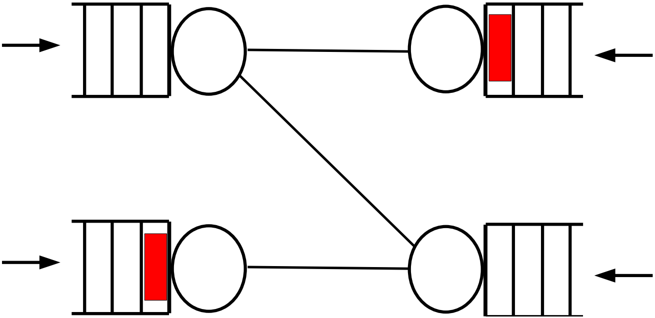
Bipartite Dynamic Matching Models

Assumption: random arrivals and in pairs



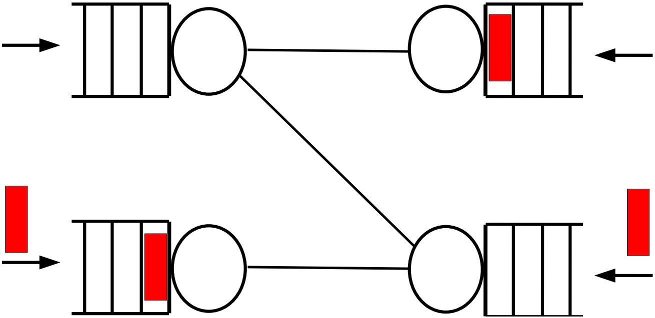
Bipartite Dynamic Matching Models

Assumption: random arrivals and in pairs



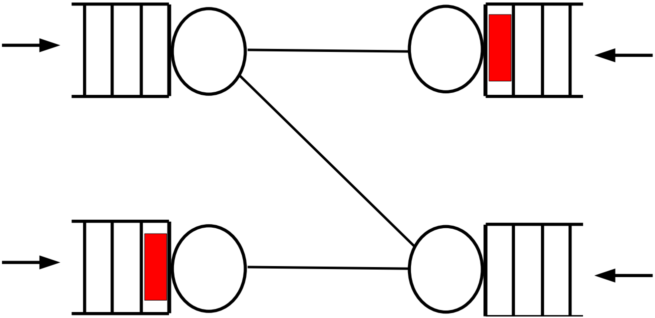
Bipartite Dynamic Matching Models

Assumption: random arrivals and in pairs



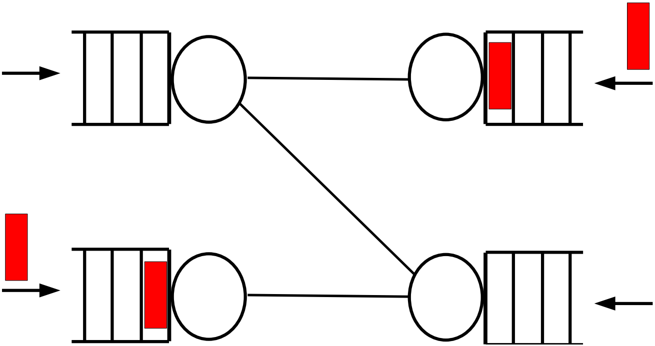
Bipartite Dynamic Matching Models

Assumption: random arrivals and in pairs



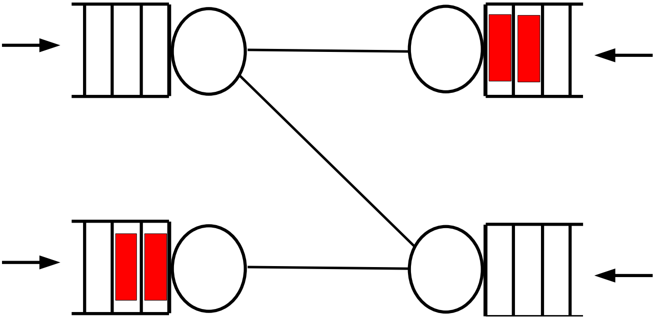
Bipartite Dynamic Matching Models

Assumption: random arrivals and in pairs



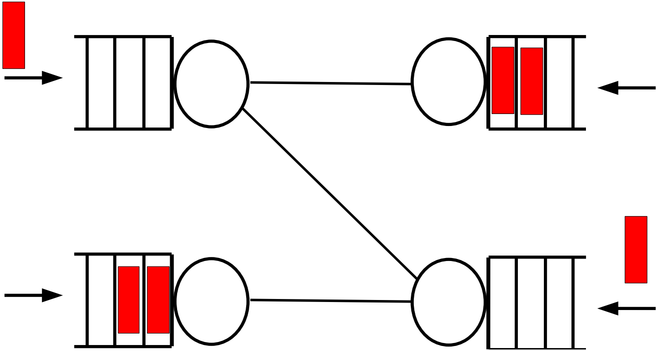
Bipartite Dynamic Matching Models

Assumption: random arrivals and in pairs



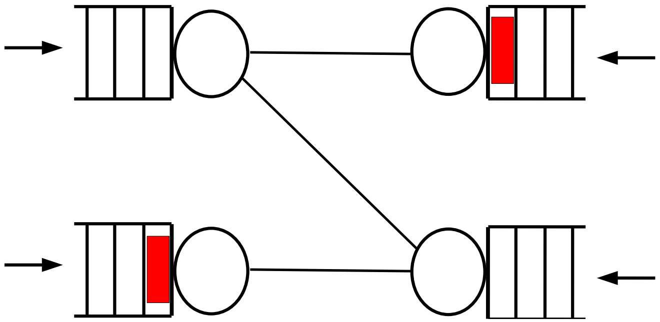
Bipartite Dynamic Matching Models

Assumption: random arrivals and in pairs



Bipartite Dynamic Matching Models

Assumption: random arrivals and in pairs



Theorem (Adan et al 2018)

The steady-state distribution of unmatched items for FCFM has a product form.

Bipartite Dynamic Matching Models

Theorem (Adan et al 2018)

The steady-state distribution of unmatched items for FCFM has a product form.

Open question

Does it exhibit the performance paradox in bipartite matching models?

Bipartite Dynamic Matching Models

Theorem (Adan et al 2018)

The steady-state distribution of unmatched items for FCFM has a product form.

Open question

Does it exhibit the performance paradox in bipartite matching models? For FCFM? For a given network topology?

Outline

Conclusions

Dynamic matching models and the properties of the derived Markov chains

Conclusions

Dynamic matching models and the properties of the derived Markov chains

- Applications in car sharing, online gaming, organ donation...

Conclusions

Dynamic matching models and the properties of the derived Markov chains

- Applications in car sharing, online gaming, organ donation...
- Performance paradox analysis: adding flexibility is not always beneficial

Conclusions

Dynamic matching models and the properties of the derived Markov chains

- Applications in car sharing, online gaming, organ donation...
- Performance paradox analysis: adding flexibility is not always beneficial
  - A necessary and sufficient condition for existence in the quasicomplete graph and any greedy matching policy

Conclusions

Dynamic matching models and the properties of the derived Markov chains

- Applications in car sharing, online gaming, organ donation...
- Performance paradox analysis: adding flexibility is not always beneficial
  - A necessary and sufficient condition for existence in the quasicomplete graph and any greedy matching policy
  - A sufficient condition for existence in arbitrary graphs and FCFM

Conclusions

Dynamic matching models and the properties of the derived Markov chains

- Applications in car sharing, online gaming, organ donation...
- Performance paradox analysis: adding flexibility is not always beneficial
  - A necessary and sufficient condition for existence in the quasicomplete graph and any greedy matching policy
  - A sufficient condition for existence in arbitrary graphs and FCFM
  - Extension using lumpability (aggregation of states of the Markov chain)

Conclusions

Dynamic matching models and the properties of the derived Markov chains

- Applications in car sharing, online gaming, organ donation...
- Performance paradox analysis: adding flexibility is not always beneficial
  - A necessary and sufficient condition for existence in the quasicomplete graph and any greedy matching policy
  - A sufficient condition for existence in arbitrary graphs and FCFM
  - Extension using lumpability (aggregation of states of the Markov chain)

Conclusions

Further research possibilities of performance paradox

Dynamic matching models with loops and bipartite dynamic matching models

Conclusions

Further research possibilities of performance paradox

Dynamic matching models with loops and bipartite dynamic matching models

Others:

Markov Decision Processes to analyze the optimal policy in matching models

T. Prieto. "Dispersion optima de poblaciones: un enfoque con procesos de decision markovianos". November 2023. Thematic Semester on Markov Processes and their Applications

Conclusions

Further research possibilities of performance paradox

Dynamic matching models with loops and bipartite dynamic matching models

Others:

Markov Decision Processes to analyze the optimal policy in matching models

T. Prieto. "Dispersion optima de poblaciones: un enfoque con procesos de decision markovianos". November 2023. Thematic Semester on Markov Processes and their Applications

Abandonments in dynamic matching models

Outline

References about the work of this presentation

- A. Cadas, J. Doncel, J.M. Fourneau, A. Busic .
  "Flexibility can hurt dynamic matching system perform".
  Proceedings of International Symposium on Computer Performance, Modeling, Measurements and Evaluation. Milano, Italy. November 2021.
- A. Cadas, J. Doncel, J.M. Fourneau, A. Busic .
  "Performance Paradox of Dynamic Matching Models under Greedy Policies". Submitted to Queueing Systems.
- A. Busic, A. Cadas, J. Doncel, J.M. Fourneau.
  "Product form solution for the steady-state distribution of a Markov chain associated with a general matching model with self-loops".
  Proceedings of 18th European Performance Engineering Workshop (EPEW 2022).

Further references about non-bipartite matching models

- J. Mairesse, and P. Moyal. 2016.
  "Stability of the stochastic matching model."
  Journal of Applied Probability 53, 4 (2016), 1064-1077.
- C. Comte.
  "Stochastic non-bipartite matching models and order-independent loss queues."
  Stochastic Models 38(1), 1-36 (2022)
- J. Begeot, I. Marcovici, P. Moyal, and Y. Rahme.
  "A general stochastic matching model on multigraphs"
  2020. ArXiv preprint: https://arxiv.org/abs/2011.05169.

Further references about bipartite matching models

- R. Caldentey, E. H. Kaplan, and G. Weiss.
  "FCFS infinite bipartite matching of servers and customers."
  Advances in Applied Probability, 41(3):695-730, 2009
- A. Busic, V. Gupta, and J. Mairesse.
  "Stability of the bipartite matching model."
  Advances in Applied Probability, 45(2):351-378, 2013.
- I. Adan, A. Busic, J. Mairesse, and G. Weiss.
  "Reversibility and further properties of FCFS infinite bipartite matching."
  Mathematics of Operations Research, 43(2):598-621, 2018.

Muchas gracias

Muchas gracias por su atención