Performance Paradox of Dynamic Bipartite Matching Models

Josu Doncel
University of the Basque Country, UPV/EHU.

Joint work with I. Iriondo (UPV/EHU)

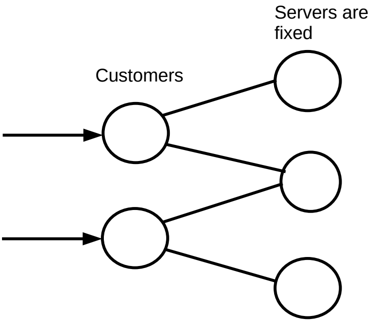
NETGCOOP 2024. Lille, France.
October 9, 2024

Outline

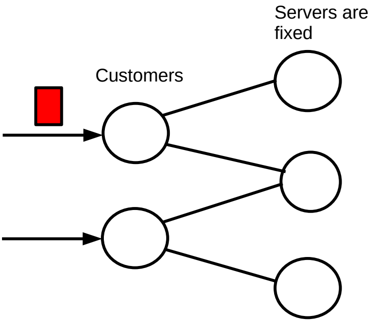
Outline

Introduction

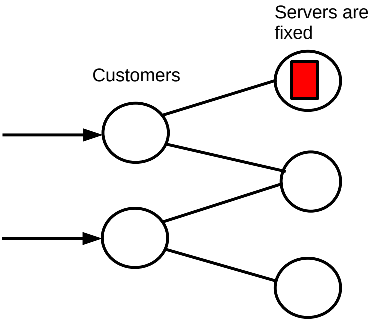
Multiskilled systems (call centers)



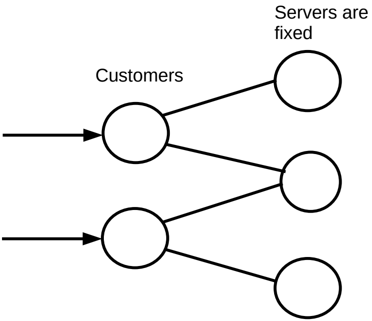
Multiskilled systems (call centers)



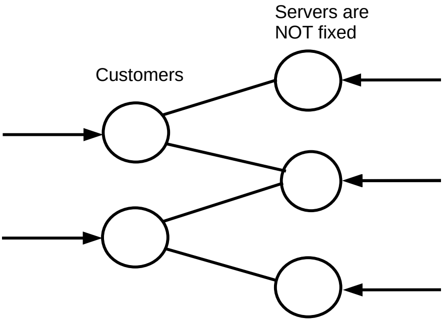
Multiskilled systems (call centers)



Multiskilled systems (call centers)

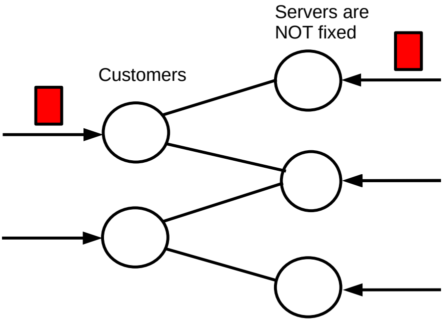


Matching models



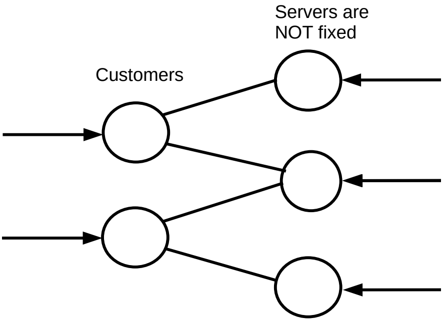
Introduction

Matching models



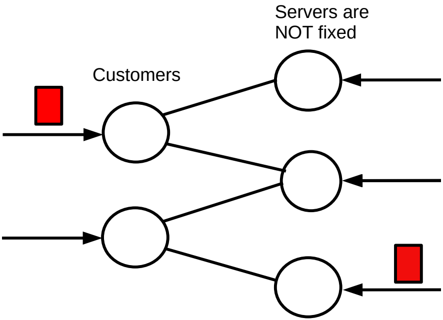
Matched customers and servers leave immediately

Matching models



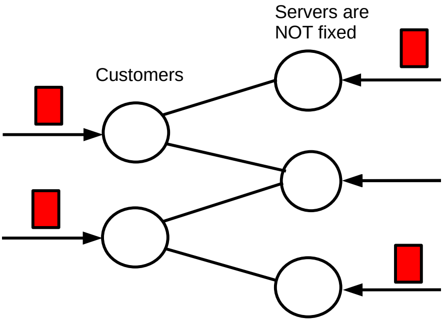
Matched customers and servers leave immediately

Matching models



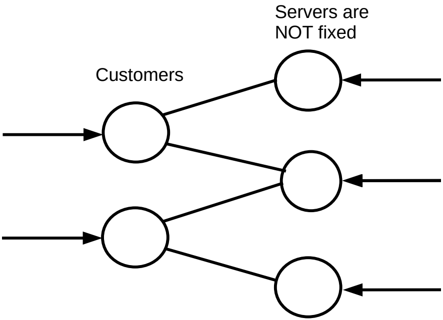
Unmatched customers and servers are stored in queues

Matching models



Unmatched customers and servers are stored in queues

Matching models



Unmatched customers and servers are stored in queues

Bipartite Matching Models

Defined by:

- Compatibility graph $\mathcal{G}$.

- Distribution of arrivals of customers and servers

- Matching policy ψ

Bipartite Matching Models

Defined by:

- Compatibility graph $\mathcal{G}$.

  Bipartite graph: defines the compatibilities of customers and servers
  $\mathcal{G} = (\mathcal{C} \cup \mathcal{S}, \mathcal{E})$, where $\mathcal{E} \subset \mathcal{C} \times \mathcal{S}$
  Example: $\mathcal{C} = \{c_1, c_2\}$, $\mathcal{S} = \{s_1, s_2, s_3\}$ and
  $\mathcal{E} = \{(c_1, s_1), (c_1, s_2), (c_2, s_2), (c_2, s_3)\}$.
- Distribution of arrivals of customers and servers
- Matching policy ψ

Bipartite Matching Models

Defined by:

- Compatibility graph $\mathcal{G}$.

  Bipartite graph: defines the compatibilities of customers and servers
  $\mathcal{G} = (\mathcal{C} \cup \mathcal{S}, \mathcal{E})$, where $\mathcal{E} \subset \mathcal{C} \times \mathcal{S}$
  Example: $\mathcal{C} = \{c_1, c_2\}$, $\mathcal{S} = \{s_1, s_2, s_3\}$ and
  $\mathcal{E} = \{(c_1, s_1), (c_1, s_2), (c_2, s_2), (c_2, s_3)\}$.

- Distribution of arrivals of customers and servers

  $\vec{\alpha}$ for customers and $\vec{\beta}$ for servers
  Example: $\vec{\alpha} = (\alpha_1, \alpha_2)$ and $\vec{\beta} = (\beta_1, \beta_2, \beta_3)$

- Matching policy ψ

Bipartite Matching Models

Defined by:

- Compatibility graph $\mathcal{G}$.

  Bipartite graph: defines the compatibilities of customers and servers
  $\mathcal{G} = (\mathcal{C} \cup \mathcal{S}, \mathcal{E})$, where $\mathcal{E} \subset \mathcal{C} \times \mathcal{S}$
  Example: $\mathcal{C} = \{c_1, c_2\}$, $\mathcal{S} = \{s_1, s_2, s_3\}$ and
  $\mathcal{E} = \{(c_1, s_1), (c_1, s_2), (c_2, s_2), (c_2, s_3)\}$.

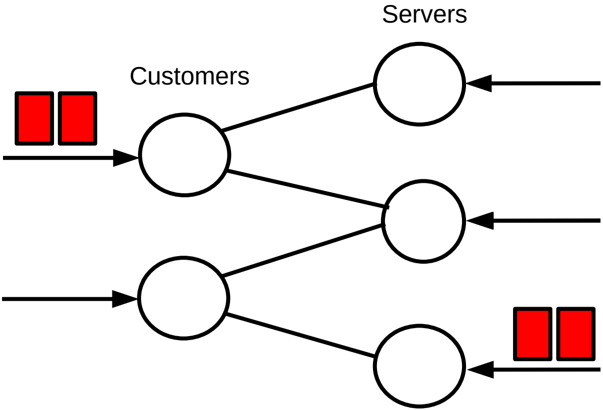
- Distribution of arrivals of customers and servers

  $\vec{\alpha}$ for customers and $\vec{\beta}$ for servers
  Example: $\vec{\alpha} = (\alpha_1, \alpha_2)$ and $\vec{\beta} = (\beta_1, \beta_2, \beta_3)$

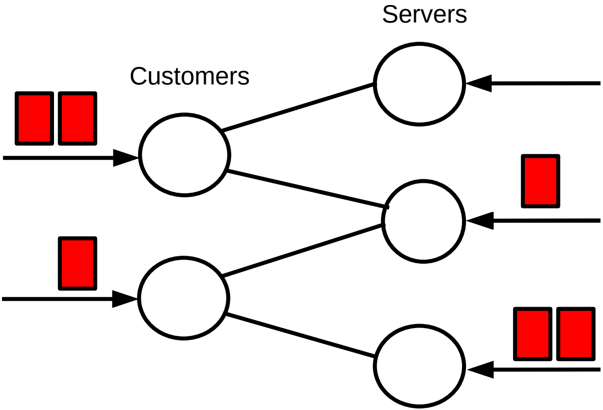
- Matching policy ψ

  How compatible customers and servers are matched?
  - In order of arrivals
  - Prioritize the class with the longest/shortest number of elements...

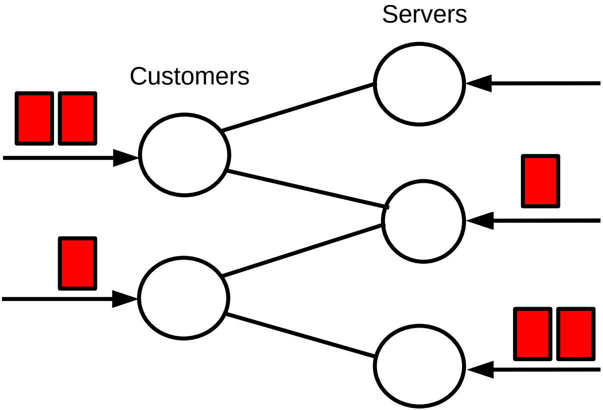
Matching Policy: Example



Matching Policy: Example

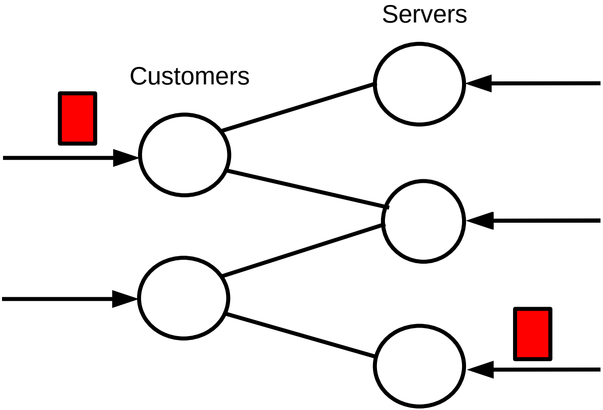


Matching Policy: Example



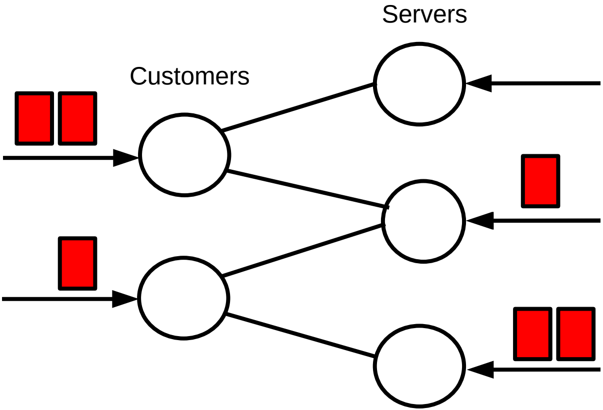
First-Come-First-Matched (FCFM)

Matching Policy: Example



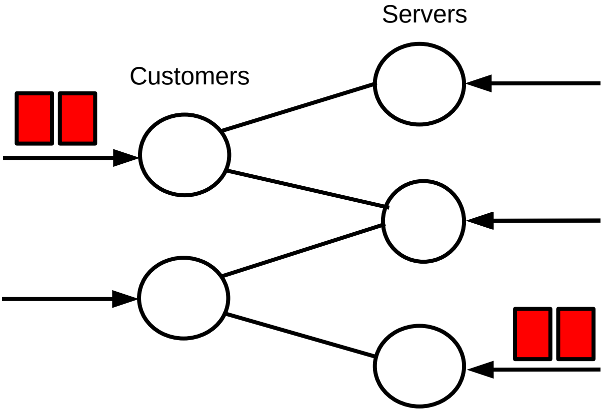
First-Come-First-Matched (FCFM)

Matching Policy: Example



Last-Come-First-Matched

Matching Policy: Example



Last-Come-First-Matched

Bipartite Matching Models

Defined by:

- Compatibility graph $\mathcal{G}$.

  Bipartite graph: defines the compatibilities of customers and servers
  $\mathcal{G} = (\mathcal{C} \cup \mathcal{S}, \mathcal{E})$, where $\mathcal{E} \subset \mathcal{C} \times \mathcal{S}$
  Example: $\mathcal{C} = \{c_1, c_2\}$, $\mathcal{S} = \{s_1, s_2, s_3\}$ and
  $\mathcal{E} = \{(c_1, s_1), (c_1, s_2), (c_2, s_2), (c_2, s_3)\}$.

- Distribution of arrivals of customers and servers (independent)
  $\vec{\alpha}$ for customers and $\vec{\beta}$ for servers

  Example: $\vec{\alpha} = (\alpha_1, \alpha_2)$ and $\vec{\beta} = (\beta_1, \beta_2, \beta_3)$
  c_1 and s_3 arrive with probability $\alpha_1\beta_3$.

- Matching policy ψ

  How compatible customers and servers are matched?
  - In order of arrivals
  - Prioritize the class with the longest/shortest number of elements...

When $\mathcal{G}$, $(\vec{\alpha}, \vec{\beta})$ and ψ are fixed

The number of unmatched items is a Discrete Time Markov Chain.

Stability of Markov Chain

Busic, Gupta and Mairesse, 2013

For FCFM, the Markov chain is stable iff $\forall C \subset \mathcal{C}\ \forall S \subset \mathcal{S}$

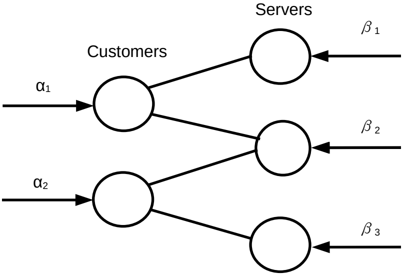
$$\sum_{c_i \in C} \alpha_i < \sum_{s_i \in S(C)} \beta_i \text{ and } \sum_{s_i \in S} \beta_i < \sum_{c_i \in C(S)} \alpha_i,$$

Stability of Markov Chain

Busic, Gupta and Mairesse, 2013

For FCFM, the Markov chain is stable iff $\forall C \subset \mathcal{C}\ \forall S \subset \mathcal{S}$

$$\sum_{c_i \in C} \alpha_i < \sum_{s_i \in S(C)} \beta_i \text{ and } \sum_{s_i \in S} \beta_i < \sum_{c_i \in C(S)} \alpha_i,$$

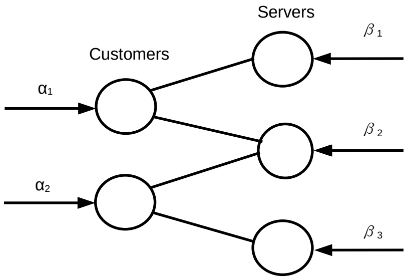


Stability of Markov Chain

Busic, Gupta and Mairesse, 2013

For FCFM, the Markov chain is stable iff $\forall C \subset \mathcal{C}\ \forall S \subset \mathcal{S}$

$$\sum_{c_i \in C} \alpha_i < \sum_{s_i \in S(C)} \beta_i \text{ and } \sum_{s_i \in S} \beta_i < \sum_{c_i \in C(S)} \alpha_i,$$

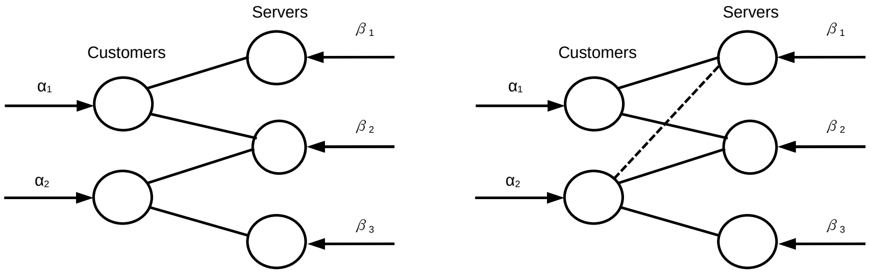


- $\alpha_1 < \beta_1 + \beta_2$
- $\alpha_2 < \beta_2 + \beta_3$
- $\beta_1 < \alpha_1$
- $\beta_2 < \alpha_1 + \alpha_2 = 1$
- $\beta_3 < \alpha_3$
- $\beta_1 + \beta_2 < \alpha_1 + \alpha_2 + \alpha_3 = 1$
- $\beta_2 + \beta_3 < \alpha_1 + \alpha_2 + \alpha_3 = 1$

Outline

FCFM matching policy

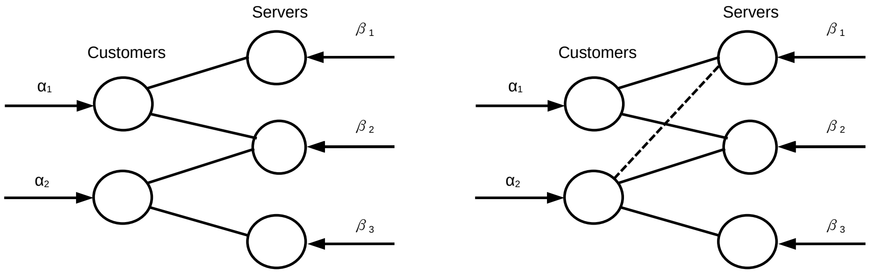
Objective: study the influence of adding an edge to the compatibility graph



Model Description

FCFM matching policy

Objective: study the influence of adding an edge to the compatibility graph



Definition: Performance paradox

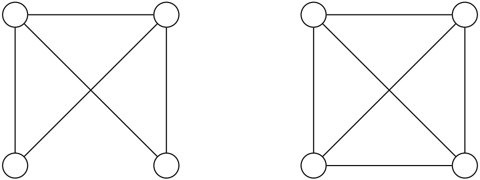
Adding an edge the mean number of unmatched customers and servers increases (analogue of Braess paradox).

Performance Paradox

Definition: Performance paradox

Adding an edge the mean number of unmatched customers and servers increases (analogue of Braess paradox).

Previous works of performance paradox in matching models focus on a general matching models

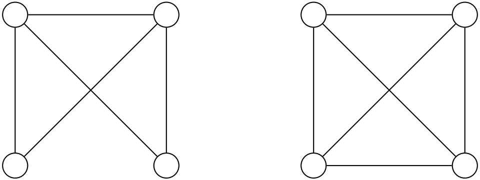


Performance Paradox

Definition: Performance paradox

Adding an edge the mean number of unmatched customers and servers increases (analogue of Braess paradox).

Previous works of performance paradox in matching models focus on a general matching models



Necessary and sufficient conditions on the arrivals such that performance paradox exists in a quasicomplete graph under FCFM (Cadas et al, 2021) and under greedy policies (Busic et al, 2024)

Outline

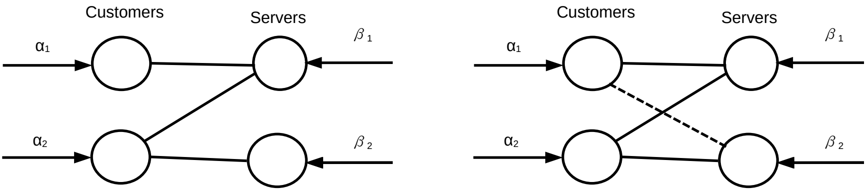
Performance Paradox Analysis

Objective: study the existence of the performance paradox in bipartite matching models

Performance Paradox Analysis

Objective: study the existence of the performance paradox in bipartite matching models

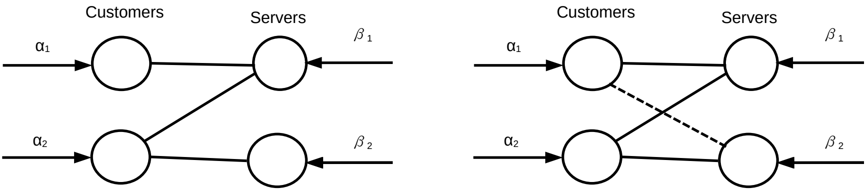
First (and simplest) approach:



Performance Paradox Analysis

Objective: study the existence of the performance paradox in bipartite matching models

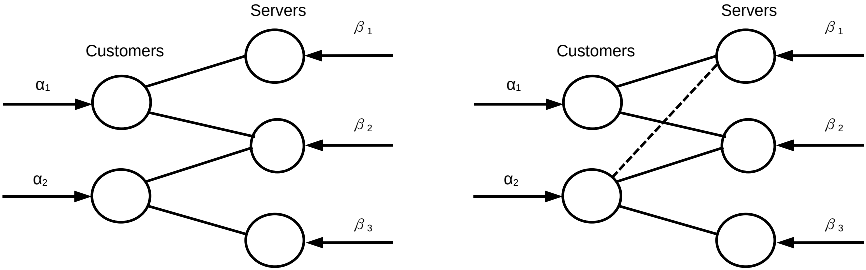
First (and simplest) approach:



The performance paradox does NOT exist for this case

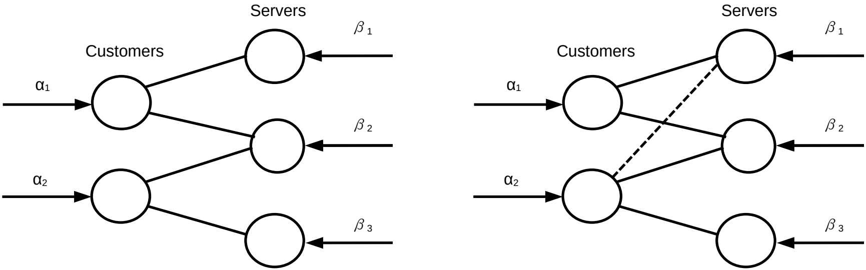
Performance Paradox Analysis

Second approach:



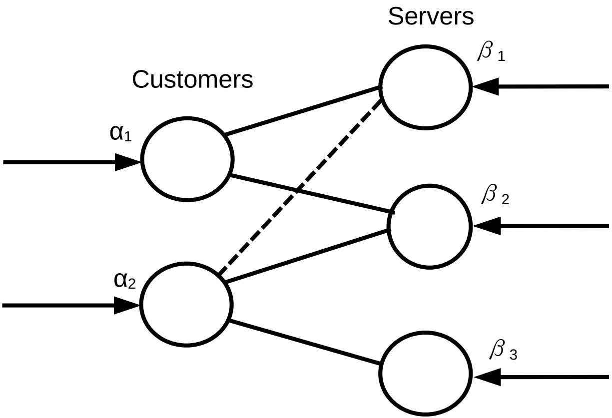
Performance Paradox Analysis

Second approach:

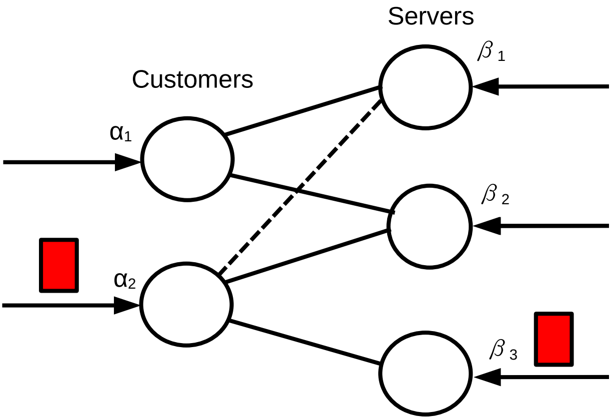


Under FCFM, the Markov chains of the unmatched elements are not difficult to analyze

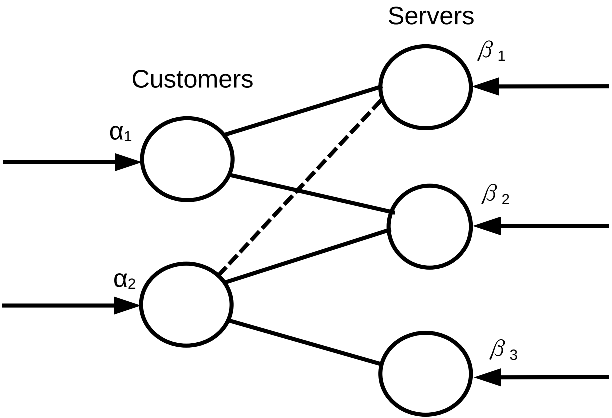
Performance Paradox Analysis



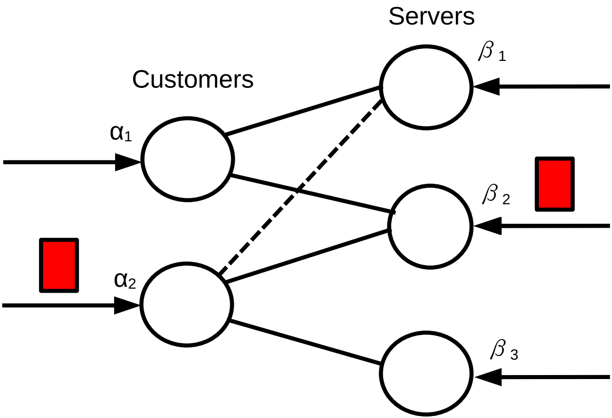
Performance Paradox Analysis



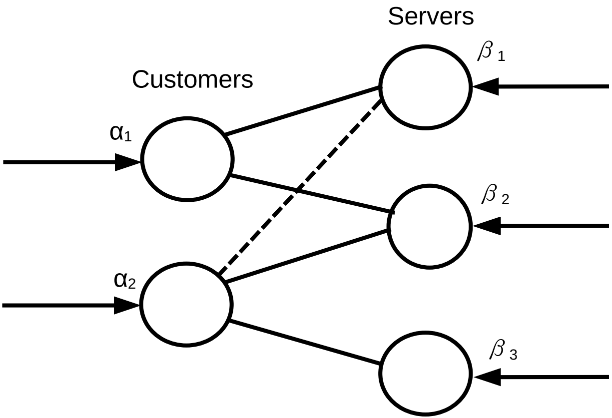
Performance Paradox Analysis



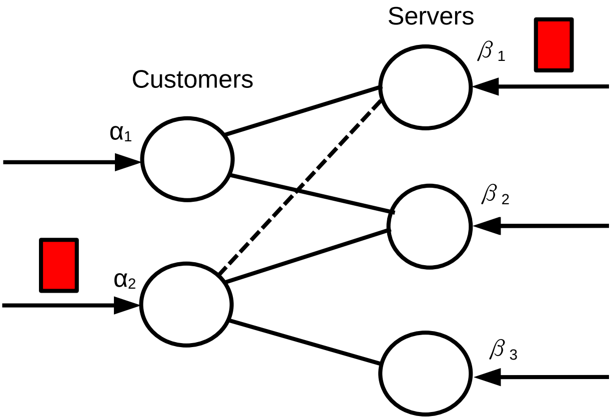
Performance Paradox Analysis



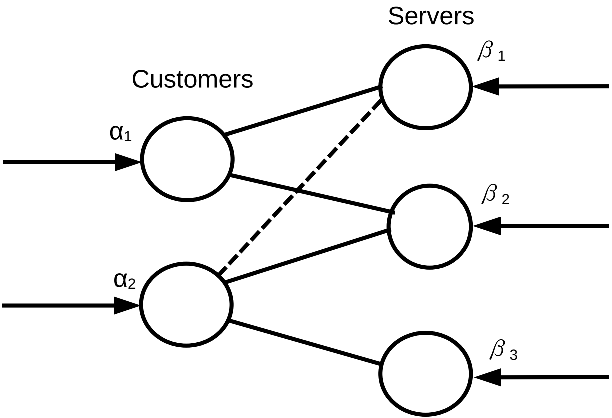
Performance Paradox Analysis



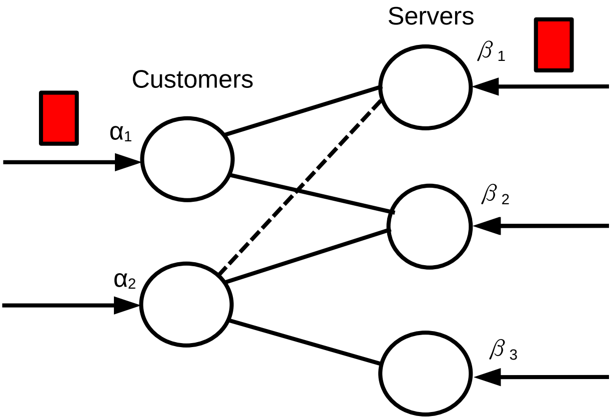
Performance Paradox Analysis



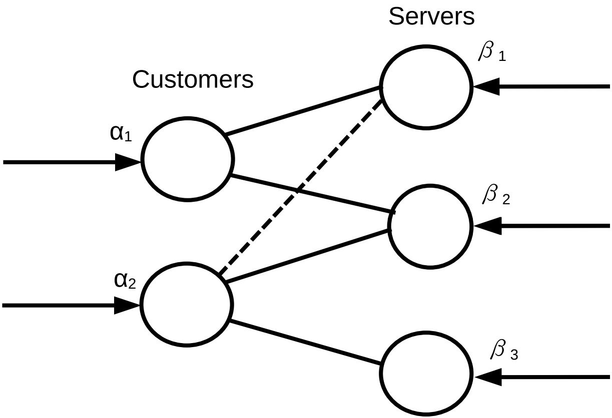
Performance Paradox Analysis



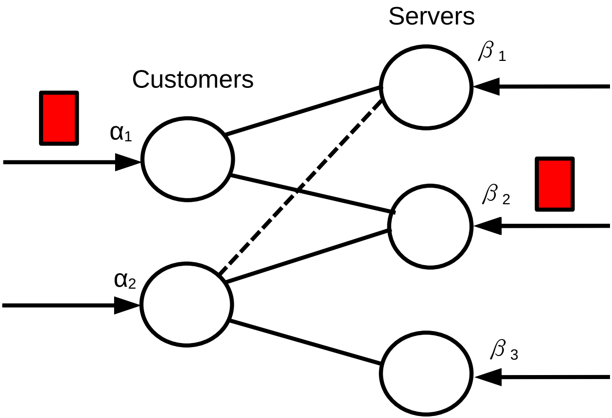
Performance Paradox Analysis



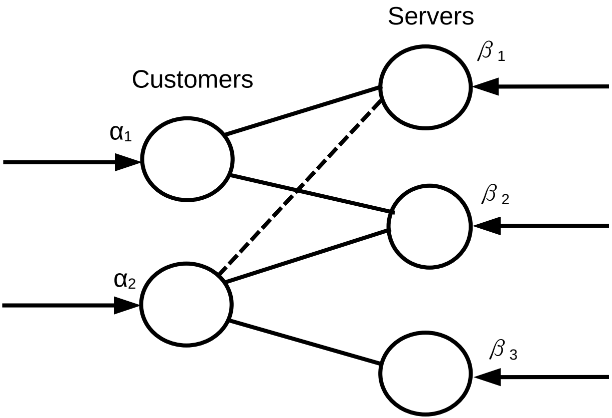
Performance Paradox Analysis



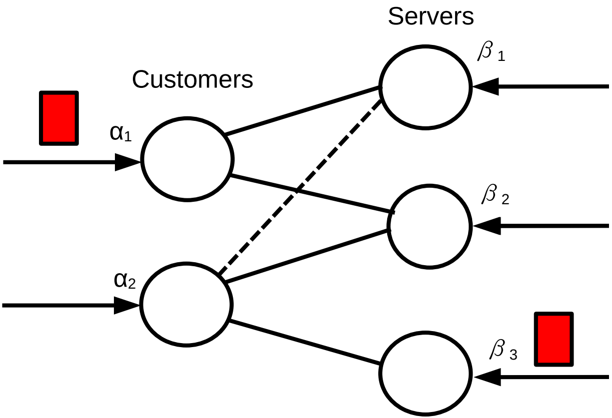
Performance Paradox Analysis



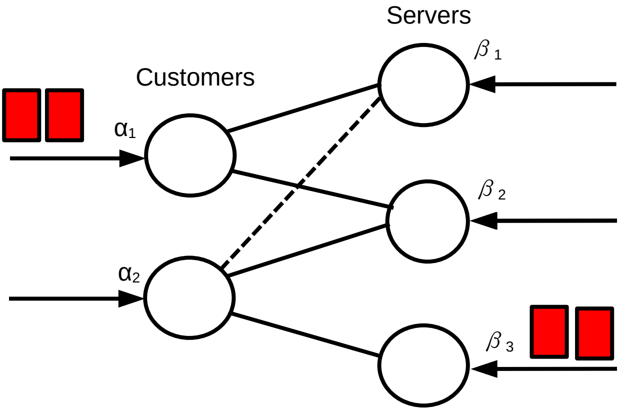
Performance Paradox Analysis



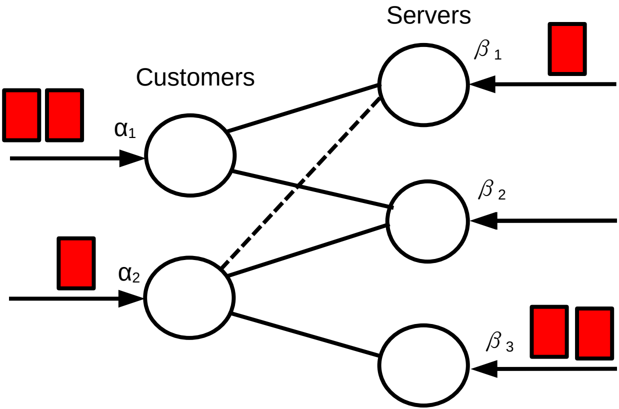
Performance Paradox Analysis



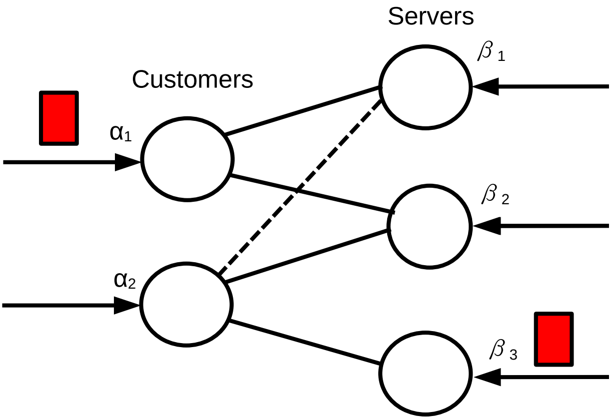
Performance Paradox Analysis



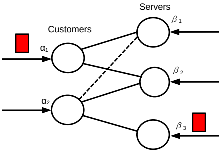
Performance Paradox Analysis



Performance Paradox Analysis

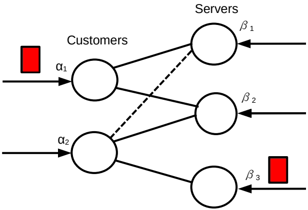


Performance Paradox Analysis



The couple (c_1, s_3) is the only possible unmatched pair of customer and server.

Performance Paradox Analysis

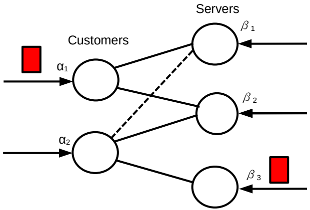


The couple (c_1, s_3) is the only possible unmatched pair of customer and server.

The Markov chain is a birth-death process:

Birth probability: $\alpha_1 \beta_3$
Death probability: $\alpha_2(1 - \beta_3)$

Performance Paradox Analysis



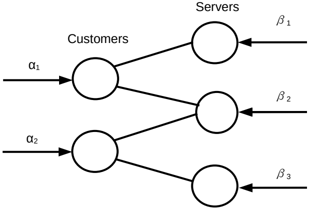
The couple (c_1, s_3) is the only possible unmatched pair of customer and server.

The Markov chain is a birth-death process:

Birth probability: $\alpha_1\beta_3$
Death probability: $\alpha_2(1-\beta_3)$

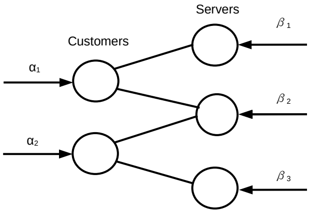
Let $\rho = \frac{\alpha_1\beta_3}{\alpha_2(1-\beta_3)}$. If $\rho < 1$, the mean number of unmatched elements is $2\frac{\rho}{1-\rho}$

Performance Paradox Analysis



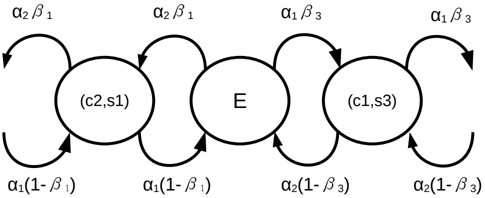
The couples (c_1, s_3) and (c_2, s_1) are the only possible unmatched pair of customer and server.

Performance Paradox Analysis

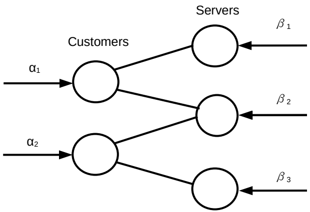


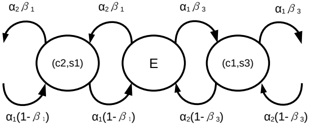
The couples (c_1, s_3) and (c_2, s_1) are the only possible unmatched pair of customer and server.

The Markov chain of unmatched couples is formed by two birth-death process connected in the empty state.



Performance Paradox Analysis





Let $\rho_1 = \frac{\alpha_1 \beta_3}{\alpha_2(1-\beta_3)}$ and $\rho_2 = \frac{\alpha_2 \beta_1}{\alpha_1(1-\beta_1)}$. If $\rho_1 < 1$ and $\rho_2 < 1$, the mean number of unmatched elements is

$$\frac{2(1-\rho_1)(1-\rho_2)}{1-\rho_1\rho_2}\left(\frac{\rho_1^2}{(1-\rho_1)^2} + \frac{\rho_2^2}{(1-\rho_2)^2}\right)$$

Performance Paradox Analysis

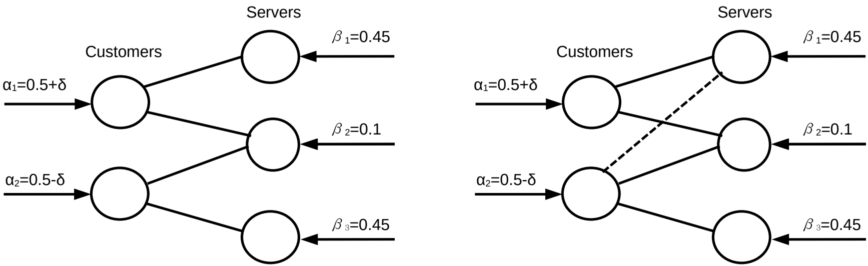
Objective: study the existence of the performance paradox

$$\frac{2(1-\rho_1)(1-\rho_2)}{1-\rho_1\rho_2}\left(\frac{\rho_1^2}{(1-\rho_1)^2}+\frac{\rho_2^2}{(1-\rho_2)^2}\right) \overset{?}{>} \frac{2\rho_1}{1-\rho_1}.$$

Performance Paradox Analysis

Objective: study the existence of the performance paradox

$$\frac{2(1-\rho_1)(1-\rho_2)}{1-\rho_1\rho_2}\left(\frac{\rho_1^2}{(1-\rho_1)^2}+\frac{\rho_2^2}{(1-\rho_2)^2}\right)\overset{?}{>}\frac{2\rho_1}{1-\rho_1}.$$

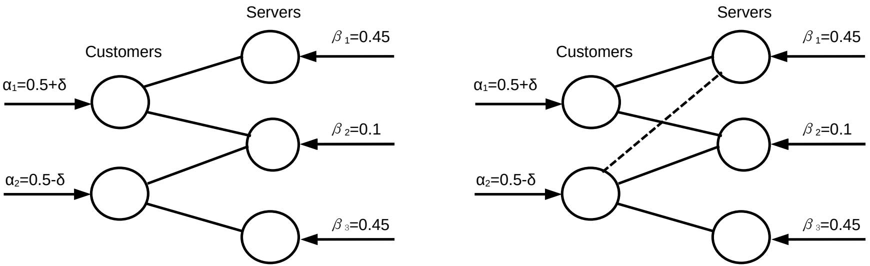


where $\delta \in (0, 0{,}05)$.

Performance Paradox Analysis

Objective: study the existence of the performance paradox

$$\frac{2(1-\rho_1)(1-\rho_2)}{1-\rho_1\rho_2}\left(\frac{\rho_1^2}{(1-\rho_1)^2}+\frac{\rho_2^2}{(1-\rho_2)^2}\right) \overset{?}{>} \frac{2\rho_1}{1-\rho_1}.$$



where $\delta \in (0, 0{,}05)$.

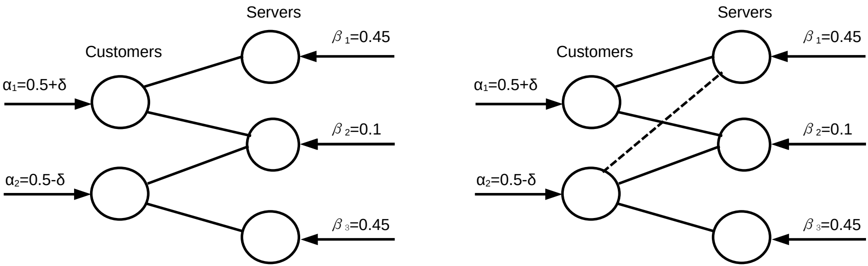
Stability

We check that $\beta_1 < \alpha_1$ and $\beta_3 < \alpha_2$ for all $\delta \in (0, 0{,}05)$

Performance Paradox Analysis

Objective: study the existence of the performance paradox

$$\frac{2(1-\rho_1)(1-\rho_2)}{1-\rho_1\rho_2}\left(\frac{\rho_1^2}{(1-\rho_1)^2}+\frac{\rho_2^2}{(1-\rho_2)^2}\right)\overset{?}{>}\frac{2\rho_1}{1-\rho_1}.$$



where $\delta \in (0, 0{,}05)$.

Stability

We check that $\beta_1 < \alpha_1$ and $\beta_3 < \alpha_2$ for all $\delta \in (0, 0{,}05)$ and also that $\rho < 1, \rho_1 < 1$ and $\rho_2 < 1$.

Performance Paradox Analysis

Objective: study the existence of the performance paradox

$$\frac{99}{10}\frac{1+400\delta^2}{1-400\delta^2} \stackrel{?}{>} \frac{9(1+2\delta)}{1-20\delta}.$$

Performance Paradox Analysis

Objective: study the existence of the performance paradox

$$\frac{99}{10}\frac{1+400\delta^2}{1-400\delta^2} \overset{?}{>} \frac{9(1+2\delta)}{1-20\delta}.$$

Theorem

The performance paradox exists iff $\delta \in (0{,}005, 0{,}05)$

Performance Paradox Analysis

Objective: study the existence of the performance paradox

$$\frac{99}{10}\frac{1+400\delta^2}{1-400\delta^2} \stackrel{?}{>} \frac{9(1+2\delta)}{1-20\delta}.$$

Theorem

The performance paradox exists iff $\delta \in (0{,}005, 0{,}05)$

Conclusion

As in previous work, the performance paradox is given when the stability condition ($\beta_3 < \alpha_2$) is marginally satisfied ($\delta \to 0{,}05$)

Performance Paradox Analysis

Objective: study the existence of the performance paradox

$$\frac{99}{10}\frac{1+400\delta^2}{1-400\delta^2} \overset{?}{>} \frac{9(1+2\delta)}{1-20\delta}.$$

Theorem

The performance paradox exists iff $\delta \in (0{,}005, 0{,}05)$

Conclusion

As in previous work, the performance paradox is given when the stability condition ($\beta_3 < \alpha_2$) is marginally satisfied ($\delta \to 0{,}05$)

$\Rightarrow$ When $\delta \to 0{,}05$, the mean number of unmatched elements tends to infinity

Performance Paradox Analysis

Objective: study the existence of the performance paradox

$$\frac{99}{10}\frac{1+400\delta^2}{1-400\delta^2} \overset{?}{>} \frac{9(1+2\delta)}{1-20\delta}.$$

Theorem

The performance paradox exists iff $\delta \in (0{,}005, 0{,}05)$

Conclusion

As in previous work, the performance paradox is given when the stability condition ($\beta_3 < \alpha_2$) is marginally satisfied ($\delta \to 0{,}05$)

$\Rightarrow$ When $\delta \to 0{,}05$, the mean number of unmatched elements tends to infinity

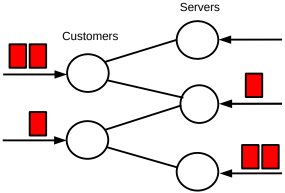
Proposition

When $\delta \to 0{,}05$, the difference due to the performance paradox tends to infinity.

Extensions

Matching policy: FCFM

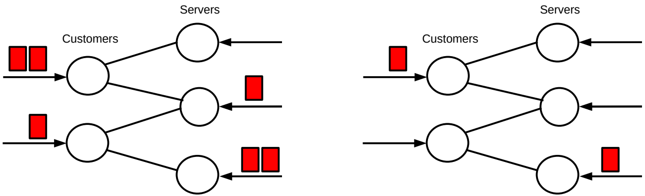
Other matching policies lead to the same Markov chains: MaxWeight



Extensions

Matching policy: FCFM

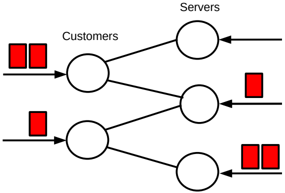
Other matching policies lead to the same Markov chains: MaxWeight

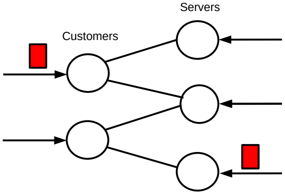


Extensions

Matching policy: FCFM

Other matching policies lead to the same Markov chains: MaxWeight





The W-shaped compatibility graph

The same results are obtained for any compatibility graph which is complete minus two edges

Outline

Conclusions and Future Work

Adding an edge in bipartite matching models might hurt the performance of the system

Conclusions and Future Work

Adding an edge in bipartite matching models might hurt the performance of the system

Questions for future work

- Is the existence of a performance paradox related to this particular compatibility graph?
- Can we provide sufficient conditions on the existence of a performance paradox in arbitrary compatibility graphs (and FCFM)?
- Does the performance paradox exist in multigraphs? And when we consider self-loops?

References

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.

References

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.

Performance paradox in general matching models:

- A. Cadas, J. Doncel, J.-M. Fourneau, and A. Busic. Flexibility can hurt dynamic matching system performance. ACM SIGMETRICS Performance Evaluation Review, 49(3):37-42, 2022.
- A. Busic, A. Cadas, J. Doncel, and J.-M. Fourneau. Performance paradox of dynamic matching models under greedy policies. Queueing Systems 107, 257–293 (2024).

Thanks and questions?

THANKS FOR YOUR ATTENTION

QUESTIONS?