Online Matching with Delays and Stochastic Arrival Times

Mathieu Mari, **Michał Pawłowski**, Runtian Ren, Piotr Sankowski

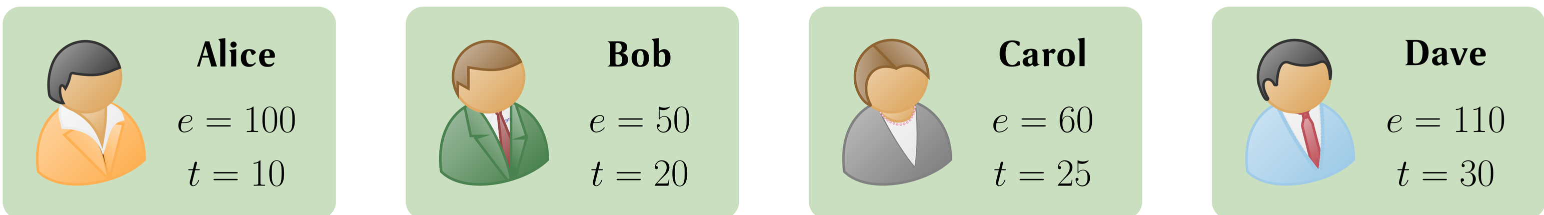
Keywords: **online algorithms, matchings, stochastic model, Poisson arrivals**

Motivation

Imagine an **online chess gaming platform** that wants to **maximize** the overall **satisfaction** of the game for its users. Two main factors contribute to this — for each pair of players, we need to look at

- the **experience difference** (the smaller, the better)
- the **waiting time** to start a game (the shorter, the better).

Problem: each of these parameters needs to be minimized at the expense of the other. For example, finding an opponent with a similar level of experience results in a longer waiting time. Thus, the question arises of **how to balance these two parameters**.



In the figure above, e and t represent the experience level and arrival time, respectively. If we decided to match the players immediatelly (Alice with Bob, Carol with Dave), we would generate a total **cost** (e and t differences) of 115. However, if we **postponed** our decisions and matched Bob with Carol (and Alice with Dave), we would generate a cost of 45.

Problem statement

Min-Cost Perfect Matching with Delays (MPMD) [3]

- m **requests**, each representing an independent player/agent
- they arrive at arbitrary times in a **metric space** $\mathcal{M} = (\mathcal{X}, d)$
- the metric is equipped with a **distance function** d
- m is an **even integer**, and $\mathcal{X}$ is a set of n points in $\mathcal{M}$
- request r: **location** $\ell(r) \in \mathcal{X}$, **arrival time** $t(r) \in \mathbb{R}^+$
- matching r and r' at time $t \geq \max\{t(r), t(r')\}$ **inquiries**: **connection cost** $d(\ell(r), \ell(r'))$ and **delay cost** $(t - t(r)) + (t - t(r'))$

Target: **minimize the total cost** produced by the online algorithm for matching all the requests into pairs

Previous results

Approach: match two requests whenever their **total delay cost exceeds the connection cost** (balanced greedy, ALG).

Input: a sequence σ of requests.
Output: a perfect matching of the requests.
for *any time* t **do**
  if *there exist pending requests* r, r' *such that* $(t - t(r)) + (t - t(r')) \geq d(\ell(r), \ell(r'))$ **then**
    match them into a pair with ties broken arbitrarily.

- greedy in **adversarial model** → $O(m^{\log(1.5+\varepsilon)})$-competitive [2]
- **any algorithm** in this model → $\Omega(\log n / \log\log n)$-competitive [1]

Beyond worst-case

Assumption: the **waiting time** between any two consecutive requests arriving at any metrical point x, **follows an exponential distribution** $\mathrm{Exp}(\lambda_x)$ with parameter $\lambda_x \geq 0$

- the platform needs to **minimize the expected cost** it produces when dealing with a random input sequence of m requests
- to evaluate the performance of any algorithm A on stochastic input, we use the **ratio-of-expectations** → the ratio of the expected costs of A and the optimal offline solution (OPT)

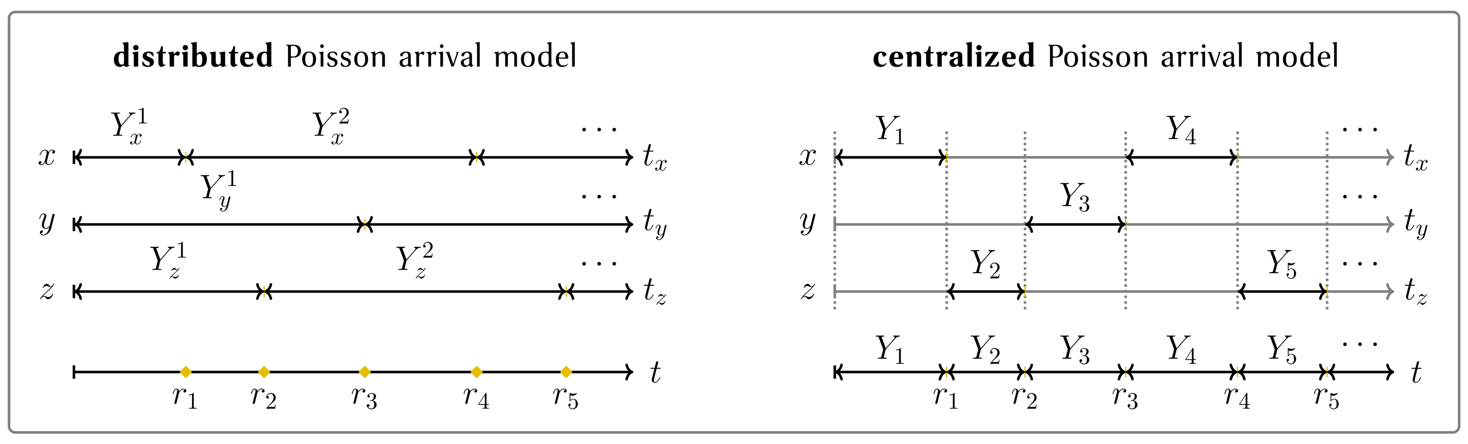
Our main result

Theorem 1 *For MPMD in the Poisson arrival model, the balanced greedy algorithm achieves a **constant** ratio-of-expectations.*

Arrival models

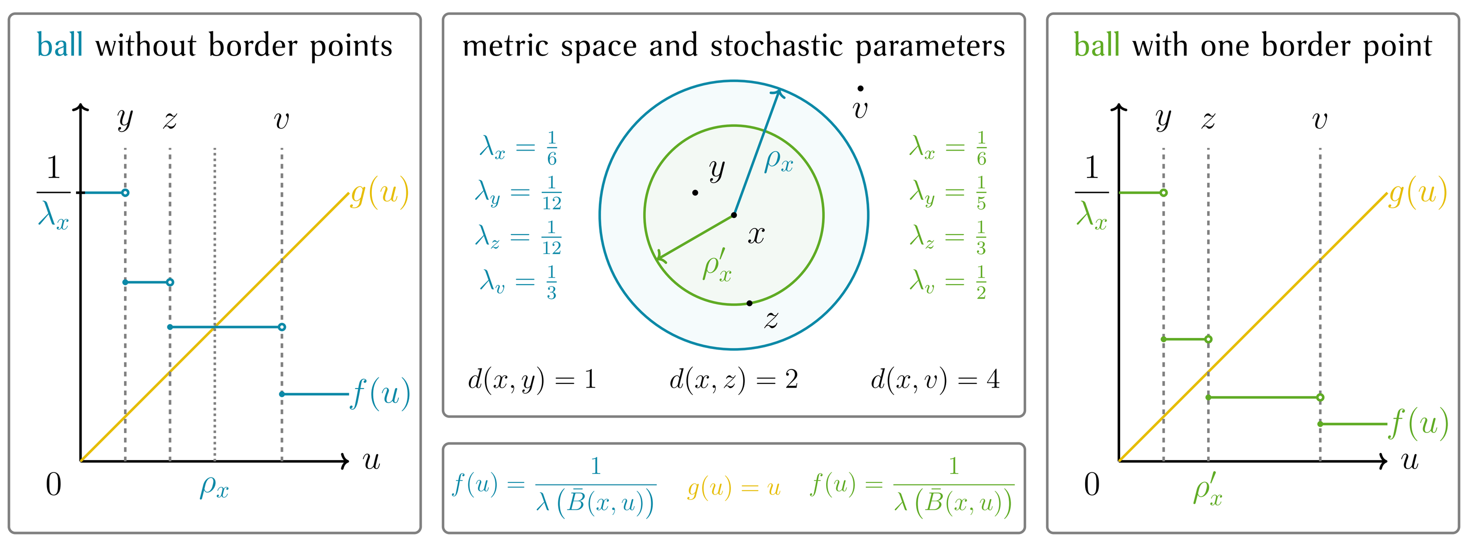
We use **two equivalent models** of stochastic arrivals:

- look at **each metric point independently**
- use a **centralized approach** → a **single process specifies request arrivals** in the whole metric (one variable determining waiting time)

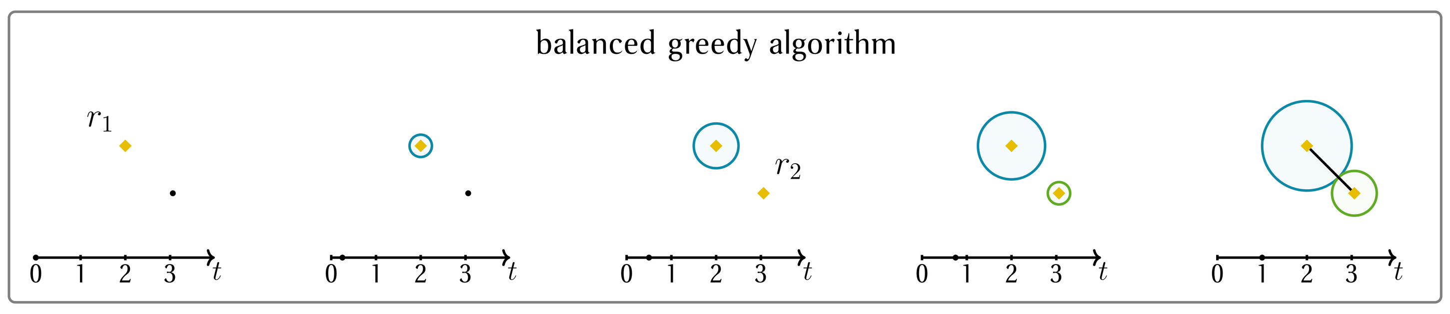


Balanced radius notion

To **estimate the costs** of our greedy algorithm and the optimal one, we introduce a **balanced radius**. For any given point x, we compute a radius ρ_x such that the **expected waiting time** between two consecutive arrivals located in the ball of radius ρ_x centred at x is **equal** to ρ_x.



With the use of both arrival models and the balanced radius, we can show that the expected cost of serving a request arriving at x is **upper-bounded** by $c_1 \rho_x$ for **ALG** and **lower-bounded** by $c_2 \rho_x$ for **OPT**, where c_1 and c_2 are some specific constants.



Generalizations and future directions

It is worth noting that the **greedy algorithm** doesn't require upfront knowledge of the metric space and stochastic parameters (it **is non-clairvoyant**). In case we **have this knowledge**, we provide **another algorithm** that achieves a **two times better competitiveness ratio**.

In our work, we also extend the results to the case where:

- the **delay cost** corresponds to an **arbitrary positive and non-decreasing function** of the waiting time
- the platform can **pay a penalty cost to clear some requests**

Future directions would be to study similar online problems, such as k-way MPMD, **multi-level aggregation**, **online service** or **facility location** with delays and stochastic arrival times.

Bibliography

[1] Itai Ashlagi, Yossi Azar, Moses Charikar, Ashish Chiplunkar, Ofir Geri, Haim Kaplan, Rahul Makhijani, Yuyi Wang, and Roger Wattenhofer. Min-cost bipartite perfect matching with delays. In *Proc. APPROX*, pages 1:1–1:20, 2017.

[2] Yossi Azar and Amit Jacob-Fanani. Deterministic min-cost matching with delays. *Theory of Computing Systems*, 64(4):572–592, 2020.

[3] Yuval Emek, Shay Kutten, and Roger Wattenhofer. Online matching: haste makes waste! In *Proc. STOC*, pages 333–344, 2016.

Contact: Michał Pawłowski
michal.pawlowski@ideas-ncbr.pl

University of Warsaw
IDEAS NCBR