



HAL
open science

Online Stochastic Matching: A Polytope Perspective

Céline Comte, Fabien Mathieu, Sushil Mahavir Varma, Ana Bušić

► **To cite this version:**

Céline Comte, Fabien Mathieu, Sushil Mahavir Varma, Ana Bušić. Online Stochastic Matching: A Polytope Perspective. 2025. <hal-03502084v6>

HAL Id: hal-03502084

<https://hal.science/hal-03502084v6>

Preprint submitted on 11 Nov 2025

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons CC BY 4.0 - Attribution - International License

Online Stochastic Matching: A Polytope Perspective

Céline Comte¹ ✉ 🏠 

Eindhoven University of Technology, Eindhoven, The Netherlands
CNRS and LAAS, Toulouse, France

Fabien Mathieu ✉ 🏠 

Swapcard, Paris, France
Sorbonne Université, CNRS, LIP6, Paris, France

Sushil Mahavir Varma ✉ 🏠 

Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, US

Ana Bušić ✉ 🏠 

Inria, Paris, France
DI ENS, CNRS, PSL University, Paris, France

Abstract

Stochastic dynamic matching problems have recently gained attention in the stochastic-modeling community due to their diverse applications, such as supply-chain management and kidney exchange programs. In this paper, we study a matching problem where items of different classes arrive according to independent Poisson processes. Unmatched items are stored in a queue, and compatibility between items is represented by a simple graph, where items can be matched if their classes are connected. We analyze matching policies in terms of stability, delay, and long-term matching rate optimization. Our approach relies on the conservation equation, which ensures a balance between arrivals and departures in any stable system. Our main contributions are as follows. We establish a link between the existence of stable policies, the dimensionality of the solution set of the conservation equation, and the compatibility graph's structure. We describe the convex polytope formed by non-negative solutions to the conservation equation, and we design policies that can achieve or closely approximate the vertices of this polytope. Lastly, we discuss potential extensions of our results beyond the main assumptions of this paper.

2012 ACM Subject Classification Mathematics of computing → Queueing theory; Mathematics of computing → Markov processes; Mathematics of computing → Matchings and factors

Keywords and phrases stochastic dynamic matching, graph theory, linear algebra, stability, matching rates, conservation equation

Supplementary Material Stochastic Matching Package

Software (Python, PyPi): Documentation [35]

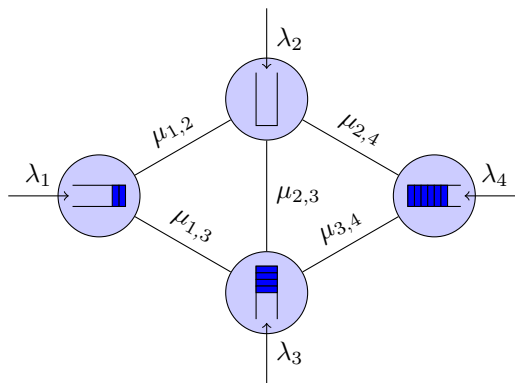
1 Introduction.

Stochastic dynamic matching problems, in which items arrive at random instants to be matched with other items, have recently attracted much attention in the stochastic-modeling community. These challenging control problems are highly relevant in many applications, including supply-chain management, pairwise kidney exchange programs, and online marketplaces. In pairwise kidney exchange programs for example, each item represents a donor-receiver pair, and two pairs can be matched if the donor of each pair is compatible with the receiver of the other pair. In online marketplaces, items are typically divided into

¹ Corresponding author.

two categories, called demand and supply, and the goal is to maximize a certain long-term performance criteria by appropriately matching demand items with supply items.

In this paper, we consider the following dynamic matching problem². Items of different classes arrive according to independent Poisson processes. Compatibility constraints between items are described by a simple graph on their classes, such that two items can be matched if their classes are neighbors in the graph. Unmatched items are stored in the queue of their class, and a matching policy decides which matches are performed and when. All in all, a stochastic matching model is described by a triplet (G, λ, Φ) , where $G = (V, E)$ is the compatibility graph, $\lambda = (\lambda_1, \lambda_2, \dots, \lambda_n)$ is the vector of per-class arrival rates, and Φ is the matching policy. Assuming stability, i.e. a steady-state behavior, the matching rate vector, denoted μ , represents the average rates at which each type of match is performed. In Figure 1 for instance, there are four item classes numbered from 1 to 4; classes 2 and 3 are compatible with all classes, while classes 1 and 4 are compatible only with classes 2 and 3.



■ **Figure 1** Illustration of a matching model (G, λ, Φ) on the diamond graph.

1.1 Motivation.

A central challenge in matching models is to characterize the set of matching rates that can be achieved by stable policies, i.e. policies that guarantee the system remains under control over time, and to understand the delay properties associated with these rates. Criteria like stability, delay, and matching rates are central to comparing the long-term performance of different matching policies and have practical relevance in real applications like organ exchanges or online matching platforms.

Many studies adopt an optimization viewpoint, where each match of type $k \in E$ gets a reward r_k , and the objective is to find a policy Φ that maximizes the long-run reward rate $\sum_{k \in E} r_k \mu_k$. This paradigm has clear practical appeal: for instance, in organ exchange programs, rewards capture metrics such as quality of life and survival rates after transplant [7]. Yet, natural reward-maximizing policies such as greedy priority-based schemes can jeopardize stability, as noted by [34, Section 5]. Moreover, there is no guarantee that the ordering of the resulting matching rates is consistent with the edge priorities.

These issues gain further complexity when a network of matching models is considered. For example, the matching rates μ from a stable first-level model (G, λ, Φ) can be the

² Detailed definitions of the concepts discussed here will be given in Section 2.

(non-Poisson) arrival rates in a second-level model with a compatibility graph $G' = (E, E')$ (the nodes of the second-level model are the edges of G).

From a broader perspective, these problems can be viewed through two complementary lenses. The first, inspired by queueing theory, focuses on stability and delay. The second, rooted in linear optimization seeks to maximize reward or minimize regret with respect to an optimal solution. As we will see, the link between these two questions is the polytope defined by the non-negativity constraints. The main goal of this paper is to bridge these perspectives by adopting a “polytope perspective” on matching models.

1.2 Contributions.

The foundation of our approach is the conservation equation: any stable policy must satisfy $A\mu = \lambda$, where A is the incidence matrix of the compatibility graph G . The properties of the linear mapping $\mu \mapsto A\mu$, particularly injectivity and surjectivity, are shown to be fundamentally linked to the behavior of matching models. We establish precise connections between these linear-algebraic properties and the structure of the underlying graph (Definitions 2–5 and Proposition 6). With some abuse of language, we say that a graph inherits a property, like bijectivity, injectivity, or surjectivity, if the associated linear map does; for example, G is referred to as bijective if $\mu \mapsto A\mu$ is bijective. Building on these concepts, our main contributions are as follows.

First, we connect stability and the conservation equation: There exists a policy Φ such that the matching model (G, λ, Φ) is stable) if and only if the graph G is surjective (Proposition 7) and the conservation equation has a solution with positive coordinates (Proposition 8). While equivalent results exist, our formulation provides a direct, intuitive, method to verify stabilizability, in polynomial time with respect to the number of classes and edges.

Second, we describe the solutions to the conservation equation, that is, the set Π of vectors $\mu \in \mathbb{R}^m$ such that $A\mu = \lambda$ (see (16) in Section 5.1). When G is bijective, we provide a closed-form expression of the unique solution (Proposition 9). When G is surjective-only, we give a parametric expression for Π (Proposition 10 and Definition 11) and characterize the polytope $\Pi_{\geq 0}$ of non-negative solutions. We show that the vertices of $\Pi_{\geq 0}$, i.e., its extreme points, correspond to injective solutions in the sense that the subgraph restricted to their support is injective (Proposition 13). Moreover, we show that the vertices are bijective with probability 1 when the vector λ is sampled randomly from a reasonable distribution (Proposition 14).

Third, we apply these results to the linear optimization of μ , which for a stable policy amounts to achieving a vertex of $\Pi_{\geq 0}$ (Propositions 15 and 16). If the target vertex is bijective, we propose an explicit optimal policy that attains it (Proposition 17). In contrast, when the vertex is injective-only, a strong trade-off emerges between regret with respect to the optimal solution and delay (Proposition 19). Motivated by this case, we introduce a sequence of policies that provably achieve arbitrarily small regret, at the cost of increasing delay (Proposition 18). While our bijective vertex results align with independently established findings in recent literature, our analysis and guarantees for the injective-only case are, to our knowledge, novel.

At a high level, all the optimizing policies we consider share a common recipe: start from a simple base policy designed for stability, and then adapt its parameters to be “vertex-aware”. While this insight may appear straightforward, it is highly effective in practice, especially for numerical evaluation. Building on this, we developed a modular simulation engine that enables easy integration of existing policies, their modification, and the creation of new heuristics. This flexible framework has been instrumental in validating our theoretical

results through extensive simulations, where our policies are systematically compared against state-of-the-art alternatives.

Finally, we outline possible extensions of our work. Through a convexity argument, we demonstrate that it is feasible to optimize nonlinear functions of μ : specifically, any μ^* in the interior of $\Pi_{\geq 0}$ can be achieved by a stable policy (Propositions 20 and 21). We further explore the optimization of μ with the additional constraint of using a *greedy* policy (Propositions 23 and 24 and Conjecture 25), present an extension of our model that accommodates arbitrary hypergraphs, allowing for the systematic discard of under-demanded items.

Note that this paper builds upon the original results reported in [14] as a preprint. The overlapping content reflects the continuity and development of our research, with the current manuscript providing a more comprehensive and peer-reviewed presentation.

1.3 State of the art.

We now review the relevant work related to (static or dynamic) matching problems.

Non-bipartite or general stochastic matching.

Our work is part of a broader research effort on the stochastic matching model that was briefly discussed earlier and will be described in details in Section 2, see [5, 6, 12, 16, 26, 34, 36]. Among these works, the following are particularly relevant because directly related to our results on stability. The paper of [34] is the earliest work on this matching model. It derives necessary and sufficient stability conditions that are instrumental in several of our results, in particular Propositions 7 and 8. This work also proves that the Match-the-Longest (ML) policy is maximally stable (in the sense that it always leads to stability whenever the matching problem (G, λ) is stabilizable), a result that is also applied in Proposition 8. [16] and [36] focus on the First-Come-First-Match (FCFM) policy. In particular, [36] proves that the FCFM policy is maximally stable, and [16] provides a new sufficient stability condition we prove to be also necessary in Proposition 8.

The recent work of [5] is perhaps the closest to ours, so we provide a detailed discussion to highlight the relation with our paper. The first statement of [5, Theorem 1] is synonym to the equivalence of statements ii and iii in Proposition 8. Our proof is significantly shorter because it relies more heavily on existing results. Our observation at the beginning of Section 4 that the conservation equation has a unique solution if and only if the graph is bijective (and not surjective-only) summarizes [5, Theorem 3]. Some formulas derived in [5, Section 9] are special cases of the formulas derived in Proposition 9. The non-bipartite matching model in [5] is slightly more general because it considers graphs with self-loops, that is, an edge can have identical endpoints, and this paper also considers the bipartite model that will be discussed in the next paragraph. However, [5] does not adopt the polytope approach that allows us to derive the necessary *and sufficient* conditions of Sections 4–6. Furthermore, although we decided to focus on nonbipartite matching models to alleviate the discussion, the general approach we develop in Section 3 equally applies to bipartite graphs and can be used to derive results for bipartite matching model such as those studied in [5].

Other variants of the model were studied recently. In particular, [26] consider item abandonment, [6] consider graphs with self-loops, and [24], [37], [30], [23], [41], and [39] allow matches not limited to two items by replacing the graph with a hypergraph, which is also called multi-way matching in the literature. The generalization of our work to multi-way matching will be discussed in Section 8.3.

Reward maximization.

A series of papers on multi-way ([37, 41, 30, 23]) and two-way matching ([29]) focus on maximizing edge-weighted rewards. These works are closely related to our analysis in Section 6, which addresses the same theme. For a thorough discussion, we postpone direct comparison until Section 6.4, following the presentation of our main findings on reward optimization.

Bipartite stochastic matching.

While our paper focuses on non-bipartite graphs, there is also an extensive literature on the bipartite case. A major complication with bipartite stochastic matching is that, in general, the model is unstable in the sense that the underlying Markov chain is at best null-recurrent, or transient, but never positive recurrent. To the best of our knowledge, the first example in the literature of a stochastic matching model with an infinite time horizon, which predated the model we consider, is the bipartite matching model introduced in [13], and later studied in by [1, 2, 9, 10, 11, 17]. In this model, the compatibility graph is bipartite, with two parts that correspond to supply and demand, respectively. Stabilizability is enforced by making a synchronization assumption: time is slotted and, during each time slot, one demand item *and* one supply item arrive. [2] and [9] obtained results on stability and [2] focused on matching rates; these results are similar to those described above regarding the literature on non-bipartite matching. Another work in the literature has obtained a stabilizable matching model by allowing renegeing or abandonment [3]. In general, the bipartite nature of the graph simplifies some calculations, for instance by allowing the application of flow-maximization algorithms to calculate optimal matching rates.

Static and fractional matching.

The static matching problem, in which the nodes of the graph represent items (rather than classes), has been extensively studied in mathematics, computer science, and economy, see [33]. Although the questions raised in static and dynamic matching are often different, the conservation equation that we obtain is reminiscent of several results on static matching. For example, finding a maximum-cardinality matching in the graph G (that is, a maximum-cardinality set of edges without common endpoints) is equivalent to finding integers $\mu_k \in \{0, 1\}$ for each edge $k \in E$ that maximize $\sum_{k \in E} \mu_k$ while satisfying the conservation equation with $\lambda_i = 1$ for each $i \in V$. The relaxation of this integer linear program leads to the so-called fractional matching problem, which has been studied in the literature, see [33, Section 7.2]. Therefore, the fractional matching polytope defined by [33, Section 7.5] is a special case of the convex polytope that we consider in Section 5.2, and our characterization of this convex polytope is a natural generalization of existing characterizations of the fractional polytope³.

1.4 Companion package

This paper is supported by the Python package *Stochastic Matching* by [35], which provides an open-source implementation of all the concepts and results developed here, together with

³ The fractional matching polytope is actually defined using non-strict inequalities rather than equalities. However, one can verify that these two convex polytopes have the same non-zero vertices.

additional tools for experimentation. The package is freely available on GitHub and PyPI, and can be readily installed on any recent Python distribution⁴.

The package is designed to remain flexible for extensions. Its main features include:

- Support for both simple graphs and hypergraphs, created either manually or through a built-in graph library, making it possible to test theoretical results on arbitrary structures;
- Automated categorization of graphs G (e.g., surjective, injective) and verification of the stabilizability of any instance (G, λ) ;
- Computation of the solution space Π and of the vertices of $\Pi_{\geq 0}$, including the reward-optimal vertex;
- Visualization tools that produce graph-based representations of Π , vertices, and policy matching rates;
- A scalable simulation framework for evaluating the long-term performance of arbitrary policies (up to 10^{10} time steps), together with a curated library of policy implementations that includes the approaches proposed in this paper and state-of-the-art baselines.

The package builds upon SciPy for linear algebra and optimization routines, VisJS Network for graph visualization, and Numba for efficient simulation.

While the package is designed as a standalone tool that can serve a broader community working on stochastic matching and related problems, a dedicated section of its documentation accompanies this article by reproducing all the results presented here and presenting additional examples⁵. We believe that this package constitutes a valuable contribution on its own, as it provides a foundation for future experimental and theoretical work.

1.5 Outline.

The remainder of the paper is organized as follows. **Section 2** gives a formal definition of the model. **Section 3** introduces the conservation equation and defines the notions of surjectivity, injectivity, and bijectivity for a graph. We use these definitions to formulate new necessary and sufficient stability conditions. In particular, we show that stability requires the compatibility graph to be surjective (that is, either bijective or surjective-only). In **Section 4**, we focus on bijective graphs and give a closed-form expression of the unique solution to the conservation equation. **Section 5** characterizes the solution set of the conservation equation for surjective-only graphs. **Section 6** focuses on the linear optimization of the matching rates (that is, optimizing a linear combination of the matching rates) under a stable policy. **Section 7** focuses on computing the numerical performance of policies through simulations. Lastly, in **Section 8**, we consider several extensions of our framework: non-linear optimization, restriction to greedy policies, and hypergraph matchings.

2 Stochastic dynamic matching.

We consider the following model: items arrive at random times to be matched with other items; each incoming item may be paired with any unmatched item of a compatible class. When a match occurs, the involved items disappear immediately, while unmatched items wait in a queue. In this paper, such a stochastic dynamic matching system is described by a triplet (G, λ, Φ) , where G is the *compatibility graph*, λ the vector of *arrival rates*, and Φ the *matching policy*.

⁴ For instance, `pip install stochastic-matching` or `uv add stochastic-matching`.

⁵ https://balouf.github.io/stochastic_matching/companion/index.html

The remainder of this section formalizes the model in detail. Our goal here is to state the model precisely so that the paper remains both rigorous and self-contained. Readers already familiar with queueing theory and matching systems may safely skip ahead to Section 3, as the definitions and notations introduced here do not extend beyond the standard framework.

For convenience, notation is summarized in Table 1.

2.1 Compatibility graph.

Compatibility constraints between items are described by a graph $G = (V, E)$, called the *compatibility graph* of the model, which is simple (undirected and without self-loop). The number of nodes is represented by n , while m denotes the number of edges.

We use $V = \{v_1, v_2, \dots, v_n\}$ to denote the set of nodes, where each node corresponds to a class in the matching model. In cases where there is no confusion, we may refer to a class v_i simply by its index i . Following the intuition conveyed in Figure 1, we will use the terms “class i ” and “queue i ” interchangeably, and refer for instance to the number of unmatched class- i items as the size of queue i ; this is a convenience of terminology, and this does not preclude the matching policy from using information not captured by this state representation, such as the arrival order of items of different classes (see Section 2.3 for more details).

The set of edges is denoted by $E = \{e_1, e_2, \dots, e_m\}$. These edges represent compatibility constraints between item classes, in the sense that a class- i item and a class- j item can be matched with one another if and only if their classes are adjacent, that is, if there is an edge with endpoints i and j . When there is no ambiguity, we may refer to an edge $e_k \in E$ with endpoints $i, j \in V$ by its index k or its set $\{i, j\}$ of endpoints. In Figure 1 for instance, there are four item classes numbered from 1 to 4. Classes 2 and 3 are compatible with all classes, but classes 1 and 4 are only compatible with classes 2 and 3. The absence of self-loop means that an item of a given class cannot be matched with other items of the same class.

Lastly, we let \mathbb{I} denote the family of independent sets of the compatibility graph G , where an independent set of G is a non-empty set of nodes that are pairwise non-adjacent. The family of independent sets in the compatibility graph of Figure 1 is $\mathbb{I} = \{\{1\}, \{2\}, \{3\}, \{4\}, \{1, 4\}\}$.

2.2 Arrival process.

Class- i items arrive according to an independent Poisson process with rate $\lambda_i > 0$, for each $i \in V$. The vector of arrival rates is denoted by $\lambda = (\lambda_1, \lambda_2, \dots, \lambda_n) \in \mathbb{R}_{>0}^n$. Scaling all coordinates of λ by the same positive constant is equivalent to changing the time unit, so we can renormalize λ without changing the dynamics. For example, we will sometimes use the *unit* normalization, in which $\sum_{i \in V} \lambda_i = 1$. We also let $I = (I_t, t \in \mathbb{N})$ denote the sequence of independent and identically distributed (i.i.d.) item classes, so that I_t is the class of the $(t + 1)$ -th item, equal to i with probability $\lambda_i / (\sum_{j \in V} \lambda_j)$, for each $t \in \mathbb{N}$. The couple (G, λ) is called a *matching problem* or simply a *problem*. Occasionally, when we need to specify the sequence of incoming items and not merely its distribution, we will also refer to the couple (G, I) as a (matching) problem.

2.3 Policy and matching dynamics.

Most of the paper focuses on deterministic size-based policies, that is, policies whereby matching decisions are deterministic functions of the queue-size vector. However, as we will briefly explain at the end of this section (and discuss in more details in the supplementary

material), our results also apply to a more general definition of a policy. Throughout the paper, we assume that the system is initially empty, meaning that it starts with no unmatched item.

2.3.1 Deterministic size-based policies.

A (deterministic) size-based *matching policy* is defined formally as a function $\Phi : \mathcal{Q} \times V \rightarrow V \cup \{\perp\}$, where \mathcal{Q} is an infinite subset of \mathbb{N}^n that contains the reachable states of the system. In-depth discussion on \mathcal{Q} will come in Sections 2.3.2 and 2.3.3. For each $q \in \mathcal{Q}$ and $i \in V$, an incoming class- i item that finds the system in state q is matched with an item of class $\Phi(q, i)$ if $\Phi(q, i) \in V$ and is added to class- i queue if $\Phi(q, i) = \perp$. The matching policy is assumed to be *adapted* to the compatibility graph G in the sense that

$$\Phi(q, i) \in \{j \in V_i : q_j \geq 1\} \cup \{\perp\}, \quad q \in \mathcal{Q}, \quad i \in V, \quad (1)$$

where V_i is the neighbor set of node i in G . The system dynamics are described by a Markov chain $Q = (Q_t, t \in \mathbb{N})$, called the *queue-size process*. For each $t \in \mathbb{N}$, $Q_t = (Q_{t,1}, Q_{t,2}, \dots, Q_{t,n})$ is an n -dimensional vector giving the number of unmatched items of each class right after the arrival of the t -th item, with the assumption that the system is initially empty, that is, $Q_0 = 0$. The system dynamics satisfy the recursion

$$Q_{t+1} = \begin{cases} Q_t + \mathbb{1}_{I_t} & \text{if } J_t = \perp, \\ Q_t - \mathbb{1}_{J_t} & \text{if } J_t \neq \perp, \end{cases} \quad (2)$$

where $J_t = \Phi(Q_t, I_t)$ for each $t \in \mathbb{N}$, and $\mathbb{1}_i$ is the n -dimensional vector with one in coordinate i and zero elsewhere, for each $i \in V$. We assume that the policy Φ is such that the Markov chain Q has state space \mathcal{Q} and is irreducible. By unfolding the recursion (2), we obtain that, for each $t \in \mathbb{N}$,

$$Q_{t,i} = L_{t,i} - \sum_{k \in E_i} M_{t,k}, \quad t \in \mathbb{N}, \quad i \in V, \quad (3)$$

where $E_i \subseteq E$ is the set of edges that are incident to node i in the graph G , for each $i \in V$, $L_{t,i}$ is the number of class- i items among the first t arrivals, for each $t \in \mathbb{N}$ and $i \in V$, and $M_{t,k}$ is the number of times that classes i and j are matched over the first t arrivals, for each $t \in \mathbb{N}$ and $\{i, j\} = e_k \in E$:

$$L_{t,i} = \sum_{s=0}^{t-1} \mathbb{1}_{\{I_s=i\}}, \quad t \in \mathbb{N}, \quad i \in V, \quad (4)$$

$$M_{t,k} = \sum_{s=0}^{t-1} \mathbb{1}_{\{\{I_s, J_s\}=e_k\}}, \quad t \in \mathbb{N}, \quad k \in E, \quad (5)$$

with the convention that the sums are zero if $t = 0$. The triplet (G, λ, Φ) is called a *matching model*, or simply a *model*. Occasionally, when specifying the sequence of incoming item classes is useful, we will also refer to the triplet (G, I, Φ) as a (matching) model.

2.3.2 Greedy policies.

A policy Φ is called *greedy* if an incoming item is matched whenever possible, that is, if there is an unmatched item that is compatible. More formally, the policy Φ is greedy if

$$\Phi(q, i) \neq \perp \text{ for each } (q, i) \in \mathcal{Q} \times V \text{ such that } \{j \in V_i : q_j \geq 1\} \neq \emptyset. \quad (6)$$

Equivalently, a policy Φ is greedy if the set of unmatched item classes under this policy is an independent set of the compatibility graph, meaning that the state space \mathcal{Q} of the queue-size process is equal to

$$\mathcal{Q}_{\mathcal{G}} = \{q \in \mathbb{N}^n : q_i q_j = 0 \text{ for each } i, j \in V \text{ such that } \{i, j\} \in E\}. \quad (7)$$

Here are two examples of greedy matching policies that will appear later in the paper:

- **Match-the-Longest (ML)**: For each $(q, i) \in \mathcal{Q} \times V$ such that $\sum_{j \in V_i} q_j \geq 1$, we choose $\Phi(q, i) \in \arg \max_{j \in V_i} (q_j)$ (ties are broken arbitrarily). This policy was considered by [34, 6, 26, 5].
- **Highest-Reward-First (HRF)**: This policy selects matches according to rewards defined on edges. If r denotes a vector of \mathbb{R}^m (indexed by the edges) with distinct coordinates, we let $\Phi(q, i) = \arg \max_{j \in V_i: q_j \geq 1} r_{i,j}$ for each $(q, i) \in \mathcal{Q} \times V$ such that $\sum_{j \in V_i} q_j \geq 1$. This definition remains valid if the coordinates of r have ties that do not impact the decision, i.e., if all pairs of edges that are incident to the same node and are not part of a triangle have distinct rewards⁶. Note that the order induced by r fully defines an HRF policy.

2.3.3 Non-greedy policies.

The state space \mathcal{Q} under non-greedy policies is a strict superset of the set $\mathcal{Q}_{\mathcal{G}}$ defined in (7). In Section 6, non-greedy policies are obtained by applying the following modifications to greedy policies:

- **Filtering**: Given a subset $E^* \subsetneq E$ of edges, replace V_i (the neighbors of node i) with $V_i^* = \{j \in V_i : \{i, j\} \in E^*\}$ in the definition (1) of Φ . Intuitively, we eliminate the edges of $E \setminus E^*$ and follow a (greedy or non-greedy) policy on the subgraph $G^* = (V, E^*)$.
- **Semi-filtering**: We consider a variant of filtering policies in which the restriction of selecting matches from a subset E^* is not always enforced. Examples of semi-filtering policies will be given in Sections 6 and 7.

2.3.4 Other policies.

Although our results are stated for deterministic size-based policies because they are notationwise convenient, our results apply to a broader family of policies that are either random or require a more complex state descriptor, or both.

The extension to random policies is standard. A random (size-based) policy Φ can be defined as a function $\Phi : \mathcal{Q} \times V \times (V \cup \{\perp\}) \rightarrow [0, 1]$ such that, for each $t \in \mathbb{N}$, J_t is sampled according to the distribution $\Phi(q, i, \cdot)$ given that $Q_t = q$ and $I_t = i$. Saying that the policy is adapted to the compatibility graph G is then equivalent to saying that, for each $q \in \mathcal{Q}$ and $i \in V$, the support of $\Phi(q, i, \cdot)$ is included into $\{j \in V_i : q_j \geq 1\} \cup \{\perp\}$. The policy is greedy if $\Phi(q, i, \perp) = 0$ for each $(q, i) \in \mathcal{Q} \times V$ such that $\{j \in V_i : q_j \geq 1\} \neq \emptyset$ or, equivalently, if $\mathcal{Q} = \mathcal{Q}_{\mathcal{G}}$.

An even broader set of policies that fit into our framework is described in Appendix A. In a nutshell, the extended definition of a policy starts with a pair $(\mathcal{S}, |\cdot|)$, where \mathcal{S} is a countably infinite set and $|\cdot| : \mathcal{S} \rightarrow \mathbb{N}^n$ is a function that maps any state $s \in \mathcal{S}$ to the corresponding queue-size vector. The policy is then a function $\Phi : \mathcal{S} \times V \times (V \cup \{\perp\}) \times \mathcal{S} \rightarrow [0, 1]$ such

⁶ If three classes form a triangle in the compatibility graph, at most one of them can be non-empty under a greedy policy, so at most one edge of the triangle can be in the input of $\arg \max$.

that $\Phi(s, i, j, s')$ is the conditional probability that, given an incoming class- i item finds the system in state s , the matching decision is j and the new state is s' . The policy is assumed to be such that the Markov chain $S = (S_t, t \in \mathbb{N})$ defined on \mathcal{S} by the evolution of the system state is irreducible. The stochastic process $Q = (Q_t = |S_t|, t \in \mathbb{N})$ is called the *queue-size process*. It no longer satisfies the Markov property in general, but it does satisfy the evolution equations (2) and (3), with $L_i = (L_{t,i}, t \in \mathbb{N})$ and $M_k = (M_{t,k}, t \in \mathbb{N})$ defined by (4) and (5) for each $i \in V$ and $k \in E$. The state space of the queue-size process is given by $\mathcal{Q} = \{|s|, s \in \mathcal{S}\}$. The policy is greedy if $\mathcal{Q} = \mathcal{Q}_G$ and non-greedy if $\mathcal{Q} \supsetneq \mathcal{Q}_G$, where \mathcal{Q}_G is still given by (7). First-Come-First-Match (FCFM) (e.g., see [36, 16]) is a classical example of a deterministic policy that requires an expanded state descriptor (remembering the arrival order of items of different classes).

This extended policy definition can initially be ignored, with S_t (resp. \mathcal{S}) understood as Q_t (resp. \mathcal{Q}). All results for extended policies remain valid if restated for deterministic size-based policies, except for the convexity result in Proposition 20, which specifically relies on extended policies.

2.4 Stability.

We now define the notions of stability and stabilizability, which will be explored further in Section 3.

► **Definition 1** (Stability and stabilizability).

- (i) A model (G, λ, Φ) is stable if the Markov chain $(S_t, t \in \mathbb{N})$ is positive recurrent. In this case, we say that the policy Φ stabilizes the problem (G, λ) .
- (ii) A problem (G, λ) is stabilizable if there exists a policy that stabilizes it.
- (iii) A compatibility graph G is stabilizable if there exists a vector $\lambda \in \mathbb{R}_{>0}^n$ such that the problem (G, λ) is stabilizable.

The ML greedy policy, discussed in Section 2.3.2, has the property of stabilizing all stabilizable matching problems ([34]). The First-Come-First-Match (FCFM) greedy policy has the same property (see [36] for details). When the matching problem (G, λ) is clear from the context, we will simply call a policy (adapted to G) *stable* if the corresponding model (G, λ, Φ) is stable.

Stability is a central property in our framework. In addition to ensuring finite waiting times—a desirable feature for dynamic matching applications such as organ donation, assemble-to-order systems, and online marketplaces—it also supports a rigorous theoretical analysis. In particular, it guarantees that matching rates can be properly defined.

Matching rates.

Consider a stable matching model (G, λ, Φ) . We define the *matching rate* μ_k along an edge $e_k \in E$, with endpoints i and j , as the long-run average number of matches between a class- i item and a class- j item per unit of time, given by:

$$\left(\sum_{i \in V} \lambda_i \right) \times \frac{1}{t} M_{t,k} \xrightarrow[t \rightarrow +\infty]{\text{almost surely}} \mu_k, \quad k \in E. \quad (8)$$

This quantity is uniquely defined according to the ergodic theorem, see [38, Theorem 1.10.2]. Note that $M_{t,k}/t$ represents the average number of matches *per arrival* (out of the first t arrivals), and the factor $(\sum_{i \in V} \lambda_i)$ converts this into the average number of matches per

unit of time. The vector of matching rates associated with the model (G, λ, Φ) is denoted by $\mu = (\mu_1, \dots, \mu_m) \in \mathbb{R}_{\geq 0}^m$. When necessary, we write $\mu(\Phi)$ or $\mu(G, \lambda, \Phi)$ to indicate the dependence explicitly. Sections 3–5 will investigate the structural constraints that a vector of matching rates must satisfy.

General notation	
$\mathbb{N}, \mathbb{N}_{>0}, \mathbb{R}, \mathbb{R}_{\geq 0}, \mathbb{R}_{>0}$	Sets of non-negative integers, positive integers, real numbers, non-negative real numbers, positive real numbers.
$\geq, \leq, >, <$	Coordinate-wise comparison in \mathbb{R}^n .
$ \mathcal{A} $	Cardinality of the set \mathcal{A} .
Graph notation	
$G = (V, E)$	Simple graph G with $ V = n$ vertices and $ E = m$ edges.
v_i	Vertex indexed by i (denoted i if there is no ambiguity).
e_k, k , or $\{i, j\}$	Edge indexed by k , with endpoint vertices i and j .
$V_i, V(\mathcal{I})$	Neighbor set of a node i and of node set \mathcal{I} .
E_i	Set of edges incident to node i .
Matching notation	
$\lambda = (\lambda_i)_{1 \leq i \leq n}$	Vector of arrival rates of the item classes.
Φ	A matching policy.
$\mu = (\mu_k)_{1 \leq k \leq m} = (\mu_{i,j})_{\{i,j\} \in E}$	Vector of matching rates along the edges.
Linear-algebra notation	
$A = (a_{i,k})_{i \in V, k \in E}$	Incidence matrix of the graph G .
$A^\top = (a_{k,i})_{k \in E, i \in V}$	Transpose of the matrix A .
$\ker(A) = \{y \in \mathbb{R}^m : Ay = 0\}$	Right kernel of the matrix A . Its dimension is called the nullity of A .
$\ker(A^\top) = \{x \in \mathbb{R}^n : A^\top x = 0\}$	Left kernel of the matrix A . Its dimension is the nullity of A^\top .
$d = m - n$	Dimension of the right kernel of the matrix A if G is surjective.

■ **Table 1** Table of key notation.

3 Graph theory and linear algebra.

In this section, we present a central result of this paper, namely, the connection between stability and the structure of the compatibility graph. Section 3.1 introduces the conservation equation, a system of linear equations satisfied by all matching rate vectors. Section 3.2 introduces the related concepts of surjectivity, injectivity, and bijectivity that will play a key role throughout the paper. In Section 3.3, we combine these concepts to formulate new necessary and sufficient conditions under which a compatibility graph G or a matching problem (G, λ) is *stabilizable* in the sense of Definition 1. Section 3.4 illustrates these results with early examples.

3.1 Conservation equation.

Computing the matching rate vector achieved by a given stable policy or characterizing the set of matching rate vectors that can be achieved by stable policies is a difficult problem *a priori*. To circumvent this difficulty, we first establish a necessary condition known as the *conservation equation*. This equation is satisfied by the matching rate vectors achieved by all

stable policies. It asserts that, in a stable system, the arrival of items and their departure due to matches balance each other in the long run. More formally, given a stable matching model (G, λ, Φ) , the conservation equation (CE) can be derived by dividing (3) by t and taking the limit as t tends to infinity, and it can be written in two equivalent forms, either as a system of linear equations (CE-1) or in matrix form (CE-2):

$$\sum_{k \in E_i} \mu_k = \lambda_i, \quad i \in V, \quad (\text{CE-1})$$

$$A\mu = \lambda, \quad (\text{CE-2})$$

where the $n \times m$ matrix $A = (a_{i,k})_{i \in \{1,2,\dots,n\}, k \in \{1,2,\dots,m\}}$ is the *incidence* matrix of the graph G , defined by $a_{i,k} = 1$ if edge e_k is incident to node v_i and $a_{i,k} = 0$ otherwise. Using a conservation equation is common in queueing theory, but our contributions in this paper primarily stem from the novel mixed approach that combines graph theory and linear algebra, which we will elaborate on in the following sections.

The conservation equation (CE) holds significant importance throughout this paper. While our primary focus is to understand the matching rate vector, we often find it beneficial to temporarily depart from interpreting λ and μ as vectors of arrival and matching rates in a matching model. Instead, we view the conservation equation as a linear equation with λ as a free variable and μ as an unknown. This perspective allows us to explore various aspects of the equation and its implications. In this context, we will sometimes allow the coordinates of λ and μ to be negative, even if the vectors of arrival and matching rates in a matching model have non-negative coordinates.

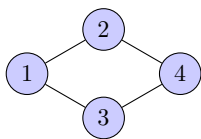
3.2 Surjectivity, injectivity, and bijectivity.

Definitions 2–5 below introduce the notions of *surjectivity*, *injectivity*, and *bijectivity* of a graph. In a nutshell, a compatibility graph G is said to be surjective (resp. injective, bijective) if the linear application $\mu \mapsto A\mu$ defined by its incidence matrix A is surjective (resp. injective, bijective). Interestingly, simple equivalent conditions exist in terms of the graph structure. As we will see later, these notions are fundamental to study the stability of matching models and the associated matching rate vector. In particular, we will see that (i) a compatibility graph G is stabilizable if and only if G is surjective, and (ii) the matching rates in a stabilizable matching problem (G, λ) are independent of the policy Φ if and only if G is bijective. Toy examples are shown in Figure 2, and other illustrative examples are discussed in Section 3.4. The equivalence of the conditions given in Definitions 2–4 and the implications in Proposition 6 are proved in Appendix B.1.

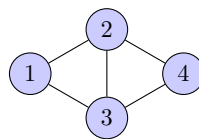
► **Definition 2** (Surjective graph). *Consider a simple graph $G = (V, E)$ with n nodes and m edges. Let A denote the $n \times m$ incidence matrix of G . The graph G is called surjective if one of the following equivalent conditions is satisfied:*

- (i) *The function $\mu \in \mathbb{R}^m \mapsto A\mu \in \mathbb{R}^n$ is surjective.*
- (ii) *For each $\lambda \in \mathbb{R}^n$, the equation $A\mu = \lambda$ of unknown $\mu \in \mathbb{R}^m$ has at least one solution.*
- (iii) *The left kernel of the matrix A is trivial.*
- (iv) *Each connected component of the graph G is non-bipartite; in other words, each connected component of G contains at least one odd cycle.*

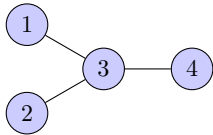
► **Definition 3** (Injective graph). *Consider a simple graph $G = (V, E)$ with n nodes and m edges. Let A denote the $n \times m$ incidence matrix of G . The graph G is called injective if one of the following equivalent conditions is satisfied:*



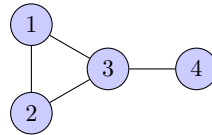
(a) Graph that is neither surjective nor injective. Both A^\top and A have nullity 1.



(b) Surjective-only graph. The nullity of A^\top is 0 and that of A is 1.



(c) Injective-only graph. The nullity of A^\top is 1 and that of A is 0.



(d) Bijective graph. Both A^\top and A have nullity 0.

■ **Figure 2** Examples of graphs. The nullity of a matrix is the dimension of its kernel.

- (i) The function $\mu \in \mathbb{R}^m \mapsto A\mu \in \mathbb{R}^n$ is injective.
- (ii) For each $\lambda \in \mathbb{R}^n$, the equation $A\mu = \lambda$ of unknown $\mu \in \mathbb{R}^m$ has at most one solution.
- (iii) The right kernel of the matrix A is trivial.
- (iv) Each connected component of the graph G contains at most one odd cycle and no even cycle; in other words, each connected component of G is either a tree or a unicyclic graph with an odd cycle.

► **Definition 4** (Bijective graph). Consider a simple graph $G = (V, E)$ with n nodes and m edges. Let A denote the $n \times m$ incidence matrix of G . The graph G is called bijective if the following equivalent conditions are satisfied:

- (i) The function $\mu \in \mathbb{R}^m \mapsto A\mu \in \mathbb{R}^n$ is bijective.
- (ii) For each $\lambda \in \mathbb{R}^n$, the equation $A\mu = \lambda$ of unknown $\mu \in \mathbb{R}^m$ has exactly one solution.
- (iii) The matrix A is invertible.
- (iv) Each connected component of the graph G contains one cycle and this cycle is odd.

► **Definition 5** (Surjective-only graph, injective-only graph). A simple graph G is called surjective-only (resp. injective-only) if G is surjective but not injective (resp. injective but not surjective).

The following proposition gives necessary conditions for surjectivity and injectivity in terms of the number of nodes and edges in the graph.

► **Proposition 6.** Consider an undirected graph $G = (V, E)$ with n nodes and m edges.

- (i) If G is surjective, then $n \leq m$.
- (ii) If G is injective, then $n \geq m$.
- (iii) If G is bijective, then $n = m$.
- (iv) If G is surjective, then G is also injective if and only if $n = m$.
- (v) If G is injective, then G is also surjective if and only if $n = m$.

3.3 Necessary and sufficient conditions for stabilizability

The definitions and results of Sections 3.1 and 3.2 allow us to formulate new necessary and sufficient conditions for stabilizability.

3.3.1 Stabilizable compatibility graph.

The following proposition gives necessary and sufficient conditions for a graph G to be stabilizable, in terms of either its structure or its incidence matrix.

► **Proposition 7.** *Let G be a compatibility graph. The following conditions are equivalent:*

- (i) *The graph G is stabilizable.*
- (ii) *The graph G is surjective.*

Proof. Equivalence between Proposition 7 i and Definition 2iv has been proved by [34, Theorem 1]. ◀

Although the equivalence between Proposition 7i and Definition 2iv is a known result, no prior literature on stochastic matching models (to the best of our knowledge) has explored the connection between Proposition 7i and the alternative definitions of surjectivity introduced in Definition 2. As we will see later, this new characterization of the stabilizability of a graph G will be useful to analyze the relationship between delay and matching rates.

In the remainder, we will use the words “stabilizable” and “surjective” interchangeably. Furthermore, unless stated otherwise, we will assume that the graph G is surjective.

3.3.2 Stabilizable matching problem.

We now turn to the stabilizability of a matching problem (G, λ) . As recalled in Section 2.4, two examples of greedy policies that stabilize the model whenever this matching problem is stabilizable are ML and FCFM, as shown by [34] and [36], respectively. Proposition 8 below provides necessary and sufficient conditions for the matching problem (G, λ) to be stabilizable; condition ii was already derived by [34], but condition iii is new⁷.

► **Proposition 8.** *Consider a matching problem (G, λ) with a surjective graph G . The following conditions are equivalent:*

- (i) *The matching problem (G, λ) is stabilizable.*
- (ii) *For each independent set $\mathcal{I} \in \mathbb{I}$, we have $\sum_{i \in \mathcal{I}} \lambda_i < \sum_{i \in V(\mathcal{I})} \lambda_i$, where $V(\mathcal{I}) = \bigcup_{i \in \mathcal{I}} V_i$ is the neighbor set of \mathcal{I} .*
- (iii) *The conservation equation (CE) admits a solution $\mu \in \mathbb{R}_{>0}^m$ (i.e., a solution with strictly positive components).*

Proof. This result is proved in Appendix B.2 using results from [34, 16]. ◀

One might expect that the time complexity to verify condition ii in Proposition 8 is exponential in the general case, considering that the number of independent sets grows exponentially fast with the number n of classes. However, [34, Proposition 1] proved that there exists a polynomial algorithm for verifying this condition. It is worth noting that this verification process is indirect in that it involves constructing a bipartite double cover of G . In contrast, condition iii offers a more direct, polynomial-time method to verify the stabilizability of a matching model (G, μ) . To present this approach more explicitly, we differentiate between two cases based on whether the graph G , assumed to be surjective, is surjective-only or bijective.

⁷ [36] concurrently proved a similar result, see Section 1.3.

► **Remark 1.** As observed by [16, Lemma 12], if the graph G is surjective, condition iii in Proposition 8 gives a simple way of generating vectors $\lambda \in \mathbb{R}_{\geq 0}^n$ such that the problem (G, λ) is stabilizable: it suffices to take $\lambda = A\mu$ for some $\mu \in \mathbb{R}_{> 0}^m$. For instance, if $\mu = (\beta, \dots, \beta)$ for some $\beta > 0$, then the coordinates of λ are proportional to the degree of each node.

Verify stabilizability when G is bijective.

If the graph G is bijective, then the matrix A is invertible, and (CE) has a unique solution, namely $A^{-1}\lambda$. Proposition 8 implies that the matching problem (G, λ) is stabilizable if and only if all coordinates of $A^{-1}\lambda$ are positive. The special case of bijective graphs will be investigated in detail in Section 4, where we will provide a more direct expression for $A^{-1}\lambda$.

Verify stabilizability when G is surjective-only.

If the compatibility graph G is surjective-only, (CE) has multiple solutions. To determine if one of these solutions is positive, it suffices to solve a linear optimization problem that searches for a solution to (CE) whose smallest coordinate is as large as possible:

$$\begin{aligned} & \underset{z=(z_1, z_2, \dots, z_{m+1}) \in \mathbb{R}^{m+1}}{\text{Maximize}} && z_{m+1}, \\ & \text{Subject to} && A(z_1, z_2, \dots, z_m)^\top = \lambda, \\ & && z_i \geq z_{m+1}, \quad i \in \{1, 2, \dots, m\}. \end{aligned} \tag{10}$$

Here, the first m coordinates of the vector z are the coordinates of a vector $\mu \in \mathbb{R}^m$ that satisfies (CE), and the last coordinate of z is a lower bound of the coordinates of this vector μ . Indeed, the equality constraint means that μ satisfies (CE), and the inequality constraint means that the last coordinate of z is less than or equal to its other coordinates. The value to maximize is the last coordinate of the vector z .

If z is a solution to (10), we call the corresponding vector $\mu \in \mathbb{R}^m$ a *maximin* solution to (CE). The optimization problem (10) is a textbook linear optimization problem. It can be solved with a time complexity that is polynomial in the number n of nodes and the number m of edges using many methods, for instance the interior-point-method, see [27].

The linear optimization problem (10) has a solution with positive coordinates if and only if (CE) has a solution with positive coordinates. According to Proposition 8, this is equivalent to saying that the matching problem (G, λ) is stabilizable. Therefore, to verify if a matching problem (G, λ) is stabilizable, it suffices to find a solution to the linear optimization problem (10) and to check if its last coordinate is positive.

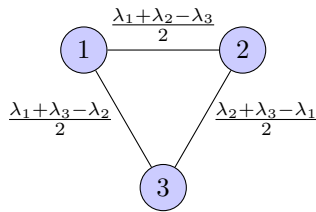
► **Remark 2.** Observe that the optimization problem (10) always has solutions with finite coordinates. Indeed, the set of vectors that satisfy the constraints of (10) contains at least one valid solution with real-valued coordinates (this is again a consequence of the surjectivity of G). We just need to consider an arbitrary solution μ of (CE) (see Section 5.1.2 for a concrete example using the Moore-Penrose inverse) and to let $z_\mu = (\mu_1, \mu_2, \dots, \mu_m, \min(\mu))$, where $\min(\mu)$ is the smallest coordinate of the vector μ . Any solution better than z_μ has all its coordinates lower-bounded by $\min(\mu)$ and upper-bounded by $\max(\lambda) - \min(0, (n-2)\min(\mu))$. The latter bound is obtained by observing that, if edge k is incident to node i and if $(\mu'_1, \mu'_2, \dots, \mu'_m, x')$ is a solution to (10) such that $x' \geq \min(\mu)$, then $\mu'_k = \lambda_i - \sum_{\ell \in E_i \setminus k} \mu'_\ell$ by (CE). We then use the inequalities $\lambda_i \leq \max(\lambda)$ and $\sum_{\ell \in E_i \setminus k} \mu'_\ell \geq \min(0, (n-2)x') \geq \min(0, (n-2)\min(\mu))$ (this latter inequality is obtained by distinguishing two cases, depending on whether $\min(\mu) \geq 0$ or $\min(\mu) < 0$). Therefore, the solutions better than z_μ belong to a compact set of \mathbb{R}^{m+1} , which ensures the existence of an optimal solution with finite coordinates.

3.4 Early examples.

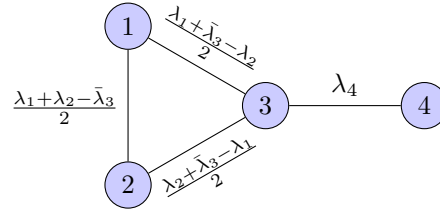
We now provide examples that illustrate the stabilizability results of Propositions 7 and 8 as well as our definitions of surjectivity, injectivity, and bijectivity. These examples will also introduce useful notions that will be further explored in Sections 4–6.

3.4.1 Bijective graphs.

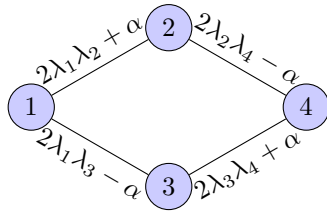
We first consider connected compatibility graphs G that are both surjective and injective: the graph contains exactly one cycle, and this cycle is odd. According to Definition 4, (CE) has a unique solution for each vector $\lambda \in \mathbb{R}^n$ of arrival rates. Proposition 8 implies that the matching problem (G, λ) is stabilizable if and only if the coordinates of this solution are positive, in which case this solution gives the matching rates achieved by *all* stable matching policies. By Remark 1, one can always exhibit a vector $\lambda \in \mathbb{R}_{>0}^n$ of arrival rates such that the matching problem (G, λ) is stable.



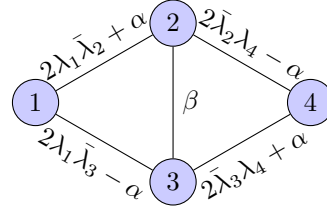
(a) Matching rates in the triangle graph C_3 .



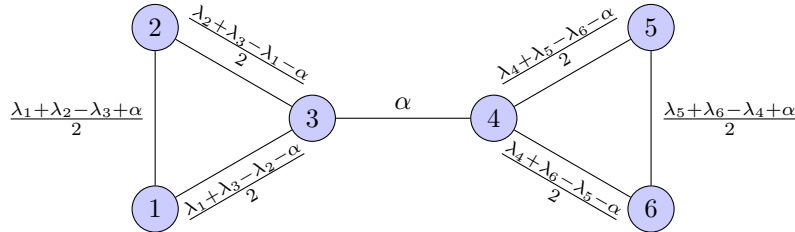
(b) Matching rates in the paw graph. $\bar{\lambda}_3 = \lambda_3 - \lambda_4$ denotes the residual rate that class 3 can provide to classes 1 and 2.



(c) Matching rates in the square graph C_4 with the normalization $\lambda_1 + \lambda_4 = \frac{1}{2}$. $2\beta = \lambda_2 + \lambda_3 - \lambda_1 - \lambda_4 = \lambda_2 + \lambda_3 - \frac{1}{2}$ is the difference $\lambda_1 + \lambda_4 = \lambda_2 + \lambda_3 = \frac{1}{2}$. This graph between the arrival rates of the inner part $\{2, 3\}$ and the outer part $\{1, 4\}$.



(d) Matching rates in the diamond graph with the normalization graph C_4 with the normalization $\lambda_1 + \lambda_4 = \frac{1}{2}$. $2\beta = \lambda_2 + \lambda_3 - \lambda_1 - \lambda_4 = \lambda_2 + \lambda_3 - \frac{1}{2}$ is the difference $\lambda_1 + \lambda_4 = \lambda_2 + \lambda_3 = \frac{1}{2}$. This graph between the arrival rates of the inner part $\{2, 3\}$ and the outer part $\{1, 4\}$. $\bar{\lambda}_2 = \lambda_2 - \beta$ and $\bar{\lambda}_3 = \lambda_3 - \beta$ represent the residual rates that classes 2 and 3 can provide to classes 1 and 4, and they are such that $\lambda_1 + \lambda_4 = \bar{\lambda}_2 + \bar{\lambda}_3 = \frac{1}{2}$, as in Example 3.



(e) Matching rates in the kayak paddle $KP_{3,3,1}$.

■ **Figure 3** Examples of Section 3.4.

► **Example 1 (Triangle).** *If the graph G is a triangle graph C_3 , and the vector $\lambda = (\lambda_1, \lambda_2, \lambda_3)$ is given, the solution to (CE) is unique and shown in Figure 3a. Indeed, a triangle graph contains a unique cycle which is odd, hence it is bijective according to Definition 2. According to Proposition 8iii, (G, λ) is stabilizable if and only if all coordinates of the unique solution of (CE) are positive. This condition is equivalent to Proposition 8ii, which states that $\lambda_1 < \lambda_2 + \lambda_3$, $\lambda_2 < \lambda_1 + \lambda_3$, and $\lambda_3 < \lambda_1 + \lambda_2$. Alternatively, this condition can be expressed as λ_1 , λ_2 , and λ_3 being the lengths of the sides of a non-degenerate triangle (i.e., they satisfy the triangular inequality). Under these conditions, the model (G, λ, Φ) is stable if Φ is for instance the (unique) greedy policy adapted to G (Proposition 34 shows that a complete graph G admits a unique greedy policy Φ).*

► **Example 2 (Paw graph).** *If G is a paw graph, the solution to (CE) is again unique and shown in Figure 3b. Here, $\bar{\lambda}_3 = \lambda_3 - \lambda_4$ is the remaining rate of class 3 after accounting for the needs of class 4. After this subtraction, the matching rates along edges $\{1, 2\}$, $\{1, 3\}$, and $\{2, 3\}$ are defined as in the triangle graph of Figure 3a.*

It is important to note that, while the existence of a matching rate vector with positive coordinates guarantees that a greedy policy like ML is stable, there may still exist greedy policies that are unstable. Consider, for instance, the paw graph of Figure 3b and a Highest-Reward-First policy where the edges $\{1, 3\}$ and $\{2, 3\}$ offer higher rewards than $\{3, 4\}$. Even if the problem is stabilizable, this policy might select the edge $\{3, 4\}$ at a slower rate than class-4 items arrive, causing instability (see [34, Section 5]).

3.4.2 Bipartite graph (neither injective nor surjective).

Besides explaining intuitively why bipartite graphs cannot be stabilized, the following example will prepare the ground for Example 4.

► **Example 3 (Square graph).** *Figure 3c shows a square graph C_4 . This graph is not surjective because it is bipartite with parts $\{1, 4\}$ (called the outer part) and $\{2, 3\}$ (inner part). Therefore, according to Proposition 7, this graph is not stabilizable. Yet, given a vector $\lambda = (\lambda_1, \lambda_2, \lambda_3, \lambda_4)$ of arrival rates, (CE) may still have a solution with positive coordinates. This does not contradict Proposition 8, as Statement iii is equivalent to Statements i and ii only if the graph G is surjective. Assuming unit normalization, the conservation equation (CE-2) has a solution if and only if*

$$\lambda_1 + \lambda_4 = \lambda_2 + \lambda_3 = \frac{1}{2}. \quad (11)$$

If (11) is satisfied, the solutions to (CE) can be described with a parameter α as shown in Figure 3c: starting from the particular solution $\mu = (2\lambda_1\lambda_2, 2\lambda_1\lambda_3, 2\lambda_2\lambda_4, 2\lambda_3\lambda_4)$, all solutions can be generated by alternately adding and subtracting α along the (even) cycle 1-2-4-3. The positive solutions correspond to values of α such that $-2 \min(\lambda_1\lambda_2, \lambda_3\lambda_4) < \alpha < 2 \min(\lambda_1\lambda_3, \lambda_2\lambda_4)$.

To understand why the matching problem (C_4, λ) is not stabilizable even when (CE) has a solution with positive coordinates, let us focus on the system dynamics. We can use (3) to show that the difference in total queue size between the outer part $\{1, 4\}$ and the inner part $\{2, 3\}$ satisfies $Q_{t,1} + Q_{t,4} - (Q_{t,2} + Q_{t,3}) = L_{t,1} + L_{t,4} - (L_{t,2} + L_{t,3})$ for each $t \in \mathbb{N}$. Therefore, the Markov chain $(Q_{t,1} + Q_{t,4} - (Q_{t,2} + Q_{t,3}), t \in \mathbb{N})$ is a random walk on the integer number line $\{\dots, -2, -1, 0, 1, 2, \dots\}$, with transition probability proportional to $\lambda_1 + \lambda_4$ in the increasing direction and to $\lambda_2 + \lambda_3$ in the decreasing direction. If (11) is not satisfied, this random walk is transient, and the difference between the queue sizes of the two parts

grows linearly with time. On the other hand, if (11) is satisfied, then the random walk does not have this bias, but the model is still unstable because the corresponding Markov chain is null recurrent⁸.

3.4.3 Surjective-only graphs.

We finally consider a compatibility graph G that is surjective but not injective. In other words, the graph G is stabilizable and (CE) has an infinite number of solutions. The achievability of these solutions by a stable matching policy will be discussed in Section 6.

► **Example 4** (Diamond (double-fan) graph). Figure 3d shows the diamond graph D , that is, a square graph with an additional edge between nodes 2 and 3. Compared to Example 3, this additional edge makes the graph non-bipartite, and therefore surjective, so that the graph is stabilizable according to Proposition 7. For ease of computation, we assume that the vector $\lambda = (\lambda_1, \lambda_2, \lambda_3, \lambda_4)$ of arrival rates is normalized so that $\lambda_1 + \lambda_4 = \frac{1}{2}$. Under this assumption, and with $\beta = \frac{1}{2}(\lambda_2 + \lambda_3 - \lambda_1 - \lambda_4) = \frac{1}{2}(\lambda_2 + \lambda_3) - \frac{1}{4}$, $\bar{\lambda}_2 = \lambda_2 - \beta$, and $\bar{\lambda}_3 = \lambda_3 - \beta$, the general solution to (CE) can be described with a parameter α as shown in Figure 3d. After subtracting β from λ_2 and λ_3 , the solutions to (CE) for all edges but $\{2, 3\}$ are exactly the same as in the square graph of Example 3.

According to Proposition 8ii, the matching problem (D, λ) is stabilizable if and only if

$$\lambda_2 < \lambda_1 + \lambda_3 + \lambda_4, \quad \lambda_3 < \lambda_1 + \lambda_2 + \lambda_4, \quad \lambda_1 + \lambda_4 < \lambda_2 + \lambda_3, \quad (12)$$

that is, $\bar{\lambda}_3 > 0$, $\bar{\lambda}_2 > 0$, and $\beta > 0$. If these inequalities are satisfied, the positive solutions correspond to values of α such that $-2 \min(\lambda_1 \bar{\lambda}_2, \bar{\lambda}_3 \lambda_4) < \alpha < 2 \min(\lambda_1 \bar{\lambda}_3, \bar{\lambda}_2 \lambda_4)$. Intuitively, compared to the square graph, stabilizable matching problems (D, λ) have a positive difference of 2β between the arrival rates of the inner part $\{2, 3\}$ and the outer part $\{1, 4\}$. This difference is absorbed by the central edge $\{2, 3\}$, which has matching rate β . Like Example 1 and unlike Example 2, the matching model (D, λ, Φ) is stable for every greedy policy Φ provided that (12) is satisfied (as shown in Corollary 27 of Appendix B.3).

► **Example 5** (Kayak paddle graph). Figure 3e shows a kayak paddle $KP_{3,3,1}$, consisting of two triangles linked by an edge. According to Proposition 8, the matching problem (G, λ) is stabilizable if and only if there exists $\alpha > 0$ such that $(\lambda_1, \lambda_2, \lambda_3 - \alpha)$ and $(\lambda_4 - \alpha, \lambda_5, \lambda_6)$ are the arrival rate vectors of two stabilizable triangle graphs \mathcal{C}_3 . The solutions to (CE) can be described by varying α as shown in Figure 3e. Assuming the matching problem (G, λ) is stabilizable, the solutions to (CE) with positive coordinates correspond to the values of α such that

$$0 < \alpha < \min(\lambda_3 - |\lambda_2 - \lambda_1|, \lambda_4 - |\lambda_5 - \lambda_6|).$$

Intuitively, solutions with positive coordinates have a positive matching rate α along edge $\{3, 4\}$. After subtracting this rate from λ_3 and λ_4 , the subgraphs restricted to nodes 1, 2, and 3 and to nodes 4, 5, and 6, respectively, behave like the triangle of Figure 3a. Like Example 2 and unlike Examples 1 and 4, the fact that (G, λ) is stabilizable does not guarantee the stability of all greedy policies.

⁸ Some existing studies of matching in bipartite graphs solve this issue by coupling arrivals in the two components, see [2, 9, 13], or by assuming that items have a finite patience time, as in [26].

4 Matching rates in bijective graphs.

In the remainder, we consider exclusively compatibility graphs G that are stabilizable. According to Definition 2 and Proposition 7, this implies that the graph G is surjective, i.e., that each connected component of G is non-bipartite. By Proposition 6iv and Proposition 8, there are only two possible cases:

1. If $n = m$, the graph G is also bijective: each connected component of G is a unicyclic graph, and its (only) cycle is odd. The conservation equation (CE) has a unique solution given by $\mu = A^{-1}\lambda$. The matching problem (G, λ) is stabilizable if and only if all components of this solution are positive, in which case the solution provides the matching rates achievable by any stable policy.
2. If $n < m$, the graph G is surjective-only: each connected component of G is non-bipartite, and at least one of these connected components contains an even cycle or a pair of odd cycles. The conservation equation (CE) has an infinite number of solutions. The matching problem (G, λ) is stabilizable if and only if one of them has all-positive components.

Case 1 is studied in this section, while case 2 will be studied in Sections 5 and 6.

In Proposition 9 below, we give a simpler expression for the unique solution $\mu = A^{-1}\lambda$ of (CE) in terms of the arrival rate vector λ , under the assumption that the graph G is bijective. We assume without loss of generality that G is connected, as otherwise we can consider each connected component independently. Compared to the expression $\mu = A^{-1}\lambda$, the advantage of Proposition 9 is twofold: it does not require calculating a matrix inversion, and it highlights the monotonicity of the matching rates with respect to the arrival rates. This result will be illustrated in Examples 6 and 7. Note that a similar result was independently derived in [29, Theorem 4.1]. As discussed in Section 1.3, [29] considers a different setting where the only objective is to maximize reward, without enforcing stability. The analysis in [29] exploits a closed-form characterization of the solution to the conservation equation for both bijective and injective-only graphs. In contrast, Proposition 9 focuses exclusively on the bijective case, which simplifies the proof and discussion.

► **Proposition 9.** *Consider a problem (G, λ) with a compatibility graph $G = (V, E)$ that is connected and bijective, and consider an edge $k \in E$.*

- (i) *If edge k does not belong to the (unique odd) cycle of the graph G , then edge k separates the graph G into two parts, namely a tree and a unicyclic graph. If $T_k \subset V$ denotes the set of nodes that belong to the tree (including one endpoint of edge k), then the matching rate along edge k is given by*

$$\mu_k = \sum_{i \in T_k} (-1)^{d_{i,k}} \lambda_i, \quad (13)$$

where $d_{i,k}$ is the distance between node i and edge k , defined as the distance between node i and the closest endpoint of edge k .

- (ii) *If edge k belongs to the (unique odd) cycle of the graph G , then the matching rate along edge k is*

$$\mu_k = \frac{1}{2} \left(\sum_{i \in V} (-1)^{d_{i,k}} \lambda_i \right). \quad (14)$$

Proof. We first prove (13) for every edge k that does not belong to the cycle. As observed in the proposition, each edge k that does not belong to the cycle separates the graph into two parts, one of which is a tree with node set T_k ; the rooted tree associated with k is obtained

by designating the corresponding endpoint of edge k as the root. We now prove (13) by induction on the height of this rooted tree. Equation (13) is true if the height of this tree is zero. Indeed, in this case, the endpoint of edge k that belongs to the tree, say node i , has no other incident edge, so that applying (CE-1) to node i yields $\mu_k = \lambda_i$, which is consistent with (13). Now assume that the assumption is satisfied for each edge whose associated rooted tree has height at most $h - 1$ for some $h \geq 0$, and consider an edge k whose associated rooted tree has height h . By applying (CE-1) to the root i of this associated rooted tree, we obtain

$$\mu_k = \lambda_i - \sum_{\ell \in E_i \setminus \{k\}} \mu_\ell. \quad (15)$$

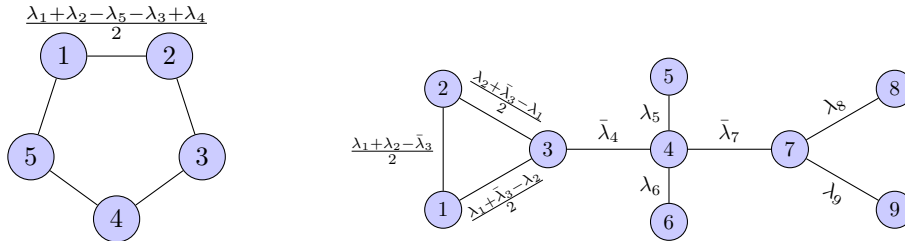
The induction hypothesis guarantees that (13) is satisfied for every $\ell \in E_i \setminus \{k\}$ (as the height of the associated rooted tree is at most $h - 1$). After injecting this observation to (15), the result for edge k follows by observing that $d_{j,k} = d_{j,\ell} + 1$ for each $\ell \in E_i \setminus \{k\}$ and $j \in T_\ell$, and that $T_k = \{i\} \cup (\bigcup_{\ell \in E_i \setminus \{k\}} T_\ell)$ (all sets being disjoint).

We now prove (14) for each edge k that belongs to the cycle. Since the graph G is unicyclic, deleting edge k from G yields a (connected) tree, and therefore a bipartite graph. We let V_+ denote set of nodes in the part that contains both endpoints of edge k (that both endpoints belong to the same part follows from the fact that the cycle is odd) and V_- the set of nodes in the other part. We obtain

$$\sum_{i \in V_+} \lambda_i - \sum_{i \in V_-} \lambda_i = \sum_{i \in V_+} \sum_{\ell \in E_i} \mu_\ell - \sum_{i \in V_-} \sum_{\ell \in E_i} \mu_\ell = 2\mu_k.$$

The first equality follows from (CE-1). The second equality holds because each edge $\ell \in E \setminus \{k\}$ has one endpoint in V_+ and another in V_- , so that μ_ℓ appears once in the first nested sum and once in the second; on the contrary, since both endpoints of edge k belong to V_+ , μ_k appears twice in the first nested sum and zero times in the second. Equation (14) follows by observing that $d_{i,k}$ is even if and only if $i \in V_+$. \blacktriangleleft

We remark that the influence of the arrival rate of a node on the matching rate along an edge only depends on the parity of the distance between the edge and the node. The actual distance does not. In particular, even in a very large (bijective) graph, a node far away from an edge has the same (although possibly reversed) impact on this edge's matching rate as an endpoint of this edge.



(a) Matching rates in the pentagon graph C_5 . Only rate $\mu_{1,2}$ is shown for ease of display. The other rates are deduced by permutation. (b) Matching rates in a “lying puppet” graph with $n = 9$ nodes and $m = 9$ edges. The differences $\bar{\lambda}_7 = \lambda_7 - \lambda_8 - \lambda_9$, $\bar{\lambda}_4 = \lambda_4 - \lambda_5 - \lambda_6 - \bar{\lambda}_7$, and $\bar{\lambda}_3 = \lambda_3 - \bar{\lambda}_4$ are the residual rates that classes 7, 4, and 3 provide to their neighbors of lower index.

■ **Figure 4** Matching rates in bijective graphs.

► **Example 6** (Cycle graph with 5 nodes). A cycle graph is the simplest bijective graph that we can consider, as it contains an odd cycle and no other edges. In the cycle graph of Figure 4a, a direct application of Proposition 9ii yields $\mu_{1,2} = \frac{1}{2}(\lambda_1 + \lambda_2 - \lambda_3 + \lambda_4 - \lambda_5)$, as we have $d_{1,\{1,2\}} = d_{2,\{1,2\}} = 0$, $d_{3,\{1,2\}} = d_{5,\{1,2\}} = 1$, and $d_{4,\{1,2\}} = 3$; regarding node 3 for instance, the endpoint of edge $\{1,2\}$ that is closest to node 3 is node 2, hence $d_{3,\{1,2\}}$ is equal to the distance between nodes 2 and 3, which is 1. Matching rates along other edges follow by symmetry. From the point of view of edge $\{1,2\}$, we can partition nodes into two sets, namely $\{1,2,4\}$ and $\{3,5\}$. The former (resp. latter) set contains nodes at an even (resp. odd) distance of edge $\{1,2\}$, and increasing the arrival rate of these nodes increases (resp. decreases) the matching rate along edge $\{1,2\}$.

► **Example 7** (Lying puppet). We now consider the graph of Figure 4b. Edges $\{1,2\}$, $\{1,3\}$, and $\{2,3\}$ belong to the cycle, and the other edges do not. According to Proposition 9, we have

$$\mu_{1,2} = \frac{\lambda_1 + \lambda_2 - \bar{\lambda}_3}{2}, \quad \mu_{1,3} = \frac{\lambda_1 - \lambda_2 + \bar{\lambda}_3}{2}, \quad \mu_{2,3} = \frac{-\lambda_1 + \lambda_2 + \bar{\lambda}_3}{2},$$

where $\bar{\lambda}_3 = \lambda_3 - \mu_{3,4}$, and

$$\begin{aligned} \mu_{4,5} &= \lambda_5, & \mu_{4,6} &= \lambda_6, & \mu_{7,8} &= \lambda_8, \\ \mu_{7,9} &= \lambda_9, & \mu_{4,7} &= \lambda_7 - \mu_{7,8} - \mu_{7,9}, & \mu_{3,4} &= \lambda_4 - \mu_{4,5} - \mu_{4,6} - \mu_{4,7}. \end{aligned}$$

This second set of equations can be obtained either by a direct application of (13) or by applying (CE-1) recursively from the leaves. Indeed, applying (CE-1) to nodes 5, 6, 8, and 9 gives directly the values of $\mu_{4,5}$, $\mu_{4,6}$, $\mu_{7,8}$, and $\mu_{7,9}$, then applying (CE-1) to node 7 gives the value of $\mu_{4,7}$, and finally applying (CE-1) to node 4 gives the value of $\mu_{3,4}$. The values of $\mu_{1,2}$, $\mu_{1,3}$, and $\mu_{2,3}$ are similar to Example 2, where the arrival rate λ_3 is again replaced with the effective arrival rate $\bar{\lambda}_3$ from the point of view of classes 1 and 2.

5 Polytope of solutions in surjective-only graphs.

Section 4 considered the case where the compatibility graph G was bijective, so that the vector of matching rates for any stable policy was the unique solution to (CE). Let us now assume that G is surjective-only, meaning that: each connected component of G contains at least one odd cycle; at least one component contains either an even cycle or a pair of odd cycles (cf. Definitions 2, 3, and 5). In this case, the solution set of (CE) is infinite. The matching rate vector under any stable policy belongs to this set and has non-negative coordinates.

To better control the matching rates, we first need to understand the structure of the solution set of (CE). To this end, Section 5.1 generalizes the earlier examples from Section 3.4 by systematically characterizing the affine space of all real-valued solutions to (CE). Following this, Section 5.2 describes the convex polytope formed by the solutions with non-negative coordinates, in particular its vertices, that is its extreme points. Whether these solutions can be achieved through a stable matching policy is explored in Section 6.

5.1 Affine space of real-valued solutions.

We first consider the set of solutions to (CE) with real-valued (positive, zero, or negative) coordinates:

$$\Pi = \{\mu \in \mathbb{R}^m : A\mu = \lambda\}. \quad (16)$$

We now delve into the properties of Π . In Section 5.1.1, we recall that Π is an affine space of dimension $d = m - n$ that can be described as a translation of the kernel of the incidence matrix A by a particular solution to (CE). Section 5.1.2 gives examples of particular solutions that can be used. Section 5.1.3 gives an algorithm to construct a basis for the kernel of the incidence matrix using a spanning tree of the graph G .

5.1.1 Edge basis, kernel basis.

The following proposition characterizes the solution set Π of (CE) using the incidence matrix of the compatibility graph. Equation (17) shows in particular that, up to translation, this solution set depends only on the structure of the compatibility graph G , while the arrival rate vector λ impacts only the translation vector.

► **Proposition 10.** *Consider a matching problem (G, λ) with a surjective-only compatibility graph G , and let A denote the incidence matrix of G . The solution set Π of (CE) is the affine space obtained by translating the kernel $\ker(A)$ of the matrix A by a particular solution μ° of (CE), that is,*

$$\Pi = \{\mu^\circ + \mu : \mu \in \ker(A)\}. \quad (17)$$

Furthermore, the vector space $\ker(A)$ and the affine space Π have dimension $d = m - n$.

Proof. That the set Π is of the form (17) is a well-known result in linear algebra. Definition 2 about surjectivity implies that the rank of A is n , and we conclude from the rank-nullity theorem that the nullity of A is $d = m - n$. The affine space Π has the same dimension according to (17). ◀

Thanks to Proposition 10, given a particular solution μ° of (CE) and a basis $\mathcal{B} = (b_1, b_2, \dots, b_d)$ of $\ker(A)$, we can rewrite the affine space Π as

$$\Pi = \{\mu^\circ + \alpha_1 b_1 + \alpha_2 b_2 + \dots + \alpha_d b_d : \alpha_1, \alpha_2, \dots, \alpha_d \in \mathbb{R}\}.$$

In fact, we can define the following affine isomorphism between the coordinate space \mathbb{R}^d and the d -dimensional affine space Π :

$$\alpha = (\alpha_1, \alpha_2, \dots, \alpha_d) \in \mathbb{R}^d \mapsto \mu = \mu^\circ + \alpha_1 b_1 + \alpha_2 b_2 + \dots + \alpha_d b_d \in \Pi. \quad (18)$$

This allows us to define two coordinate systems, *edge coordinates* and *kernel coordinates*.

► **Definition 11 (Edge basis, kernel basis).** *Consider a matching problem (G, λ) with a surjective-only graph G . Given a particular solution μ° of (CE) and a basis $\mathcal{B} = (b_1, b_2, \dots, b_d)$ of $\ker(A)$, there are two natural bases to represent vectors in Π :*

- **Edge basis:** *A vector of Π is described by its canonical coordinates $\mu = (\mu_1, \mu_2, \dots, \mu_m) \in \mathbb{R}^m$, solution to (CE), where μ_k represents a candidate matching rate along edge k , for each $k \in E$.*
- **Kernel basis:** *A vector of Π is described by its coordinates $\alpha = (\alpha_1, \alpha_2, \dots, \alpha_d) \in \mathbb{R}^d$ in the basis \mathcal{B} , where $d = m - n$ is the dimension of the affine space Π .*

If B is the $m \times d$ matrix giving the coordinates of the vectors of the basis \mathcal{B} in the edge basis, the change-of-basis formulas are as follows:

- *A vector of Π with coordinates α in kernel basis has coordinates $\mu = \mu^\circ + B\alpha$ in edge basis;*

- A vector of Π with coordinates μ in edge basis has coordinates $\alpha = B^+(\mu - \mu^\circ)$ in kernel basis, where B^+ is the pseudo-inverse⁹ (or Moore-Penrose inverse) of B .

Both bases have advantages. The edge basis, by definition, gives directly the candidate matching rates. The kernel basis allows us to work in lower dimension (d instead of m) and to ignore the conservation equation (CE), which is implicitly enforced. In the remainder, we will often use interchangeably the edge coordinates $\mu = (\mu_1, \mu_2, \dots, \mu_m)$ and the kernel coordinates $\alpha = (\alpha_1, \alpha_2, \dots, \alpha_d)$ to describe a given vector in Π . Which basis we are actually using will be made clear by our choice of letters (either μ or α).

For graphs that have a low kernel dimension d , it is convenient to mix both approaches by representing a generic vector of Π , i.e., a generic solution to (CE), in the form $\mu^\circ + \alpha_1 b_1 + \alpha_2 b_2 + \dots + \alpha_d b_d$. The solutions in Examples 4 and 5 displayed in Figures 3d and 3e follow this convention. This representation, along with the possibility to switch between edge basis and kernel basis, will be used extensively in Sections 5.2 and 6.

5.1.2 Particular solution.

We propose two ways of computing a particular solution to (CE).

Maximin solution.

A solution to the linear optimization problem (10) from Section 3.3.2 is a particular solution. Recall that such a solution allows us to determine whether the matching problem (G, λ) is stabilizable by checking if all its coordinates are positive.

Pseudoinverse.

Alternatively, a standard approach to simultaneously finding a particular solution μ° and characterizing $\ker(A)$ makes use of the pseudoinverse (or Moore-Penrose inverse) of the matrix A . Since Definition 2 on surjectivity implies that the rows of A are linearly independent, the pseudoinverse A^+ of A has the following simple form:

$$A^+ = A^\top(AA^\top)^{-1},$$

where the $n \times n$ matrix AA^\top is invertible because $\ker(AA^\top) = \ker(A^\top) = \{0\}$. We can then describe a particular solution μ^\perp and the kernel $\ker(A)$ as follows:

$$\mu^\perp = A^+\lambda, \quad \ker(A) = \{(\text{Id}_{m \times m} - A^+A)\mu : \mu \in \mathbb{R}^m\}, \quad (19)$$

where $\text{Id}_{m \times m}$ denotes the m -dimensional identity matrix. The vector μ^\perp is the *least-squares solution* to (CE), and it is orthogonal to $\ker(A)$. In general, some coordinates of this solution can be negative even if non-negative solutions exist. For example, if G is the diamond graph D of Example 4, then the matching problem (D, λ) with $\lambda = (4, 5, 2, 1)$ is stabilizable ($\mu = (\frac{7}{2}, \frac{1}{2}, 1, \frac{1}{2}, \frac{1}{2})$ is a solution to (CE) with positive coordinates), but the solution given by the pseudoinverse is $\mu^\perp = (\frac{11}{4}, \frac{5}{4}, 1, \frac{5}{4}, -\frac{1}{4})$.

Equation (19) shows that the pseudoinverse also provides an implicit characterization of $\ker(A)$, though this is not very practical, as it relies on a projection from \mathbb{R}^m to $\ker(A)$. Section 5.1.3 offers a more direct characterization by constructing a basis for $\ker(A)$ based on the structure of the compatibility graph G .

⁹ The columns of the matrix B are linearly independent because \mathcal{B} is a basis, so that the pseudo-inverse B^+ has the simple expression $B^+ = (B^\top B)^{-1}B^\top$, where the $d \times d$ matrix $B^\top B$ is invertible because $\ker(B^\top B) = \ker(B) = \{0\}$.

5.1.3 Basis of the kernel of the incidence matrix.

Recall that a vector $\mu \in \mathbb{R}^m$ belongs to $\ker(A)$ if and only if $A\mu = 0$, which reads $\sum_{k \in E_i} \mu_k = 0$ for each $i \in V$. In other words, a vector $\mu \in \mathbb{R}^m$ belongs to $\ker(A)$ if and only if, for each $i \in V$, the sum of the coordinates of μ associated with the edges that are incident to node i is zero. Using this observation, we first give examples of vectors that belong to $\ker(A)$, and then we give an algorithm that generates a basis $\mathcal{B} = (b_1, b_2, \dots, b_d)$ of $\ker(A)$.

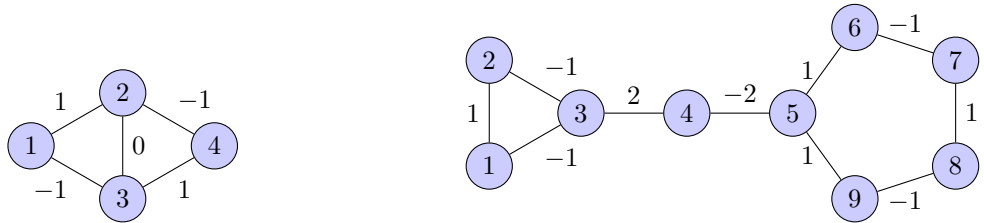
First observe that an even cycle, if it exists, is the support of a vector in $\ker(A)$: it suffices to assign alternatively the values $+1$ and -1 to the edges of this cycle and the value 0 to all other edges. In the diamond graph of Example 4, if edges are numbered in lexicographical order, then $y = (1, -1, 0, -1, 1)$ is a vector of the unidimensional kernel ($d = m - n = 1$), with support the even cycle $1-2-4-3$ (see Figure 5a). Intuitively, even cycles can be used to move weight between “odd” and “even” edges of the cycle without modifying the value of the product $A\mu$. Actually, in this example, Figure 3d shows that the only way to increase the matching rate along edges $\{1, 2\}$ and $\{3, 4\}$ is if we reduce the matching rate along edges $\{1, 3\}$ and $\{2, 4\}$, and conversely.

Apart from even cycles, other structures of interest are kayak paddles $KP_{\ell,r,p}$, made of two odd cycles (of lengths ℓ and r) connected by a path (of length p). Such graphs have a unidimensional kernel, and a base vector can be found by assigning properly the values $+1$ and -1 along the cycles and the values $+2$ and -2 along the path. Figure 5b shows such an assignment for $KP_{3,5,2}$.

Surprisingly, for any surjective graph G , one can build a basis of $\ker(A)$ using only subgraphs of G that are even cycles and kayak paddles. Algorithm 1, derived from [21], describes such a construction. Appendix C.1 gives a more detailed description of this algorithm and proof that it terminates and returns the desired result.

Figure 6 shows a possible run of Algorithm 1 for the codomino graph: in Figure 6a, a spanning tree \mathcal{T} (plain back edges) and a pivotal edge a such that $\mathcal{T} \cup \{a\}$ contains an odd cycle (dotted edge) are selected. Then, for each edge s_i that does not belong to $\mathcal{T} \cup \{a\}$, we build a base vector of the kernel space with support included into $\mathcal{T} \cup \{a, s_i\}$, after observing that $\mathcal{T} \cup \{a, s_i\}$ contains either an even cycle or a kayak paddle similar to those shown in Figure 5. Other examples are shown in Appendix C.1.

► **Remark 3.** Equation (17) implies that, given an edge $k \in E$, all solutions to (CE) have the same value along edge k if and only if edge k does not belong to the support of any basis vector. According to Algorithm 1, this is equivalent to saying that edge k belongs neither to an even cycle nor to a kayak paddle. In the diamond graph of Example 4 for instance, the edge $\{2, 3\}$ is the only one that does not belong to the even cycle $1-2-4-3$, and it is indeed the only one with a fixed rate β . In general, if an edge $k \in E$ satisfies this unicity condition,



(a) Vector of the kernel space of the diamond graph.

(b) Vector of the kernel space of the kayak paddle $KP_{3,5,2}$.

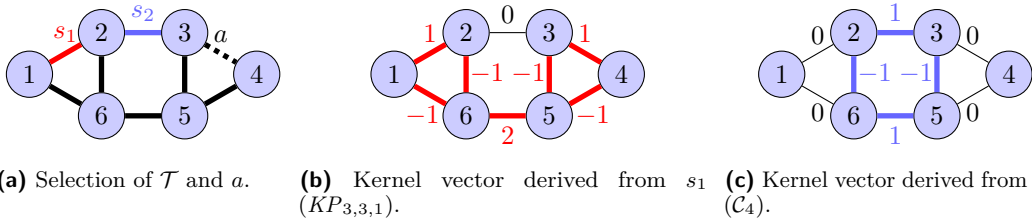
■ **Figure 5** Base vectors of kernel spaces of two toy graphs.

Data: A connected surjective-only compatibility graph $G = (V, E)$

Result: A basis \mathcal{B} of the kernel of the incidence matrix A of G

- 1 $\mathcal{T} \leftarrow$ Edges of a spanning tree of G
- 2 $a \leftarrow$ An edge in $E \setminus \mathcal{T}$ such that $\mathcal{T} \cup \{a\}$ contains an odd cycle
- 3 $\mathcal{B} \leftarrow \emptyset$
- 4 **for** $s \in E \setminus (\mathcal{T} \cup \{a\})$ **do**
- 5 Select from $\mathcal{T} \cup \{a, s\}$ an even cycle or a kayak paddle made of two odd cycles
- 6 Weight the selected edges like in Figure 5 (unselected edges have weight 0)
- 7 Add resulting kernel vector to \mathcal{B}
- 8 **return** \mathcal{B}

■ **Algorithm 1** High-level description of the construction of a basis of the kernel of the incidence matrix A of a compatibility graph G . See Appendix C.1 for detailed description.



■ **Figure 6** A possible execution of Algorithm 1 for the codomino graph.

then the matching rate along edge k in a stable matching model (G, λ, Φ) is independent of the policy Φ . Note that there is no straightforward relation between the number of edges with uniquely-defined matching rates and the dimensionality d of the affine space Π .

5.2 Polytope description

We continue to focus exclusively on matching problems (G, λ) that are stabilizable. Let $\Pi_{\geq 0}$ denote the set of solutions to (CE) that have non-negative coordinates, defined as

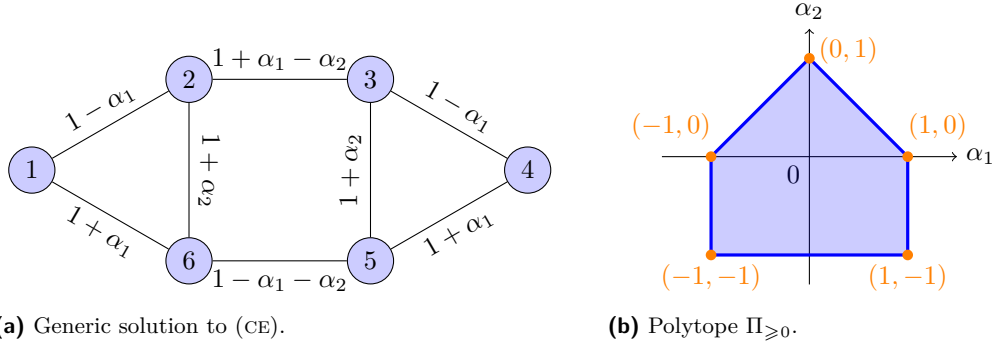
$$\Pi_{\geq 0} = \Pi \cap \mathbb{R}_{\geq 0}^m = \{\mu \in \mathbb{R}^m : A\mu = \lambda, \mu \geq 0\}. \quad (20)$$

The set $\Pi_{\geq 0}$ is a d -dimensional convex polytope in \mathbb{R}^m , as it is the intersection of a d -dimensional affine space with the positive orthant $\mathbb{R}_{\geq 0}^m$, both of which are convex. This set $\Pi_{\geq 0}$ is neither empty nor degenerated to a dimension lower than d because the matching problem (G, λ) is assumed to be stabilizable, which by Proposition 8 means that Π contains a vector with positive coordinates (i.e., in the interior of the positive orthant). It is bounded because each $\mu \in \Pi_{\geq 0}$ satisfies $0 \leq \mu_{i,j} \leq \min\{\lambda_i, \lambda_j\}$ for each $\{i, j\} \in E$.

Equation (20) describes $\Pi_{\geq 0}$ in the edge basis. As $\Pi_{\geq 0}$ is a subset of Π , we can also express its elements in the kernel basis introduced in Section 5.1.1. In the kernel basis, $\Pi_{\geq 0}$ is defined by the vectors whose coordinates belong to

$$\tilde{\Pi}_{\geq 0} = \{\alpha \in \mathbb{R}^d : \mu^\circ + \alpha_1 b_1 + \alpha_2 b_2 + \dots + \alpha_d b_d \geq 0\}. \quad (21)$$

As (20) and (21) basically represent the same polytope up to the change-of-basis formulas of Definition 11, in the remainder, we will use the same notation $\Pi_{\geq 0}$ to describe both sets; the underlying basis will be made clear by our choice of letters (as before, μ for the edge basis and α for the kernel basis).



■ **Figure 7** Matching problem (G, λ) , where G is the codomino graph and $\lambda = (2, 3, 3, 2, 3, 3) \in \mathbb{R}^6$.

5.2.1 Vertex characterization

The vertices of a convex polytope, also known as its *corners* or extreme points, are instrumental in optimization, which will be the object of Section 6. Definition 12 provides the formal definitions of vertices, along with faces and facets.

► **Definition 12** (Vertices, faces, and facets; adapted from [42]). *Let Υ denote a convex polytope of dimension $d \in \mathbb{N}_{>0}$. A (non-empty) face of Υ is a non-empty intersection of Υ with a hyperplane such that Υ is included into one of the two halfspaces defined by the hyperplane. A vertex of Υ is a face of dimension 0. Equivalently, a vector $\mu \in \Upsilon$ is a vertex of Υ if, and only if, it cannot be written as a convex combination of points in $\Upsilon \setminus \{\mu\}$. A facet of Υ is a face of dimension $d - 1$.*

Proposition 13 below gives a simple yet powerful characterization of the vertices of $\Pi_{\geq 0}$.

► **Proposition 13.** *Consider a vector $\mu \in \Pi_{\geq 0}$. Let $E^* = \{k \in E : \mu_k > 0\}$ denote the support of the vector μ and $G^* = (V, E^*)$ its support graph. The following statements are equivalent:*

- (i) *The vector μ is a vertex of $\Pi_{\geq 0}$.*
- (ii) *The graph G^* is injective.*

In particular, if μ is a vertex, we can distinguish two cases depending on the value of $|E^|$:*

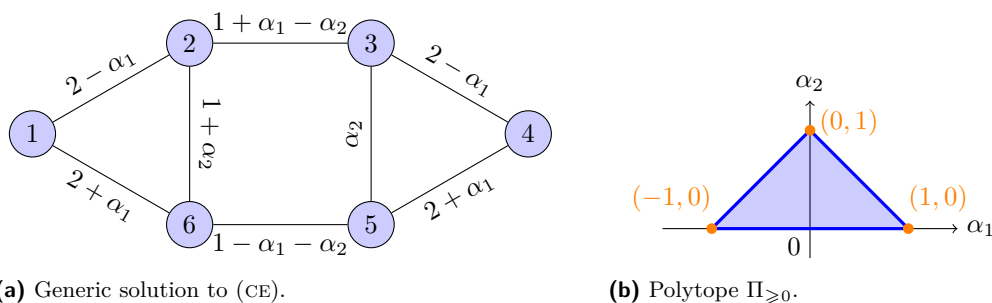
1. *If $|E^*| = n$ then G^* is bijective.*
2. *If $|E^*| < n$ then G^* is injective-only.*

With a slight abuse of notation, we say that a vertex $\mu \in \Pi_{\geq 0}$ is bijective (resp. injective-only) if the support graph G^ of μ is bijective (resp. injective-only).*

Proof (borrowed from [15]). See Appendix C.2. ◀

Figures 7 and 8 shows $\Pi_{\geq 0}$ for the codomino graph. In each figure, the left part displays a generic solution to (CE) on the graph, while the right part shows the polytope $\Pi_{\geq 0}$ and its vertices in the kernel basis. In Figure 7, $\Pi_{\geq 0}$ has five vertices. The vertex $\alpha = (0, 1)$ is bijective, as $|E^*| = |E \setminus \{\{2, 3\}, \{5, 6\}\}| = 6$. One can verify that the remaining four vertices are injective-only (as at least three edge coordinates are null). In Figure 8, $\Pi_{\geq 0}$ has three vertices, all of which are bijective.

For further details, we encourage the reader to consult Appendix C.3, which provides additional results on the relationships between vertex properties and the edge positivity inequalities from Equation (21), along with more detailed examples of polytopes and their vertices.



■ **Figure 8** Matching problem (G, λ) , where G is the codomino graph and $\lambda = (4, 4, 3, 4, 3, 4) \in \mathbb{R}^6$.

5.2.2 Probability of bijectivity

Section 6 will show that the bijectivity of vertices is central for optimizing the matching rates achieved by a stable policy. It is thus natural to wonder how frequent (or rare) bijective vertices are. Proposition 14 gives a part of the answer.

► **Proposition 14.** *Let G be a surjective-only graph, and assume that the arrival rate vector λ is drawn according to a positive Lebesgue density over Δ^{n-1} , the standard simplex of \mathbb{R}^n . Then,*

$$\mathbb{P}(\text{All vertices of } \Pi_{\geq 0} \text{ are bijective} | (G, \lambda) \text{ stabilizable}) = 1.$$

Proof. See Appendix C.4. ◀

Proposition 14 suggests that under random arrival rates, the polytope $\Pi_{\geq 0}$ is likely to have only bijective vertices. This result also holds if each coordinate of λ follows a continuous, independent probability distribution, provided that the distribution's projection on Δ^{n-1} has a positive measure and includes at least one vector λ for which (G, λ) is stabilizable. In practice, injective-only vertices may still occur, especially when the coordinates of λ display certain regularities, such as being proportional to node degrees (as in Figure 7). Moreover, injective-only vertices offer unique theoretical challenges worth exploring.

In a sense, Proposition 14 resembles the statement *with probability 1, a square matrix is invertible*: while generally true under suitable conditions, this does not imply that non-invertible matrices can be ignored.

6 Stable policy optimization.

In Section 5, we identified the matching rate vectors μ that *may* be achieved by a stable policy when the compatibility graph G is surjective-only. We now build on these results to characterize and optimize the performance *actually* achieved by stable policies.

The remainder of this section is organized as follows. Section 6.1 introduces our two main performance metrics—*delay* and *regret*. Section 6.2 recalls that optimizing linear rewards amounts to reaching a vertex of the polytope. We show that when the vertex is bijective, the optimal solution is trivially achieved by a stable policy; in contrast, if the vertex is injective-only, no stable policy can achieve it. Section 6.3 presents ϵ -filtering, a parameterized policy that can get arbitrarily close to a vertex, even when it is injective-only, with a guaranteed trade-off in performance. Finally, Section 6.4 relates our results to the most recent work on matching policy optimization.

6.1 Performance criteria.

Delay and regret are our two main performance indicators. Later in this section, we will show that these indicators can sometimes be antagonistic, meaning that certain matching problems may require trade-offs between them.

Delay.

In a stable matching model (G, λ, Φ) , the *delay* is defined as the long-term average waiting times of the items (with waiting equal to zero if the item is matched immediately upon arrival). By Little's law [31, 32], the delay $D(\Phi)$ is given by

$$D(\Phi) = \frac{\sum_{i \in V} \mathbb{E}[Q_i^\infty]}{\sum_{i \in V} \lambda_i},$$

where $Q^\infty = (Q_1^\infty, Q_2^\infty, \dots, Q_n^\infty)$ is a random vector distributed according to the stationary distribution of the Markov chain $Q = (Q_t, t \in \mathbb{N})$ under policy Φ .

Regret.

To turn a matching rate vector into a scalar performance metric, we assume that the *quality* of a matching rate vector can be evaluated by some *reward function* $f : \mu \in \mathbb{R}^m \rightarrow f(\mu) \in \mathbb{R}$. The *regret* of a policy Φ is then defined as

$$R(\Phi) = f_{\text{sup}} - f(\mu(\Phi)),$$

where $f_{\text{sup}} = \sup_{\Psi: (G, \lambda, \Psi) \text{ stable}} f(\mu(\Psi))$ is the supremum of the rewards achievable by any stable policy.

Unless otherwise stated, we consider *linear* reward functions of the form

$$f : \mu \in \mathbb{R}^m \rightarrow r^\top \mu,$$

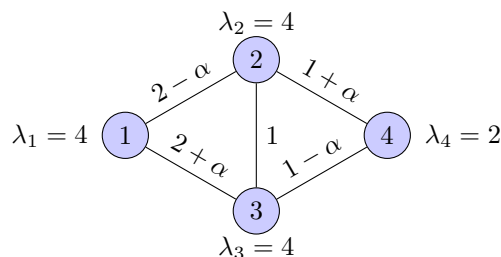
where $r = (r_1, \dots, r_m) \in \mathbb{R}^m$ is a vector of rewards associated with the edges of G .

► **Remark 4.** *In reward-driven analyses of matching models, a single regret metric is commonly used to quantify total reward loss compared to a hindsight-optimal policy. In effect, this global regret encompasses both metrics considered in our study: delay, which reflects queue sizes and is related to short-term regret ([30, Theorem 3.1 and discussion]), and regret as we define it, which coincides with the long-term regret metric in [29, Definition 2.2, Equation (9)].*

Delay fundamentally measures the stability of the system. It depends solely on the actual dynamics of the network, irrespective of the reward vector, and is therefore critical in applications such as organ donation, assemble-to-order systems, and online marketplaces, where reducing the waiting time of agents is essential. In contrast, (long-term) regret captures the cumulative shortfall in reward that is due to suboptimal matching decisions. Recognizing this essential difference, which sets up inherent tensions explored in this Section, we deliberately keep these metrics separate throughout our analysis.

As a first example, consider the matching problem in Figure 9 with associated rewards $(1, 1, 1, 0, 0)$, i.e. matches involving class 4 are not rewarded. In this case, one can verify that r is orthogonal to $\ker(A)$, so that all stable policies yield the same reward, equal to 5. On the other hand, if matching class-4 items are abandoned and a Match-the-Longest (ML) policy restricted to the triangle $\{1, 2, 3\}$ is applied, the resulting matching model has infinite delay (class 4 items never leave the system) but achieves a higher reward of 6.

This example highlights the trade-off between maximizing rewards and ensuring system stability, or more generally, keeping the expected total queue length small. The polytope approach provides a systematic way to identify and analyze such trade-offs. For readers specifically interested in reward maximization without stability guarantees, we refer to [29, 30]. In Section 8.3.2, we further demonstrate that the issue illustrated by Figure 9 actually has a simple solution within the hypergraph formalism: the addition of a discarding mono-edge.



■ **Figure 9** A matching problem where all stable policies have the same reward if $r = (1, 1, 1, 0, 0)$.

6.2 Vertex optimality

We first recall a classical result from convex optimization; namely, a convex optimization problem defined on a convex polytope always admits a vertex of the polytope as solution. More specifically, Proposition 15 states that for optimizing a linear reward function, it is sufficient to be able to reach the vertices of $\Pi_{\geq 0}$. Proposition 16 shows that, most of the time, it is also necessary.

► **Proposition 15.** *Let (G, λ) a stabilizable matching problem with a surjective-only compatibility graph G , and let $r = (r_1, \dots, r_m) \in \mathbb{R}^m$ be a vector of rewards associated with the edges of G . Consider the problem of finding a solution to (CE) with non-negative components that maximizes the reward rate $r^\top \mu$, and let $F = \{\mu \in \Pi_{\geq 0} : r^\top \mu = \max_{z \in \Pi_{\geq 0}} r^\top z\}$. Then F is a non-empty face of $\Pi_{\geq 0}$. In particular, there exists a vertex $\mu \in \Pi_{\geq 0}$ that maximizes the reward (i.e., $\mu \in F$).*

Proof. This is a standard result from convex optimization. Since $\Pi_{\geq 0}$ is closed and bounded, there is a maximum r_{\max} among the rewards associated with vectors inside $\Pi_{\geq 0}$. The set F is the intersection of the hyperplane $\{y \in \mathbb{R}^m : r^\top y = r_{\max}\}$ with $\Pi_{\geq 0}$. Therefore, it is a non-empty face of $\Pi_{\geq 0}$. That any non-empty face contains a vertex follows from the lattice structure of the faces of polytopes. ◀

► **Proposition 16.** *Let (G, λ) a stabilizable matching problem with a surjective-only compatibility graph G , and let $r \in \Delta^{m-1}$ be a vector of rewards drawn according to a positive Lebesgue density over Δ^{m-1} , the standard simplex of \mathbb{R}^m . Let $F = \{\mu \in \Pi_{\geq 0} : r^\top \mu = \max_{z \in \Pi_{\geq 0}} r^\top z\}$ be the set of the optimal matching rate vectors for r . Then, with probability 1, $F = \{\mu_r\}$, where μ_r is a vertex of $\Pi_{\geq 0}$.*

Proof. As in Proposition 15, the non-empty face F is the intersection of the hyperplane $\{y \in \mathbb{R}^m : r^\top y = \max_{z \in \Pi_{\geq 0}} r^\top z\}$ with $\Pi_{\geq 0}$. If its dimension is more than 0, it contains a 1-dimensional face of $\Pi_{\geq 0}$ (an edge). Let $b \in \mathbb{R}^m \setminus \{0\}$ be a direction of that edge. We have $b^\top r = 0$. The set $\{r \in \Delta^{m-1} : b^\top r = 0\}$, if it exists, has a dimension at most $m - 2$, so the probability that $b^\top r = 0$ is 0. As the number of edges of $\Pi_{\geq 0}$ is finite, we conclude that, with

probability 1, F does not contain any edge of $\Pi_{\geq 0}$, which means it is reduced to a single vertex. ◀

Building on Propositions 15 and 16, a natural question is: *Can a given vertex be achieved by a stable policy?* It is answered by Proposition 17.

► **Proposition 17.** *Let (G, λ) be a stabilizable matching problem with a surjective-only compatibility graph G . Let $\Pi_{\mathcal{P}}$ be the set of matching rate vectors achieved by stable policies adapted to (G, λ) , μ a vertex of the polytope $\Pi_{\geq 0}$ associated to (G, λ) , and $E^* \subset E$ the support of μ . We distinguish two cases, depending on whether μ is bijective or injective-only (cf Proposition 13).*

- (i) *If μ is bijective (meaning that $|E^*| = n$), then $\mu \in \Pi_{\mathcal{P}}$. In fact, $\mu = \mu(\Phi_{E^*})$, where Φ_{E^*} is the Match-the-Longest (ML) policy adapted to G with a filter on E^* , i.e. ML applied to the subgraph $G = (V, E^*)$.*
- (ii) *If μ is injective-only (meaning that $|E^*| < n$), then $\mu \notin \Pi_{\mathcal{P}}$: no stable policy can achieve μ .*

Proof. Let $G^* = (V, E^*)$ be the subgraph of G associated with μ , $p = |E^*|$ be the number of positive coordinates of μ , and let A^* denote the $n \times p$ incidence matrix of G^* . As G^* is injective (Proposition 13), the restriction $\tilde{\mu}$ of the vector μ to its positive coordinates is the only solution of the conservation equation $A^* \tilde{z} = \lambda$, of unknown $\tilde{z} \in \mathbb{R}^p$. We now consider the two cases separately:

- (i) If G^* is bijective, Proposition 8 implies that the matching problem (G^*, λ) is stabilized by the ML policy: G^* is surjective and $\tilde{\mu}$, the unique solution to the conservation equation $A^* z = \lambda$, has positive coordinates. The unicity of $\tilde{\mu}$ ensures that it is the matching rate achieved by that policy. To prove $\mu \in \Pi_{\mathcal{P}}$, we consider the ML policy with a filter on E^* on the matching problem (G, λ) , denoted Φ_{E^*} . Φ_{E^*} behaves exactly like the greedy ML policy on (G^*, λ) , which is stable with matching rate $\tilde{\mu}$, as we just saw. Hence, the model (G, λ, Φ_{E^*}) is stable, and its matching rate, which is $\tilde{\mu}$ on E^* and 0 elsewhere, coincides with μ . This proves that $\mu \in \Pi_{\mathcal{P}}$.
- (ii) If G^* is injective-only, Proposition 7 implies that the matching problem (G^*, λ) is not stabilizable. We prove that $\mu \notin \Pi_{\mathcal{P}}$ by contradiction. Suppose that $\mu \in \Pi_{\mathcal{P}}$, and let Φ be a stable policy on (G, λ) such that $\mu(\Phi) = \mu$. We have $\mu_k(\Phi) = 0$ for each $k \in E \setminus E^*$. But the matching rate $\mu_k(\Phi)$ along an edge $e_k \in E$ with endpoints i and j is also given by

$$\mu_k(\Phi) = \lambda_i \sum_{s, s' \in \mathcal{S}} \pi(s) \Phi(s, i, j, s') + \lambda_j \sum_{s, s' \in \mathcal{S}} \pi(s) \Phi(s, j, i, s'),$$

where π is the equilibrium distribution of the Markov chain $(S_t, t \in \mathbb{N})$. Since $\lambda_i > 0$, $\lambda_j > 0$, and the distribution π is positive on its support \mathcal{S} , it follows that the matching rate $\mu_k(\Phi)$ is zero if and only if $\Phi(s, i, j, s') = \Phi(s, j, i, s') = 0$ for each $s, s' \in \mathcal{S}$, that is, items of classes i and j are *never* matched with one another under policy Φ . In this case, Φ also defines a stable policy on the matching problem (G^*, λ) , which contradicts the instability of the matching problem (G^*, λ) . Therefore, $\mu \notin \Pi_{\mathcal{P}}$. ◀

Proposition 17 shows that optimizing linear rewards is straightforward when the vertex is bijective: we simply need to forbid the edges outside the support of the vertex. Thus, no inherent trade-off exists between reward maximization and system stability in this case. Note that a trade-off can still exist between optimizing the reward and minimizing the

expected sum of queue lengths, as forbidding a subset of edges limits the flexibility of the policy. In contrast, for the injective-only case, optimality requires a trade-off with stability. To navigate this balance between these two key objectives, we now present the so-called ϵ -filtering policies.

6.3 ϵ -filtering policy

Given an injective vertex μ of $\Pi_{\geq 0}$, we now construct a sequence of policies, that we call ϵ -filtering policies, which converge to the vertex μ as $\epsilon \downarrow 0$ while also upper bounding the expected queue lengths. Note that the distance to μ and the expected queue length are directly tied to our main scalar performance metrics, namely regret and delay¹⁰. The principle of these policies is as follows: when a new item arrives, it is marked with low probability as eligible for all matches in E . For a match outside E^* to be made, the selected pair must contain at least one such marked item. Marks allow some matches to be made outside E^* , which ensures stability. Their low probability ensures that the resulting matching rate vector is close to the target vertex μ .

Formally, given $0 < \epsilon < 1$, we label each incoming arrival by ‘-’ (marked) with probability ϵ and ‘+’ (unmarked) otherwise, independently of everything else. An item with class i and label $\ell \in \{+, -\}$ is said to have type- i^ℓ . We denote the number of items of type i^ℓ at time t by Q_{t,i^ℓ} . Thus, the total number of class- i items at time t is $Q_{t,i} = Q_{t,i^+} + Q_{t,i^-}$. Now, we restrict our attention to policies that do *not* match type- i^+ items to type- j^+ items for all $\{i, j\} \in E \setminus E^*$. The policies that follow this restriction in the original graph G can be viewed as policies operating on the augmented graph $G' = (V', E')$, with $V' = \{i^\ell : i \in V, \ell \in \{+, -\}\}$ and $E' = E^+ \cup E^\pm \cup E^-$, where

$$E^+ = \{\{i^+, j^+\} : \{i, j\} \in E^*\}, \quad E^\pm = \{\{i^+, j^-\} : \{i, j\} \in E\}, \quad E^- = \{\{i^-, j^-\} : \{i, j\} \in E\}.$$

The arrival-rate vector in the augmented graph is given by $\lambda'_\epsilon \in \mathbb{R}_{\geq 0}^{2n}$ with $\lambda'_{\epsilon, i^+} = (1 - \epsilon)\lambda_i$ and $\lambda'_{\epsilon, i^-} = \epsilon\lambda_i$ for each $i \in V$. An ϵ -filtering policy adapted to (G, λ) with filter E^* refers to a policy adapted to (G', λ'_ϵ) . For example, an ϵ -filtering ML policy adapted to (G, λ) with filter E^* mimics the ML policy adapted to (G', λ'_ϵ) . To be more precise, for a given state $q \in \mathbb{R}_{\geq 0}^{2n}$, an incoming item of type i^ℓ is matched with a type in the set $\arg \max_{j^{\tilde{\ell}} \in V' : \{i^\ell, j^{\tilde{\ell}}\} \in E'} q_{j^{\tilde{\ell}}}$.

Note that, for any greedy policy adapted to (G', λ'_ϵ) , most of the arrivals are matched using edges in E^+ , or equivalently E^* , as items are labeled ‘+’ with probability $1 - \epsilon$. As G^* is the support graph of the vertex μ , we can then show that such a policy ensures that the matching rate on $\{i^+, j^+\}$ is $O(\epsilon)$ close to $\mu_{i,j}$ for all $\{i, j\} \in E^*$. Next, to bound the expected queue lengths, we define $\delta(G, \lambda)$, the Complete Resource Pooling Gap (CRPG), as follows:

$$\delta(G, \lambda) = \min_{\mathcal{I} \in \mathbb{I}} \left\{ \sum_{j \in V(\mathcal{I})} \lambda_j - \sum_{i \in \mathcal{I}} \lambda_i \right\}. \quad (22)$$

¹⁰ Queue length and delay are related by Little’s law (see Section 2.4). Consider a stable policy Φ with matching rate $\mu(\Phi) \in \Pi_{\geq 0}$ and regret $R(\Phi)$. We have $R(\Phi) = r^\top(\mu - \mu(\Phi)) = \cos(\theta) \|r\|_2 \cdot \|\mu - \mu(\Phi)\|_2$, where θ denotes the angle between r and $\mu - \mu(\Phi)$. In particular, $R(\Phi) \leq \|r\|_2 \cdot \|\mu - \mu(\Phi)\|_2$: if you are close to μ , your regret is low. Conversely, if α denotes the minimum possible cosine between r and $\mu - \mu'$ for $\mu' \in \Pi_{\geq 0} \setminus \{\mu\}$ (i.e., $\alpha = \min_{\mu' \in \Pi_{\geq 0} \setminus \{\mu\}} \frac{r^\top(\mu - \mu')}{\|r\|_2 \cdot \|\mu - \mu'\|_2}$), we obtain $R(\Phi) \geq \alpha \|r\|_2 \cdot \|\mu - \mu(\Phi)\|_2$. Therefore, $\alpha \|r\|_2 \cdot \|\mu - \mu(\Phi)\|_2 \leq R(\Phi) \leq \|r\|_2 \cdot \|\mu - \mu(\Phi)\|_2$. Note that if μ is the unique optimal solution (as is often the case; see Proposition 16), then $\alpha > 0$: if you are far from μ , your regret is high.

Note that, by Proposition 8, the matching problem (G, λ) is stabilizable if and only if $\delta(G, \lambda) > 0$. Intuitively, $\delta(G, \lambda)$ characterizes the minimum slack between the arrival rate of any independent set \mathcal{I} and its neighbors $V(\mathcal{I})$. The arrival rate of the neighbors acts as a service rate for \mathcal{I} , and the CRPG $\delta(G, \lambda)$ is reminiscent of the heavy-traffic parameter for a single-server queue. The CRPG is also related to the General Position Gap (GPG) (KAG23), which we discuss in Section 6.4. Now, we are ready to state the main result of this section that formalizes the trade-off between reward and expected queue lengths.

► **Proposition 18.** *Consider a vertex μ of $\Pi_{\geq 0}$ with support graph $G^* = (V, E^*)$, where $E^* = \{k \in E : \mu_k > 0\}$. For each $\epsilon > 0$, let Φ_ϵ denote the ϵ -filtering ML policy adapted to (G, λ) with filter E^* . Then, $(G, \lambda, \Phi_\epsilon)$ is stable and there exist $C_1, C_2 \in \mathbb{R}_{\geq 0}$ such that, for all $\epsilon \in (0, 0.5)$, we have*

$$\|\mu(\Phi_\epsilon) - \mu\|_1 \leq C_2 \epsilon, \quad \mathbb{E} \left[\sum_{i \in V} Q_i \right] \leq \frac{C_1}{\min\{\epsilon \min_{i \in V} \lambda_i + \delta(G^*, \lambda), \delta(G, \lambda)\}}.$$

Now we present a sketch of the proof of Proposition 18 that highlights the intuition behind the result.

Sketch of the Proof (See Appendix D.2 for the complete proof). We first consider an arbitrary matching problem (G, λ) . Note that the CRPG $\delta(G, \lambda)$ is reminiscent of the heavy-traffic parameter for a single-server queue. We make this intuition rigorous by showing that, under the ML policy, we have

$$\mathbb{E} \left[\sum_{i \in V} Q_i \right] = O \left(\frac{1}{\delta(G, \lambda)} \right). \quad (23)$$

The above result holds for any matching problem (G, λ) operating under the ML policy as long as $\delta(G, \lambda) > 0$ (equivalently, whenever the matching problem (G, λ) is stabilizable). Now, we use the above result to analyze the ϵ -filtering ML policy adapted to (G, λ) with filter E^* . Recalling that this is exactly the ML policy adapted to (G', λ'_ϵ) , we turn our attention to the augmented matching problem (G', λ'_ϵ) and prove the following:

Expected queue length bound: We show that $\delta(G', \lambda'_\epsilon) = \Omega(\min\{\epsilon + \delta(G^*, \lambda), \delta(G, \lambda)\})$, and so, the expected queue length is $O(1/\min\{\epsilon + \delta(G^*, \lambda), \delta(G, \lambda)\})$ by (23). The idea is that, the edges in E^+ contribute $\delta(G^*, \lambda)$ to the CRPG because E^+ mimics E^* . In addition, matching $\Theta(\epsilon)$ arrivals using the edges in $E^\pm \cup E^-$ contributes an additional $\Omega(\epsilon)$ to the CRPG because $E^\pm \cup E^-$ mimics E . Thus, we have $\delta(G', \lambda'_\epsilon) = \Omega(\min\{\epsilon + \delta(G^*, \lambda), \delta(G, \lambda)\})$, which by (23) implies the expected queue length in steady state is at most $O(1/\min\{\epsilon + \delta(G^*, \lambda), \delta(G, \lambda)\})$. The minimum with $\delta(G, \lambda)$ is intuitive as $\delta(G, \lambda)$ is the CRPG of the original graph G , which is the best case scenario of using all edges to make matches.

Matching rate is $O(\epsilon)$ close to μ : We show that we are $O(\epsilon)$ close to the vertex μ by bounding the frequency of “bad” matches, i.e., the matches that do not correspond to the support graph G^* . Such edges are of the form (i^+, j^-) or (i^-, j^-) . We use the conservation equations given by (CE-1) along with the fact that the arrival rate to vertices $\{i^- : i \in V\}$ is $O(\epsilon)$ to prove this statement.

Combining the two steps above completes the proof of the proposition. ◀

Now we discuss the implications of the above proposition separately for bijective and injective vertex.

6.3.1 Approaching bijective vertex

When μ is a bijective vertex, we have $\delta(G^*, \lambda) > 0$ as (G^*, λ) is stabilizable (Proposition 17). Thus, we can let $\epsilon \rightarrow 0$ in Proposition 18 to conclude $\mu(\Phi_0) = \mu$ and $\mathbb{E} [\sum_{i \in V} Q_i] \leq \frac{C_1}{\delta(G^*, \lambda)} < \infty$. Thus, we observe no trade-off between stability and approaching a bijective vertex. In fact, the ϵ -filtering ML policy converges to the ML policy with filter E^* as $\epsilon \rightarrow 0$, which is stable by Proposition 17. Moreover, Proposition 18 implies that the trade-off between the reward and the expected queue lengths still exists. In particular, if $\delta(G^*, \lambda) < \delta(G, \lambda)$, then, as ϵ increases, the expected queue lengths decreases at the expense of moving ϵ away from the vertex μ . Intuitively, as ϵ increases, we more frequently use all the edges in the graph to make matches, which helps to reduce the expected queue lengths. At the same time, we more often use the edges $E \setminus E^*$ resulting in moving away from the vertex μ . Note that, the expected queue lengths is always $O(1/\delta(G, \lambda))$ for any choice of $\epsilon > 0$. This upper bound corresponds to the best case scenario (see Lemma 31 in Appendix D.2) of using all the edges in E to make matches. In Section 7, we analyze this trade-off via simulations and observe a mild trade-off (compared to the injective-only case) between the two objectives, aligning with the result of Proposition 18.

6.3.2 Approaching injective-only vertex

For an injective-only vertex, we have $\delta(G^*, \lambda) = 0$, and so, Proposition 18 implies the following:

$$\|\mu(\Phi_\epsilon) - \mu\|_1 \leq C_2 \epsilon, \quad \mathbb{E} \left[\sum_{i \in V} Q_i \right] \leq \frac{C_1}{\min \{ \epsilon \min_{i \in V} \lambda_i, \delta(G, \lambda) \}}. \quad (24)$$

As $\epsilon \rightarrow 0$, the expected queue lengths increases to infinity, substantiating that the matching problem (G^*, λ) is unstable as shown in Proposition 17. In other words, (24) establishes the trade-off between approaching an injective-only vertex and ensuring a stable system. Moreover, (24) also establishes a $O(1/\epsilon)$ expected queue length bound, while being ϵ close to the injective-only vertex μ . We now show that this trade-off is fundamental: no sequence of matching policies can converge to an injective-only vertex of $\Pi_{\geq 0}$ while maintaining finite queue lengths. The next proposition captures this fundamental trade-off.

► **Proposition 19.** *Consider an injective-only vertex μ of $\Pi_{\geq 0}$. For each policy Φ adapted to G and such that the matching model (G, λ, Φ) is stable, if $\|\mu(\Phi) - \mu\|_1 \leq \epsilon$, then $\mathbb{E} [\|Q\|_2^2] \geq \Omega(1/\epsilon)$.*

Proof. See Appendix D.1. ◀

The above proposition provides a bound of the order $\Omega(1/\epsilon)$ on the second moment of the queue length. We believe such a lower bound is not tight. In particular, we expect an $\Omega(1/\epsilon)$ lower bound on the first moment. Nonetheless, the result demonstrates that the trade-off between approaching an injective-only vertex and keeping the queue lengths small is fundamental. Remind, however, that such a trade-off does not exist for bijective vertices, as discussed before.

► **Remark 5.** *Appendix D.1 actually proves a more general version of Proposition 19 where we consider matching policies that can form at most M pairs of items at each time step, for some $M \in \mathbb{N}_{>0}$.*

6.4 Connections to the Literature

Maximizing edge-weighted rewards has recently received considerable attention in the literature ([37, 41, 23, 29, 30]). Among these, our work introduces a key novelty: leveraging stability, specifically via the conservation law (CE), to provide a global characterization of optimal solutions through the polytope. With the notable exception of [37], most related results were developed independently of our own, initiated with [14], resulting in some overlap in findings. In this section, we highlight both the originality of our approach and its strong parallels with related contributions in policy optimization. A detailed comparison of numerical performance is provided in Section 7.

6.4.1 Comparison with [29, 30, 23]

A recent line of work on stochastic matching focuses on maximizing a finite-time reward allows the system to be unstable. Specifically, this literature considers the Static Planning Problem (SPP), defined as

$$\max_{Az+s=\lambda, z, s \geq 0} r^T z, \quad (25)$$

where s represents slack variables allowing a fraction of the arrivals to be “thrown away” ([29]). A key assumption in this framework is the General Position Condition (GPC), which states that the optimal solution (z^*, s^*) to Equation (25) is unique and non-degenerate ([29, Definition 3.1]).

Our framework is closely connected to this approach through several structural and conceptual parallels:

SPP and polytope vertices. Consider a matching problem (G, λ) where G is surjective, and a reward vector r such that Equation (25) admits one unique solution (z^*, s^*) with $s^* = 0$. In this case, (G, λ) is stabilizable, z^* is a vertex of the polytope $\Pi_{\geq 0}$, and GPC is satisfied if and only if z^* is bijective.

Filtering policies. When GPC is satisfied and $s^* = 0$, our optimal policy Φ_{E^*} proposed in Proposition 17 (originally introduced in [14, Proposition 6.10]) coincides exactly with the match-the-longest-queue optimal policy in [29].

Similarity between GPG and CRPG. When GPC is satisfied, the smallest non-zero element of (z^*, s^*) , denoted $\tilde{\epsilon} > 0$ and called General Position Gap (GPG), is used to study the performance of the system¹¹. Both $\tilde{\epsilon}$ and the CRPG $\delta(G, \lambda)$ measure, in different ways, the robustness of an optimal policy. If $s^* = 0$ and if $G^* = (V, E^*)$ denotes the support graph of the vertex z^* , they are related through the following equation:

$$\delta(G, \lambda) \geq \delta(G^*, \lambda) \geq \tilde{\epsilon}. \quad (26)$$

Differences between GPG and CRPG. While $\delta(G, \lambda)$ is a function only of the graph and arrival rates, $\tilde{\epsilon}$ additionally captures the effect of rewards. The two parameters are only connected (through Equation (26)) when GPC is satisfied and $s^* = 0$. For example, consider the diamond graph from Figure 1 with arrival rate $\lambda = (1, 2, 2, 1)$. In this case, one can check that $\delta = 2 > 0$. For a reward vector $r = (1, 0, 1, 0, 1)$, the SPP solution is $z^* = (1, 0, 1, 0, 1)$, $s^* = 0$, which is degenerate (i.e. injective-only) and thus $\tilde{\epsilon}$ is not defined. Conversely, one can have $\delta \leq 0$ (meaning that the problem (G, λ) is not stabilizable) and

¹¹We denote the GPG by $\tilde{\epsilon}$ to avoid conflict with our ϵ -filtering notation.

$\tilde{\epsilon} > 0$ (meaning that arrival rates are “imbalanced”). Consider for example $V = \{1, 2\}$ and $E = \{\{1, 2\}\}$ with arrival rates $\lambda_1 > \lambda_2 > 0$. G is not stabilizable, so $\delta \leq 0$ (more precisely, $\delta = \lambda_2 - \lambda_1 < 0$). On the other hand, the SPP solution is $z^* = \lambda_2$, $s^* = (\lambda_1 - \lambda_2, 0)$, so $\tilde{\epsilon} = \min\{\lambda_1 - \lambda_2, \lambda_2\} > 0$.

Elimination of slack variables. [23] extends the model from [29, 30] by considering multi-way matching and variable arrival rates, and explores the meaning of the GPC in that context. Assuming that the solution of SPP is a bijective vertex with support E^* , they propose to use a filtering policy on E^* . Since ML can become unstable in multi-way matching, a *Sum-of-Squares* policy operating on a virtual queue is instead employed. They also eliminate the need for a slack variable s by augmenting the graph with mono-edges of zero reward for each node. Applied to our framework, or more exactly its hypergraph extension presented in Section 8.3, this trick ensures that SPP solutions always correspond to a polytope face, and GPC is satisfied if and only if that face is a bijective vertex.

Proof of Equation (26). We first prove $\delta(G, \lambda) \geq \delta(G^*, \lambda)$: as $E^* \subseteq E$, if \mathcal{I} is an independent set of G , it is also an independent set of G^* , and $V_{G^*}(\mathcal{I}) \subseteq V_G(\mathcal{I})$, where $V_G(\mathcal{I})$ (resp. $V_{G^*}(\mathcal{I})$) is the neighbor set of \mathcal{I} in G (resp. G^*). We conclude by comparing Equation (22) applied to $\delta(G, \lambda)$ and $\delta(G^*, \lambda)$.

For the second part of the inequality, by the definition of $\tilde{\epsilon}$, we have $z_k^* \geq \tilde{\epsilon}$ for all $k \in E^*$. For any $\epsilon < \tilde{\epsilon}$, we define $z(\epsilon) \in \mathbb{R}_+^{|E^*|}$ as $z_k(\epsilon) = z_k^* - \epsilon > 0$ for all $k \in E^*$, and note that $z(\epsilon)$ is the unique solution to $A^*z = \lambda(\epsilon)$ with $\lambda_j(\epsilon) = \lambda_j - \epsilon d_j^* > 0$ for all $j \in V$ (d_j^* denotes the degree of j in G^*). Thus, by Proposition 8iii, we have

$$\sum_{j \in V_{G^*}(\mathcal{I})} \lambda_j - \sum_{i \in \mathcal{I}} \lambda_i > \epsilon \left(\sum_{j \in V_{G^*}(\mathcal{I})} d_j^* - \sum_{i \in \mathcal{I}} d_i^* \right), \quad \forall \mathcal{I} \in \mathbb{I}_{G^*}.$$

As G^* is bijective, it has no bipartite component, which implies

$$\sum_{j \in V_{G^*}(\mathcal{I})} d_j^* - \sum_{i \in \mathcal{I}} d_i^* \geq 1, \quad \forall \mathcal{I} \in \mathbb{I}_{G^*}.$$

By minimizing over all independent sets $\mathcal{I} \subseteq \mathbb{I}_{G^*}$, we obtain

$$\epsilon < \min_{\mathcal{I} \in \mathbb{I}_{G^*}} \left(\sum_{j \in V_{G^*}(\mathcal{I})} \lambda_j - \sum_{i \in \mathcal{I}} \lambda_i \right) = \delta(G^*, \lambda).$$

As $\epsilon < \tilde{\epsilon}$ is arbitrary, we get

$$\tilde{\epsilon} \leq \delta(G^*, \lambda).$$

◀

6.4.2 Comparison with [37, 41]

[37] introduced the Extended Greedy Primal-Dual (EGPD) policies, a family of reward-based stable policies developed for general multi-way matching settings. Under their stabilizability condition ([37, Assumption 5]), which is equivalent to our condition Proposition 8 (i.e., surjective compatibility graph and existence of a positive solution to (CE)), they prove that EGPD asymptotically maximizes a concave function of the matching rewards while ensuring system stability. Notably, EGPD provides a canonical alternative to ϵ -filtering for approaching any vertex (bijective or injective) of $\Pi_{\geq 0}$ under the stability constraint.

Their work is instrumental for extending our results beyond simple graphs to multi-way matching (cf Section 8.3). In particular, the EGPD algorithm is built on a variant of the ML policy that operates over a virtual queue system. This variant stabilizes any stabilizable problem (G, λ) , regardless of whether G is simple or contains multi-edges, and can thus be used as a direct replacement for the standard ML policy in multi-way matching contexts. We call this policy Virtual-Queue Match-the-Longest (VQML).

While EGPD ensures stability and asymptotic optimality, it does not provide explicit guarantees on queue lengths. In contrast, our analysis establishes concrete bounds on queue lengths, a contribution beyond [37]. Specifically, we show that for bijective vertices, the reward is achieved exactly (rather than just asymptotically) with bounded queue lengths. For injective-only vertices, we prove that one can approach the optimal reward to within ϵ at a cost of $O(1/\epsilon)$ for queue lengths. These explicit bounds are, to our knowledge, new relative to [37].

More recently, [41] introduced the Constant-Regret Primal-Dual (CRPD) policy, a variant of EGPD. They use an alternative, but related, GPC based on the dual of the reward-maximization problem $\max_{z \in \Pi_{\geq 0}} r^T z$, which excludes the case where the primal solution is a unique but degenerate (i.e. injective-only) vertex. CRPD features two key modifications compared to the original policy:

1. The edge reward vector r is replaced by $r' = r - UA$, where U represents shadow prices on the nodes, chosen to solve $\min_{U, A \geq r} U\lambda$. If the arrival rates λ are not known, they are replaced by empirical estimates $\hat{\lambda}$ based on observed arrivals so far. Under the assumption that the optimal point corresponds to a bijective vertex of support $G^* = (V, E^*)$, it can be shown that $r' \leq 0$ and $r'_k = 0$ if and only if $k \in E^*$. Thus, r' acts as a re-weighted reward that penalizes edges outside the optimal support.
2. The parameter β controlling the rewards-versus-queues trade-off is not static but decreases over time, typically $\beta(t) = t^{-\alpha}$ for some $\alpha > 0$. As time progresses, this makes the algorithm increasingly reward-centric.

When α is large, CRPD effectively estimates E^* (using either λ or empirical $\hat{\lambda}$) and restricts matching exclusively to E^* . In effect, this approach is a filtering policy on E^* , as in [14, 29], with the main difference being the use of a virtual queue mechanism (VQML) instead of ML.

Interestingly, our numerical experiments revealed that using values of α outside the range originally proposed in [41] (typically $\alpha < 1/2$) produces a new family of policies parameterized by α , exhibiting a nuanced delay–regret trade-off reminiscent of EGPD or ϵ -filtering. This phenomenon and its implications will be discussed in the next section.

6.4.3 Summary

Our work main focus is the study of stable matching policies and of the space of matching rates they can achieve. It coincides with the related work on reward optimization in the important case where the GPC is satisfied, which corresponds to considering a bijective vertex. In fact, most results on bijective vertices can be translated between the two frameworks. For example, Proposition 17(i)) can be directly inferred from [29, Lemma 5.4] along with the Foster-Lyapunov theorem.

Conversely, our work explicitly handles the cases where the GPC does not hold. Specifically, when SPP has a unique, degenerate, solution (injective-only vertex), we propose a parameterized stable policy that tunes the delay-reward trade-off, while a naive application of the ML policy as in [29] would result in an unstable system. If the SPP solution is not

unique (i.e., the maximizing set is a face of $\Pi_{\geq 0}$ rather than a point), we can look for an element of the face whose support is a surjective graph, and if none exists, choose one vertex of the face (it is injective-only) and approach it, for example using ϵ -filtering.

7 Practical performance

Let (G, λ) be a stabilizable problem, $r \in \mathbb{R}^m$ a vector of rewards, and μ a vertex of $\Pi_{\geq 0}$ that optimizes the average reward $r^\top \mu$. Let $E^* \subsetneq E$ denote the edges of G that support μ . In this section, we use simulations to quantitatively evaluate the performance of various policies in achieving μ . Section 7.1 introduces the policies under consideration, while Section 7.2 outlines the simulation methodology. Sections 7.3 and 7.4 analyze the injective-only and bijective cases, respectively. Finally, Section 7.5 presents a discussion of the results.

7.1 Considered policies

For ease of display, we use Φ to denote all matching policies, distinguishing them by a subscript indicating a specific policy (e.g., ϵ for the ϵ -filtering, β for EGPD). More specifically, we consider the following policies:

Filtering ML policy The ML policy restricted to E^* , denoted by Φ_{E^*} , is stable only if μ is bijective (see Proposition 17). In this case, it is optimal (i.e., incurs no regret), and its delay serves as a benchmark for other policies. In figures of simulations, Φ_{E^*} is represented as a dotted line marking its delay.

ϵ -filtering ML policy The ϵ -filtering ML policy with filter E^* , denoted by Φ_ϵ for $\epsilon \in (0, 1)$, is stable for any $\epsilon > 0$ and converges to μ as $O(\epsilon)$, with bounded delay in $O(1/\epsilon)$ (Proposition 18).

k -filtering ML policy The k -filtering ML policy with filter E^* , denoted by Φ_k for $k \in \mathbb{N}$, applies the filter E^* depending on the state: it uses Φ_{E^*} if the length of the longest queue is less than k ; otherwise it applies the ML policy without filter.

EGPD policy The Extended Greedy Primal-Dual (EGPD) algorithm, denoted by Φ_β for $\beta \in (0, +\infty)$, is a stable policy that does not require explicit knowledge of λ or E^* . It is based on the Virtual-Queue Match-the-Longest (VQML) policy.

EGPD+ policy Denoted by Φ'_β for $\beta \in (0, +\infty)$, EGPD+ is variant of EGPD with rewards are adapted to E^* .

CRPD policy The Constant-Regret Primal-Dual (CRPD) is variant of EGPD where the rewards are adapted to E^* and parameter β decreases over time. If β goes to 0 the policy is unstable on injective-only vertices. If it decreases sufficiently fast, Φ_α behaves like a filtering policy. The CRPD policy that uses $\beta(t) = 1/t^\alpha$ for $\alpha > 0$ is denoted Φ_α .

Table 2 summarizes the selected policies and their key features.

► **Remark 6.** *Two new policies are proposed for numerical evaluation, Φ_k and Φ'_β .*

The motivation for Φ_k is our expectation that it outperforms Φ_ϵ in practical scenarios. While Φ_ϵ consistently allows a small fraction of items to be matched outside E^ , Φ_k relaxes the constraint on allowed edges only when necessary — specifically when delay is significantly affected. We provide in Appendix E.1 elements to substantiate the conjecture that Φ_k offers theoretical guarantees similar to those of Φ_ϵ .*

We also introduce Φ'_β as it is a natural intermediate between Φ_β and its variant Φ_α . The policy Φ_α incorporates two new features: adapted rewards and a decreasing β parameter. Consider the addition of only one of these features, we discard the option of using a decreasing

Policy	Base	Parameter	Principle
Φ_{E^*}	ML	N/A	No matching outside E^* .
Φ_ϵ		ϵ	Φ_{E^*} , matchings outside E^* allowed for a fraction of items.
Φ_k		k	Φ_{E^*} , matchings outside E^* allowed if the queue is large.
Φ_β	VQML	β	Edge scores based on rewards and queue sizes.
Φ'_β			Φ_β , with rewards adapted to E^* (positive on E^* , negative outside).
Φ_α		α	Φ_β , with adapted rewards $r' = r - UA$ and vanishing queue size impact.

■ **Table 2** Overview of the matching policies evaluated.

β without reward adaptation as it usually makes the system unstable. The Φ'_β policies considered here use simplified reward adaptation.

7.2 Methodology

For measuring the performance criteria presented in Section 6.1, we used the Python package *Stochastic Matching* of [35]. This package enables us to verify the stabilizability of a matching problem (G, λ) , analyze the vertices of $\Pi_{\geq 0}$, and measure the regret and delay of policies through simulations.

For policies with parameters, we selected the following ranges:

- $k \in [1, 2^{13}]$ for k -filtering;
- $\epsilon \in [10^{-3}, 1]$ for ϵ -filtering;
- $\beta \in [10^{-3}, 10]$ for EGPD and EGPD+;
- $\alpha \in [0, 1]$ for CRPD.

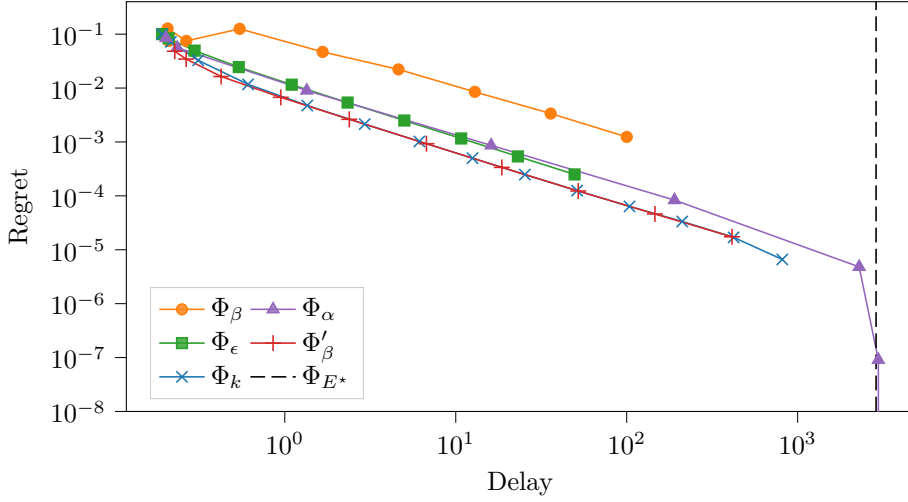
Unless stated otherwise, our evaluation protocol is as follows: after selecting G and λ , we choose a vertex μ of $\Pi_{\geq 0}$. Next we choose a reward vector r such that μ is the unique optimal, using *adversarial* rewards: weights are adjusted to make a direct identification of μ 's support E^* difficult, while ensuring that μ remains the unique optimum. Note that only Φ_β depends on the exact reward weights; other policies are directly driven by E^* . The rationale for using adversarial rewards is to highlight the performance difference between Φ_β and Φ'_β . Once parameters are fixed, all policies are evaluated by simulating $T = 10^{10}$ arrivals and measuring regret and delay.

We focus here on the diamond graph from Figure 1, but we conducted experiments on other graphs of various sizes (up to 100 nodes) with qualitatively similar results. Some of them are available in Appendix E.2 and in the package documentation¹².

7.3 Approaching an injective-only vertex

To begin, we consider degree-proportional arrival rates, i.e., $\lambda = (2, 3, 3, 2)$. Our objective is to approach the injective-only vertex $\mu = (2, 0, 1, 0, 2)$, which effectively disables edges $\{1, 3\}$ and $\{2, 4\}$. We select the adversarial reward vector $r = (1, 2.9, 1, -1, 1)$, assigning the highest reward to edge $\{1, 3\}$. The results are presented in Figure 10.

¹²https://balouf.github.io/stochastic_matching/companion/simulations.html#Vertices-of-simple-graphs



■ **Figure 10** G : diamond graph; $\lambda = (2, 3, 3, 2)$; $\mu = (2, 0, 1, 0, 2)$; $r = (1, 2.9, 1, -1, 1)$.

First, we emphasize that any simulation, regardless of length, provides only a finite-horizon estimate of a policy’s long-term behavior. In particular, it will always yield a finite delay, even for policies proven to be unstable here, like Φ_{E^*} and Φ_α for $\alpha > 0$. This explains why Φ_α , when converging to Φ_{E^*} , appears to achieve zero regret with constant delay. In reality, the delay is unbounded and grows like \sqrt{T}^{13} . Therefore, values observed in the far right portion of the figure must be interpreted with caution, as they reflect finite-horizon effects rather than the true long-term behavior.

With that in mind, one can observe that all policies clearly demonstrate the $(\epsilon, 1/\epsilon)$ trade-off between delay and regret, as predicted by Proposition 18.

Specifically, Φ_k and Φ'_β exhibit very similar performance, while Φ_ϵ and Φ_α perform slightly worse. Due to the adversarial reward vector, the performance of Φ_β degrades by more than one order of magnitude, incurring more than ten times the regret of Φ_k for the same target delay, or equivalently, requiring a delay ten times longer to achieve the same regret.

7.4 Approaching a bijective vertex

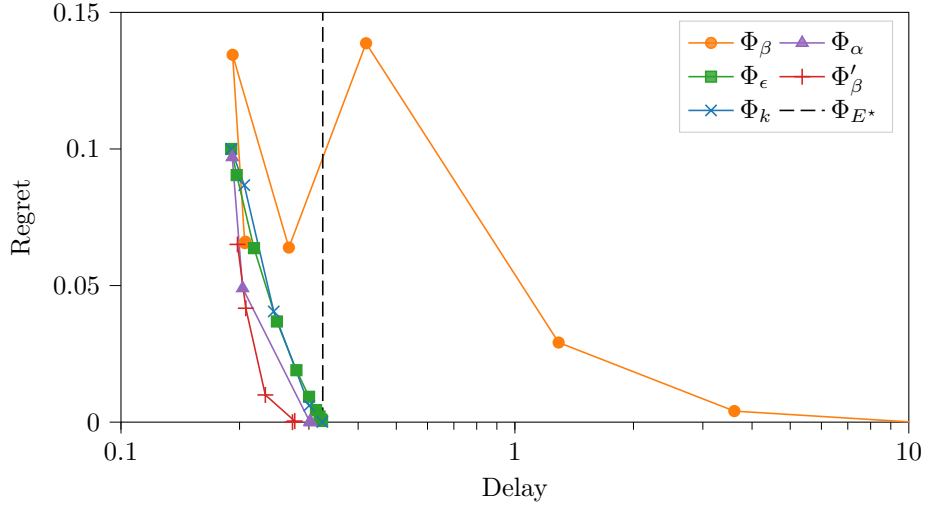
Still focusing on the diamond graph, we now set $\lambda = (4, 4, 4, 2)$ and examine the vertex $\mu = (1, 3, 1, 2, 0)$, which disables edge $\{3, 4\}$. We select the adversarial reward vector $r = (-1, 1, 1, 1, 2.9)$, assigning the highest reward to edge $\{3, 4\}$ and a negative reward to the positive edge $\{1, 2\}$.

Since μ is bijective, we know from Proposition 17 that Φ_{E^*} is optimal in terms of regret. However, it is interesting to assess how vertex-approaching policies perform in this case. The results are presented in Figure 11.

All policies continue to exhibit a trade-off between regret and delay. Except for Φ_β , they all achieve delays lower than that Φ_{E^*} at the cost of a positive regret.

Specifically, Φ_ϵ and Φ_k perform very similarly and converge precisely to Φ_{E^*} , as predicted by their design. Φ_α and Φ'_β outperform the others; notably, they achieve zero regret with

¹³https://balouf.github.io/stochastic_matching/companion/simulations.html#Drift-of-optimal-solution-for-injective-only-vertices



■ **Figure 11** G : diamond graph; $\lambda = (4, 4, 4, 2)$; $\mu = (1, 3, 1, 2, 0)$; $r = (-1, 1, 1, 1, 2.9)$.

delays lower than Φ_{E^*} . We interpret this as resulting from their base policy, VQML, which may perform better than Φ_{E^*} 's base policy ML in the model where edge $\{3, 4\}$ is removed.

As before, Φ_β performs poorly due to the adversarial reward vector. While its regret can approach zero arbitrarily closely, as predicted by theory, this comes at the cost of a large delay that appears to grow asymptotically like $1/\beta$, effectively unbounded.

7.5 Discussion

The observations made in Sections 7.3 and 7.4 are not specific to the diamond graph. Other studied examples show qualitatively similar behavior. Our main conclusion is that no single policy consistently outperforms all others across all scenarios. Nevertheless, some patterns emerge:

- Pure filtering should be avoided if the vertex is injective-only. For bijective vertices, if minimizing regret is the absolute priority, pure filtering is a natural choice; otherwise, more flexible policies may perform better.
- ϵ -filtering ML is a robust vertex-approaching method effective on both injective-only and bijective vertices. It offers theoretical guarantees but can often be outperformed numerically by state-dependent policies.
- EGPD does not require knowledge of the vertex μ corresponding to the arrival rate and reward vectors. However, it performs poorly when the reward vector is misaligned with the support E^* of μ .
- EGPD+ addresses the drawbacks of EGPD by aligning the rewards with E^* . Note that this requires knowledge or estimation of the arrival rate λ .
- Although theoretical guarantees for k -filtering ML remain conjectural, simulations suggest it strikes a favorable balance between performance and robustness. Alongside EGPD+, it is our recommended policy when arrival rates are known or stable enough.
- CRPD delivers good performance and theoretical guarantees; however, when compared with Φ'_β , the practical benefit of introducing a time-decreasing parameter remains unclear.

in many scenarios¹⁴.

8 Extensions of our results

This article focuses on the matching rates of stable policies in the context of two-way matching, emphasizing linear optimization. However, our approach extends beyond this setting. In this section, we present several extensions: Section 8.1 addresses the case of non-linear reward functions; Section 8.2 explores the (im)possibility of achieving a vertex using only stable *greedy* policies; and Section 8.3 briefly outlines the adaptation of our work to hypergraphs, enabling the possibility of discarding items and multi-way matching.

8.1 Non-linear optimization

Proposition 17 introduced $\Pi_{\mathcal{P}}$, the set of matching rates achievable by a stable policy, and compared it to the vertices of $\Pi_{\geq 0}$, which contain the optimal solutions for linear reward optimization.

However, when considering a non-linear reward function $r(\mu)$, an optimal solution can lie anywhere within $\Pi_{\geq 0}$. For example, one might want a stable policy to be as close as possible to a target matching rate vector μ_0 and use the reward function $r(\mu) = -\|\mu - \mu_0\|_1$. Ideally, we would like $\Pi_{\mathcal{P}}$ to be as close to $\Pi_{\geq 0}$ as possible. The following proposition demonstrates that this is indeed the case.

► **Proposition 20.** *Let $\Pi_{\mathcal{P}}$ be the set of matching rate vectors achievable by a stable policy adapted to a given stabilizable matching problem (G, λ) . Then, $\Pi_{\mathcal{P}}$ is convex. Furthermore, any positive solution to the conservation equation (CE) can be obtained by a stable policy, that is,*

$$\Pi_{>0} \subseteq \Pi_{\mathcal{P}} \subseteq \Pi_{\geq 0}, \text{ where } \Pi_{>0} = \{\mu \in \mathbb{R}_{>0}^m : A\mu = \lambda\}.$$

In the particular case where all vertices of $\Pi_{\geq 0}$ are bijective, we have $\Pi_{\mathcal{P}} = \Pi_{\geq 0}$.

Proof. Convexity of $\Pi_{\mathcal{P}}$ is shown in Appendix F.1. The inclusion $\Pi_{\mathcal{P}} \subseteq \Pi_{\geq 0}$ follows from the discussion in Section 3.1: any stable policy yields a matching rate vector that satisfies (CE) with non-negative coordinates. We now prove that $\Pi_{>0} \subseteq \Pi_{\mathcal{P}}$. Since $\Pi_{\mathcal{P}}$ is convex, its closure is also convex. By Proposition 18 and the convexity of $\Pi_{\mathcal{P}}$, the closure of $\Pi_{\mathcal{P}}$ contains all vertices of $\Pi_{\geq 0}$, and, by convexity, encompasses $\Pi_{\geq 0}$ as well. As a result, by virtue of convexity, $\Pi_{\mathcal{P}}$ contains the interior¹⁵ of its own closure, which includes $\Pi_{>0}$, the interior of $\Pi_{\geq 0}$. In the particular case where all vertices are bijective, they all belong to $\Pi_{\mathcal{P}}$, leading to the equality $\Pi_{\mathcal{P}} = \Pi_{\geq 0}$. ◀

Proposition 20 is essentially an existence result, as the policy constructed to show convexity of $\Pi_{\mathcal{P}}$ in Appendix F.1 may be both hard to implement (as it requires computing the mean return time to a particular state of two Markov chains) and undesirable in practice (as the matching rate will have high variance). An interesting follow-up question is thus to explore practical methods to reach an arbitrary vector of $\Pi_{\mathcal{P}}$.

¹⁴Counter-examples exist, cf https://balouf.github.io/stochastic_matching/companion/simulations.html#Larger-graphs

¹⁵Here, the notion of *interior* refers to the canonical \mathbb{R}^d topology of the d -dimensional affine space of solutions to the conservation equation.

8.2 Greedy policies

Greedy policies are appealing candidates for controlling matching problems, and they have been extensively studied in the literature. Some, such as ML and FCFM, are direct to implement and stabilize all stabilizable matching problems. Moreover, greedy policies may be more socially acceptable, particularly in scenarios involving human participants. However, as we will demonstrate in this section, greedy policies are generally not well-suited for optimizing matching rate vectors. They can never achieve a vertex, and examples suggest that in most cases, they are unable to even approach one.

In what follows, consider a stabilizable problem (G, λ) with a surjective-only graph G , and let Π_G denote the set of matching rate vectors achieved by stable *greedy* policies adapted to the problem (G, λ) . As in Proposition 20, let $\Pi_{>0} = \{\mu \in \mathbb{R}_{>0}^m : A\mu = \lambda\}$ denote the set of the positive solutions to the conservation equation (CE), i.e., those with positive coordinates, which form the interior of $\Pi_{\geq 0}$.

8.2.1 Convexity and impossibility result

We first demonstrate that, unlike stable policies in general, stable *greedy* policies can never reach the boundary of the convex polytope $\Pi_{\geq 0}$, irrespective of whether the vertices of $\Pi_{\geq 0}$ are bijective or injective-only.

► **Proposition 21.** *Let Π_G denote the set of matching rate vectors achievable by a stable greedy policy for a given stabilizable matching problem (G, λ) , where G is surjective-only. Then, Π_G forms a non-empty convex subset of $\Pi_{>0}$.*

Proof. The set Π_G is non-empty because, as recalled in Section 2.4, the greedy ML and FCFM policies are stable. Its convexity is shown in Appendix F.1. We now prove that $\Pi_G \subseteq \Pi_{>0}$. Consider a stable greedy policy Φ and let μ denote the matching-rate vector in the model (G, λ, Φ) . Consider an edge $e_k = \{i, j\}$. Since the policy Φ is greedy, two items of classes i and j are always matched if the following sequence of events occurs: the system is in the empty state \emptyset , then a class- i item arrives, and then a class- j item arrives. Let p_\emptyset denote the stationary probability that the model (G, λ, Φ) is in the empty state¹⁶ \emptyset . We know that $p_\emptyset > 0$ because the model is stable, and the previous remark implies that $\mu_k \geq p_\emptyset \lambda_i \lambda_j / (\sum_{\ell \in V} \lambda_\ell)^2 > 0$. Since this is true for each edge $e_k \in E$ and each $\mu \in \Pi_G$, we conclude that $\Pi_G \subseteq \Pi_{>0}$. ◀

Proposition 21 has the following consequence regarding greedy policies and linear optimization.

► **Corollary 22.** *Let $r_{\max} = \max_{\mu \in \Pi_{\geq 0}} r^\top \mu$ be the optimal reward in the linear optimization problem defined by the reward vector r in Proposition 15. One of the following must hold:*

- (i) *All stable policies (greedy or not) are optimal, i.e., $r^\top \mu = r_{\max}$ for each $\mu \in \Pi_G$.*
- (ii) *All stable greedy policies are suboptimal, i.e., $r^\top \mu < r_{\max}$ for each $\mu \in \Pi_G$.*

Proof. We know from Proposition 15 that the set of vectors $\mu \in \Pi_{\geq 0}$ maximizing the reward forms a non-empty face F of $\Pi_{\geq 0}$. If $F = \Pi_{\geq 0}$, then we are in case (i), meaning

¹⁶As mentioned in Appendix A, we assume that there exists a unique state $s \in \mathcal{S}$ such that $|s| = 0$. This state is called the *empty state* and denoted by \emptyset . This assumption guarantees that the intuitive notion of system stability is captured by the positive recurrence of the Markov chain describing the evolution of the system state. If the policy is queue-based (so that $\mathcal{S} = \mathcal{Q}$), this empty state is simply the n -dimensional zero vector.

all $\mu \in \Pi_{\mathcal{P}} \subset \Pi_{\geq 0}$ are optimal. This occurs when the vector r is orthogonal to $\ker(A)$ ¹⁷. Otherwise, by the lattice structure of polytope faces, F is contained in a facet of $\Pi_{\geq 0}$, implying there is at least one edge $k \in E$ where the k -th coordinate of all vectors in F is zero. As stable greedy policies produce matching rate vectors with all positive coordinates, no greedy policy can be optimal in this case. ◀

8.2.2 Achievability results

We now explore the relationship $\Pi_G \subseteq \Pi_{>0}$ through several examples. In particular, Propositions 23 and 24 illustrate situations where Π_G is a strict subset of $\Pi_{>0}$, suggesting that the greedy constraint imposes significant limitations on the set of achievable matching rates. However, this is not universally true, as Conjecture 25 provides a (carefully chosen) example where $\Pi_G = \Pi_{>0}$.

► **Proposition 23.** *Let (K_n, λ) be a stabilizable matching problem, where K_n is the complete graph with $n \geq 3$ nodes. All greedy policies adapted to K_n are stable and yield the same matching-rate vector, denoted μ_G . In particular, we have $\Pi_G = \{\mu_G\} \subsetneq \Pi_{>0}$ whenever $n \geq 4$.*

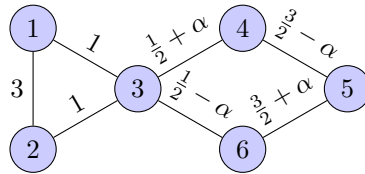
Proof. See Appendix F.2.1 for proof and discussion. ◀

► **Proposition 24.** *Let (D, λ) be a stabilizable matching problem, where D is the diamond graph from Figure 3d. All greedy policies adapted to D are stable. In kernel coordinates, there exist $\alpha_- < \alpha_+$ such that $\Pi_G = [\alpha_-, \alpha_+] \subsetneq \Pi_{>0}$.*

Proof. See Appendix F.2.2 for proof, discussion, and numerical results. ◀

► **Conjecture 25.** *Let (G, λ) be the stabilizable matching problem depicted in Figure 12 (the Fish matching problem). For this problem, we have $\Pi_G = (-1/2, 1/2) = \Pi_{>0}$.*

Proof. See Appendix F.2.3 for sketch of proof, discussion, and numerical results. ◀

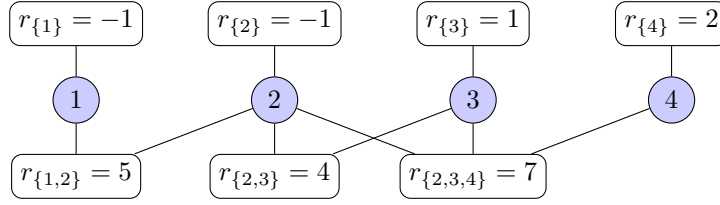


■ **Figure 12** Generic solution to (CE) in the Fish matching ($\lambda = (4, 4, 3, 2, 3, 2)$).

8.3 Hypergraphs

A hypergraph $G = (V, E)$ consists of a set of nodes V and a set of hyperedges E , where each hyperedge is a multi-subset (i.e. potentially allowing multiplicities) of V of arbitrary size, starting from one. This contrasts with simple graphs, where each edge is a pair of distinct nodes. Stochastic matching models extend naturally to hypergraphs as follows: when a

¹⁷An example of this is shown in Figure 9, or when all coordinates of r are equal, making all edges equivalent.



■ **Figure 13** Hypergraph example studied in [37]. The rewards associated with each hyperedge are indicated.

matching decision is made, items corresponding to the selected hyperedge are removed from the system.

A complete, formal, extension of our results to hypergraphs is beyond the scope of this paper, but our approach adapts with limited changes. The key elements are:

- The incidence matrix A for hypergraphs is defined analogously to simple graphs but can be any matrix in $\mathbb{N}^{n \times m}$, with $a_{i,k}$ indicating whether node i belongs to hyperedge k , counting multiplicities in case of repeated nodes. The connection between properties of A and stability remains, particularly Proposition 7. However, some graph-specific results, such as the connection between instability and bipartite structures or the description of $\ker(A)$ in terms of cycles and kayak paddles, do not easily generalize to hypergraphs.
- Proposition 8 remains valid. In particular, the stability condition iii can still be verified by computing $A^{-1}\lambda$ for bijective hypergraphs, or solving (10) for surjective-only hypergraphs.
- Some policies from Sections 6 and 7 that optimize the matching-rate vector, specifically Φ_{E^*} , Φ_ϵ , and Φ_k , rely on the maximal stability property of the ML policy in simple graphs. This property does not extend to hypergraphs¹⁸, necessitating alternative maximally stable policies. We chose VQML, the rewardless version of EGPD, although other choices are possible.

We illustrate these adaptations through numerical results on example cases. We apply the policies from Table 2, adapted to hypergraphs by always using VQML as the base policy, following the simulation methodology described in Section 7.2. These examples highlight the performance of the proposed policies in the hypergraph context.

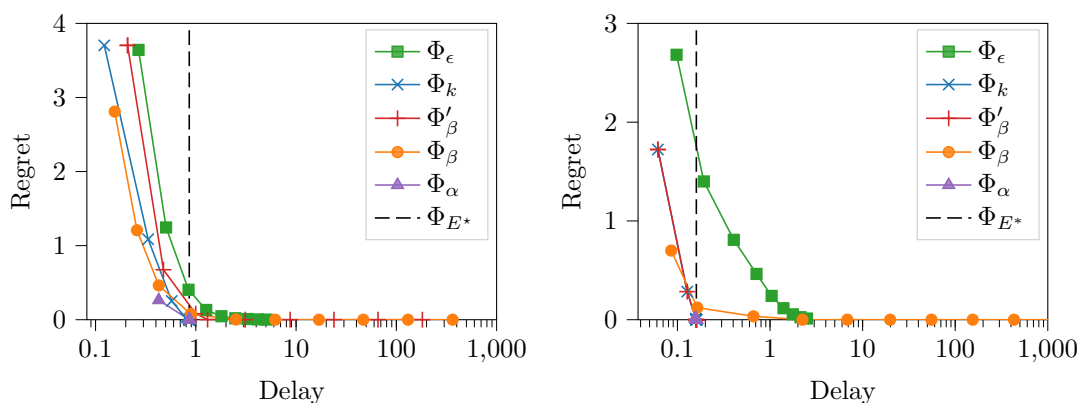
8.3.1 Original example from [37]

[37] considered the hypergraph shown in Figure 13 for the numerical evaluation of their policy Φ_β . This hypergraph consists of four nodes, each with a mono-edge (i.e., a 1-edge), two regular 2-edges, and one 3-edge. It can be verified that this hypergraph is surjective-only, and the solution space to (CE) has dimension 3.

The authors consider a single reward vector (shown in Figure 13) and two distinct arrival-rate vectors: $\lambda = (1.2, 1.5, 2, 0.8)$ and $\lambda = (1.8, 0.8, 1.4, 1)$. Both vectors are stabilizable. The first generates a polytope $\Pi_{\geq 0}$ with 8 vertices, while the second generates a polytope with 4 vertices. All vertices are bijective, indicating that the reward can always be optimized using the filtering policy Φ_{E^*} . The performance of the considered policies under these conditions is shown in Figure 14. The key observations are:

- All parameterized policies can minimize the regret when their parameter is aggressive enough.

¹⁸In hypergraphs, stabilizable matching problems exist where no greedy policy is stable.



(a) $\lambda = (1.2, 1.5, 2, 0.8)$, $\mu = (0, 0, 1.7, 0.5, 1.2, 0, 0.3)$ (b) $\lambda = (1.8, 0.8, 1.4, 1)$, $\mu = (1, 0, 1.4, 1, 0.8, 0, 0)$

■ **Figure 14** Reaching the vertex corresponding to $r = (-1, -1, 1, 2, 5, 4, 7)$ in the hypergraph from Figure 13.

- As noted in [37], the delay experienced by Φ_β continues to grow as $1/\beta$, even after the regret reaches zero.
- Φ'_β only keeps a bounded delay in the second setting.
- The delay experienced by Φ_ϵ appears bounded but remains higher than that of Φ_{E^*} . This contrasts with the expected convergence to Φ_{E^*} , as observed in Section 7.4. We attribute this to unforeseen interactions between the expanded graph, on which Φ_ϵ dispatches arrivals unevenly, and the virtual queue reservation mechanism.
- For both Φ_k and Φ_α , the delay converges to that of Φ_{E^*} , which is with Section 7.4. Adjusting the parameters allow to control the delay/regret trade-off.

8.3.2 Augmenting stability

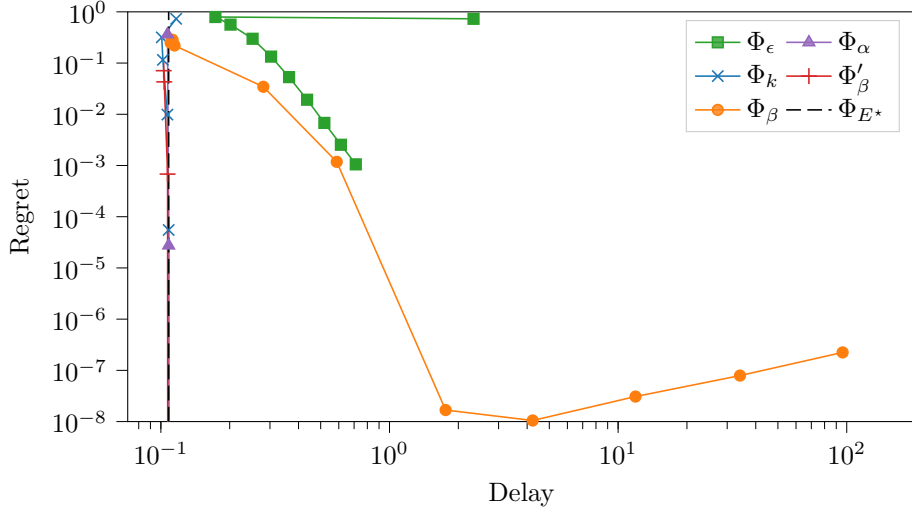
Figure 9 in Section 6 presented a toy example illustrating that the optimal stable policy does not always achieve the optimal achievable reward. This demonstrates that imposing stability can limit performance, as the optimal matching-rate may require discarding a fraction of arrivals and thus be outside of $\Pi_{\geq 0}$. Following [23], this issue can be addressed by augmenting each under-demanded node with a zero-reward mono-edge, thereby allowing the discard of excess items.

In the toy example of Figure 9, adding a mono-edge to class 4 creates a third vertex in the polytope, which zeros the edges $\{2, 4\}$ and $\{3, 4\}$. This vertex is bijective, making Φ_{E^*} both optimal and stable.

The performance of the considered policies under this setting is shown in Figure 15. The most noteworthy findings are:

- Φ_β and Φ_ϵ can have a low regret, but at the cost of high delay.
- All other parameterized policies rapidly converge to Φ_{E^*} , with negligible regret/delay trade-off. We believe this is because the target vertex (a perfectly balanced triangle plus one discarded node) already yields low delay, so relaxing the constraints on edges $\{2, 4\}$ and $\{3, 4\}$ has minimal impact.

The case analyzed above is illustrative, but the role of mono-edges extends beyond this specific example. In general, adding mono-edges to a graph increases the dimensionality of the polytope, so that the original feasible region becomes a face of the enlarged polytope.



■ **Figure 15** G : diamond augmented with $\{4\}$; $\lambda = (4, 4, 4, 2)$; $\mu = (2, 2, 2, 0, 0, 2)$; $r = (-1, 1, 1, 1, 3, 2)$ (adversarial).

For instance, augmenting the toy example transforms a segment into a triangle, with one side representing the original feasible set.

Providing one mono-edge per node guarantees that any problem (G, λ) is trivially stabilizable, for example by discarding all incoming items. This stabilization mechanism, however, only remedies instability that results from an imbalance leading to systemic over-demanded nodes; vertices that are injective-only remain injective-only after the addition of mono-edges.

It is crucial that mono-edges carry non-negative rewards to ensure that $\Pi_{\geq 0}$ contains the optimal solution. If mono-edges have negative rewards, the discrepancy between stability and optimality may persist. For example, in Figure 14b, the best stable solutions are actually suboptimal due to the discarding of class 1 items with negative reward. In that case, absolute optimality requires the queue for class 1 to grow without bound.

► **Remark 7.** *If G is a simple graph augmented with mono-edges, Definitions 2–5 remain entirely valid, under the assumption that one mono-edge is an odd cycle. In particular, if each connected component of G contains at least one mono-edge, G is surjective.*

Other definitions and formulas can be likewise adapted. For example, Equation (22), which defines $\delta(G, \lambda)$, works with the convention that a node with a mono-edge does not belong to any independent set. As illustration, one can check that the problem studied in Figure 15 verifies $\delta(G, \lambda) = \delta(G^, \lambda) = 2$. A graph with mono-edges everywhere translates to $\delta(G, \lambda) = +\infty$ (all incoming items can be processed instantly).*

References

- 1 Ivo Adan, Ana Bušić, Jean Mairesse, and Gideon Weiss. Reversibility and Further Properties of FCFS Infinite Bipartite Matching. *Mathematics of Operations Research*, 43(2):598–621, December 2017. Publisher: INFORMS. doi:10.1287/moor.2017.0874.
- 2 Ivo Adan and Gideon Weiss. Exact FCFS matching rates for two infinite multitype sequences. *Operations Research*, 60(2):475–489, 04 2012. Publisher: INFORMS. doi:10.1287/opre.1110.1027.
- 3 Angelos Aveklouris, Levi DeValve, Maximiliano Stock, and Amy Ward. Matching impatient and heterogeneous demand and supply. *Operations Research*, 73(3):1637–1658, may 2025. Publisher: INFORMS. doi:10.1287/opre.2022.0005.
- 4 Egon Balas and William R. Pulleyblank. The perfectly matchable subgraph polytope of a bipartite graph. *Networks*, 13(4):495–516, 1983. doi:10.1002/net.3230130405.
- 5 Jocelyn Begeot, Irène Marcovici, and Pascal Moyal. Stability regions of systems with compatibilities and ubiquitous measures on graphs. *Queueing Systems*, 103(3):275–312, 2023. doi:10.1007/s11134-023-09872-0.
- 6 Jocelyn Begeot, Irène Marcovici, Pascal Moyal, and Youssef Rahme. A general stochastic matching model on multigraphs. *arXiv:2011.05169 [math]*, November 2020. arXiv: 2011.05169. URL: <http://arxiv.org/abs/2011.05169>.
- 7 Péter Biró, Joris van de Klundert, David Manlove, William Pettersson, Tommy Andersson, Lisa Burnapp, Pavel Chromy, Pablo Delgado, Piotr Dworzak, Bernadette Haase, Aline Hemke, Rachel Johnson, Xenia Klimentova, Dirk Kuypers, Alessandro Nanni Costa, Bart Smeulders, Frits Spieksma, María O. Valentín, and Ana Viana. Modelling and optimisation in european kidney exchange programmes. *European Journal of Operational Research*, 291(2):447–456, 2021. doi:10.1016/j.ejor.2019.09.006.
- 8 Pierre Bremaud. *Markov Chains: Gibbs Fields, Monte Carlo Simulation, and Queues*. Texts in Applied Mathematics. Springer-Verlag, New York, 1999. URL: <https://www.springer.com/gp/book/9780387985091>, doi:10.1007/978-1-4757-3124-8.
- 9 Ana Bušić, Varun Gupta, and Jean Mairesse. Stability of the Bipartite Matching Model. *Advances in Applied Probability*, 45(2):351–378, June 2013. Publisher: Cambridge University Press. doi:10.1239/aap/1370870122.
- 10 Ana Bušić and Sean Meyn. Approximate optimality with bounded regret in dynamic matching models. *arXiv:1411.1044 [cs, math]*, June 2016. arXiv: 1411.1044. URL: <http://arxiv.org/abs/1411.1044>.
- 11 Arnaud Cadas, Ana Bušić, and Josu Doncel. Optimal control of dynamic bipartite matching models. In *Proceedings of the 12th EAI International Conference on Performance Evaluation Methodologies and Tools, VALUETOOLS 2019, Universitat de les Illes Balears, Palma de Mallorca, Spain, March 12-15, 2019*, pages 39–46. ACM, 2019. doi:10.1145/3306309.3306317.
- 12 Arnaud Cadas, Josu Doncel, Jean-Michel Fourneau, and Ana Bušić. Flexibility can hurt dynamic matching system performance. *arXiv:2009.10009 [cs, math]*, September 2020. arXiv: 2009.10009. URL: <http://arxiv.org/abs/2009.10009>.
- 13 René Caldentey, Edward H. Kaplan, and Gideon Weiss. FCFS infinite bipartite matching of servers and customers. *Advances in Applied Probability*, 41(3):695–730, 09 2009. Publisher: Cambridge University Press. doi:10.1239/aap/1253281061.
- 14 Céline Comte, Fabien Mathieu, and Ana Bušić. Stochastic dynamic matching: A mixed graph-theory and linear-algebra approach. working paper or preprint, December 2021. URL: <https://hal.science/hal-03502084v1>.
- 15 Céline Comte and Jaap Storm. Global and local stability in parallel service systems with redundancy, 2022. Unpublished.
- 16 Céline Comte. Stochastic non-bipartite matching models and order-independent loss queues. *Stochastic Models*, 38(1):1–36, January 2022. doi:10.1080/15326349.2021.1962352.
- 17 Céline Comte and Jan-Pieter Dorsman. Performance Evaluation of Stochastic Bipartite Matching Models. In *Performance Engineering and Stochastic Modeling*, Lecture Notes in

- Computer Science, pages 425–440. Springer International Publishing, 2021. doi:10.1007/978-3-030-91825-5_26.
- 18 Dragoš Cvetkovic, Peter Rowlinson, and Slobodan Simic. *Spectral Generalizations of Line Graphs: On Graphs with Least Eigenvalue -2*. London Mathematical Society Lecture Note Series. Cambridge University Press, 2004. doi:10.1017/CB09780511751752.
 - 19 J. G. Dai. On positive harris recurrence of multiclass queueing networks: A unified approach via fluid limit models. *The Annals of Applied Probability*, 5(1):49–77, 1995. Publisher: Institute of Mathematical Statistics. doi:10.1214/aoap/1177004828.
 - 20 Reinhard Diestel. *Graph Theory*. Springer Publishing Company, Incorporated, 5th edition, August 2017.
 - 21 Michael Doob. An interrelation between line graphs, eigenvalues, and matroids. *Journal of Combinatorial Theory, Series B*, 15(1):40–50, 08 1973. doi:10.1016/0095-8956(73)90030-0.
 - 22 David Gregory. Hall conditions for edge-weighted bipartite graphs. Technical report, Queen’s University, 2010.
 - 23 Varun Gupta. Greedy algorithm for multiway matching with bounded regret. *Operations Research*, 72(3):1139–1155, 2024.
 - 24 Itai Gurvich and Amy Ward. On the Dynamic Control of Matching Queues. *Stochastic Systems*, 4(2):479–523, October 2014. Publisher: INFORMS. doi:10.1287/13-SSY097.
 - 25 B. Hajek. *Random Processes for Engineers*. Cambridge university press, 2015.
 - 26 Matthieu Jonckheere, Pascal Moyal, Claudia Ramírez, and Nahuel Soprano-Loto. Generalized max-weight policies in stochastic matching. *arXiv:2011.04535 [math]*, November 2020. arXiv: 2011.04535. URL: <http://arxiv.org/abs/2011.04535>.
 - 27 N. Karmarkar. A new polynomial-time algorithm for linear programming. In *Proceedings of the sixteenth annual ACM symposium on Theory of computing*, STOC ’84, pages 302–311, New York, NY, USA, December 1984. Association for Computing Machinery. doi:10.1145/800057.808695.
 - 28 F. P. Kelly. *Reversibility and Stochastic Networks*. Cambridge University Press, 2011.
 - 29 Süleyman Kerimov, Itai Ashlagi, and Itai Gurvich. On the optimality of greedy policies in dynamic matching. *Operations Research*, 2023.
 - 30 Süleyman Kerimov, Itai Ashlagi, and Itai Gurvich. Dynamic matching: Characterizing and achieving constant regret. *Management Science*, 70(5):2799–2822, 2024.
 - 31 John D. C. Little. A proof for the queuing formula: $l = \lambda w$. *Operations Research*, 9(3):383–387, 1961. doi:10.1287/opre.9.3.383.
 - 32 John D. C. Little. Little’s law as viewed on its 50th anniversary. *Operations Research*, 59(3):536–549, 2011. doi:10.1287/opre.1110.0940.
 - 33 László Lovász and M. D. Plummer. *Matching Theory*. American Mathematical Soc., 2009.
 - 34 Jean Mairesse and Pascal Moyal. Stability of the stochastic matching model. *Journal of Applied Probability*, 53(4):1064–1077, 12 2016. Publisher: Cambridge University Press. doi:10.1017/jpr.2016.65.
 - 35 Fabien Mathieu. Stochastic Matching, 2022. URL: https://balouf.github.io/stochastic_matching/index.html.
 - 36 Pascal Moyal, Ana Bušić, and Jean Mairesse. A product form for the general stochastic matching model. *Journal of Applied Probability*, 58(2):449–468, 06 2021. doi:10.1017/jpr.2020.100.
 - 37 Mohammadreza Nazari and Alexander L. Stolyar. Reward maximization in general dynamic matching systems. *Queueing Systems*, 91(1):143–170, February 2019. doi:10.1007/s11134-018-9593-y.
 - 38 J. R. Norris. *Markov Chains*. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press, Cambridge, 1997. doi:10.1017/CB09780511810633.
 - 39 Youssef Rahme and Pascal Moyal. A stochastic matching model on hypergraphs. *Advances in Applied Probability*, 53(4):951–980, December 2021. doi:10.1017/apr.2021.8.

- 40 R. Srikant and Lei Ying. *Communication Networks: An Optimization, Control and Stochastic Networks Perspective*. Cambridge University Press, New York, NY, USA, 2014.
- 41 Yehua Wei, Jiaming Xu, and Sophie H Yu. Constant regret primal-dual policy for multi-way dynamic matching. In *Abstract Proceedings of the 2023 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems*, pages 79–80, 2023.
- 42 Günter M. Ziegler. *Lectures on polytopes*. Springer-Verlag, New York, 1995. URL: http://www.worldcat.org/search?qt=worldcat_org_all&q=9780387943657.

A

 Supplementary material of Section 2.3.4 (Other policies).

Our results are not limited to deterministic size-based policies but apply to a broader family of policies that are either random or require a more complex state descriptor, or both. We now introduce this family of policies, with the goal of being as general as possible.

A.1 Extended definition.

Under this more general definition, the match-maker makes decisions based not only on the vector of queue sizes, but also (possibly) on additional information that is captured by the system state. The state space is a couple $(\mathcal{S}, |\cdot|)$, where \mathcal{S} is a countably infinite set and $|\cdot| : \mathcal{S} \rightarrow \mathbb{N}^n$ is a function that maps any state $s \in \mathcal{S}$ to the vector giving the number of unmatched items of each class in that state, denoted by $|s| = (|s|_1, |s|_2, \dots, |s|_n)$. The existence of the function $|\cdot|$ guarantees that the system state contains enough information to retrieve the number of unmatched items of each class, which is a classical assumption in queueing theory (see for instance [28, Section 3.2]). We assume that there exists a unique state $s \in \mathcal{S}$ such that $|s| = 0$. This state is called the *empty state* and denoted by \emptyset . This assumption guarantees that the intuitive notion of system stability will be captured by the positive recurrence of the Markov chain describing the evolution of the system state¹⁹.

The policy is now a function $\Phi : \mathcal{S} \times V \times (V \cup \{\perp\}) \times \mathcal{S} \rightarrow [0, 1]$ such that $\Phi(s, i, j, s')$ is the conditional probability that, given an incoming class- i item finds the system in state s , the matching decision is j and the new state is s' . More formally, the dynamics are described by a Markov chain $((S_t, I_t, J_t), t \in \mathbb{N})$, where $I = (I_t, t \in \mathbb{N})$ is the sequence of incoming item classes and, for each $t \in \mathbb{N}$, $i \in V$, $j \in V \cup \{\perp\}$, and $s, s' \in \mathcal{S}$, we have

$$\mathbb{P}(J_t = j, S_{t+1} = s' \mid S_t = s, I_t = i) = \Phi(s, i, j, s').$$

The stochastic process $S = (S_t, t \in \mathbb{N})$ is also a Markov chain, with transition probabilities

$$\mathbb{P}(S_{t+1} = s' \mid S_t = s) = \frac{\sum_{i \in V} \lambda_i \sum_{j \in V \cup \{\perp\}} \Phi(s, i, j, s')}{\sum_{i \in V} \lambda_i}, \quad t \in \mathbb{N}, \quad s, s' \in \mathcal{S}.$$

We assume that the Markov chain S has state space \mathcal{S} and is irreducible, and that $S_0 = \emptyset$. The policy is assumed to be *adapted* to the compatibility graph G and *consistent* in the sense that, for each $(s, i, j, s') \in \mathcal{S} \times V \times (V \cup \{\perp\}) \times \mathcal{S}$, we have $\Phi(s, i, j, s') > 0$ only if

$$j \in \{j' \in V_i : |s|_{j'} \geq 1\} \cup \{\perp\}, \quad \text{and} \quad |s'| = \begin{cases} |s| + \mathbf{1}_i & \text{if } j = \perp, \\ |s| - \mathbf{1}_j & \text{if } j \neq \perp. \end{cases}$$

Using this extended definition, the previously defined policy models can be easily expressed. For example, the matching policy Φ is called *size-based* if $|\cdot|$ is the identity (implying $\mathcal{S} \subset \mathbb{N}^n$) and *deterministic* if, for each $s \in \mathcal{S}$ and $i \in V$, there exists $(j, s') \in (V \cup \{\perp\}) \times \mathcal{S}$ such that $\Phi(s, i, j, s') = 1$. The policy is called *greedy* if $\sum_{s' \in \mathcal{S}} \Phi(s, i, \perp, s') = 0$ for each $(s, i) \in \mathcal{S} \times V$ such that $\{j \in V_i : |s|_j \geq 1\} \neq \emptyset$, and *non-greedy* otherwise. FCFM (e.g., see [16, 36]) is a classical example of a deterministic policy that is not size-based: its state space is

¹⁹Without this assumption, one may construct two Markov chains associated with the same system, one positive recurrent and the other transient, for example if the state of the latter Markov chain embeds the time t . This assumption is used only in the proofs of Propositions 20 and 26. In both cases, we can verify that the same conclusion holds as long as the set of states $s \in \mathcal{S}$ such that $|s| = 0$ is finite. Assuming that this set is reduced to a singleton is merely a notational convenience.

a couple $(\mathcal{S}, |\cdot|)$ where \mathcal{S} is a subset of the set of sequences $c = (c_1, c_2, \dots, c_p)$ made of a finite but arbitrarily large number p of elements of \mathcal{I} , and $|c|_i$ is the cardinality of the set $\{q \in \{1, 2, \dots, p\} : c_q = i\}$, for each $i \in \mathcal{I}$. A policy that is neither size-based nor deterministic will appear in the proof of Proposition 20.

The stochastic process $Q = (Q_t, t \in \mathbb{N})$ defined by $Q_t = |S_t|$ for each $t \in \mathbb{N}$ is called the *queue-size process*. This process does not satisfy the Markov property in general, but it does satisfy the evolution equations (2) and (3), with $L_i = (L_{t,i}, t \in \mathbb{N})$ and $M_k = (M_{t,k}, t \in \mathbb{N})$ defined by (4) and (5) for each $i \in V$ and $k \in E$. The state space of the queue-size process is given by $\mathcal{Q} = \{|s|, s \in \mathcal{S}\}$. The policy is greedy if $\mathcal{Q} = \mathcal{Q}_G$ and non-greedy if $\mathcal{Q} \supsetneq \mathcal{Q}_G$, where \mathcal{Q}_G is still given by (7).

► **Remark 8** (Arrival rates vs. arrival sequence). *We will often identify the matching model (G, λ, Φ) with the Markov chain S . This is a slight abuse of language: the triplet (G, λ, Φ) specifies the transition diagram of this Markov chain but, even if Φ is deterministic, characterizing its sample paths requires specifying the sequence I , sampled according to $\lambda = (\lambda_1, \lambda_2, \dots, \lambda_n)$. This slight abuse of language will not cause confusion when discussing stability and matching rates, but the distinction will matter in Section 6.*

► **Remark 9** (Discrete time vs. continuous time). *The discrete-time Markov chain S gives the sequence of states observed by incoming items, and it was analyzed under various policies [34, 26]. Yet, in queueing theory, it is more common to consider the continuous-time Markov chain describing the system state over time. However, as observed in [16, Section 2.2.2], S is the jump chain of this continuous-time Markov chain, and both Markov chains have the same stationary measures because the departure rate from each state in the continuous-time Markov chain is constant equal to $\sum_{i \in V} \lambda_i$. Therefore, our results are equally relevant to study performance metrics like the mean queue size or the mean waiting time of items.*

A.2 Equivalent policies.

With our extended definition, a decision rule can be associated with an infinite number of policies. For instance, it is always possible to artificially expand the state definition, resulting in an unlimited range of policies. We define here an equivalence relation between policies that captures the intuitive concepts of *yielding identical distributions of matching decisions* (for random policies) and *making the same decisions* (for deterministic policies). This discussion will also prepare the ground for Propositions 34 and 36.

Consider a policy Φ_1 adapted to a compatibility graph $G = (V, E)$, and let $(\mathcal{S}_1, |\cdot|_1)$ denote its state space. Assume that the function $|\cdot|_1 : \mathcal{S}_1 \rightarrow \mathbb{N}^n$ can be written as a composition of two functions, $\langle \cdot \rangle : \mathcal{S}_1 \rightarrow \mathcal{S}_2$ and $|\cdot|_2 : \mathcal{S}_2 \rightarrow \mathbb{N}^n$, such that \mathcal{S}_2 is the image of \mathcal{S}_1 through $\langle \cdot \rangle$. Moreover, assume that there exists a policy Φ_2 with state space $(\mathcal{S}_2, |\cdot|_2)$, adapted to the graph G , such that for each $s_2, s'_2 \in \mathcal{S}_2$, $i \in V$, and $j \in V \cup \{\perp\}$, we have

$$\sum_{s'_1 \in \mathcal{S}_1 : \langle s'_1 \rangle = s'_2} \Phi_1(s_1, i, j, s'_1) = \Phi_2(s_2, i, j, s'_2) \quad \text{for each } s_1 \in \mathcal{S}_1 \text{ such that } \langle s_1 \rangle = s_2. \quad (27)$$

We say that policy Φ_1 can be *reduced* to policy Φ_2 and that $\langle \cdot \rangle$ is a *reduction* function. If $((S_{1,t}, I_t, J_{1,t}), t \in \mathbb{N})$ and $((S_{2,t}, I_t, J_{2,t}), t \in \mathbb{N})$ denote the Markov chains associated with policies Φ_1 and Φ_2 , respectively, under the same sequence $(I_t, t \in \mathbb{N})$ of incoming item classes, then for each $t \in \mathbb{N}$, (i) the conditional distribution of $(J_{1,t}, \langle S_{1,t+1} \rangle)$ given that $S_{1,t} = s_1$ and $I_t = i$ is the same for all states $s_1 \in \mathcal{S}_1$ that have the same image $s_2 = \langle s_1 \rangle$, and (ii) $((S_{1,t}, I_t, J_{1,t})$ and $(S_{2,t}, I_t, J_{2,t})$ have the same distribution. Conclusion (ii) follows from an

inductive argument and implies that policies Φ_1 and Φ_2 are stable or unstable under the same conditions and, if stable, yield the same matching rate vector.

The special case where the policy Φ_2 is deterministic will be useful in Propositions 34 and 36. In this case, (27) says that, for each $s_2 \in \mathcal{S}_2$ and $i \in V$, there exist $j \in V \cup \{\perp\}$ and $s'_2 \in \mathcal{S}_2$ such that $\sum_{s'_1 \in \mathcal{S}_1: \langle s'_1 \rangle = s'_2} \Phi_1(s_1, i, j, s'_1) = \Phi_2(s_2, i, j, s'_2) = 1$ for each $s_1 \in \mathcal{S}_1$ such that $\langle s_1 \rangle = s_2$. This condition implies that the Markov chains under Φ_1 and Φ_2 are equivalent pathwise (i.e., $\langle S_{1,t} \rangle = S_{2,t}$ and $J_{1,t} = J_{2,t}$ for each $t \in \mathbb{N}$) and not just in distribution.

In general, we say that two policies Φ_1 and Φ_2 adapted to the graph G are *equivalent* if there exists a policy Φ adapted to the graph G such that both Φ_1 and Φ_2 can be reduced to Φ . This equivalence between Φ_1 and Φ_2 can be interpreted as indicating that the two policies, when they are in equivalent states, i.e. states that have the same reduction, yield identical distributions of matching decisions. It follows that, if we let $(Q_{1,t}, t \in \mathbb{N})$ and $(Q_{2,t}, t \in \mathbb{N})$ denote the queue-size processes under Φ_1 and Φ_2 with the same sequence $(I_t, t \in \mathbb{N})$ of incoming item classes, then we have $\mathbb{P}(Q_{1,t} = q) = \mathbb{P}(Q_{2,t} = q)$ for each $t \in \mathbb{N}$ and $q \in \mathcal{Q}$.

B Supplementary material of Section 3 (Graph theory and linear algebra).

B.1 Proofs of Section 3.2 (Surjectivity, injectivity, and bijectivity)

Proof of Definition 2. The equivalence of i, ii, and iii is a well-known result in linear algebra. We prove that conditions iii and iv are equivalent. This proof is adapted from [18, Lemma 2.2.3].

The key argument consists of observing that a vector $x = (x_1, x_2, \dots, x_n) \in \mathbb{R}^n$ belongs to the left kernel of the matrix A if and only if

$$\sum_{i=1}^n x_i a_{i,k} = 0, \quad k \in \{1, 2, \dots, m\}.$$

For each $k \in \{1, 2, \dots, m\}$, the k -th equation reads $x_j = -x_i$, where i and j are the endpoints of edge k . An induction argument shows that, for every path i_1, i_2, \dots, i_k in the graph G , we have $x_{i_p} = (-1)^{p-1} x_{i_1}$ for each $p \in \{1, 2, \dots, k\}$.

First assume that condition iv is satisfied. Let $x \in \mathbb{R}^n$ be a vector of the left kernel of the matrix A . Since each connected component of G is non-bipartite, for each $i \in V$, there exists a path of length say ℓ that connects node i to a cycle $i_1, i_2, \dots, i_p, i_{p+1} = i_1$ consisting of an odd number p of nodes. We then obtain $x_i = (-1)^{\ell+p+\ell} x_{i_1} = -x_{i_1}$, which implies that $x_i = 0$. Therefore, the left kernel of A is trivial, meaning that condition iii is satisfied.

On the contrary, if condition iv is not satisfied, then there exists a connected component of G that is bipartite with parts V_+ and V_- . We build a non-zero vector in the left kernel of A by choosing $x_i = 1$ for each $i \in V_+$, $x_i = -1$ for each $i \in V_-$, and $x_i = 0$ for each $i \in V \setminus (V_+ \cup V_-)$. This implies that condition iii is not satisfied. ◀

Proof of Definition 3. The equivalence of conditions i, ii, and iii is a well-known result in linear algebra. We now prove that conditions iii and iv are equivalent.

We first assume that the graph G is connected, and we distinguish the following two cases:

- If G is non-bipartite, according to Definition 2, the nullity of A^\top is 0. The rank-nullity theorem implies that the rank of A^\top is n , so that the rank of A is also n . A second application of the rank-nullity theorem implies that the nullity of A is $m - n$. In particular, $\ker(A) = \{0\}$ if and only if $m = n$.
- If G is bipartite, any non-zero vector of the left kernel of A must be parallel (collinear) to the non-zero vector x constructed in the proof of Definition 2. This parallelism is due to the constraints $x_i = -x_j$ for all edges $\{i, j\}$. Based on this, the nullity of A^\top is 1, and we conclude from another double application of the rank-nullity theorem that the nullity of A is $m - n + 1$. In particular, $\ker(A) = \{0\}$ if and only if $m = n - 1$.

All in all, we obtain that condition iii is true if and only if either the graph G is non-bipartite and contains as many edges as nodes, or the graph G is bipartite and contains one less edge than it contains nodes. This, in turn, is equivalent to condition iv.

If the graph G is not connected, we can rewrite A as a bloc matrix in which each bloc corresponds to a connected component, and we can then use the previous argument to prove the equivalence for each connected component. ◀

Proof of Definition 4. The function $\mu \in \mathbb{R}^m \mapsto A\mu \in \mathbb{R}^n$ is bijective if and only if it is both surjective and injective. Hence, the equivalence of conditions i to iv follows directly from Definitions 2 and 3. ◀

Proof of Proposition 6. These statements (transposed to A) are again well-known in linear algebra. ◀

B.2 Proof of Proposition 8 in Section 3.3.2 (Stabilizable matching problem)

Equivalence of i and ii follows from [34, Proposition 2 and Theorem 2]. For completeness, we observe that [34, Proposition 2] is proved under the assumption that the matching policy Φ is greedy and deterministic and that the state space $(\mathcal{S}, |\cdot|)$ has a particular form, but we can verify that the argument remains valid under the assumptions of Section 2. We now prove that ii and iii are equivalent. Condition ii implies condition iii because (a) according to [34], under condition ii, (G, λ, Φ) is stable when Φ is the ML policy, and (b) the associated vector μ of matching rates satisfies condition iii by ergodicity. That condition iii implies condition ii was proved by [16, Lemma 12]. ◀

B.3 Minimal stability region for greedy matching policies (Section 3.4.3 and Appendices F.2.1 and F.2.2).

The following result gives a sufficient stability condition for greedy matching policies. The proof relies on a linear Lyapunov function. This result can be seen as the counterpart of [9, Proposition 5.1] for non-bipartite matching models.

► **Proposition 26.** *Consider a matching problem (G, λ) with a connected graph G . If*

$$\sum_{i \in V(\mathcal{I})} \lambda_i > \frac{1}{2} \sum_{i \in V} \lambda_i, \quad \mathcal{I} \in \mathbb{I}, \quad (28)$$

then the matching model (G, λ, Φ) is stable for every greedy matching policy Φ .

Proof. Consider a matching problem (G, λ) that satisfies (28) and a greedy matching policy Φ adapted to the graph G . Since the Markov chain $(S_t, t \in \mathbb{N})$ associated with the matching model (G, λ, Φ) depends on the vector λ only up to a positive multiplicative constant, we can assume without loss of generality that $\sum_{i \in V} \lambda_i = 1$. Let \mathcal{S} denote the state space of this Markov chain and $|\cdot|$ the corresponding queue-size function (as defined in Appendix A). We consider the Lyapunov function $F : \mathcal{S} \rightarrow \mathbb{R}$ defined by $F(s) = \sum_{i \in V} |s|_i$ (that is, $F(s)$ is the number of unmatched items in state s) for each $s \in \mathcal{S}$. For each $t \in \mathbb{N}$ and $s \in \mathcal{S}$, we have

$$\mathbb{E}(F(S_{t+1}) | S_t = s) - F(s) = \sum_{i \in V \setminus V(\mathcal{I})} \lambda_i - \sum_{i \in V(\mathcal{I})} \lambda_i = - \left(\sum_{i \in V(\mathcal{I})} \lambda_i - \sum_{i \in V \setminus V(\mathcal{I})} \lambda_i \right),$$

where $\mathcal{I} = \{i \in V : |s|_i \geq 1\}$ is the set of classes of unmatched items in state s . Importantly, if $F(s) > 0$ (that is, $s \neq \emptyset$), then \mathcal{I} is an independent set of the compatibility graph G because it is non-empty and the policy Φ is greedy. It follows that, for each $s \in \mathcal{S} \setminus \{\emptyset\}$,

$$\mathbb{E}(F(S_{t+1}) | S_t = s) - F(s) \leq -\varepsilon, \quad \text{with } \varepsilon = \min_{\mathcal{I} \in \mathbb{I}} \left(\sum_{i \in V(\mathcal{I})} \lambda_i - \sum_{i \in V \setminus V(\mathcal{I})} \lambda_i \right).$$

Equation (28) implies that $\varepsilon > 0$. Using the Lyapunov-Foster theorem, see [8, Theorem 1.1 in Chapter 5], we conclude that the matching model (G, λ, Φ) is stable. ◀

As one would expect, any matching problem (G, λ) that satisfies (28) is stabilizable in the sense of Definition 1. Indeed, (28) implies Proposition 8ii because $\mathcal{I} \subseteq V \setminus V(\mathcal{I})$ for each $\mathcal{I} \in \mathbb{I}$. Corollary 27 below shows that, conversely, whether a stabilizable matching problem satisfies (28) depends on the structure of the graph G : conditions i and ii exhibit compatibility graphs G such that (28) is satisfied whenever the matching problem (G, λ) is stabilizable, while conditions iii and iv exhibit stabilizable compatibility graphs G for which (28) is never satisfied.

► **Corollary 27.** *Consider a matching problem (G, λ) .*

Under the following two conditions, the stabilizability of the matching model (G, λ) implies that (28) is satisfied, and therefore that the matching model (G, λ, Φ) is stable under every greedy policy Φ adapted to G :

- (i) G is a complete graph with $n \geq 3$ nodes.
- (ii) G is the diamond graph of Example 4.

Under the following conditions, (28) is never satisfied:

- (iii) The graph G has diameter greater than or equal to 3.
- (iv) The graph G contains a leaf (that is, a node with degree 1).

Proof. We first need to prove that, under either condition i or condition ii, the matching model (G, λ) is stabilizable if and only if (28) is satisfied. We proceed by verifying that, under either of these two conditions, Proposition 8ii and (28) are equivalent:

- (i) First assume that condition i is satisfied. The independent sets of a complete graph K_n are the singletons. Using this observation, we can verify that Proposition 8ii and (28) are both equivalent to $\lambda_i < \frac{1}{2} \sum_{i' \in V} \lambda_{i'}$ for each $i \in V$.
- (ii) Now assume that condition ii is satisfied, that is, G is the diamond graph. The conclusion follows by recalling that Proposition 8ii simplifies to (12), and then by observing that (12) and (28) are equivalent.

To prove that (28) cannot be satisfied under either condition iii or iv, we proceed by contradiction:

- (iii) First assume that condition iii is satisfied, and let i and j denote two nodes that are at distance 3 or more. In particular, the sets V_i and V_j are disjoint. If (28) is satisfied, then applying this equation to both $\{i\}$ and $\{j\}$ and summing the inequalities yields $\sum_{i' \in V_i \cup V_j} \lambda_{i'} > \sum_{i' \in V} \lambda_{i'}$, which is a contradiction since $V_i \cup V_j \subseteq V$. Hence, (28) cannot be satisfied by both $\{i\}$ and $\{j\}$.
- (iv) Now assume that condition iv is satisfied. Let i denote a leaf node of G and j the (only) neighbor of i . Then again, applying (28) to both $\{i\}$ and $\{j\}$ and summing the inequalities yields $\sum_{i' \in V_j \cup \{j\}} \lambda_{i'} > \sum_{i' \in V} \lambda_{i'}$, which is again a contradiction. ◀

C

Supplementary material of Section 5 (Polytope of solutions in surjective-only graphs).

C.1 Algorithm of Section 5.1.3 (Basis of the kernel of the incidence matrix).

Given a surjective-only graph G with an incidence matrix denoted by A , Algorithm 1 builds a basis of the kernel of A as follows. The algorithm first identifies a spanning tree \mathcal{T} of G (Line 1) and an edge $a \in E \setminus \mathcal{T}$ such that the set $E \setminus (\mathcal{T} \cup \{a\})$ (of cardinality $d = m = n$) contains an odd cycle (Line 2). Then, for each edge $s \in E \setminus (\mathcal{T} \cup \{a\})$, the algorithm builds (Lines 5–26) a base vector b whose support (i) is either an even cycle or a kayak paddle and (ii) contains s and is included into $\mathcal{T} \cup \{a, s\}$. We assume without loss of generality that the graph G is connected (in addition to being surjective-only). If not, we can apply the algorithm to each connected component separately, and then we embed the obtained vectors to \mathbb{R}^n via zero padding.

We now verify that Algorithm 1 terminates and yields the desired result.

► **Proposition 28.** *Algorithm 1 terminates and returns a basis of the kernel of the incidence matrix A of the compatibility graph G .*

Proof. The proof is based mainly on the notion of *cycle space* in a graph. We briefly summarize the concepts that are useful to understand the proof (see [20, Section 1.9] for details).

A *spanning subgraph* of a graph $G = (V, E)$ is a subgraph $G^\circ = (V, E^\circ)$ with $E^\circ \subseteq E$. Importantly, G and G° have the same set of nodes. A subgraph is *Eulerian* if every vertex has an even degree (possibly zero). In particular, if E° is a set of edges that form a cycle in G , then the graph (V, E°) is Eulerian. The *cycle space* of G is the vector space made of all Eulerian spanning subgraphs of G , using the symmetric difference of the edge sets for addition and the two-element field for scalar multiplication. Equivalently, the cycle space can be described as a vector space of the finite field $\mathbb{Z}/2\mathbb{Z}$: each vector $g = (g_1, g_2, \dots, g_m)$ in this vector space satisfies $\sum_{k \in E_i} g_k = 0$ (modulo 2) for each $i \in V$, and the addition and multiplication are the usual operations in $\mathbb{Z}/2\mathbb{Z}$. For example, if G is the codomino graph of Figure 17, the spanning subgraphs G_1, G_2 , and G_3 of G with edge sets $E_1 = \{\{1, 2\}, \{1, 6\}, \{2, 3\}, \{3, 4\}, \{4, 5\}, \{5, 6\}\}$, $E_2 = \{\{2, 3\}, \{2, 6\}, \{3, 5\}, \{5, 6\}\}$, and $E_3 = \{\{1, 2\}, \{1, 6\}, \{2, 6\}\}$, respectively, belong to the cycle space of G . The edge set of the addition $G_1 + G_2$ is the set $\{\{1, 2\}, \{1, 6\}, \{2, 6\}, \{3, 4\}, \{3, 5\}, \{4, 5\}\}$ of edges that are either in E_1 or in E_2 , but not in both. Similarly, the edge set of $G_1 + G_3$ is $\{\{2, 3\}, \{2, 6\}, \{3, 4\}, \{4, 5\}, \{5, 6\}\}$, and the edge set of $G_2 + G_3$ is $\{\{1, 2\}, \{1, 6\}, \{2, 3\}, \{3, 5\}, \{5, 6\}\}$. One can verify that $\{G_1, G_2, G_3\}$ forms a basis of the cycle space of G . Importantly, in general, the dimension of the cycle space is $m - n + 1$.

Algorithm 1 terminates. We first prove the existence of edge a defined on Line 2 of Algorithm 1. By definition of a spanning tree, \mathcal{T} contains $n - 1$ edges and, for each edge $a \in E \setminus \mathcal{T}$, $\mathcal{T} \cup \{a\}$ contains a unique cycle. The $m - n + 1 = |E \setminus \mathcal{T}|$ cycles thus obtained are independent in the sense that each cycle contains at least one edge (a) that is not contained in the other cycles. Therefore, these $m - n + 1$ cycles form a basis of the cycle space of G . Since a linear combination of even cycles cannot produce a subgraph consisting of a single odd cycle²⁰, and since G contains an odd cycle (as it is non-bipartite), then at least one of the $m - n + 1$ basis cycles is odd.

²⁰A linear combination of even cycles may produce a subgraph consisting of an even number of disjoint

Data: A connected surjective-only compatibility graph $G = (V, E)$

Result: A basis \mathcal{B} of the kernel of the incidence matrix A of G

```

1  $\mathcal{T} \leftarrow$  the set of edges of a spanning tree of  $G$ 
2  $a \leftarrow$  an edge in  $E \setminus \mathcal{T}$  such that  $\mathcal{T} \cup \{a\}$  contains an odd cycle
3  $\mathcal{B} \leftarrow \emptyset$ 
4 for  $s \in E \setminus (\mathcal{T} \cup \{a\})$  do
5    $b \leftarrow (0, 0, \dots, 0) \in \mathbb{R}^m$ 
6   if  $\mathcal{T} \cup \{a, s\}$  contains an even cycle  $C_\ell$  then
7      $c_1, \dots, c_\ell \leftarrow$  consecutive edges of  $C_\ell$ 
8     for  $d \in \{1, \dots, \ell\}$  do
9        $k \leftarrow$  index of  $c_d$  in  $E$ 
10       $b_k \leftarrow (-1)^d$ 
11   else
12      $\mathcal{T} \cup \{a, s\}$  contains a kayak paddle  $KP_{\ell, r, p}$  with  $\ell$  odd,  $r$  odd, and  $p \geq 0$ 
13      $v_i \leftarrow$  node connecting the kayak's cycle  $C_\ell$  to the kayak central path  $P_p$ 
14      $v_j \leftarrow$  node connecting the kayak's cycle  $C_r$  to the kayak central path  $P_p$ 
15      $c_1, \dots, c_\ell \leftarrow$  consecutive edges of  $C_\ell$ , starting and ending at node  $v_i$ 
16     for  $d \in \{1, \dots, \ell\}$  do
17        $k \leftarrow$  index of  $c_d$  in  $E$ 
18        $b_k \leftarrow (-1)^d$ 
19      $c_1, \dots, c_p \leftarrow$  consecutive edges of  $P_p$ , starting at node  $v_i$  and ending at node  $v_j$ 
20     for  $d \in \{1, \dots, p\}$  do
21        $k \leftarrow$  index of  $c_d$  in  $E$ 
22        $b_k = 2(-1)^{d+1}$ 
23      $c_1, \dots, c_r \leftarrow$  consecutive edges of  $C_r$ , starting and ending at node  $v_j$ 
24     for  $d \in \{1, \dots, r\}$  do
25        $k \leftarrow$  index of  $c_d$  in  $E$ 
26        $b_k \leftarrow (-1)^{d+p+1}$ 
27    $\mathcal{B} \leftarrow \mathcal{B} \cup \{b\}$ 
28 return  $\mathcal{B}$ 

```

■ **Algorithm 1** Construction of a basis of the kernel of the incidence matrix A of the compatibility graph G . This algorithm was initially introduced by [21, Section 3] to build a basis of the eigenspace associated with the eigenvalue -2 of the adjacency matrix of a line graph (i.e., a graph whose nodes and edges represent, respectively, the edges and their incidence relations in another graph).

We now verify that, for each $s \in E \setminus (\mathcal{T} \cup \{a\})$, $\mathcal{T} \cup \{a, s\}$ contains either (i) an even cycle C_ℓ or (ii) a kayak paddle $KP_{\ell, r, p}$ with two odd cycles. By construction, $\mathcal{T} \cup \{a\}$ contains a unique cycle C_r , which is odd, and $\mathcal{T} \cup \{s\}$ contains a unique cycle C_ℓ . $\mathcal{T} \cup \{a, s\}$ contains both C_r and C_ℓ . We now proceed by elimination:

- If C_ℓ is even, then C_ℓ is an even cycle included into $\mathcal{T} \cup \{a, s\}$, and we are in case (i).
- If C_ℓ is odd and shares at least one edge with C_r , then the symmetric difference of C_r and C_ℓ is an even cycle, and it is again included into $\mathcal{T} \cup \{a, s\}$, so we are again in case (i).
- If C_ℓ is odd and C_r and C_ℓ have no edge in common, then we are in case (ii).

odd cycles, as illustrated by $G_1 + G_2$ in the example above, but not a subgraph consisting of one odd cycle. Indeed, the symmetric difference of two edge sets contains an even number of edges if both edge sets contain an even number of edges.

Algorithm 1 returns the correct result. We finally prove that the family \mathcal{B} returned by Algorithm 1: (i) has cardinality $m - n$, (ii) is linearly independent, and (iii) is included into the kernel of A . We prove each item one after another:

- (i) The family \mathcal{B} has cardinality $m - n$: It suffices to observe that \mathcal{B} has same cardinality as $E \setminus (\mathcal{T} \cup \{a\})$, which we already mentioned has cardinality $m - n$.
- (ii) The family \mathcal{B} is linearly independent: For each $s \in E \setminus (\mathcal{T} \cup \{a\})$, the basis vector constructed from edge s is the only vector in \mathcal{B} whose support contains this edge.
- (iii) The family \mathcal{B} is included into the kernel of A : Let $b \in \mathcal{B}$. Our goal is to prove that b belongs to the kernel of A , i.e., that $\sum_{k \in E_i} b_k = 0$ for each $i \in V$. First observe that, for each $i \in V$, we have $\sum_{k \in E_i} b_k = \sum_{k \in E_i \cap S} b_k$, where S is the support of the vector b . In particular, we have immediately $\sum_{k \in E_i} b_k = 0$ for each $i \in V$ such that $E_i \cap S = \emptyset$. Now consider a node $i \in V$ such that $E_i \cap S \neq \emptyset$. We make a case disjunction depending on the support S of b :

- If S is an even cycle C_ℓ , then $E_i \cap S = \{k_1, k_2\}$, where k_1 and k_2 are two consecutive edges of the cycle C_ℓ . Line 10 in the algorithm implies that $b_{k_1} = -b_{k_2} \in \{1, -1\}$. It follows that $\sum_{k \in E_i} b_k = b_{k_1} + b_{k_2} = 0$.
- If S is a kayak paddle $KP_{\ell,r,p}$ with odd cycles C_ℓ and C_r and central path P_p , $p \in \mathbb{N}$, we again distinguish several cases:

Node i does not belong to the central path: If $E_i \cap S \subseteq C_\ell$ or $E_i \cap S \subseteq C_r$, we conclude as before.

Node i does not belong to a cycle: If $E_i \cap S \subseteq P_p$, then $E_i \cap S = \{k_1, k_2\}$ where k_1 and k_2 are two consecutive edges of the path P_p . Line 22 of the algorithm implies that $b_{k_1} = -b_{k_2} \in \{2, -2\}$. It follows that $\sum_{k \in E_i} b_k = b_{k_1} + b_{k_2} = 0$.

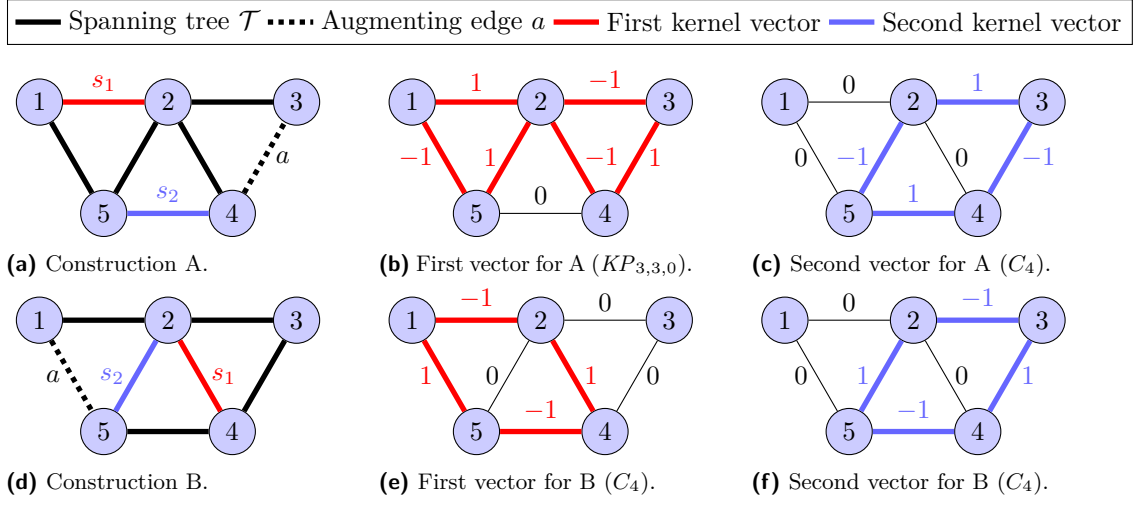
Node i belongs to a cycle and the central path: The only remaining case is when $E_i \cap S$ intersects several sets among C_ℓ , C_r , and P_p . If $p = 0$, that is, if the central path is the node i , then $E_i \cap S = \{k_1, k_2, k_3, k_4\}$, where k_1 and k_2 (resp. k_3 and k_4) are two consecutive edges in C_ℓ (resp. C_r). Lines 18 and 26 yield $b_{k_1} = b_{k_2} = -1$ and $b_{k_3} = b_{k_4} = 1$, which implies that $\sum_{k \in E_i} b_k = b_{k_1} + b_{k_2} + b_{k_3} + b_{k_4} = 0$. If $p \geq 1$, then $E_i \cap S = \{k_1, k_2, k_3\}$, where k_1 and k_2 are two consecutive edges of either C_ℓ or C_r , and k_3 is an edge in P_p . Lines 18, 22, and 26 of the algorithm imply that $b_{k_1} = b_{k_2} \in \{1, -1\}$ and that $b_{k_3} = -2b_{k_1}$, so that we conclude again that the desired sum is zero. ◀

Figures 16 and 17 shows possible runs of Algorithm 1 on the triamond and codomino graphs, which both have a two-dimensional kernel. Note that the basis is not unique and depends on our choice of the spanning tree \mathcal{T} and the augmenting edge a (see Lines 1 and 2 in Algorithm 1).

C.2 Proof of Proposition 13 in Section 5.2.1 (Vertex characterization) (borrowed from [15]).

We prove that the negations of i and ii are equivalent. Let A^* denote the incidence matrix of G^* .

By Definition 12, if μ is not a vertex of $\Pi_{\geq 0}$, there exist $z_1, z_2 \in \Pi_{\geq 0} \setminus \{\mu\}$ and $0 < \theta < 1$ such that $\mu = \theta z_1 + (1 - \theta) z_2$. The coordinates of the vectors z_1 and z_2 are non-negative, so this equality implies that their supports are included into the support E^* of the vector μ . In particular, if $\tilde{\mu}$ and \tilde{z}_1 denote the restrictions of μ and z_1 to coordinates in E^* , respectively,



■ **Figure 16** Two possible constructions of a kernel basis for the triamond graph. Construction A yields the basis vectors $b_1 = (1, -1, -1, -1, 1, 1, 0)$ and $b_2 = (0, 0, 1, 0, -1, -1, 1)$. Construction B yields the basis vectors $b_1 = (-1, 1, 0, 1, 0, 0, -1)$ and $b_2 = (0, 0, -1, 0, 1, 1, -1)$.

then $A^* \tilde{\mu} = A\mu = \lambda = Az_1 = A^* \tilde{z}_1$ with $\tilde{\mu} \neq \tilde{z}_1$, which means that $G^* = (V, E^*)$ is not injective.

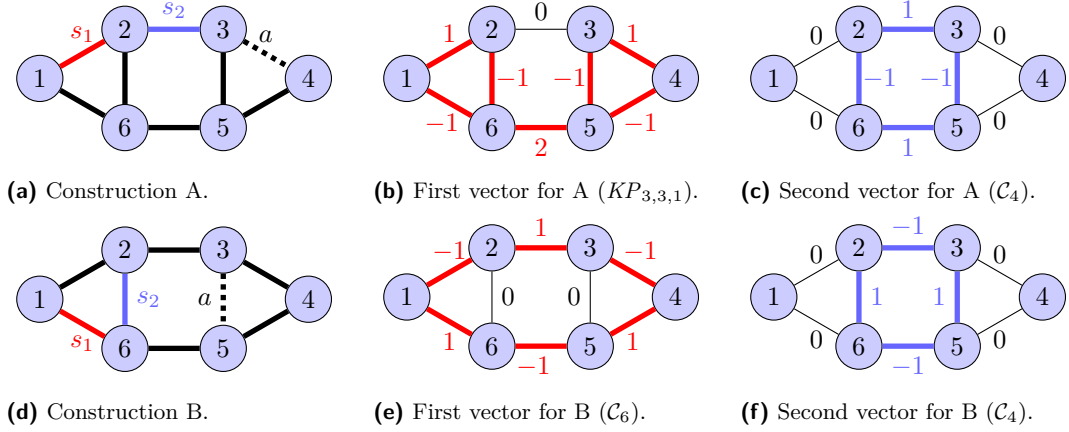
Conversely, if G^* is not injective, there exists a non-zero vector \tilde{z} in $\mathbb{R}^{|E^*|}$ such that $A^* \tilde{z} = 0$. If we embed \tilde{z} into $\mathbb{R}^{|E|}$ with zero-padding, we obtain a non-zero vector z such that $Az = 0$, and whose support is included into that of the vector μ . This implies that there exists $\varepsilon > 0$ such that both $\mu - \varepsilon z$ and $\mu + \varepsilon z$ belong to $\Pi_{\geq 0}$. The convex combination $\mu = \frac{1}{2}(\mu - \varepsilon z) + \frac{1}{2}(\mu + \varepsilon z)$ proves that the vector μ is not a vertex of $\Pi_{\geq 0}$.

The last part directly derives from Statements ii and v in Proposition 6. ◀

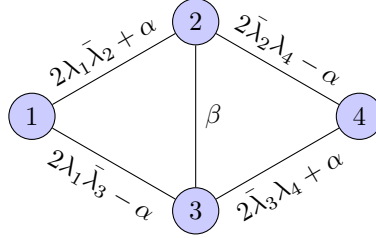
C.3 Additional examples and results for Section 5.2.1 (Vertex characterization).

Following Proposition 13, the bijectivity of a vertex is determined by the number of its coordinates that are positive in edge coordinates, that is, by the cardinality of the set of edges that form its support. Recall that the d -dimensional polytope $\Pi_{\geq 0}$ is actually characterized by the m inequalities $\mu_k \geq 0$ for each $k \in E$. In particular, this polytope has at most m facets, one for each inequality, but it typically has fewer. Indeed, some inequalities may be redundant and/or not tight, in a sense that will be defined in Definition 29 below. For example, by looking more closely at the general solution obtained for the diamond graph in Figure 3d, we conclude that:

- The inequality $\mu_{2,3} \geq 0$ is satisfied trivially by every vector $\mu \in \Pi$, as we have $\mu_{2,3} = \beta > 0$. Therefore, this inequality does not define a facet of $\Pi_{\geq 0}$.
- If $\lambda_1 \bar{\lambda}_2 < \bar{\lambda}_3 \lambda_4$, the inequality $\mu_{1,2} \geq 0$ supersedes the inequality $\mu_{3,4} \geq 0$, and conversely. If $\lambda_1 \bar{\lambda}_2 = \bar{\lambda}_3 \lambda_4$, these two inequalities are equivalent. In both cases, the inequalities $\mu_{1,2} \geq 0$ and $\mu_{3,4} \geq 0$ lead to a single facet of $\Pi_{\geq 0}$.
- If $\lambda_1 \bar{\lambda}_3 < \bar{\lambda}_2 \lambda_4$, the inequality $\mu_{1,3} \geq 0$ supersedes the inequality $\mu_{2,4} \geq 0$, and conversely. If $\lambda_1 \bar{\lambda}_3 = \bar{\lambda}_2 \lambda_4$, these two inequalities are equivalent. In both cases, the inequalities $\mu_{1,3} \geq 0$ and $\mu_{2,4} \geq 0$ lead to a single facet of $\Pi_{\geq 0}$.



■ **Figure 17** Two possible constructions of a kernel basis for the codomino graph. Construction A yields the vectors $b_1 = (1, -1, 0, -1, 1, -1, -1, 2)$ and $b_2 = (0, 0, 1, -1, 0, -1, 0, 1)$. Construction B yields the vectors $b_1 = (-1, 1, 1, 0, -1, 0, 1, -1)$ and $b_2 = (0, 0, -1, 1, 0, 1, 0, -1)$.



■ **Figure 3d (repeated)** Matching rates in the diamond graph with the normalization $\lambda_1 + \lambda_4 = \frac{1}{2}$. $2\beta = \lambda_2 + \lambda_3 - \frac{1}{2}$ is the difference between the arrival rates of $\{2, 3\}$ and $\{1, 4\}$. $\bar{\lambda}_2 = \lambda_2 - \beta$ and $\bar{\lambda}_3 = \lambda_3 - \beta$ represent the residual rates that classes 2 and 3 can provide to classes 1 and 4.

All in all, the 1-dimensional convex polytope $\Pi_{\geq 0}$ associated with the diamond graph of Figure 3d has two facets, even if it is defined by five inequalities. Definition 29 below will help us relate these notions to the number of zero coordinates of the vertices of the convex polytope $\Pi_{\geq 0}$.

► **Definition 29** (Adapted from [4, 42]). Let $k \in E$.

- (i) The inequality $\mu_k \geq 0$ is said to be tight if there exists a vector $\mu \in \Pi_{\geq 0}$ such that $\mu_k = 0$, in which case we also say that this inequality is tight for the vector μ .
- (ii) The inequality $\mu_k \geq 0$ is said to be redundant if removing this inequality does not change the polytope $\Pi_{\geq 0}$, in the sense that

$$\Pi_{\geq 0} = \{\mu \in \mathbb{R}^m : A\mu = \lambda \text{ and } \mu_\ell \geq 0 \text{ for each } \ell \in E \setminus \{k\}\}.$$

Otherwise, this inequality is called *irredundant*.

- (iii) The matching problem (G, λ) is called *essential* if all tight inequalities are irredundant.
- (iv) The polytope $\Pi_{\geq 0}$ is said to be *simple* if every vertex of $\Pi_{\geq 0}$ belongs to exactly d facets, which is the minimal number of facets a vertex belongs to.

Importantly, the number of positive coordinates of a vertex μ (considered in Proposition 17) is the number of inequalities that are not tight for this vertex. More generally, Definition 29 has the following intuitive interpretation. An inequality is tight if the convex polytope $\Pi_{\geq 0}$

intersects the hyperplane obtained by transforming this inequality into an equality. Non-tight inequalities are “useless” (and redundant) because they are never satisfied as equalities by any vector in $\Pi_{\geq 0}$. The matching problem (G, λ) is essential if each tight inequality defines a distinct facet of the convex polytope $\Pi_{\geq 0}$. Under this condition, the number of facets that contain a vertex is equal to the number of inequalities that are tight for this vertex. In particular, as we will see in Proposition 30, if the matching problem (G, λ) is essential and the polytope $\Pi_{\geq 0}$ is simple, then every vertex satisfies exactly d (tight) inequalities as equalities, which means that this vertex has d zero coordinates, and therefore $n = m - d$ positive coordinates, so that this vertex is bijective.

All these notions are illustrated in Examples 8–10 below, which show in particular that a matching problem (G, λ) may be essential even if the polytope $\Pi_{\geq 0}$ is not simple, and conversely. Consistently with the discussion above on Figure 3d, these examples use a kernel basis to verify effortlessly whether an inequality is tight and/or irredundant.

► **Example 8** (Essential matching problem). *Figure 18 considers a codomino graph with the vector of arrival rates $\lambda = (4, 5, 3, 2, 3, 5)$. A particular solution to (CE) is $\mu^\circ = (2, 2, 1, 2, 1, 1, 1, 1) \in \mathbb{R}^8$, and the basis of $\ker(A)$ consists of the vectors $b_1 = (-1, 1, 1, 0, -1, 0, 1, -1)$ and $b_2 = (0, 0, -1, 1, 0, 1, 0, -1)$ obtained in construction B of Figure 17. The generic solution to (CE) is shown in Figure 18b.*

The inequalities are listed in Figure 18a. The 2-dimensional polytope $\Pi_{\geq 0}$, shown in Figure 18c in kernel basis, is characterized by five tight inequalities which are also irredundant:

$$-1 \leq \alpha_1 \leq 1, \quad \alpha_2 \geq -1, \quad \alpha_1 - \alpha_2 \geq -1, \quad \alpha_1 + \alpha_2 \leq 1.$$

The matching problem (G, λ) is essential. In kernel basis, the vertices of the convex polytope $\Pi_{\geq 0}$ are $(0, 1)$, $(-1, 0)$, $(1, 0)$, $(-1, -1)$, and $(1, -1)$, and we can verify on Figure 18c that each vertex belongs to exactly 2 facets. Therefore $\Pi_{\geq 0}$ is simple (more generally, all 2-dimensional polytopes are simple). All in all, each vertex of $\Pi_{\geq 0}$ has 2 zero coordinates and 6 positive coordinates in edge coordinates, so that this vertex is bijective. These vertices are represented in edge basis in Figures 18d–18h.

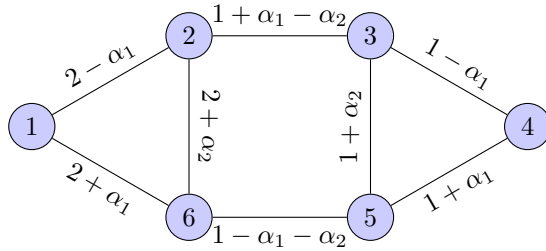
► **Example 9** (Non-essential matching problem). *Figure 19 shows the same codomino graph as in Example 8, with the same basis of $\ker(A)$, but with the vector of arrival rates $\lambda = (2, 4, 4, 2, 2, 2)$. A particular solution to (CE) is $\mu^\circ = (1, 1, 2, 1, 1, 1, 1, 0)$, and the general solution is shown in Figure 19b.*

The inequalities are listed in Figure 19a. The 2-dimensional convex polytope $\Pi_{\geq 0}$ is shown in kernel basis in Figure 19c. All inequalities are tight, but only one is irredundant, so we conclude that the matching problem (G, λ) is not essential, even if the polytope $\Pi_{\geq 0}$ is still simple. Correspondingly, even if each vertex belongs to exactly two facets, they all have more than 2 zero coordinates, so none of them is bijective. For example, the vertex $(1, -1)$ in kernel basis has coordinates $(0, 2, 4, 0, 0, 0, 2, 0)$ in edge basis (Figure 19f). This vertex has 5 zero coordinates in edge coordinates (and only 3 positive coordinates) even if it belongs to only 2 facets.

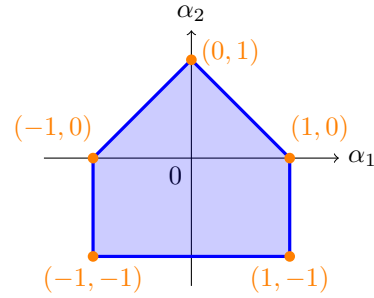
► **Example 10** (Non-simple polytope). *We finally exhibit an essential matching problem with a non-simple associated polytope. As 2-dimensional polytopes are simple, we need to consider a more complex example. We consider the matching problem of Figure 20a. The arrival rate is $\lambda = (3, 3, 6, 3, 4, 4, 6, 3, 4, 4) \in \mathbb{R}^{10}$. The particular solution and kernel basis are shown*

Edge basis	Kernel basis	Tight?	Irredundant?
$\mu_{1,2} \geq 0$	$\alpha_1 \leq 2$	✗	✗
$\mu_{1,6} \geq 0$	$\alpha_1 \geq -2$	✗	✗
$\mu_{2,3} \geq 0$	$\alpha_1 - \alpha_2 \geq -1$	✓	✓
$\mu_{2,6} \geq 0$	$\alpha_2 \geq -2$	✗	✗
$\mu_{3,4} \geq 0$	$\alpha_1 \leq 1$	✓	✓
$\mu_{3,5} \geq 0$	$\alpha_2 \geq -1$	✓	✓
$\mu_{4,5} \geq 0$	$\alpha_1 \geq -1$	✓	✓
$\mu_{5,6} \geq 0$	$\alpha_1 + \alpha_2 \leq 1$	✓	✓

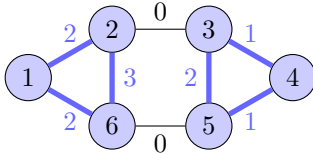
(a) Inequalities.



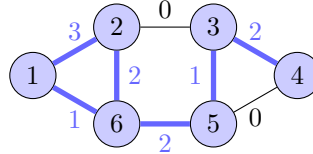
(b) Generic solution to (CE).



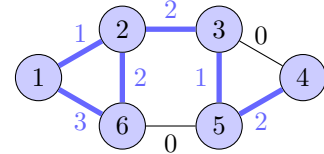
(c) Polytope $\Pi_{\geq 0}$ in kernel basis.



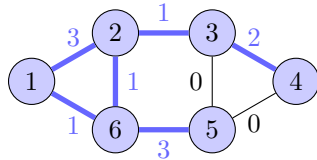
(d) Edge coordinates of $(0, 1)$.



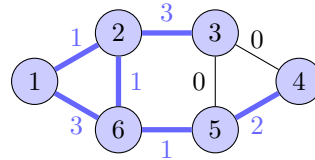
(e) Edge coordinates of $(-1, 0)$.



(f) Edge coordinates of $(1, 0)$.



(g) Edge coordinates of $(-1, -1)$.



(h) Edge coordinates of $(1, -1)$.

■ **Figure 18** An essential matching problem (G, λ) with a simple polytope $\Pi_{\geq 0}$. The vector of arrival rates is $\lambda = (4, 5, 3, 2, 3, 5) \in \mathbb{R}^6$, a particular solution to (CE) is $\mu^\circ = (2, 2, 1, 2, 1, 1, 1, 1) \in \mathbb{R}^8$, and the chosen base vectors for $\ker(A)$ are $b_1 = (-1, 1, 1, 0, -1, 0, 1, -1)$ and $b_2 = (0, 0, -1, 1, 0, 1, 0, -1)$.

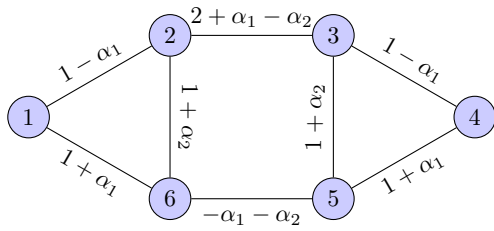
on the edges. The set $\Pi_{\geq 0}$, shown in Figure 20b in kernel basis, is an Egyptian pyramid characterized by the following tight inequalities:

$$\alpha_3 \geq 0, \quad 1 + \alpha_1 - \alpha_3 \geq 0, \quad 1 - \alpha_1 - \alpha_3 \geq 0, \quad 1 + \alpha_2 - \alpha_3 \geq 0, \quad 1 - \alpha_2 - \alpha_3 \geq 0.$$

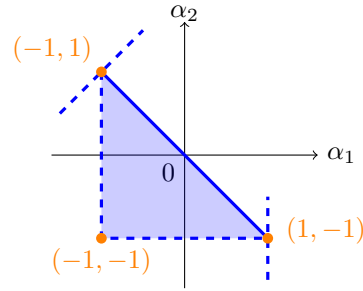
These five inequalities are irredundant (each one corresponds to exactly one of the five facets), so we conclude that the matching problem (G, λ) is essential. In kernel basis, the vertices of this convex polytope are $(-1, -1, 0)$, $(1, -1, 0)$, $(1, 1, 0)$, $(-1, 1, 0)$, and $(0, 0, 1)$. These vertices are shown in edge basis in Figures 20c–20g. The polytope $\Pi_{\geq 0}$ is not simple because the vertex $(0, 0, 1)$ (the “top” of the pyramid) belongs to 4 facets, while the polytope has dimension 3. Consistently, we can see in Figure 20g that this vertex has 4 zero coordinates

Edge basis	Kernel basis	Tight?	Irredundant?
$\mu_{1,2} \geq 0$	$\alpha_1 \leq 1$	✓	✗
$\mu_{1,6} \geq 0$	$\alpha_1 \geq -1$	✓	✗
$\mu_{2,3} \geq 0$	$\alpha_1 - \alpha_2 \geq -2$	✓	✗
$\mu_{2,6} \geq 0$	$\alpha_2 \geq -1$	✓	✗
$\mu_{3,4} \geq 0$	$\alpha_1 \leq 1$	✓	✗
$\mu_{3,5} \geq 0$	$\alpha_2 \geq -1$	✓	✗
$\mu_{4,5} \geq 0$	$\alpha_1 \geq -1$	✓	✗
$\mu_{5,6} \geq 0$	$\alpha_1 + \alpha_2 \leq 0$	✓	✓

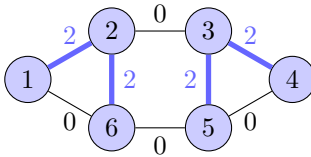
(a) Inequalities



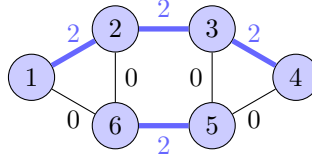
(b) Generic solution to (CE).



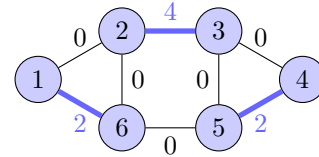
(c) Polytope $\Pi_{\geq 0}$ in kernel basis. Dashed lines show tight redundant inequalities.



(d) Edge coordinates of $(-1, 1)$.



(e) Edge coordinates of $(-1, -1)$.



(f) Edge coordinates of $(1, -1)$.

■ **Figure 19** Non-essential matching problem (G, λ) with a simple polytope $\Pi_{\geq 0}$. The vector of arrival rates is $\lambda = (2, 4, 4, 2, 2, 2) \in \mathbb{R}^6$, a particular solution to (CE) is $\mu^\circ = (1, 1, 2, 1, 1, 1, 1, 0) \in \mathbb{R}^8$, and the chosen base vectors for $\ker(A)$ are $b_1 = (-1, 1, 1, 0, -1, 0, 1, -1)$ and $b_2 = (0, 0, -1, 1, 0, 1, 0, -1)$.

and only 9 positive coordinates in edge basis; the subgraph defined by the support of this vertex is injective-only.

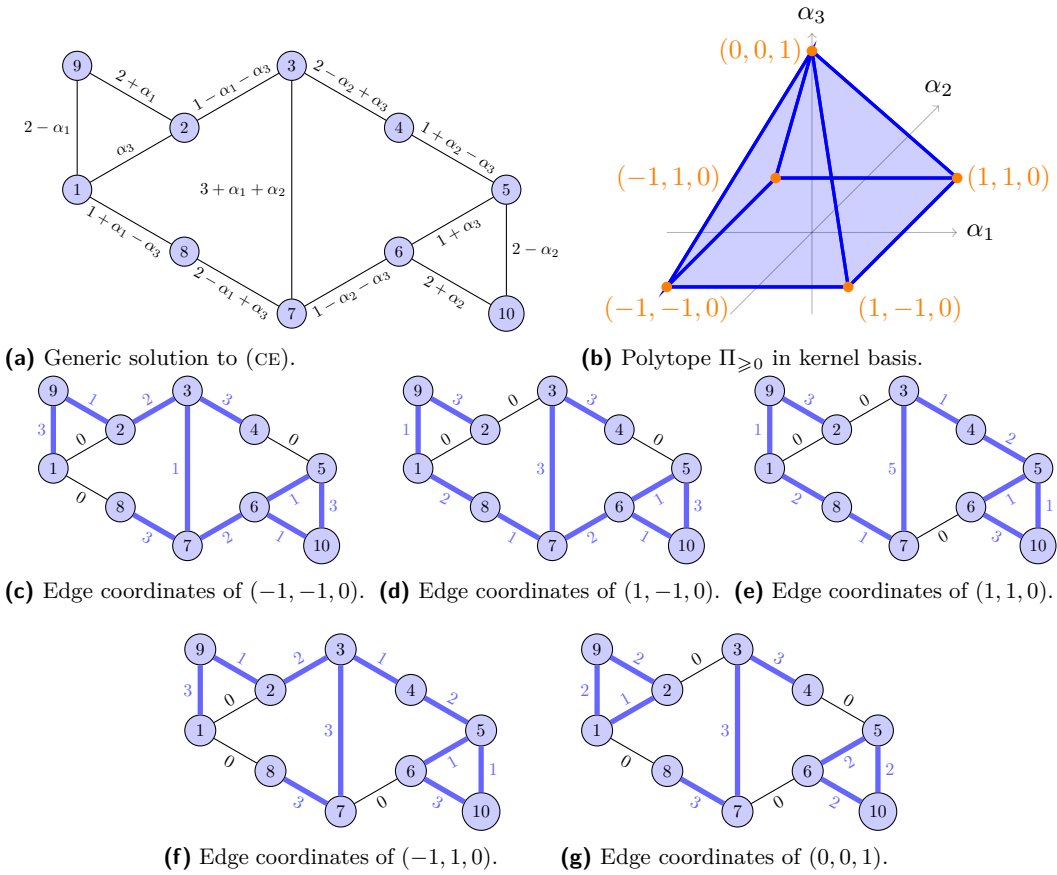
In light of these examples, we can give the following characterization of the bijective vertices of $\Pi_{\geq 0}$.

► **Proposition 30.** *Let μ be a vertex of $\Pi_{\geq 0}$. The following statements are equivalent:*

(i) μ is bijective.

(ii) μ belongs to exactly d facets of $\Pi_{\geq 0}$ and none of the inequalities tight for μ is redundant. In particular, all vertices of $\Pi_{\geq 0}$ are bijective if, and only if, the matching problem (G, λ) is essential and the polytope $\Pi_{\geq 0}$ is simple.

Proof. To prove the equivalence of i and ii, we first remark that the number of zero edge coordinates of a vector $\mu \in \Pi_{\geq 0}$ is by definition the number of inequalities that are tight for μ . It is in particular at least the number of facets that intersect μ , with equality if, and only if, none of the inequalities tight for μ is redundant.



■ **Figure 20** Essential matching problem (G, λ) with a non-simple polytope $\Pi_{\geq 0}$. The arrival rate is $\lambda = (3, 3, 6, 3, 4, 4, 6, 3, 4, 4) \in \mathbb{R}^{10}$. A particular solution and the chosen base vectors for $\ker(A)$ are implicitly shown on the edges of Figure 20a.

If a vector μ is a vertex of $\Pi_{\geq 0}$, then μ belongs to at least d facets of $\Pi_{\geq 0}$. Now, the vector μ is bijective if, and only if, d of its coordinates are zero, which in view of the remark above is equivalent to say that μ belongs to exactly d facets of $\Pi_{\geq 0}$ and none of the inequalities tight for μ is redundant.

As for the last statement, it follows directly from Definition 29. ◀

► **Remark 10.** We can further explore the relationship between vertices and their support graph. Again consider a vertex μ of $\Pi_{\geq 0}$, and let G^* denote the support graph of μ and A^* its incidence matrix. According to Definitions 2, 3, and 5, the subgraph G^* is injective if and only if each connected component of G^* is either a tree or a unicyclic graph with an odd cycle. For each connected component of G^* that is a tree, and therefore a bipartite graph with parts V_+ and V_- , the existence of vertex μ implies that

$$\sum_{i \in V_+} \lambda_i = \sum_{i \in V_-} \lambda_i. \tag{29}$$

This equation follows by summing (CE-1) over the nodes in V_+ on the one hand, summing (CE-1) over the nodes in V_- on the other hand, and verifying that the left-hand sides of both equations are equal. In fact, one can verify that the vector λ belongs to the image of A^* if and only if λ satisfies (29) for each connected component of G^* that is a tree. Note that

this condition is void if G^* is bijective because, in this case, none of the connected components of G^* is a tree.

Conversely, one can wonder which injective subgraph G^* of G defines a vertex of $\Pi_{\geq 0}$. Satisfying (29) for each tree connected component of G^* only guarantees the existence of a (unique) solution $z \in \mathbb{R}^p$ to the conservation equation $A^*z = \lambda$. If each coordinate of z is positive, then we indeed obtain a vertex of $\Pi_{\geq 0}$ by embedding z in \mathbb{R}^m with zero padding; otherwise, G^* does not define a vertex of $\Pi_{\geq 0}$.

C.4 Proof of Proposition 14 in Section 5.2.2 (Probability of bijectivity).

Let C denote the set of normalized arrival rate vectors λ such that (G, λ) is stabilizable:

$$C = \{\lambda \in \Delta^{n-1} : (G, \lambda) \text{ stabilizable}\}.$$

To prove Proposition 14, we proceed as follows:

- We prove that C is non-empty and convex, hence measurable.
- We prove that C is an open set (for the Lebesgue measure), so its probability is positive and the conditional probability given C is well-defined.
- We prove that if a vertex is injective-only, the vector λ must respect a constraint that reduces the dimension of the possible solutions, meaning that the measure of the event is null.

We first prove the different properties of C .

C is not empty. We use Remark 1: consider $\lambda = A\mu$, with $\mu = (\frac{1}{2m}, \dots, \frac{1}{2m})$. The matching problem (G, λ) is stabilizable as all coordinates of μ are positive. Moreover, one can check that $\lambda = (\frac{d_1}{2m}, \dots, \frac{d_m}{2m})$, where d_i is the degree of node i , so $\lambda \in \Delta^{n-1}$ ($\sum_i d_i = 2m$). Hence, $\lambda \in C$.

C is convex. Let λ_a and λ_b be two vectors of C . Let $\mu_a \in \mathbb{R}_{>0}^m$ and $\mu_b \in \mathbb{R}_{>0}^m$ be positive solutions to $A\mu_a = \lambda_a$ and $A\mu_b = \lambda_b$ respectively (such solutions exist as stated in Proposition 8). For $0 < x < 1$, consider $\lambda = x\lambda_a + (1-x)\lambda_b$. $\lambda \in \Delta^{n-1}$ because Δ^{n-1} is convex. Moreover, we have $A\mu = \lambda$, with $\mu = x\mu_a + (1-x)\mu_b$. As $\mathbb{R}_{>0}^m$ is convex, $\mu \in \mathbb{R}_{>0}^m$, so (G, λ) is stabilizable (Proposition 8), hence $\lambda \in C$.

C is open in Δ^{n-1} . Let $\lambda \in C$. For some $\epsilon > 0$, let λ_ϵ be an element of Δ^{n-1} such that $\|\lambda - \lambda_\epsilon\|_1 < \epsilon$. To prove that C is open, we just need to show that $\lambda_\epsilon \in C$ if ϵ is small enough. Let $\mu \in \mathbb{R}_{>0}^m$ be a positive solution to $A\mu = \lambda$. Using Equations (18) and (19), we can write $\mu = A^+\lambda + \mu_\alpha$, with $\mu_\alpha = \mu - A^+\lambda = (\text{Id}_{m \times m} - A^+A)\mu \in \ker(A)$. Consider now $\mu_\epsilon = A^+\lambda_\epsilon + \mu_\alpha$. We have $A\mu_\epsilon = AA^+\lambda_\epsilon + A\mu_\alpha = \lambda_\epsilon$ so $(\lambda_\epsilon, \mu_\epsilon)$ checks (CE). Moreover, $\|\mu - \mu_\epsilon\|_1 = \|A^+(\lambda - \lambda_\epsilon)\|_1 < \|A^+\|_1 \epsilon$, where $\|A^+\|_1 < +\infty$ is the operator norm of A^+ associated to the 1-norm. All coordinates of μ are positive, so if ϵ is small enough, it also the case for μ_ϵ , meaning that (G, λ_ϵ) is stabilizable (Proposition 8). In other words, $\lambda_\epsilon \in C$.

This establishes that C (i.e., the stabilizability of (G, λ)) defines a proper conditional probability. We now examine the conditions under which the polytope $\Pi_{\geq 0}$ of a stabilizable problem (G, λ) possesses an injective-only vertex.

Consider $\lambda \in \Delta^{n-1}$. Let μ be a solution to $A\mu = \lambda$. Using Definition 11 and the pseudoinverse approach from Section 5.1.2, the kernel coordinates of μ are $\alpha = B^+(\text{Id}_{m \times m} - A^+A)\mu$, so we can write

$$\mu = B\alpha + A^+\lambda = D\gamma,$$

where $D = (B|A^+)$ is the $m \times m$ horizontal concatenation of B and A^+ , and $\gamma = (\alpha_1, \dots, \alpha_d, \lambda_1, \dots, \lambda_n)$ is the concatenation of α and λ .

Notice that D depends solely on G and is invertible. Its inverse can be verified as the vertical concatenation of $B^+(\text{Id}_{m \times m} - A^+A)$ and A : any matching rate vector μ uniquely defines λ and α .

We now assume that one vertex of the polytope is injective-only and look for the implications for λ .

The existence of an injective-only vertex implies that there exists a subset of edges $K \subset E$, with $|K| > d$, such that μ is null on K (remind that $d = m - n$ is the dimension of $\ker(A)$). Let $D|_K$ denote the $|K| \times m$ submatrix of D formed by the rows corresponding to K , and let $B|_K$ denote the d first columns of $D|_K$, that is the part of $D|_K$ that handles α . As $|K| > d$, we have $\ker((B|_K)^\top) \neq \{0\}$, meaning that there exists a non-trivial linear combination $v \in \mathbb{R}^{|K|} \setminus \{0\}$ of the rows of $B|_K$ such that $vB|_K = 0$.

Let $A^+|_K$ denotes the n last columns of $D|_K$, that is the part of $D|_K$ that handles λ . We know that $vD|_K \neq 0$ because D is invertible, so its rows are linearly independent. As $vB|_K = 0$, this means that $vA^+|_K \neq 0$. The vector $u = vA^+|_K$ can be seen as a non-trivial linear combination of the coordinates of λ .

We can now conclude: if the polytope $\Pi_{\geq 0}$ associated to some $\lambda \in C$ admits an injective-only vertex, we have a subset of edges $K \subset E$, with $|K| > d$ such that $D|_K \gamma = 0$. In particular, we must have $vD|_K \gamma = 0$, which simplifies as $u\lambda = 0$. In other words, for μ to be null over K , the vector λ must belong to some hyperplane L_u of dimension $n - 1$ that can be built solely from G and K . As $\Delta^{n-1} \neq L_u$ (one is affine, the other is not), the intersection $\Delta^{n-1} \cap L_u$, if it exists, has a dimension at most $n - 2$. In particular, the Lebesgue measure of $\Delta^{n-1} \cap L_u$ (over Δ^{n-1}) is null, meaning that the probability to nullify all edges of K is null.

The number of subsets $K \subset E$ such that $|K| > d$ is finite, so the probability to nullify all edges of any of them is also null, so the probability that the polytope $\Pi_{\geq 0}$ associated to some $\lambda \in C$ admits an injective-only vertex is null. \blacktriangleleft

D Supplementary material of Section 6 (Stable policy optimization).

D.1 Proof of Proposition 19

We consider an arbitrary Markovian, stationary matching policy that matches at most $\mathcal{M} \in \mathbb{N}$ pairs of items in one time step. In particular, we consider a matching policy that only depends on the current state of the underlying Markov chain of the matching problem. To be more precise, consider an irreducible, aperiodic, and positive recurrent DTMC $(S_t, t \in \mathbb{N})$ defined in a similar way as in Appendix A, such that Q_t and Φ_t are functions of S_t , where $\Phi_{t,i}$ is the number of class- i items matched at time t for all $i \in V$. If $Q_t = S_t$, then the matching policy simply depends on the current queue lengths, e.g., ML policy. Now, given an injective-only vertex μ , the following proposition formalizes the trade-off between closeness to μ and the queue lengths in the steady-state (denoted by Q_∞) for any such matching policy. Proposition 19 is a special case of Extended Proposition 19 with $\mathcal{M} = 1$. The proof of Extended Proposition 19 is given in the rest of the appendix.

► **Extended Proposition 19.** *Consider an injective-only vertex μ of $\Pi_{\geq 0}$. Then, under any Markovian, stationary matching policy Φ that matches at most $\mathcal{M} \in \mathbb{N}$ pairs of items in one-time step, if $\|\mu(\Phi) - \mu\|_1 \leq \epsilon$, then $\mathbb{E}[\|Q_\infty\|_2^2] \geq \Omega(1/\epsilon)$, where Q_∞ is the queue length under the steady state of $(S_t, t \in \mathbb{N})$.*

As μ is injective-only, there exists a connected component of $G^* = (V, E^*)$ that is bipartite by Definitions 3 and 4. Denote such a connected component by $G^\circ = (V^+ \cup V^-, E^\circ)$, where V^+ and V^- are the (disjoint) parts of the connected component, and $E^\circ \subseteq E^*$. Now, consider the following test function:

$$W(q) = \left\langle \mathbf{1}^{(V^+, V^-)}, q \right\rangle^2, \quad q \in \mathcal{Q},$$

where $\mathbf{1}_i^{(V^+, V^-)} = 1$ for $i \in V^+$, -1 for $i \in V^-$, and 0 otherwise. For each $t \in \mathbb{N}$, let A_t denote the class of the incoming item at time t , so that $A_t = \mathbf{1}_{I_t}$, where I_t is the class of $(t+1)$ -th item. Thus, $\lambda_i / \sum_{j \in V} \lambda_j$ is the probability that $A_t = \mathbf{1}_i$. We also define $\delta M_{t,k} = M_{t,k} - M_{t-1,k}$ to be the number of items matched using edge $k \in E$ at time t . Thus, we have $\sum_{k \in E_i} \delta M_{t,k} = \Phi_{t,i}$ for all $i \in V$. For the rest of the proof, we consider the steady-state of the Markov chain underlying the matching problem. In particular, we consider the steady-state of the (discrete-time) Markov chain $(S_t, t \in \mathbb{N})$, of which $Q = (Q_t, t \in \mathbb{N})$ and $\Phi = (\Phi_t, t \in \mathbb{N})$ are functions.

If $\mathbb{E}[W(Q)] = \infty$, then we have $\mathbb{E}[\sum_{i \in V} Q_i^2] = \infty$ which completes the proof of the proposition. Indeed, for each $q \in \mathbb{N}^n$,

$$\begin{aligned} W(q) &= \left(\sum_{i \in V^+} q_i \right)^2 + \left(\sum_{i \in V^-} q_i \right)^2 - 2 \left(\sum_{i \in V^+} q_i \right) \left(\sum_{i \in V^-} q_i \right), \\ &\leq \left(\sum_{i \in V^+} q_i \right)^2 + \left(\sum_{i \in V^-} q_i \right)^2, \\ &\leq |V^+| \sum_{i \in V^+} q_i^2 + |V^-| \sum_{i \in V^-} q_i^2. \end{aligned}$$

Now, assume instead that $\mathbb{E}[W(Q)] < \infty$. In stationarity, we have $\mathbb{E}[W(Q_{t+1})] = \mathbb{E}[W(Q_t)]$. In particular, we take the expectation under the steady state distribution of $(S_t, t \in \mathbb{N})$.

Using this, we get

$$\begin{aligned}
0 &= \mathbb{E} \left[\left\langle \mathbf{1}^{(V^+, V^-)}, Q_{t+1} \right\rangle^2 - \left\langle \mathbf{1}^{(V^+, V^-)}, Q_t \right\rangle^2 \right], \\
&= \mathbb{E} \left[\left\langle \mathbf{1}^{(V^+, V^-)}, Q_t + A_t - \Phi_t \right\rangle^2 - \left\langle \mathbf{1}^{(V^+, V^-)}, Q_t \right\rangle^2 \right], \\
&= \mathbb{E} \left[\left\langle \mathbf{1}^{(V^+, V^-)}, A_t - \Phi_t \right\rangle^2 \right] + 2\mathbb{E} \left[\left\langle \mathbf{1}^{(V^+, V^-)}, Q_t \right\rangle \left\langle \mathbf{1}^{(V^+, V^-)}, A_t - \Phi_t \right\rangle \right], \\
&\stackrel{(a)}{=} \mathbb{E} \left[\left\langle \mathbf{1}^{(V^+, V^-)}, A_t - \Phi_t \right\rangle^2 \right] - 2\mathbb{E} \left[\left\langle \mathbf{1}^{(V^+, V^-)}, Q_t \right\rangle \left\langle \mathbf{1}^{(V^+, V^-)}, \Phi_t \right\rangle \right], \\
&\stackrel{(b)}{\geq} \frac{\sum_{i \in V^+ \cup V^-} \lambda_i}{\sum_{i \in V} \lambda_i} - 2\mathbb{E} \left[\left\langle \mathbf{1}^{(V^+, V^-)}, Q_t \right\rangle \left\langle \mathbf{1}^{(V^+, V^-)}, \Phi_t \right\rangle \right], \\
&\geq \frac{\sum_{i \in V^+ \cup V^-} \lambda_i}{\sum_{i \in V} \lambda_i} - 2\sqrt{\mathbb{E} \left[\left\langle \mathbf{1}^{(V^+, V^-)}, Q_t \right\rangle^2 \right] \mathbb{E} \left[\left\langle \mathbf{1}^{(V^+, V^-)}, \Phi_t \right\rangle^2 \right]} \quad (\text{by Cauchy-Schwarz}), \\
&\stackrel{(c)}{\geq} \frac{\sum_{i \in V^+ \cup V^-} \lambda_i}{\sum_{i \in V} \lambda_i} - 4\sqrt{\mathbb{E} \left[\left(\sum_{i \in V} Q_{t,i} \right)^2 \right] \mathbb{E} \left[\left(\sum_{k \in E \setminus E^*} \delta M_{t,k} \right)^2 \right]}, \\
&\stackrel{(d)}{\geq} \frac{\sum_{i \in V^+ \cup V^-} \lambda_i}{\sum_{i \in V} \lambda_i} - 4\sqrt{\mathcal{M} \mathbb{E} \left[\left(\sum_{i \in V} Q_{t,i} \right)^2 \right] \sum_{k \in E \setminus E^*} \mathbb{E} [\delta M_{t,k}]}, \\
&\stackrel{(e)}{\geq} \frac{\sum_{i \in V^+ \cup V^-} \lambda_i}{\sum_{i \in V} \lambda_i} - 4\sqrt{\mathcal{M} \epsilon \mathbb{E} \left[\left(\sum_{i \in V} Q_{t,i} \right)^2 \right]}. \tag{30}
\end{aligned}$$

Here, (a) follows by noting that $\mathbb{E} \left[\left\langle \mathbf{1}^{(V^+, V^-)}, A_t \right\rangle \right] = \mathbb{E} \left[\left\langle \mathbf{1}^{(V^+, V^-)}, \lambda \right\rangle \right] = 0$, as μ is a feasible solution to the conservation equations (CE) and $G^\circ = (V^+ \cup V^-, E^\circ)$ is a connected component of the subgraph of G induced by support of μ . Note that we also use the independence of A_t and Q_t in this step. Next, (b) follows by noting that $\left\langle \mathbf{1}^{(V^+, V^-)}, A_t - \Phi_t \right\rangle^2 \geq 1$ whenever $\left\langle \mathbf{1}^{(V^+, V^-)}, A_t \right\rangle = \pm 1$ (because $\left\langle \mathbf{1}^{(V^+, V^-)}, \Phi_t \right\rangle$ is an even number, as each match is counted twice), and that the event $\left\langle \mathbf{1}^{(V^+, V^-)}, A_t \right\rangle = \pm 1$ happens with probability $\frac{\sum_{i \in V^+ \cup V^-} \lambda_i}{\sum_{i \in V} \lambda_i}$. Further, (c) follows by noting that

$$\begin{aligned}
\left| \left\langle \mathbf{1}^{(V^+, V^-)}, \Phi_t \right\rangle \right| &= \left| \sum_{i \in V^+} \Phi_{t,i} - \sum_{i \in V^-} \Phi_{t,i} \right| = \left| \sum_{i \in V^+} \sum_{j \in V_i} \delta M_{t,\{i,j\}} - \sum_{i \in V^-} \sum_{j \in V_i} \delta M_{t,\{i,j\}} \right|, \\
&\leq \sum_{i \in V^+} \sum_{j \in V_i \setminus V^-} \delta M_{t,\{i,j\}} + \sum_{i \in V^-} \sum_{j \in V_i \setminus V^+} \delta M_{t,\{i,j\}}, \\
&\quad + \left| \sum_{i \in V^+} \sum_{j \in V_i \cap V^-} \delta M_{t,\{i,j\}} - \sum_{i \in V^-} \sum_{j \in V_i \cap V^+} \delta M_{t,\{i,j\}} \right|, \quad (\text{Triangle Inequality}) \\
&\stackrel{(*)}{=} \sum_{i \in V^+} \sum_{j \in V_i \setminus V^-} \delta M_{t,\{i,j\}} + \sum_{i \in V^-} \sum_{j \in V_i \setminus V^+} \delta M_{t,\{i,j\}}, \\
&\stackrel{(**)}{\leq} 2 \sum_{k \in E \setminus E^*} \delta M_{t,k},
\end{aligned}$$

where (*) follows by noting that $\sum_{i \in V^+} \sum_{j \in V_i \cap V^-} \delta M_{t, \{i, j\}} = \sum_{i \in V^-} \sum_{j \in V_i \cap V^+} \delta M_{t, \{i, j\}}$ as the set of edges that we are summing over are the same in the left and right hand side. In particular, we are adding over all the edges between V^+ and V^- . Next, (**) follows by noting that all edges connecting $i \in V^+$ to $j \in V \setminus V^-$ are not in E^* . Similarly, all edges connecting $i \in V^-$ to $j \in V \setminus V^+$ are also not in E^* by the definition of G° . Now, (d) follows by the assumption that at most \mathcal{M} pairs of items can be matched in a single time step. Lastly, (e) follows as $\mathbb{E}[\delta M_t] = \mu(\Phi)$ in the steady state and

$$\|\mu(\Phi) - \mu\|_1 \leq \epsilon \implies \sum_{k \in E \setminus E^*} \mu(\Phi)_k \leq \epsilon.$$

Now, to complete the proof, we use (30) which implies

$$\mathbb{E} \left[\sum_{i \in V} Q_{\infty, i}^2 \right] \geq \frac{1}{|V|} \mathbb{E} \left[\left(\sum_{i \in V} Q_{\infty, i} \right)^2 \right] \stackrel{(30)}{\geq} \frac{(\sum_{i \in V^+ \cup V^-} \lambda_i)^2}{16\mathcal{M}|V| (\sum_{i \in V} \lambda_i)^2} \cdot \frac{1}{\epsilon} = \Omega \left(\frac{1}{\epsilon} \right).$$

Note that the first inequality follows as $\|q\|_2^2 \geq \|q\|_1^2 / |V|$ for all $q \in \mathbb{R}^{|V|}$. This completes the proof. \blacktriangleleft

D.2 Proof of Proposition 18

We first prove two intermediary results, Lemmas 31 and 32, that will be instrumental in proving the upper bound for the mean queue length in Proposition 18.

► **Lemma 31.** *Consider a matching problem (G, λ) . If the CRPG $\delta(G, \lambda)$ is positive, then the ML policy stabilizes (G, λ) , and*

$$\mathbb{E} \left[\sum_{i \in V} Q_i \right] \leq \frac{n}{2\delta(G, \lambda)} \sum_{i \in V} \lambda_i.$$

Proof. Consider the quadratic test function W defined on \mathcal{Q}_G by

$$W(q) = \sum_{i \in V} q_i^2, \quad q \in \mathcal{Q}_G.$$

We define the drift of this test function by $\Delta W(q) = \mathbb{E}[W(Q_{t+1}) - W(Q_t) | Q_t = q]$ for each $q \in \mathcal{Q}_G$. Lastly, we let $\mathcal{I}_q = \{i \in V : q_i > 0\}$ denote the support of q , for each $q \in \mathcal{Q}_G$. Now, we simplify the drift under the ML policy using the queue evolution recursion given by (2). For all $q \in \mathcal{Q}_G \setminus \{0\}$, we have

$$\begin{aligned} \Delta W(q) &= \mathbb{E} \left[\sum_{i \in V} (Q_{t,i} + \mathbf{1}\{i = I_t\})^2 \mathbf{1}\{J_t = \perp\} + (Q_{t,i} - \mathbf{1}\{i = J_t\})^2 \mathbf{1}\{J_t \neq \perp\} - Q_{t,i}^2 \mid Q_t = q \right], \\ &= \mathbb{E} \left[\sum_{i \in V} \mathbf{1}\{i = I_t, J_t = \perp\} + \mathbf{1}\{i = J_t\} + 2(\mathbf{1}\{i = I_t, J_t = \perp\} - \mathbf{1}\{i = J_t\}) Q_{t,i} \mid Q_t = q \right], \\ &\stackrel{(a)}{=} 1 + 2 \sum_{i \in V} q_i \mathbb{E} \left[\mathbf{1}\{i = I_t, J_t = \perp\} - \mathbf{1}\{i = J_t\} \mid Q_t = q \right], \\ &\stackrel{(b)}{=} 1 + \frac{2}{\sum_{i \in V} \lambda_i} \sum_{i \in V} \lambda_i q_i - 2 \sum_{i \in V} q_i \mathbb{E} \left[\mathbf{1}\{i = J_t\} \mid Q_t = q \right], \end{aligned}$$

$$\begin{aligned}
&\stackrel{(c)}{=} 1 + \frac{2}{\sum_{i \in V} \lambda_i} \sum_{i \in V} \lambda_i q_i - \frac{2}{\sum_{i \in V} \lambda_i} \sum_{j \in V} \lambda_j \max_{i \in V_j} q_i, \\
&\stackrel{(d)}{=} 1 + \frac{2}{\sum_{i \in V} \lambda_i} \sum_{i \in \mathcal{I}_q} \lambda_i q_i - \frac{2}{\sum_{i \in V} \lambda_i} \sum_{j \in V(\mathcal{I}_q)} \lambda_j \max_{i \in V_j} q_i, \\
&\stackrel{(e)}{\leq} 1 - \frac{2\delta(G, \lambda)}{|\mathcal{I}_q| \sum_{i \in V} \lambda_i} \sum_{i \in \mathcal{I}_q} q_i, \\
&= 1 - \frac{2\delta(G, \lambda)}{|\mathcal{I}_q| \sum_{i \in V} \lambda_i} \sum_{i \in V} q_i, \\
&\leq 1 - \frac{2\delta(G, \lambda)}{n \sum_{i \in V} \lambda_i} \sum_{i \in V} q_i. \tag{31}
\end{aligned}$$

Here, (a) follows as

$$\sum_{i \in V} \mathbb{1}\{i = I_t, J_t = \perp\} + \sum_{i \in V} \mathbb{1}\{i = J_t\} = \mathbb{1}\{J_t = \perp\} + \mathbb{1}\{J_t \neq \perp\} = 1.$$

Next, (b) follows as for all $i \in V$, we have

$$\begin{aligned}
q_i \mathbb{E}[\mathbb{1}\{i = I_t, J_t = \perp\} | Q_t = Q] &= q_i \mathbb{P}\{i = I_t, J_t = \perp | Q_t = Q\}, \\
&= q_i \mathbb{P}\{J_t = \perp | Q_t = Q, i = I_t\} \mathbb{P}\{i = I_t | Q_t = Q\}, \\
&= q_i \frac{\lambda_i}{\sum_{i \in V} \lambda_i} \mathbb{1}\left\{\max_{j \in V_i} q_j = 0\right\} = q_i \frac{\lambda_i}{\sum_{i \in V} \lambda_i}.
\end{aligned}$$

The last equality follows from the fact that the ML policy is greedy, which implies that, if $q_i > 0$, then $q_j = 0$ for all $j \in V_i$. Now, (c) follows by using the definition of ML policy. In particular, we have

$$\begin{aligned}
\sum_{i \in V} q_i \mathbb{E}[\mathbb{1}\{i = J_t\} | Q_t = q] &= \frac{1}{\sum_{j \in V} \lambda_j} \sum_{j \in V} \lambda_j \sum_{i \in V} q_i \mathbb{E}[\mathbb{1}\{i = J_t\} | Q_t = q, j = I_t], \\
&= \frac{1}{\sum_{j \in V} \lambda_j} \sum_{j \in V} \lambda_j \max_{i \in V_j} q_i.
\end{aligned}$$

The equality (d) follows by noting that \mathcal{I}_q is equal to the support of q , i.e., we have $\mathcal{I}_q = \{i \in V : q_i > 0\}$. The rest of the paragraph is dedicated to proving (e). Consider the bipartite graph

$$G_q = G(\mathcal{I}_q \cup V(\mathcal{I}_q), E_q) \quad \text{with} \quad E_q = \{\{i, j\} \in E : i \in \mathcal{I}_q, j \in V(\mathcal{I}_q)\}.$$

By definition of $\delta(G, \lambda)$, we have, for each $\mathcal{I} \subseteq \mathcal{I}_q$,

$$\sum_{i \in \mathcal{I}} \lambda_i \leq \sum_{j \in V(\mathcal{I})} \lambda_j - \delta(G, \lambda) = \sum_{\substack{j \in V(\mathcal{I}): \\ \exists i \in \mathcal{I}, \{i, j\} \in E_q}} \lambda_j - \delta(G, \lambda),$$

where the equality follows as $V(\mathcal{I}) \subseteq V(\mathcal{I}_q)$, and so, $V(\mathcal{I}) = V(\mathcal{I}) \cap \{j : \exists i \in \mathcal{I}, \{i, j\} \in E_q\}$ by the definition of G_q . Now, the above inequality implies that

$$\sum_{i \in \mathcal{I}} \left(\lambda_i + \frac{\delta(G, \lambda)}{|\mathcal{I}_q|} \right) \leq \sum_{\substack{j \in V(\mathcal{I}): \\ \exists i \in \mathcal{I}, \{i, j\} \in E_q}} \lambda_j \quad \forall \mathcal{I} \subseteq \mathcal{I}_q.$$

So, by the weighted version of Hall's marriage theorem [22, Lemma 2.5], there exists $\tilde{\mu} \in \mathbb{R}_{\geq 0}^{|E_Q|}$ such that

$$\lambda_i + \frac{\delta(G, \lambda)}{|\mathcal{I}_q|} = \sum_{\substack{j \in V(\mathcal{I}_q): \\ \{i, j\} \in E_q}} \tilde{\mu}_{i, j}, \quad \forall i \in \mathcal{I}_q, \quad \lambda_j \geq \sum_{\substack{i \in \mathcal{I}_q: \\ \{i, j\} \in E_q}} \tilde{\mu}_{i, j}, \quad \forall j \in V(\mathcal{I}_q).$$

Using the above, we get

$$\sum_{i \in \mathcal{I}_q} \lambda_i q_i - \sum_{j \in V(\mathcal{I}_q)} \lambda_j \max_{i' \in V_j} q_{i'} \leq \sum_{\{i, j\} \in E_q} \tilde{\mu}_{i, j} \left(q_i - \max_{i' \in V_j} q_{i'} \right) - \frac{\delta(G, \lambda)}{|\mathcal{I}_q|} \sum_{i \in \mathcal{I}_q} q_i \leq -\frac{\delta(G, \lambda)}{|\mathcal{I}_q|} \sum_{i \in \mathcal{I}_q} q_i,$$

where the last inequality follows as $\max_{i' \in V_j} q_{i'} \geq q_i$ for all $\{i, j\} \in E_q$. This completes the proof of (e).

Now, by (31) we immediately conclude that

$$\Delta W(q) \leq -1 \quad \text{for each } q \in \mathcal{Q}_G \text{ such that } \sum_{i \in V} q_i \geq \frac{n \sum_{i \in V} \lambda_i}{\delta(G, \lambda)}.$$

Thus, by the Foster-Lyapunov theorem [40], we conclude from (31) that the Markov chain under the ML policy is positive recurrent. Now, by the moment bound theorem [25, Proposition 6.14], we further conclude that in steady state, we have

$$\mathbb{E} \left[\sum_{i \in V} Q_i \right] \leq \frac{n}{2\delta(G, \lambda)} \sum_{i \in V} \lambda_i.$$

This completes the proof of the lemma. \blacktriangleleft

Now we turn our attention to studying the CRPG of (G', λ'_ϵ) . We start by defining additional notations for convenience. Define the set of “positive” vertices by $V^+ = \{i^+ : i \in V\}$ and the set of “negative” vertices by $V^- = \{i^- : i \in V\}$. Let $\mathbb{I}_{G'}$ be the set of all independent sets in G' . Note that, each independent set $\mathcal{I} \in \mathbb{I}_{G'}$ can be uniquely written as $\mathcal{I} = J^- \cup K^+$ where $J^- = \mathcal{I} \cap V^-$ and $K^+ = \mathcal{I} \cap V^+$ (with at least one of them nonempty). Observe we have $J \in \{\emptyset\} \cup \mathbb{I}_G$ and $K \in \{\emptyset\} \cup \mathbb{I}_{G^*}$. Now, we lower bound the CRPG of the augmented graph in the following lemma. In particular, combining the following lemma with Proposition 8 implies that (G', λ'_ϵ) is stabilizable whenever (G, λ) is.

► **Lemma 32.** *For the augmented problem (G', λ'_ϵ) , we have*

$$\delta(G', \lambda'_\epsilon) \geq \min \left\{ \epsilon \min_{i \in V} \lambda_i + \delta(G^*, \lambda), (1 - \epsilon) \min_{i \in V} \lambda_i + \delta(G^*, \lambda), \delta(G, \lambda) \right\}.$$

Proof. Consider $\mathcal{I} \in \mathbb{I}_{G'}$ and let $J \in \{\emptyset\} \cup \mathbb{I}_G$ and $K \in \{\emptyset\} \cup \mathbb{I}_{G^*}$ such that $J^- = \mathcal{I} \cap V^-$ and $K^+ = \mathcal{I} \cap V^+$, so that $\mathcal{I} = J^- \cup K^+$. Using the definition of G' , we can verify that, for each $0 < \epsilon < 1$, we have

$$\begin{aligned} \sum_{i \in V_{G'}(\mathcal{I})} \lambda'_{\epsilon, i} - \sum_{i \in \mathcal{I}} \lambda'_{\epsilon, i} &= \epsilon \left(\sum_{i \in V_G(J)} \lambda_i - \sum_{i \in J} \lambda_i \right) + (1 - \epsilon) \sum_{i \in V_G(J) \setminus V_{G^*}(K)} \lambda_i \\ &\quad + (1 - \epsilon) \left(\sum_{i \in V_{G^*}(K)} \lambda_i - \sum_{i \in K} \lambda_i \right) + \epsilon \sum_{i \in V_G(K) \setminus V_G(J)} \lambda_i. \end{aligned} \quad (32)$$

Now, we consider the following two cases.

Case I: Either $V_G(K) \not\subseteq V_G(J)$ or $V_G(J) \not\subseteq V_{G^*}(K)$. In this case, either the second term or the fourth term in (32) is non-zero; they are for instance bounded from below by $\min\{\epsilon, 1 - \epsilon\} \min_{i \in V} \lambda_i$. In addition, the first and the third term are bounded below by $\epsilon \delta(G^*, \lambda)$ and $(1 - \epsilon) \delta(G^*, \lambda)$ by the definition of δ (22). So, we have

$$\sum_{i \in V_{G'}(\mathcal{I})} \lambda'_{\epsilon, i} - \sum_{i \in \mathcal{I}} \lambda'_{\epsilon, i} \geq \min\{\epsilon, 1 - \epsilon\} \min_{i \in V} \lambda_i + \delta(G^*, \lambda) \quad \forall \mathcal{I} \in \mathbb{I}_{G'}. \quad (33)$$

Case II: We have $V_G(K) \subseteq V_G(J)$ and $V_G(J) \subseteq V_{G^*}(K)$. In this case, we must have $J \neq \emptyset$ and $K \neq \emptyset$. Thus, we have

$$V_{G^*}(K) \subseteq V_G(K) \subseteq V_G(J) \subseteq V_{G^*}(K) \implies V_{G^*}(K) = V_G(K).$$

Now, as $K \in \mathbb{I}_{G^*}$, we have $K \cap V_{G^*}(K) = \emptyset$. Thus, we have $K \cap V_G(K) = \emptyset$ which implies that $K \in \mathbb{I}_G$. Now, using the above, we get

$$\sum_{i \in V_{G^*}(K)} \lambda_i - \sum_{i \in K} \lambda_i = \sum_{i \in V_G(K)} \lambda_i - \sum_{i \in K} \lambda_i \geq \delta(G, \lambda), \quad \text{and} \quad \sum_{i \in V_G(J)} \lambda_i - \sum_{i \in J} \lambda_i \geq \delta(G, \lambda),$$

where the inequalities above follow by the definition of $\delta(G, \lambda)$ and noting that $J, K \in \mathbb{I}_G$. Now, using (32), we get

$$\sum_{i \in V_{G'}(\mathcal{I})} \lambda'_{\epsilon, i} - \sum_{i \in \mathcal{I}} \lambda'_{\epsilon, i} \geq \delta(G, \lambda). \quad (34)$$

Now, by combining (33) and (34), for all $\mathcal{I} \in \mathbb{I}_{G'}$ we get

$$\sum_{i \in V_{G'}(\mathcal{I})} \lambda'_{\epsilon, i} - \sum_{i \in \mathcal{I}} \lambda'_{\epsilon, i} \geq \min \left\{ \epsilon \min_{i \in V} \lambda_i + \delta(G^*, \lambda), (1 - \epsilon) \min_{i \in V} \lambda_i + \delta(G^*, \lambda), \delta(G, \lambda) \right\},$$

which completes the proof by definition of $\delta(G', \lambda'_\epsilon)$. \blacktriangleleft

Now that we have stated and proved Lemmas 31 and 32, we are ready to prove Proposition 18 below.

Proof of Proposition 18. We divide the proof into two parts. In the first part, we show that the expected queue length is $O(1/\epsilon)$. Then, the second part deals with the matching rates.

Controlling the expected queue length: By Lemma 32, we know that $\delta(G', \lambda'_\epsilon) > 0$. The corresponding queueing system is stable by Proposition 8. Furthermore, we have

$$\mathbb{E} \left[\sum_{i \in V} Q_i \right] = \mathbb{E} \left[\sum_{i \in V+UV^-} Q_i \right] \leq \frac{2n}{2\delta(G', \lambda'_\epsilon)} \sum_{i \in V} \lambda_i \leq \frac{n}{\min\{\epsilon \min_{i \in V} \lambda_i + \delta(G^*, \lambda), \delta(G, \lambda)\}} \sum_{i \in V} \lambda_i,$$

where the first inequality follows by applying Lemma 31 to the extended matching problem (G', λ'_ϵ) , and the second inequality follows by applying Lemma 32 and noting that, for all $\epsilon \in (0, 0.5)$, we have $\epsilon \leq 1 - \epsilon$. Now, by defining $C_1 = n \sum_{i \in V} \lambda_i$, the proof of the first part is complete.

Controlling the matching rates: Now, we show the second part of the proposition. Let Φ'_ϵ denote the ML policy adapted to (G', λ'_ϵ) . As explained at the beginning of Section 6.3, the policies Φ_ϵ and Φ'_ϵ are practically equivalent, except that Φ_ϵ is seen as a (non-greedy ϵ -dependent) policy adapted to the original problem (G, λ) , while Φ'_ϵ is seen as a (greedy) policy adapted to the (ϵ -dependent) extended problem (G', λ'_ϵ) . Consistently, $\mu(\Phi_\epsilon)$ is the

matching-rate vector (indexed by E) under the ϵ -filtering ML policy adapted to (G, λ) with filter E^* , while $\mu(\Phi'_\epsilon)$ denotes the matching-rate vector (indexed by E') under the ML policy adapted to (G', λ'_ϵ) . The rest of the proof is divided mainly in two parts: first we prove the following bound on our objective $\|\mu - \mu(\Phi_\epsilon)\|_1$ in terms of $\mu(\Phi'_\epsilon)$:

$$\|\mu - \mu(\Phi_\epsilon)\|_1 \leq \sum_{\{i,j\} \in E^*} |\mu_{i,j} - \mu(\Phi'_\epsilon)_{i^+,j^+}| + 2\epsilon \sum_{j \in V} \lambda_j. \quad (35)$$

Then, we show that $\sum_{\{i,j\} \in E^*} |\mu_{i,j} - \mu(\Phi'_\epsilon)_{i^+,j^+}| = O(\epsilon)$ using properties of ML policy to complete the proof. Intuitively, λ'_ϵ is designed in such a way that under any stable policy, all but $O(\epsilon)$ arrivals are matched using edges in E^+ . So, $\mu(\Phi'_\epsilon)_{i^+,j^+}$ must be close to $\mu_{i,j}$ for all $\{i,j\} \in E^*$.

We start by proving (35). As the ϵ -filtering ML policy mimics the ML policy adapted to (G', λ'_ϵ) , we have

$$\mu(\Phi_\epsilon(\mu))_{i,j} = \begin{cases} \mu(\Phi'_\epsilon)_{i^+,j^-} + \mu(\Phi'_\epsilon)_{i^-,j^+} + \mu(\Phi'_\epsilon)_{i^-,j^-} + \mu(\Phi'_\epsilon)_{i^+,j^+} & \text{if } \{i,j\} \in E^* \\ \mu(\Phi'_\epsilon)_{i^+,j^-} + \mu(\Phi'_\epsilon)_{i^-,j^+} + \mu(\Phi'_\epsilon)_{i^-,j^-} & \text{if } \{i,j\} \in E \setminus E^*, \end{cases}$$

Using the above, we can upper bound our objective $\|\mu - \mu(\Phi_\epsilon)\|_1$ in terms of $\mu(\Phi'_\epsilon)$ as follows:

$$\begin{aligned} \|\mu - \mu(\Phi_\epsilon)\|_1 &\stackrel{(a)}{\leq} \sum_{\{i,j\} \in E^*} |\mu_{i,j} - \mu(\Phi_\epsilon)_{i,j}| + \sum_{\{i,j\} \in E \setminus E^*} \mu(\Phi_\epsilon)_{i,j}, \\ &\stackrel{(b)}{\leq} \sum_{\{i,j\} \in E^*} |\mu_{i,j} - \mu(\Phi'_\epsilon)_{i^+,j^+}| + \sum_{\{i,j\} \in E} \left(\mu(\Phi'_\epsilon)_{i^+,j^-} + \mu(\Phi'_\epsilon)_{i^-,j^+} + \mu(\Phi'_\epsilon)_{i^-,j^-} \right), \\ &\stackrel{(c)}{\leq} \sum_{\{i,j\} \in E^*} |\mu_{i,j} - \mu(\Phi'_\epsilon)_{i^+,j^+}| + 2\epsilon \sum_{j \in V} \lambda_j. \end{aligned}$$

Here, (a) follows by recalling that E^* is defined as the support of μ . Next, (b) follows as, for all $\{i,j\} \in E^*$,

$$\begin{aligned} |\mu_{i,j} - \mu(\Phi_\epsilon)_{i,j}| &= |\mu_{i,j} - \mu(\Phi'_\epsilon)_{i^+,j^+} - \mu(\Phi'_\epsilon)_{i^+,j^-} - \mu(\Phi'_\epsilon)_{i^-,j^+} - \mu(\Phi'_\epsilon)_{i^-,j^-}|, \\ &\leq |\mu_{i,j} - \mu(\Phi'_\epsilon)_{i^+,j^+}| + \mu(\Phi'_\epsilon)_{i^+,j^-} + \mu(\Phi'_\epsilon)_{i^-,j^+} + \mu(\Phi'_\epsilon)_{i^-,j^-}. \end{aligned}$$

Lastly, (c) follows by noting that

$$\begin{aligned} \sum_{\{i,j\} \in E} \left(\mu(\Phi'_\epsilon)_{i^+,j^-} + \mu(\Phi'_\epsilon)_{i^-,j^+} \right) &\leq \sum_{j \in V} \sum_{\substack{i \in V: \\ \{i,j\} \in E}} \left(\mu(\Phi'_\epsilon)_{j^-,i^+} + \mu(\Phi'_\epsilon)_{j^-,i^-} \right) \stackrel{(*)}{=} \sum_{j \in V} \lambda'_{\epsilon,j^-} = \epsilon \sum_{j \in V} \lambda_j, \\ \sum_{\{i,j\} \in E} \mu(\Phi'_\epsilon)_{i^-,j^+} &\leq \sum_{i \in V} \sum_{\substack{j \in V: \\ \{i,j\} \in E}} \mu(\Phi'_\epsilon)_{i^-,j^+} \stackrel{(*)}{=} \sum_{i \in V} \lambda'_{\epsilon,i^-} = \epsilon \sum_{i \in V} \lambda_i, \end{aligned}$$

where (*) follows by applying the conservation equations (CE) to the extended matching model $(G', \lambda'_\epsilon, \Phi'_\epsilon)$. This completes the proof of (35).

Now, we proceed to show that $\sum_{\{i,j\} \in E^*} |\mu_{i,j} - \mu(\Phi'_\epsilon)_{i^+,j^+}| = O(\epsilon)$. Recall that, by Proposition 13, the rows of matrix A^* are linearly independent. This observation has multiple consequences:

- The system of linear equations $A^*y = \lambda$ of unknown y has a unique solution, which we know to be equal to μ .

- The system of linear equations $A^*y = \tilde{\lambda}$ of unknown y has a unique solution, which we know to be equal to $\tilde{\mu} = \{\mu(\Phi'_\epsilon)_{i^+,j^+}\}_{\{i,j\} \in E^*}$, where $\tilde{\lambda}$ is given by

$$\tilde{\lambda}_i = \sum_{\substack{j \in V: \\ \{i,j\} \in E^*}} \mu(\Phi'_\epsilon)_{i^+,j^+} \quad i \in V.$$

- The matrix $(A^*)^T A^*$ is invertible, and the Moore-Penrose inverse of A^* is $((A^*)^T A^*)^{-1} (A^*)^T$. Thus, we have $\mu - \tilde{\mu} = ((A^*)^T A^*)^{-1} (A^*)^T (\lambda - \tilde{\lambda})$ which implies

$$\sum_{\{i,j\} \in E^*} |\mu_{i,j} - \mu(\Phi'_\epsilon)_{i^+,j^+}| \leq \|((A^*)^T A^*)^{-1} (A^*)^T\|_1 \|\lambda - \tilde{\lambda}\|_1. \quad (36)$$

Now, we upperbound the term $\|\lambda - \tilde{\lambda}\|_1$ as follows:

$$\begin{aligned} \|\lambda - \tilde{\lambda}\|_1 &= \sum_{i \in V} \left| \lambda_i - \sum_{\substack{j \in V: \\ \{i,j\} \in E^*}} \mu(\Phi'_\epsilon)_{i^+,j^+} \right|, \\ &\leq \sum_{i \in V} \left| (1 - \epsilon) \lambda_i - \sum_{\substack{j \in V: \\ \{i,j\} \in E^*}} \mu(\Phi'_\epsilon)_{i^+,j^+} \right| + \epsilon \sum_{i \in V} \lambda_i, \\ &\stackrel{(a)}{=} \sum_{i \in V} \sum_{\substack{j \in V: \\ \{i,j\} \in E}} \mu(\Phi'_\epsilon)_{i^+,j^-} + \epsilon \sum_{i \in V} \lambda_i, \\ &\stackrel{(b)}{\leq} 2\epsilon \sum_{i \in V} \lambda_i, \end{aligned} \quad (37)$$

where (a) again follows by applying (CE) to the extended matching model $(G', \lambda'_\epsilon, \Phi'_\epsilon)$, and (b) is justified as follows:

$$\sum_{i \in V} \sum_{\substack{j \in V: \\ \{i,j\} \in E}} \mu(\Phi'_\epsilon)_{i^+,j^-} \stackrel{(i)}{=} \sum_{j \in V} \sum_{\substack{i \in V: \\ \{i,j\} \in E}} \mu(\Phi'_\epsilon)_{i^+,j^-} \stackrel{(ii)}{\leq} \sum_{j \in V} \lambda'_{\epsilon,j^-} = \epsilon \sum_{j \in V} \lambda_j.$$

Here, (i) follows by switching the order of summation, and (ii) follows by noting that $\{i^+ \in V^+ : \{i^+, j^-\} \in E^\pm\} \subseteq \{i \in V' : \{i, j\} \in E'\}$ and again applying (CE) to the matching model $(G', \lambda'_\epsilon, \Phi'_\epsilon)$. Now, by combining (37) with (36), we get

$$\sum_{\{i,j\} \in E^*} |\mu_{i,j} - \mu(\Phi'_\epsilon)_{i^+,j^+}| \leq 2 \|((A^*)^T A^*)^{-1} (A^*)^T\|_1 \sum_{i \in V} \lambda_i \epsilon.$$

Lastly, by injecting the above inequality into (35) and defining

$$C_2 = 2 \left(1 + \|((A^*)^T A^*)^{-1} (A^*)^T\|_1 \right) \sum_{i \in V} \lambda_i,$$

the proof is complete. \blacktriangleleft

E Supplementary material of Section 7 (Practical performance).

E.1 k -filtering policies – Section 7.1 (Considered policies)

We conjecture that k -filtering policies, introduced in Section 7.1, yield a sequence of matching rate vectors that are arbitrarily close to a vertex of the polytope $\Pi_{\geq 0}$, even if this vertex is injective-only. This conjecture is supported by numerical results shown in Sections 7 and 8.

► **Conjecture 33.** *Consider a vertex μ of $\Pi_{\geq 0}$ and let $G^* = (V, E^*)$ denote its support graph. For each $k \in \mathbb{N}$, consider the following semi-filtering policy, denoted by $\Phi_k(\mu)$:*

- *If the size of the longest queue is less than k , apply the filtering ML policy adapted to G with filter E^* ;*
- *Otherwise, apply the greedy ML policy adapted to G .*

$\Phi_k(\mu)$ is stable for each $k \in \mathbb{N}$ and $\lim_{k \rightarrow \infty} \mu(\Phi_k(\mu)) = \mu$.

Intuition of the proof. In essence, the approach is as follows: we take a possibly unstable policy that achieves the desired goal (a matching rate vector equal to μ), and we make it stable by reverting to a stable policy when the queue sizes become too large. If the threshold is large enough, most matches will be made under the unstable policy that achieves the desired goal.

We present here a sketch of proof for the case where G^* is injective-only with a unique connected component, i.e. when G^* is a tree. If G^* is made of multiple connected components that are connected in G , the proof needs to compare the model to a virtual model where each component acts independently and accounts for the interactions between the components. For example, when one the queue is longer than k , all components may interact when decisions follow the ML policy on the graph G .

The stability of the model $(G, \lambda, \Phi_k(\mu))$ for each $k \in \mathbb{N}$ can be proved using the Lyapunov-Foster theorem, see [8, Theorem 1.1 in Chapter 5], using the fact that, outside a finite set of states (those where all queues are shorter than k), the policy $\Phi_k(\mu)$ behaves like the stable ML policy on the graph G .

Now, consider a random vector $Q = (Q_1, Q_2, \dots, Q_n)$ distributed like the vector of queue sizes in the matching model $(G, \lambda, \Phi_k(\mu))$ in stationary regime, and let $Q_+ = \sum_{i \in V_+} Q_i$ and $Q_- = \sum_{i \in V_-} Q_i$ denote the total queue sizes in the parts V_+ and V_- of the (bipartite) graph G^* . Also let $L_+ = \max\{Q_i, i \in V_+\}$, $L_- = \max\{Q_i, i \in V_-\}$, and $L = L_+$ if $L_+ \geq L_-$ and $L = -L_-$ if $L_+ < L_-$, so that $|L|$ is the size of the longest queue and the sign of L indicates the part (V_+ or V_-) to which this queue belongs. The random variable L will play a crucial role in the proof. Lastly, we let $p_\ell = \mathbb{P}(L = \ell)$, $\ell \in \{\dots, -2, -1, 0, 1, 2, \dots\}$, denote its distribution. The dependency of these random variables on k (via the policy $\Phi_k(\mu)$) is left implicit to simplify notation.

To show that $\lim_{k \rightarrow \infty} \mu(\Phi_k(\mu)) = \mu$, we need to upperbound the probability that a match is performed along an edge that does not belong to the support of the vertex μ . To do this, we upperbound the probability of applying the greedy ML policy on the whole graph G , i.e., the probability $\sum_{|\ell| \geq k} p_\ell$ that the size of the longest queue is k or more.

Intuitively, we expect that the stability of the ML policy implies a uniform drift of the size of the longest queue towards the origin whenever this size is larger than k , provided that k is large enough. Formally, we conjecture that there exist $K \in \mathbb{N}$ and $0 < \rho < 1$ such that, for each $k \geq K$, we have under Φ_k that $p_{\ell+1} \leq \rho p_\ell$ for each $\ell \geq k$ and $p_{\ell-1} \leq \rho p_\ell$ for each $\ell \leq -k$. In particular, $\sum_{\ell \geq k} p_\ell \leq \frac{1}{1-\rho} p_k$ and $\sum_{\ell \leq -k} p_\ell \leq \frac{1}{1-\rho} p_{-k}$. This implies that, if p_k and p_{-k} go to 0 as k goes to infinity, the probability that the filter on E^* is disabled goes to 0 as well.

To control p_k and p_{-k} , we now need to consider the cases where the size of the longest queue is at most k . We argue that, as long as $|L| \leq k$, the dynamics of L is mainly controlled by an unbiased random walk between the two parts of the bipartite graph. More precisely, as long as $|L| \leq k$, the difference $Q_+ - Q_-$ behaves like an unbiased random walk because any arrival of a class in V_+ increases the difference by one, any arrival of a class in V_- decreases the difference by one, and the arrival rates of V_+ and V_- are equal (see Equation (29)). We conjecture that, when $|Q_+ - Q_-|$ is large enough (while still satisfying $|L| \leq k$), a distinctive structure emerges due to the filtered ML policy: the queues in one part of the graph have approximately equal sizes, while the queues in the other part are empty. In particular, we conjecture that the typical states of the system (when $|L| \leq k$) verify (in some sense) $L \approx \frac{Q_+ - Q_-}{|V_+|}$ if $Q_+ > Q_-$ and $L \approx \frac{Q_+ - Q_-}{|V_-|}$ if $Q_+ < Q_-$. In a true unbiased random walk over a bounded range of integers, all states have equal probabilities. By comparing the evolutions of L and $Q_+ - Q_-$ and taking into account the border cases ($|L| = k$), we conjecture that there exist $K' \in \mathbb{N}$ and $c > 0$ such that, if $k \geq K'$, then $p_\ell \geq c \max\{p_k, p_{-k}\}$ for all $\ell \in \{-k, \dots, -1, 0, 1, \dots, k\}$. In particular, both p_k and p_{-k} should be upperbounded by $(c(2k + 1))^{-1}$ when k is large enough.

By combining the two cases $|L| \geq k$ and $|L| \leq k$, we get $\sum_{|\ell| \geq k} p_\ell = O(1/k)$. In other words, the probability that the size of the longest queue is greater than k tends to 0 as k goes to infinity. As matchings outside E^* occur only when the size of the longest queue is greater than k , we conclude that the matching rate of an edge outside E^* goes to 0, which by continuity of the conservation principle and injectivity of G^* implies that $\lim_{k \rightarrow \infty} \mu(\Phi_k(\mu)) = \mu$. ◀

E.2 Additional examples

Section 7 evaluated the performance of matching policies on the diamond graph. We present here simulations on other graphs. For additional details, please refer to https://balouf.github.io/stochastic_matching/companion/simulations.html#Vertices-of-simple-graphs

E.2.1 Co-domino graph

We first consider the matching problem depicted in Figure 19, with the goal of approaching the injective-only vertex shown in Figure 19f. This vertex disables 5 out of 8 edges: $\{1, 2\}$, $\{2, 6\}$, $\{3, 4\}$, $\{3, 5\}$, and $\{5, 6\}$. The results are presented in Figure 21.

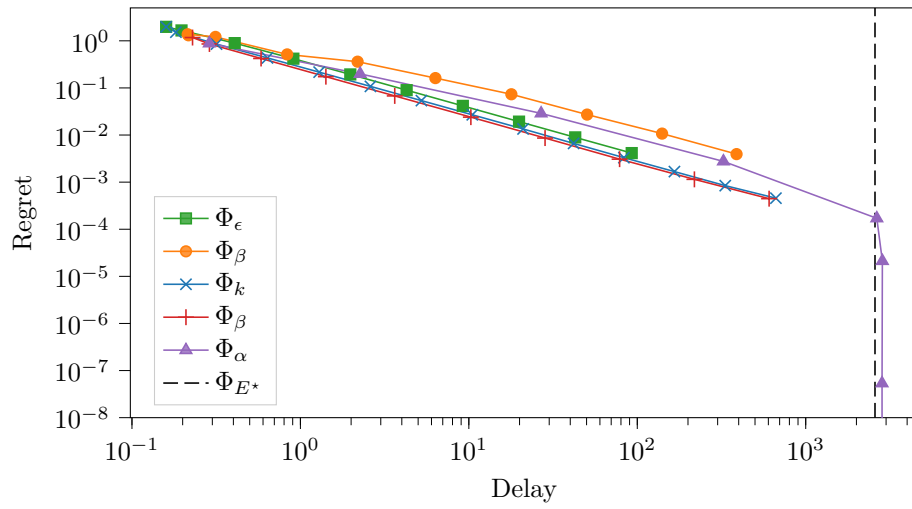
As observed in Section 7.3, The best performance is achieved by Φ'_β and Φ_k , while the performance of Φ_β significantly deteriorates under adversarial rewards.

Still focusing on the codomino graph, we now consider the matching problem from Figure 18. Our objective is to approach the bijective vertex illustrated in Figure 18e, which disables the edges $\{2, 3\}$ and $\{4, 5\}$. The results are presented in Figure 22, with a dotted line representing the delay of Φ_{E^*} , which is reward-optimal and stable for bijective vertices.

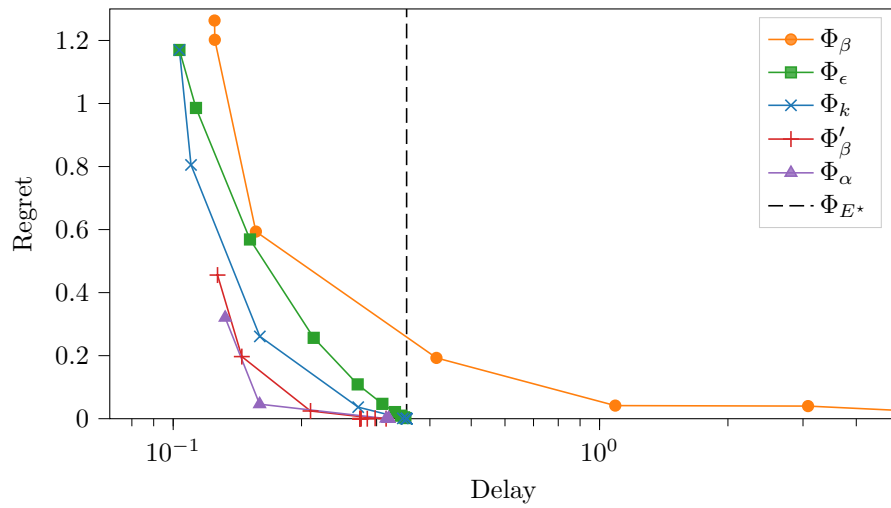
As seen in Section 7.4, most policies converge to delays at or below that of Φ_{E^*} , with the notable exception of Φ_β .

E.2.2 Larger graph

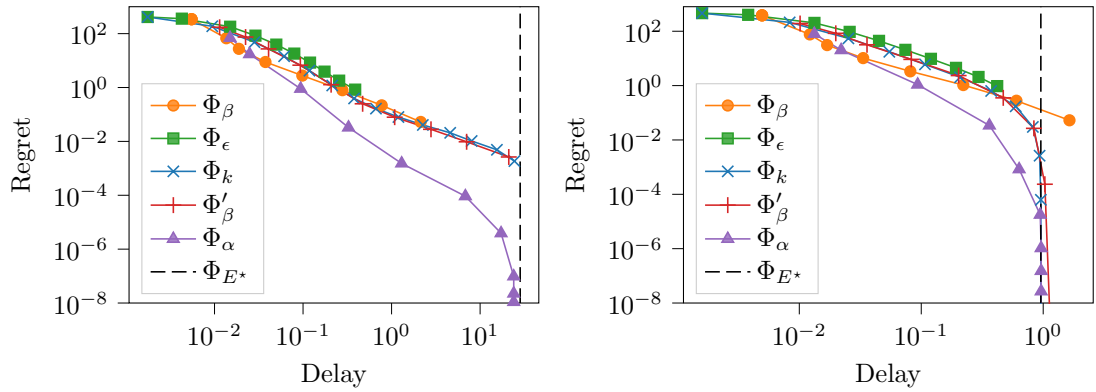
To observe the performance of our methods on a larger graph, we generated a simple Erdős-Rényi graph with $n = 100$ nodes and an edge probability of $20/(n - 1)$, resulting in $m = 1006$ edges. The edge rewards were drawn uniformly and independently.



■ **Figure 21** Matching problem from Figure 19. Targeting the injective-only vertex from Figure 19f.



■ **Figure 22** Matching problem from Figure 18. Targeting the bijective vertex from Figure 18e.



(a) Injective-only vertex (degree-proportional arrivals). (b) Bijective vertex (perturbed degree-proportional arrivals).

■ **Figure 23** Reaching a vertex on a larger graph ($n = 100, m = 1006$).

We aimed to study both an injective-only vertex and a bijective vertex. On graphs of this size, computing the exhaustive list of vertices is unrealistic, but once the reward and arrival-rate vectors are given, calculating the corresponding optimal vertex is straightforward.

To obtain an injective-only vertex, we used degree-proportional arrival rates, following the discussion in Section 5.2.2. This ensured the problem was stabilizable and, in this case, resulted in an optimal vertex with 97 positive edges (injective-only). For the bijective vertex, we added a noisy perturbation to the degree-proportional arrival rates, yielding a bijective vertex, as hinted by Proposition 14.

The performance results for both cases are presented in Figure 23. A key observation is that Φ_α performs significantly better in this scenario compared to what was observed on smaller graphs with lower-dimensional $\Pi_{\geq 0}$.

F Supplementary material of Section 8 (Extensions of our results).

F.1 Convexity proofs for $\Pi_{\mathcal{P}}$ and Π_G (Propositions 20 and 21).

We consider the extended definition of matching policies, as discussed in Appendix A. Consider a stabilizable matching problem (G, λ) and let $\Pi_{\mathcal{P}}$ (resp. Π_G) denote the set of vectors of matching rates achievable by stable policies (resp. by stable greedy policies) adapted to the compatibility graph G .

Convexity of $\Pi_{\mathcal{P}}$. Consider two (extended) policies Φ_1 and Φ_2 that stabilize the matching problem (G, λ) . The state-space, queue-size function, and empty state of Φ_1 (resp. Φ_2) are denoted by $\mathcal{S}_1, |\cdot|_1$, and \emptyset_1 (resp. $\mathcal{S}_2, |\cdot|_2$, and \emptyset_2). Given $0 < \beta < 1$, our goal is to build a matching policy Φ_β that also stabilizes the matching problem (G, λ) and satisfies $\mu(\Phi_\beta) = \beta\mu(\Phi_1) + (1 - \beta)\mu(\Phi_2)$.

Intuitively, the policy Φ_β will consistently follow the decisions of either Φ_1 or Φ_2 as long as the system is non-empty, and switch between these two policies each time the system becomes empty (therefore, at renewal times). The probability of selecting each policy after visiting the empty state will be set to achieve the desired matching rate vector in the long run. More formally, the policy Φ_β will have the following characteristics:

- State space $\mathcal{S}_\beta = (\mathcal{S}_1 \times \{\emptyset_2\}) \cup (\{\emptyset_1\} \times \mathcal{S}_2)$.
- Queue-size function $|\cdot|_\beta$ defined by $|(s_1, s_2)|_\beta = |s_1|_1 + |s_2|_2$ for each $(s_1, s_2) \in \mathcal{S}_\beta$.
- Empty state: $\emptyset_\beta = (\emptyset_1, \emptyset_2)$.

For some $0 < \gamma_\beta < 1$ that will be specified later, the matching policy Φ_β is defined as follows:

$$\begin{aligned} \Phi_\beta((\emptyset_1, \emptyset_2), i, j, (s_1, \emptyset_2)) &= \gamma_\beta \Phi_1(\emptyset_1, i, j, s_1), & s_1 \in \mathcal{S}_1, \quad i \in V, \quad j \in V \cup \{\perp\}, \\ \Phi_\beta((\emptyset_1, \emptyset_2), i, j, (\emptyset_1, s_2)) &= (1 - \gamma_\beta) \Phi_2(\emptyset_2, i, j, s_2), & s_2 \in \mathcal{S}_2, \quad i \in V, \quad j \in V \cup \{\perp\}, \\ \Phi_\beta((s_1, \emptyset_2), i, j, (s'_1, \emptyset_2)) &= \Phi_1(s_1, i, j, s'_1), & s_1 \in \mathcal{S}_1 \setminus \{\emptyset_1\}, \quad s'_1 \in \mathcal{S}_1, \\ \Phi_\beta((\emptyset_1, s_2), i, j, (\emptyset_1, s'_2)) &= \Phi_2(s_2, i, j, s'_2), & s_2 \in \mathcal{S}_2 \setminus \{\emptyset_2\}, \quad s'_2 \in \mathcal{S}_2. \end{aligned}$$

The first two equations say that, when the system is empty, we next apply Φ_1 with probability γ_β and Φ_2 with probability $1 - \gamma_\beta$. The third (resp. fourth) equation says that, once we start applying policy Φ_1 (resp. Φ_2), we keep applying this policy until we re-visit the empty state.

The stability of the policies Φ_1 and Φ_2 implies that of the policy Φ_β . According to the elementary renewal theorem for renewal reward processes, the vector giving the long-run average matching rates under the policy Φ_β is given by

$$\mu(\Phi_\beta) = \frac{\gamma_\beta T_1 \mu(\Phi_1) + (1 - \gamma_\beta) T_2 \mu(\Phi_2)}{\gamma_\beta T_1 + (1 - \gamma_\beta) T_2} = \frac{\gamma_\beta T_1}{\gamma_\beta T_1 + (1 - \gamma_\beta) T_2} \mu(\Phi_1) + \frac{(1 - \gamma_\beta) T_2}{\gamma_\beta T_1 + (1 - \gamma_\beta) T_2} \mu(\Phi_2).$$

where T_1 (resp. T_2) denotes the mean number of matches between two successive visits to the empty state \emptyset_1 (resp. \emptyset_2) in the matching model (G, λ, Φ_1) (resp. (G, λ, Φ_2)). We let the reader verify that the following value of γ_β yields $\mu(\Phi_\beta) = \beta\mu(\Phi_1) + (1 - \beta)\mu(\Phi_2)$:

$$\gamma_\beta = \frac{\beta T_2}{(1 - \beta) T_1 + \beta T_2}.$$

Convexity of Π_G . It suffices to observe that, if the policies Φ_1 and Φ_2 are greedy, so is the policy Φ_β .

F.2 Supplementary material of Section 8.2 (Greedy policies).

F.2.1 Proof of Proposition 23 (complete graphs) and discussion.

Consider a matching problem (K_n, λ) , where K_n is the complete graph with $n \geq 3$ nodes. According to Proposition 8, this matching problem is stabilizable if and only if $\lambda_i < \frac{1}{2} \sum_{j \in V} \lambda_j$ for each $i \in V$. We prove Proposition 23 by first demonstrating that all greedy policies are “equivalent”, in the sense that they always (deterministically) make the same matching decisions. The fact that we can formally define many greedy policies in this context is merely an artifact of our extended definition of a matching policy in Appendix A (see in particular Appendix A.2).

Notice that the only independent sets of a complete graph are the singletons, hence by (7) the state space of the queue-size process under greedy policies is

$$\mathcal{Q}_{\mathcal{G}}(K_n) = \{0\} \cup \left(\bigcup_{i \in V} \{\ell \mathbb{1}_i : \ell \in \mathbb{N}_{>0}\} \right), \quad (38)$$

where $\mathbb{1}_i$ is the n -dimensional vector with one in coordinate i and zero elsewhere, for each $i \in V$. This observation implies that greediness leaves no freedom in the matching decisions: all unmatched items belong to the same class, and an incoming item must be matched with one of them if its class differs. This is formalized in Proposition 34.

► **Proposition 34.** *Consider the complete graph K_n with $n \geq 3$ nodes.*

- (i) *There is a unique size-based greedy policy that is adapted to the compatibility graph K_n . This policy, called the natural greedy policy and denoted by $\Phi_{\mathcal{G}}(K_n)$, is the deterministic size-based policy defined on $\mathcal{Q}_{\mathcal{G}}(K_n) \times V$ by*

$$\Phi_{\mathcal{G}}(K_n)(q, i) = \begin{cases} j & \text{if } q_j \geq 1 \text{ with } j \neq i, \\ \perp & \text{otherwise.} \end{cases} \quad (39)$$

- (ii) *Consider a greedy policy Φ adapted to the graph K_n , and let $(\mathcal{S}, |\cdot|)$ denote its state space. The policy Φ makes the same decisions as the natural greedy policy $\Phi_{\mathcal{G}}$ in the sense that*

$$\sum_{s' \in \mathcal{S}} \Phi(s, i, \Phi_{\mathcal{G}}(K_n)(|s|, i), s') = 1, \quad s \in \mathcal{S}, \quad i \in V.$$

In other words, all greedy policies are equivalent in the sense that they can all be reduced to $\Phi_{\mathcal{G}}(K_n)$ (see “Equivalent policies” in Appendix A).

- (iii) *For each greedy policy Φ adapted to the graph K_n and each sequence $I = (I_t, t \in \mathbb{N})$ of item classes, we have $(Q_{\mathcal{G}})_t = Q_t$ for each $t \in \mathbb{N}$, where $Q_{\mathcal{G}}$ and Q are the queue-size processes of the models $(K_n, I, \Phi_{\mathcal{G}}(K_n))$ and (K_n, I, Φ) , respectively.*

Proof. We prove each statement one by one.

Proposition 34i. Consider a (possibly random) size-based greedy policy $\Phi : \mathcal{Q}_{\mathcal{G}}(K_n) \times V \times (V \cup \{\perp\}) \rightarrow [0, 1]$, as defined in Section 2.3.4, where $\mathcal{Q}_{\mathcal{G}}(K_n)$ is given in (38). Given $(q, i) \in \mathcal{Q}_{\mathcal{G}}(K_n) \times V$, the definition of $\mathcal{Q}_{\mathcal{G}}(K_n)$ implies that $\{j \in V_i : q_j \geq 1\}$ is either a singleton or the empty set. In the former case, letting j denote the unique element of the singleton, we have $\Phi(q, i, \perp) = 1$ if $i = j$, while the greediness of Φ implies that $\Phi(q, i, j) = 1$ if $i \neq j$. In the latter case, we have directly $\Phi(q, i, \perp) = 1$. In all cases, the matching decision is deterministic and makes the same decisions as in (39).

Proposition 34ii. The same argument can be repeated for an arbitrary greedy policy Φ with state space $(\mathcal{S}, |\cdot|)$. Given $(s, i) \in \mathcal{S} \times V$, we know that $|s| \in \mathcal{Q}_{\mathcal{G}}(K_n)$, so that $\{j \in V_i : |s|_j \geq 1\}$ is either a singleton or the empty set. In the former case, letting j denote the unique element of the singleton, we have $\sum_{s' \in \mathcal{S}} \Phi(s, i, \perp, s') = 1$ if $i = j$, while the greediness of Φ implies that $\sum_{s' \in \mathcal{S}} \Phi(s, i, j, s') = 1$ if $i \neq j$. In the latter case, we have directly $\sum_{s' \in \mathcal{S}} \Phi(s, i, \perp, s') = 1$. In all cases, we have $\sum_{s' \in \mathcal{S}} \Phi(s, i, \Phi_{\mathcal{G}}(|s|, i), s') = 1$.

Proposition 34iii. This statement follows directly from Proposition 34ii and from the fact that, under any matching policy, the dynamics of the queue-size process is fully specified by the sequence of incoming item classes and the matching decisions via (2). ◀

Proposition 35 below, which encompasses Proposition 23, builds on this equivalence to conclude that the set $\Pi_{\mathcal{G}}$ collapses to a single point and directly express the unique vector of matching rates achieved by (stable) greedy policies. In contrast, the polytope $\Pi_{\geq 0}$ exhibits a dimension $d = m - n = \frac{n(n-3)}{2}$, where $n = |V|$ is the number of classes and $m = |E| = \binom{n}{2} = \frac{n(n-1)}{2}$ is the number of possible matches. In particular, for $n \geq 4$, the dimension $d = m - n$ of $\Pi_{\mathcal{G}}$ is at least 2. Therefore, limiting ourselves to greedy policies dramatically narrows the range of achievable solutions on complete graphs larger than a triangle.

► **Proposition 35.** Consider a stabilizable matching problem (K_n, λ) , where K_n is the complete graph with $n \geq 3$ nodes, and let Φ denote a greedy policy adapted to K_n . Then:

- (i) The model (K_n, λ, Φ) is stable.
- (ii) The matching rate vector $\mu_{\mathcal{G}}$ in this model satisfies

$$(\mu_{\mathcal{G}})_k = \lambda_i p_j + \lambda_j p_i, \quad e_k = \{i, j\} \in E, \quad (40)$$

where, for each $i \in V$, p_i is the stationary probability that queue i is non-empty, given by

$$p_i = \frac{\lambda_i}{(\sum_{j \neq i} \lambda_j) - \lambda_i} p_{\emptyset}, \quad (41)$$

and p_{\emptyset} is the stationary probability that the system is empty, given by

$$p_{\emptyset} = \left(1 + \sum_{i \in V} \frac{\lambda_i}{(\sum_{j \neq i} \lambda_j) - \lambda_i} \right)^{-1}. \quad (42)$$

In particular, we have $\Pi_{\mathcal{G}} = \{\mu_{\mathcal{G}}\} \subsetneq \Pi_{>0}$ whenever $n \geq 4$.

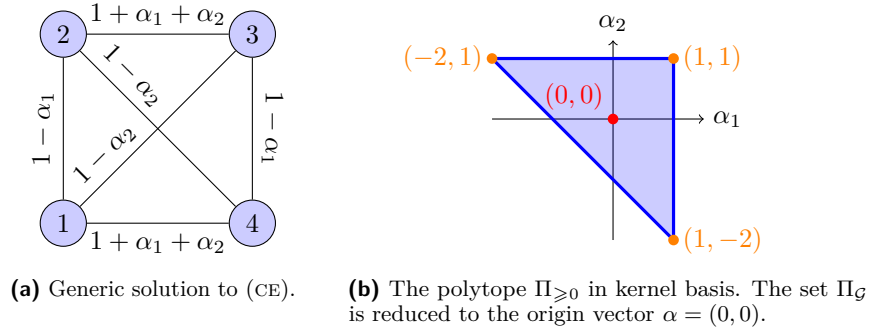
Proof. We prove each statement one by one.

Proposition 35i. Proposition 35i is a consequence of Corollary 27i (Appendix B.3).

Proposition 35ii. We can focus without loss of generality on the FCFM policy, as Proposition 34iii implies that all greedy matching policies yield the same vector of matching rates. Equation (40) follows by observing that, for each edge $e_k = \{i, j\} \in E$, a match between classes i and j happens if:

- a class- i item arrives while queue j is non-empty, which happens at rate $\lambda_i p_j$;
- a class- j item arrives while queue i is non-empty, which happens at rate $\lambda_j p_i$.

Equations (41) and (42) follow from [16, Proposition 5]. Indeed, for each $i \in V$, applying [16, Equation (10)] to the independent set $\{i\}$ yields (41), and the value of p_{\emptyset} given in (42) follows from the normalizing equation. This result may also be obtained more directly by observing that, for each $i \in V$, the restriction of the transition diagram of the Markov chain $(K_n, \lambda, \Phi_{\mathcal{G}})$ to the states where all queues but queue i are empty is similar to a (discrete-time) birth-and-death process with birth probability λ_i and death probability $\sum_{j \neq i} \lambda_j$. ◀



■ **Figure 24** Matching problem (K_4, λ) with $\lambda = (3, 3, 3, 3)$.

Figure 24 illustrates this result on a complete graph K_4 in which all arrival rates are equal to 3. The particular solution μ° and the basis $\{b_1, b_2\}$ of $\ker(A)$ that we use are shown on Figure 24a. In the corresponding kernel basis, the polytope $\Pi_{\geq 0}$ is defined by the inequalities $\alpha_1 \leq 1$, $\alpha_2 \leq 1$, and $\alpha_1 + \alpha_2 \geq -1$, that is, it is the triangle with vertices $(-2, 1)$, $(1, -2)$, and $(1, 1)$. Yet, Proposition 35 shows that only the solution $\alpha = (0, 0)$ can be achieved by a greedy policy.

F.2.2 Proof of Proposition 24 (diamond graph) and discussion.

Consider the diamond (double fan) graph introduced in Example 4, for which we will show that the set $\Pi_{\mathcal{G}}$ of matching rate vectors achieved by stable greedy policies is a strict subset of the set $\Pi_{>0}$ of positive solution to (CE). More specifically, we consider the diamond matching problem (D, λ) defined as follows:

$$\begin{aligned}
 D &= (V, E) \text{ with } V = \{1, 2, 3, 4\} \text{ and } E = \{\{1, 2\}, \{1, 3\}, \{2, 3\}, \{2, 4\}, \{3, 4\}\}, \\
 \lambda &= (\lambda_1, \bar{\lambda}_2 + \beta, \bar{\lambda}_3 + \beta, \lambda_4) \\
 &\quad \text{with } \lambda_1 > 0, \bar{\lambda}_2 > 0, \bar{\lambda}_3 > 0, \lambda_4 > 0, \beta > 0, \lambda_1 + \lambda_4 = \bar{\lambda}_2 + \bar{\lambda}_3 = \frac{1}{2}, \\
 \mathcal{B} &= \{(1, -1, 0, -1, 1)\}, \\
 \mu^\circ &= (2\lambda_1\bar{\lambda}_2, 2\lambda_1\bar{\lambda}_3, \beta, 2\bar{\lambda}_2\lambda_4, 2\bar{\lambda}_3\lambda_4), \\
 \mu &= (2\lambda_1\bar{\lambda}_2 + \alpha, 2\lambda_1\bar{\lambda}_3 - \alpha, \beta, 2\bar{\lambda}_2\lambda_4 - \alpha, 2\bar{\lambda}_3\lambda_4 + \alpha), \quad \alpha \in \mathbb{R}.
 \end{aligned} \tag{43}$$

Parameterizing λ by β ($= \mu_{2,3}$ for each $\mu \in \Pi$) is a notational convenience that does not lead to any loss of generality: as already observed in Example 4, the stabilizability condition ii writes $\bar{\lambda}_2 > 0$, $\bar{\lambda}_3 > 0$, and $\beta > 0$. We leave it to the reader to verify that our choices for \mathcal{B} and μ° are correct. Equation (43) implies in particular that the sets $\Pi_{\geq 0}$ and $\Pi_{>0}$, are real intervals, and so are $\Pi_{\mathcal{P}}$ and $\Pi_{\mathcal{G}}$ by convexity.

As in Appendix F.2.1, we first provide an equivalence result that will be useful to describe $\Pi_{\mathcal{G}}$ in details.

Equivalence of greedy policies

As in Appendix F.2.1, we first identify a common behavior shared by all greedy policies and then we exploit this behavior to characterize the matching rates achievable by stable greedy policies.

Since $\{1, 4\}$ is the only independent set of the diamond graph D that is not a singleton, the possible queue states under any greedy policy can be partitioned as follows: either all

queues are empty, or exactly one class has a non-empty queue, or (only) classes 1 and 4 have non-empty queues. In other words, the state space of the queue-size process under greedy policies adapted to the graph D is

$$\mathcal{Q}_{\mathcal{G}}(D) = \{0\} \cup \left(\bigcup_{i \in V} \{\ell \mathbf{1}_i, \ell \in \mathbb{N}_{>0}\} \right) \cup \{\ell_1 \mathbf{1}_1 + \ell_4 \mathbf{1}_4 : \ell_1, \ell_4 \in \mathbb{N}_{>0}\}. \quad (44)$$

Therefore, greediness entirely determines the decisions made by greedy policies, except if an item of class 2 or 3 enters while there are unmatched items of classes 1 and 4. Using the fact that classes 2 and 3 are both compatible with classes 1 and 4, we prove formally in Proposition 36 that all greedy policies adapted to the diamond graph make the same matching decisions as the natural greedy policy adapted to the complete graph K_3 obtained by “merging” classes 1 and 4 in the diamond graph.

► **Proposition 36.** *Given the diamond graph D , we introduce the following notation:*

- *Queue-size projection: For each $q = (q_1, q_2, q_3, q_4) \in \mathcal{Q}_{\mathcal{G}}(D)$, we let $\langle q \rangle = (q_1 + q_4, q_2, q_3)$.*
- *Class projection: We let $\langle i \rangle = i$ for each $i \in \{1, 2, 3, \perp\}$ and $\langle 4 \rangle = 1$.*

Every greedy policy Φ adapted to the diamond graph D satisfies the following properties:

- (i) *If Φ is a deterministic size-based policy, then*

$$\langle \Phi(q, i) \rangle = \Phi_{\mathcal{G}}(K_3)(\langle q \rangle, \langle i \rangle), \quad (q, i) \in \mathcal{Q}_{\mathcal{G}}(D) \times V.$$

where $\Phi_{\mathcal{G}}(K_3)$ is the natural greedy policy (39) adapted to the complete graph K_3 .

- (ii) *In general, if Φ is a policy with state space $(\mathcal{S}, |\cdot|)$, then*

$$\sum_{s' \in \mathcal{S}} \sum_{\substack{j \in V \cup \{\perp\}: \\ \langle j \rangle = \Phi_{\mathcal{G}}(K_3)(\langle |s| \rangle, \langle i \rangle)}} \Phi(s, i, j, s') = 1, \quad s \in \mathcal{S}, \quad i \in V.$$

- (iii) *For each sequence I of item classes, we have $\langle Q_t \rangle = (Q_{\mathcal{G}})_t$ for each $t \in \mathbb{N}$, where Q and $Q_{\mathcal{G}}$ are the queue-size processes of the models (D, I, Φ) and $(K_3, \langle I \rangle, \Phi_{\mathcal{G}})$, respectively, with $\langle I \rangle = (\langle I_t \rangle, t \in \mathbb{N})$.*

Loosely speaking, the main take-away of Proposition 36iii is that, in the matching model (D, I, Φ) , the process $((Q_{t,1} + Q_{t,4}, Q_{t,2}, Q_{t,3}), t \in \mathbb{N})$ does not depend on the specific greedy policy Φ that is applied, and it is in fact equal to the queue size process built by applying the natural greedy policy $\Phi_{\mathcal{G}}(K_3)$ in the complete graph K_3 obtained by merging classes 1 and 4 in the diamond graph D .

Proof of Proposition 36. We prove each statement one by one.

Proposition 36i. We leave it to the reader to verify that $\mathcal{Q}_{\mathcal{G}}(K_3)$ is the image of $\mathcal{Q}_{\mathcal{G}}(D)$ by the application $q = (q_1, q_2, q_3, q_4) \mapsto \langle q \rangle = (q_1 + q_4, q_2, q_3)$. This means in particular that $q_1 + q_4$, q_2 , and q_3 cannot be positive simultaneously if $q \in \mathcal{Q}_{\mathcal{G}}(D)$. Since the diamond graph D has only four nodes, we can then conclude by enumerating all relevant cases, depending on the support of $\langle q \rangle = (q_1 + q_4, q_2, q_3)$. For example, if $q_1 + q_4 \geq 1$ and $q_2 = q_3 = 0$, then we have immediately $\Phi(q, i) = \perp$ if $i \in \{1, 4\}$, and the greediness of the policy Φ implies that $\Phi(q, i) \in \{1, 4\}$ if $i \in \{2, 3\}$; in other words, we have $\langle \Phi(q, i) \rangle = \perp$ if $\langle i \rangle = 1$ and $\langle \Phi(q, i) \rangle = 1$ if $\langle i \rangle \in \{2, 3\}$. Similarly, if $q_1 + q_4 = q_3 = 0$ and $q_2 \geq 1$, then $\Phi(q, i) = \perp$ if $i = 2$ and $\Phi(q, i) = 2$ if $i \in \{1, 3, 4\}$, that is, $\langle \Phi(q, i) \rangle = \perp$ if $\langle i \rangle = 2$ and $\langle \Phi(q, i) \rangle = 2$ if $\langle i \rangle \in \{1, 3\}$. In all cases, we can verify that $\langle \Phi(q, i) \rangle$ is equal to $\Phi_{\mathcal{G}}(K_3)(\langle q \rangle, \langle i \rangle)$.

Alternatively, by taking a step back, we can prove Proposition 36i more directly by observing that, for each $(q, i) \in \mathcal{Q}_{\mathcal{G}}(D) \times V$, (i) the support of $\langle q \rangle$ is a singleton $\{j\}$ whenever $q \neq 0$, and (ii) whether $j \in V_i$ depends on i only via $\langle i \rangle$.

Proposition 36ii. The same argument can be repeated for an arbitrary greedy policy Φ with state space $(\mathcal{S}, |\cdot|)$. As in the proof of Proposition 34ii, the key argument consists of observing that we still have $q = |s| \in \mathcal{Q}_{\mathcal{G}}(D)$ for each $(s, i) \in \mathcal{S} \times V$, so that $q_1 + q_4$, q_2 , and q_3 cannot be positive simultaneously.

Proposition 36iii. The conclusion follows in much the same way as in the proof of Proposition 34iii, by injecting statements i and ii from Proposition 36 into (2). ◀

Matching rates under stable greedy policies.

Propositions 37 and 38, which encompass Proposition 24, use the equivalence result of Proposition 36 to characterize the matching rate vector under stable greedy policies. Proposition 37 bounds the coordinates of the matching-rate vectors achievable by a stable greedy policy, while Proposition 38 details the bounds α_+ and α_- of $\Pi_{\mathcal{G}}$. These propositions illustrate the complementarity of edge and kernel coordinates: the results of Proposition 37 are easier to state using edge coordinates, while those of Proposition 38 are easier to state using kernel coordinates. The lower bound (45) is obtained by following a similar approach as in Proposition 35ii.

► **Proposition 37** (Edge coordinates). *Consider the matching model (D, λ, Φ) , where (D, λ) is the diamond problem (43) and Φ is a greedy policy adapted to the graph D .*

(i) *This matching model is stable.*

(ii) *The matching rate vector $\mu = \mu(D, \lambda, \Phi)$ satisfies $\mu_{2,3} = \beta = \frac{1}{2}(\lambda_2 + \lambda_3 - \lambda_1 - \lambda_4)$, and*

$$\mu_{1,2} \geq \lambda_1 p_2 + \lambda_2 p_1, \quad \mu_{1,3} \geq \lambda_1 p_3 + \lambda_3 p_1, \quad \mu_{2,4} \geq \lambda_2 p_4 + \lambda_4 p_2, \quad \mu_{3,4} \geq \lambda_3 p_4 + \lambda_4 p_3, \quad (45)$$

where p_i is the stationary probability that the system contains unmatched items that belong exclusively to class i , for each $i \in V$.

(iii) *Let $p_{1,4}$ denote the stationary probability that the system contains unmatched items that belong to class 1 or 4 (or both). We have*

$$p_2 = \frac{\lambda_2}{\lambda_1 + \lambda_3 + \lambda_4 - \lambda_2} p_{\emptyset}, \quad p_3 = \frac{\lambda_3}{\lambda_1 + \lambda_2 + \lambda_4 - \lambda_3} p_{\emptyset}, \quad p_{1,4} = \frac{\lambda_1 + \lambda_4}{\lambda_2 + \lambda_3 - \lambda_1 - \lambda_4} p_{\emptyset}, \quad (46)$$

where p_{\emptyset} is the stationary probability that the system is empty, whose value follows from the normalization equation $p_{\emptyset} + p_{1,4} + p_2 + p_3 = 1$. We also have

$$p_1 > \frac{\lambda_1}{\lambda_2 + \lambda_3 + \lambda_4} p_{\emptyset}, \quad p_4 > \frac{\lambda_4}{\lambda_1 + \lambda_2 + \lambda_3} p_{\emptyset}. \quad (47)$$

Proof of Proposition 37. We prove each statement one after another.

Proposition 37i. This is a consequence of Corollary 27ii (Appendix B.3).

Proposition 37ii. The equation $\mu_{2,3} = \beta = \frac{1}{2}(\lambda_2 + \lambda_3 - \lambda_1 - \lambda_4)$ is a direct consequence of (CE). The inequalities (45) for $\mu_{1,2}$, $\mu_{1,3}$, $\mu_{2,4}$, and $\mu_{3,4}$ follow by observing that, for each edge $\{i, j\} \in \{\{1, 2\}, \{1, 4\}, \{2, 3\}, \{3, 4\}\}$, a match between classes i and j happens at least in one of the following cases:

- a class- i item arrives while the system contains unmatched items that all belong to class j ,
 - a class- j item arrives while the system contains unmatched items that all belong to class i .
- These events occur at rates $\lambda_i p_j$ and $\lambda_j p_i$, respectively. Equations (45) are not equalities in general because the above list is not exhaustive. For example, depending on the greedy

policy, a match between classes 1 and 2 may happen if a class-2 item arrives while the system contains unmatched items of class 1 and unmatched items of class 4.

Proposition 37iii. The expressions for p_\emptyset , p_2 , p_3 , and $p_{1,4}$ follow directly by combining Proposition 36iii with Equations (41) and (42) in Proposition 35. We now derive the lower bound (47) for p_1 . The one for p_4 follows by symmetry.

First assume that the greedy policy Φ is deterministic and size-based, so that it satisfies Proposition 36i. For each $q \in \mathcal{Q}_G(D)$, let π_q denote the probability that the Markov chain (D, λ, Φ) is in state q in stationary regime. The probability that we want to lower-bound is

$$p_1 = \sum_{\ell=1}^{+\infty} \pi_{\ell \mathbf{1}_1}. \quad (48)$$

Now let $\ell \in \mathbb{N}_{>0}$ and consider the balance equation for state $\ell \mathbf{1}_1$, given by

$$(\lambda_1 + \lambda_2 + \lambda_3 + \lambda_4)\pi_{\ell \mathbf{1}_1} = \lambda_1 \pi_{(\ell-1)\mathbf{1}_1} + (\lambda_2 + \lambda_3)\pi_{(\ell+1)\mathbf{1}_1} + C, \quad (49)$$

where C is a non-negative real that depends on the model parameters, the integer ℓ , and the policy Φ , and that accounts for the flow to state $\ell \mathbf{1}_1$ from state $\ell \mathbf{1}_1 + \mathbf{1}_4$, if any. It follows that $(\lambda_1 + \lambda_2 + \lambda_3 + \lambda_4)\pi_{\ell \mathbf{1}_1} > \lambda_1 \pi_{(\ell-1)\mathbf{1}_1}$. An inductive argument allows us to conclude that

$$\pi_{\ell \mathbf{1}_1} > \left(\frac{\lambda_1}{\lambda_1 + \lambda_2 + \lambda_3 + \lambda_4} \right)^\ell p_\emptyset, \quad \ell \in \mathbb{N}_{>0}. \quad (50)$$

Injecting this inequality into (48) allows us to conclude:

$$p_1 > \sum_{\ell=1}^{+\infty} \left(\frac{\lambda_1}{\lambda_1 + \lambda_2 + \lambda_3 + \lambda_4} \right)^\ell p_\emptyset = \frac{\lambda_1}{\lambda_2 + \lambda_3 + \lambda_4} p_\emptyset. \quad (51)$$

If Φ is an extended policy with state space $(\mathcal{S}, |\cdot|)$, we can still write (48)–(51) and reach the same conclusion. The only difference is that π can no longer be defined as the stationary distribution of a Markov chain: instead, we define $\pi_q = \sum_{s \in \mathcal{S}: |s|=q} \varpi_s$ for each $q \in \mathcal{Q}_G(D)$, where ϖ_s is the probability that the Markov chain (D, λ, Φ) is in state s in stationary regime, for each $s \in \mathcal{S}$. ◀

► **Proposition 38** (Kernel coordinates). *Consider the diamond matching problem (D, λ) of (43).*

(i) *The intervals $\Pi_{\geq 0}$, $\Pi_{> 0}$, and Π_G are defined as follows in the kernel basis:*

$$\begin{aligned} \Pi_{\geq 0} &= [-2 \min(\lambda_1 \bar{\lambda}_2, \bar{\lambda}_3 \lambda_4), 2 \min(\lambda_1 \bar{\lambda}_3, \bar{\lambda}_2 \lambda_4)], \\ \Pi_{> 0} &= (-2 \min(\lambda_1 \bar{\lambda}_2, \bar{\lambda}_3 \lambda_4), 2 \min(\lambda_1 \bar{\lambda}_3, \bar{\lambda}_2 \lambda_4)), \\ \Pi_G &= [\alpha_-, \alpha_+], \end{aligned}$$

with $-2 \min(\lambda_1 \bar{\lambda}_2, \bar{\lambda}_3 \lambda_4) < \alpha_- \leq \alpha_+ < 2 \min(\lambda_1 \bar{\lambda}_3, \bar{\lambda}_2 \lambda_4)$. Hence, $\Pi_G \subsetneq \Pi_{> 0} \subsetneq \Pi_{\geq 0}$.

(ii) *The coordinates α_+ and α_- satisfy the following properties:*

- a. $\alpha_+ = \alpha(\Phi_+)$, where Φ_+ is the Highest-Reward-First (HRF) policy adapted to the graph D whereby edges $\{1, 2\}$ and $\{3, 4\}$ have the highest reward.
- b. $\alpha_- = \alpha(\Phi_-)$, where Φ_- is the HRF policy adapted to the graph D whereby edges $\{1, 3\}$ and $\{2, 4\}$ have the highest reward.
- c. If $\beta \rightarrow +\infty$ while λ_1 , $\bar{\lambda}_2$, $\bar{\lambda}_3$, and λ_4 remain fixed, we have $\alpha_+ \rightarrow 0$ and $\alpha_- \rightarrow 0$.

Proof of Proposition 38. We prove each statement one after another.

Proposition 38i–iia–iib. The diamond graph D has $n = 4$ nodes and $m = 5$ edges. Therefore, according to Propositions 10 and 21, the sets $\Pi_{\geq 0}$, $\Pi_{> 0}$, and $\Pi_{\mathcal{G}}$ have dimension $d = m - n = 1$, meaning that they are intervals in \mathbb{R} . The equations for the intervals $\Pi_{\geq 0}$ and $\Pi_{> 0}$ follow directly from the change-of-basis equation $\mu = (2\lambda_1\bar{\lambda}_2 + \alpha, 2\lambda_1\bar{\lambda}_3 - \alpha, \beta, 2\bar{\lambda}_2\lambda_4 - \alpha, 2\bar{\lambda}_3\lambda_4 + \alpha)$ from (43). That $\Pi_{\mathcal{G}}$ is also an interval is a consequence of its convexity (Proposition 21). The (non-strict) inequality $-2\min(\lambda_1\bar{\lambda}_2, \bar{\lambda}_3\lambda_4) \leq \alpha_- \leq \alpha_+ \leq 2\min(\lambda_1\bar{\lambda}_3, \bar{\lambda}_2\lambda_4)$ is a consequence of Proposition 21, which states that $\Pi_{\mathcal{G}} \subseteq \Pi_{> 0}$. The first and third inequalities are also strict because α_+ and α_- belong to $\Pi_{\mathcal{G}}$ (see below), while $2\min(\lambda_1\bar{\lambda}_3, \bar{\lambda}_2\lambda_4)$ and $-2\min(\lambda_1\bar{\lambda}_2, \bar{\lambda}_3\lambda_4)$ do not belong to $\Pi_{> 0}$. That $\Pi_{\mathcal{G}}$ is a closed interval of the form $\Pi_{\mathcal{G}} = [\alpha_-, \alpha_+]$ and that α_+ and α_- are as given in Proposition 38iia–iib are consequences of Lemma 39 below, which will be proved by a coupling argument later in this appendix²¹.

► **Lemma 39.** Consider the HRF policy Φ_+ adapted to the diamond graph D whereby edges $\{1, 2\}$ and $\{3, 4\}$ have the highest reward. For each greedy policy Φ adapted to the compatibility graph D , we have

$$\begin{aligned} \mu_{1,2}(\Phi) &\leq \mu_{1,2}(\Phi_+), & \mu_{3,4}(\Phi) &\leq \mu_{3,4}(\Phi_+), \\ \mu_{1,3}(\Phi) &\geq \mu_{1,3}(\Phi_+), & \mu_{2,4}(\Phi) &\geq \mu_{2,4}(\Phi_+), \end{aligned} \quad (52)$$

Equivalently, in kernel coordinates, we have $\alpha(\Phi) \leq \alpha(\Phi_+)$.

Proposition 38iic. Most of the quantities we consider in this proof are functions of β , but this dependency is left implicit to simplify notation. In particular, we let μ denote (the edge coordinates of) the matching rate vector in the matching model (D, λ, Φ_+) , where $\lambda = (\lambda_1, \bar{\lambda}_2 + \beta, \bar{\lambda}_3 + \beta, \lambda_4)$, with $\lambda_1 + \lambda_4 = \bar{\lambda}_2 + \bar{\lambda}_3 = \frac{1}{2}$, and Φ_+ is the greedy HRF policy defined in Lemma 39. By combining (45) with the conservation equation $\mu_{1,2} + \mu_{1,3} = \lambda_1$, we obtain the following lower and upper bounds for $\mu_{1,2}$:

$$\lambda_1 p_2 + \lambda_2 p_1 \leq \mu_{1,2} \leq \lambda_1 - (\lambda_1 p_3 + \lambda_3 p_1). \quad (53)$$

In addition, injecting the definition (43) of λ into (46) and (47) shows that, in the model (D, λ, Φ_+) , we have

$$\begin{aligned} p_2 &= p_{\emptyset} \frac{\bar{\lambda}_2 + \beta}{2\bar{\lambda}_3}, & p_3 &= p_{\emptyset} \frac{\bar{\lambda}_3 + \beta}{2\bar{\lambda}_2}, & p_{1,4} &= p_{\emptyset} \frac{1}{3\beta}, \\ p_1 &> p_{\emptyset} \frac{\lambda_1}{\frac{1}{2} + \lambda_4 + 2\beta}, & p_4 &> p_{\emptyset} \frac{\lambda_4}{\frac{1}{2} + \lambda_1 + 2\beta}, \end{aligned} \quad (54)$$

with, by the normalization equation $p_{\emptyset} + p_2 + p_3 + p_{1,4} = 1$,

$$p_{\emptyset} = \left(1 + \frac{1}{3\beta} + \frac{\bar{\lambda}_2 + \beta}{2\bar{\lambda}_3} + \frac{\bar{\lambda}_3 + \beta}{2\bar{\lambda}_2} \right)^{-1}. \quad (55)$$

Taking the limit of (54) and (55) as $\beta \rightarrow +\infty$, we conclude that both the lower-bound and the upper-bound in (53) tend to $2\lambda_1\bar{\lambda}_2$ as $\beta \rightarrow +\infty$. Then combining (53) with the squeeze theorem allows us to conclude that $\mu_{1,2}$ also tends to $2\lambda_1\bar{\lambda}_2$ as $\beta \rightarrow +\infty$. By symmetry, we

²¹ Lemma 39 focuses on Proposition 38iia. The case Proposition 38iib is trivially deduced by switching the class labels 2 and 3.

obtain directly $\mu_{1,3} \rightarrow 2\lambda_1\bar{\lambda}_3$, $\mu_{2,4} \rightarrow 2\bar{\lambda}_2\lambda_4$, and $\mu_{3,4} \rightarrow 2\bar{\lambda}_3\lambda_4$ as $\beta \rightarrow +\infty$. According to the change-of-basis equation $\mu = (2\lambda_1\bar{\lambda}_2 + \alpha, 2\lambda_1\bar{\lambda}_3 - \alpha, \beta, 2\bar{\lambda}_2\lambda_4 - \alpha, 2\bar{\lambda}_3\lambda_4 + \alpha)$ from (43), this means that $\alpha_+ \rightarrow 0$ as $\beta \rightarrow +\infty$. \blacktriangleleft

Proof of Lemma 39. Consider a greedy policy Φ adapted to the graph D . We will prove the inequality relations in (52) using a coupling argument. More specifically, we will compare the matching models (D, I, Φ) and (D, I, Φ_+) , where $I = (I_t, t \in \mathbb{N})$ is a sequence of i.i.d. classes, such that $I_t = i$ with probability $\lambda_i/(\lambda_1 + \lambda_2 + \lambda_4 + \lambda_3)$ for each $t \in \mathbb{N}$ and $i \in V$.

Considering the model (D, I, Φ) , we let Q_t denote the vector of queue sizes at time t , $L_{t,i}$ the number of class- i items among the first t arrivals, and $M_{t,\{i,j\}}$ (or $M_{t,i,j}$ for short) the number of times that classes i and j are matched over the first t arrivals, for each $t \in \mathbb{N}$ and $i, j \in V$, as defined in (2)–(5). We introduce similar notation for the model (D, I, Φ^+) , the only difference being that all quantities have superscript $+$. Since both models have the same sequence of incoming items, we have $L_{t,i} = L_{t,i}^+$ for each $t \in \mathbb{N}$ and $i \in V$. As usual, we also assume that $Q_0 = Q_0^+ = 0$. Neither $(Q_t, t \in \mathbb{N})$ nor $(Q_t^+, t \in \mathbb{N})$ need to be Markov chains for our argument to hold.

Our end goal is to prove that the following inequalities are satisfied at each time $t \in \mathbb{N}$:

$$M_{t,1,2} \leq M_{t,1,2}^+, \quad (56-1,2)$$

$$M_{t,3,4} \leq M_{t,3,4}^+, \quad (56-3,4)$$

$$M_{t,1,3} \geq M_{t,1,3}^+, \quad (56-1,3)$$

$$M_{t,2,4} \geq M_{t,2,4}^+. \quad (56-2,4)$$

Injecting these inequalities into the definition (8) of the matching rates yields the inequalities (52). We will prove (56) by induction over time $t \in \mathbb{N}$. The following equations will be instrumental to prove the induction step:

$$Q_{t,1} + Q_{t,4} = Q_{t,1}^+ + Q_{t,4}^+, \quad Q_{t,2} = Q_{t,2}^+, \quad Q_{t,3} = Q_{t,3}^+, \quad (57)$$

$$L_{t,1} = Q_{t,1} + M_{t,1,2} + M_{t,1,3} = Q_{t,1}^+ + M_{t,1,2}^+ + M_{t,1,3}^+, \quad (58)$$

$$L_{4,t} = Q_{t,4} + M_{t,2,4} + M_{t,3,4} = Q_{t,4}^+ + M_{t,2,4}^+ + M_{t,3,4}^+. \quad (59)$$

Equation (57) follows from Proposition 36iii, while (58) and (59) follow from our assumption that the arrivals in both models are coupled. Furthermore, given the definition of $Q_G(D)$ in (44), we know that only one integer among $Q_{t,1} + Q_{t,4}$, $Q_{t,2}$, and $Q_{t,3}$ can be positive, for each $t \in \mathbb{N}$.

We now proceed to the induction step. Let $t \in \mathbb{N}$. We now prove that, assuming that the inequalities (56) are satisfied at time t , these inequalities are also satisfied at time $t + 1$. We distinguish several cases depending on the value of I_t :

Case $I_t = 1$: We have directly $M_{t+1,i,j} = M_{t,i,j}$ and $M_{t+1,i,j}^+ = M_{t,i,j}^+$ for $(i, j) \in \{(2, 4), (3, 4)\}$, hence the induction assumption implies that (56-2,4) and (56-3,4) hold at time $t + 1$.

Since the policy Φ is greedy, we only have three mutually-exclusive cases:

Case $Q_{t,2} \geq 1$: The class-1 item is matched with a class-2 item already present, and we obtain $M_{t+1,1,2} = M_{t,1,2} + 1$ and $M_{t+1,1,3} = M_{t,1,3}$.

Case $Q_{t,3} \geq 1$: The class-1 item is matched with a class-3 item already present, and we obtain $M_{t+1,1,2} = M_{t,1,2}$ and $M_{t+1,1,3} = M_{t,1,3} + 1$.

Case $Q_{t,2} = Q_{t,3} = 0$: The class-1 item is left unmatched, and we obtain $M_{t+1,1,2} = M_{t,1,2}$ and $M_{t+1,1,3} = M_{t,1,3}$.

Since the policy Φ^+ is also greedy, we can repeat the same argument for the quantities associated with Φ^+ . Combining this observation with (57) yields $M_{t+1,1,2} - M_{t+1,1,2}^+ = M_{t,1,2} - M_{t,1,2}^+$ and $M_{t+1,1,3} - M_{t+1,1,3}^+ = M_{t,1,3} - M_{t,1,3}^+$. Hence, the induction assumption implies directly that (56-1,2) and (56-1,3) are satisfied at time $t + 1$.

Case $I_t = 2$: We have directly $M_{t+1,i,j} = M_{t,i,j}$ and $M_{t+1,i,j}^+ = M_{t,i,j}^+$ for $(i,j) \in \{(1,3), (3,4)\}$, hence the induction assumption implies that (56-1,3) and (56-3,4) hold at time $t + 1$. Proving that (56-1,2) and (56-2,4) also hold at time $t + 1$ is more intricate, and we will distinguish three mutually-exclusive cases depending on the values of $Q_{t,1}$, $Q_{t,4}$, $Q_{t,1}^+$, and $Q_{t,4}^+$:

Case $Q_{t,1}^+ + Q_{t,4}^+ = 0$: Under both policies, the class-2 item is either matched with a class-3 item or added to the queue. In particular, we obtain $M_{t+1,i,j} = M_{t,i,j}$ and $M_{t+1,i,j}^+ = M_{t,i,j}^+$ for $(i,j) \in \{(1,2), (2,4)\}$, so that (56-1,2) and (56-2,4) are again satisfied at time $t + 1$ thanks to the induction assumption.

Case $Q_{t,1}^+ + Q_{t,4}^+ \geq 1$: We again subdivide this case into three mutually-exclusive cases:

Case $Q_{t,1}^+ \geq 1$ and $Q_{t,4}^+ \geq 1$: We have $M_{t+1,1,2}^+ = M_{t,1,2}^+ + 1$ and $M_{t+1,2,4}^+ = M_{t,2,4}^+$ by definition of the policy Φ^+ , while for the policy Φ we only know that $M_{t+1,1,2} \in \{M_{t,1,2}, M_{t,1,2} + 1\}$ and $M_{t+1,2,4} \in \{M_{t,2,4}, M_{t,2,4} + 1\}$. We can verify that (56-1,2) and (56-2,4) hold at time $t + 1$ thanks to the induction assumption.

Case $Q_{t,1}^+ = 0$ and $Q_{t,4}^+ \geq 1$: By greediness, the policy Φ^+ matches the incoming class-2 item with a class-4 item, and we obtain $M_{t+1,1,2}^+ = M_{t,1,2}^+$ and $M_{t+1,2,4}^+ = M_{t,2,4}^+ + 1$. If the policy Φ makes the same decision, then we also have $M_{t+1,1,2} = M_{t,1,2}$ and $M_{t+1,2,4} = M_{t,2,4} + 1$, hence (56-1,2) and (56-2,4) hold at time $t + 1$ thanks to the induction assumption. Otherwise, the policy Φ matches the class-2 item with a class-1 item, meaning that $M_{t+1,1,2} = M_{t,1,2} + 1$ and $M_{t+1,2,4} = M_{t,2,4}$. Importantly, this is only possible if $Q_{t,1} \geq 1$. We now prove (56-1,2) and (56-2,4) as follows:

- Proving (56-1,2) boils down to proving $M_{t+1,1,2}^+ \geq M_{t,1,2} + 1$. We have successively:

$$M_{t+1,1,2}^+ - M_{t,1,2} = (Q_{t,1} - Q_{t,1}^+) + (M_{t,1,3} - M_{t,1,3}^+) \geq 1 + 0 = 1,$$

where the equality follows from (58) and the inequality follows from the induction assumption and the fact that $Q_{t,1} \geq 1$ and $Q_{t,1}^+ = 0$.

- Proving (56-2,4) boils down to proving $M_{t+1,2,4}^+ + 1 \leq M_{t,2,4}$. We have successively

$$M_{t,2,4} - M_{t+1,2,4}^+ = (Q_{t,4}^+ - Q_{t,4}) + (M_{t,3,4}^+ - M_{t,3,4}) \geq 1 + 0 = 1,$$

where the equality follows from (59) and the inequality follows from the induction assumption and the observation that $Q_{t,4}^+ - Q_{t,4} = Q_{t,1} - Q_{t,1}^+ \geq 1$.

Intuitively, the only way that an incoming class-2 item is matched at time t with a class-1 item under the policy Φ and with a class-4 item under the policy Φ^+ is if, in the past, the policy Φ^+ had made one more match along edge $\{1,2\}$ and one less match along edge $\{2,4\}$ compared to the policy Φ .

Case $Q_{t,1}^+ \geq 1$ and $Q_{t,4}^+ = 0$: This case is symmetrical to the previous case.

Case $I_t = 3$: This case is symmetrical to the case $I_t = 2$.

Case $I_t = 4$: This case is symmetrical to the case $I_t = 1$.

◀

Proposition 38 can be interpreted as follows. Intuitively, the kernel coordinate α given in (43) (also shown in Figure 3d) acts like a slider that determines how much edges $\{1,2\}$ and $\{3,4\}$ are (dis)avored compared to edges $\{1,3\}$ and $\{2,4\}$ on the long run. Remarkably,

Proposition 38 shows that the greedy policy that favors edges $\{1, 2\}$ and $\{3, 4\}$ (resp. $\{1, 3\}$ and $\{2, 4\}$) the most *in the long run* is also the policy that favors these edges the most *in the short run*. This result is proved by a coupling argument. In addition, Proposition 38 uses the lower bound (45) in Proposition 37 to prove that, in the limit as $\beta \rightarrow +\infty$, all greedy policies yield the same matching rate vector.

Taken together, Statements i and iic in Proposition 38 show that, as $\beta \rightarrow +\infty$, the interval $\Pi_{\mathcal{G}}$ becomes reduced to a single point $\alpha = 0$, meaning that all greedy policies yield the same vector of matching rates, with edge coordinates $\mu = (2\lambda_1\bar{\lambda}_2, 2\lambda_1\bar{\lambda}_3, \beta, 2\bar{\lambda}_2\lambda_4, 2\bar{\lambda}_3\lambda_4)$. The rationale behind this result is the following. In the regime where $\beta = \mu_{2,3} \rightarrow +\infty$, we see in (43) that the arrival rates of classes 2 and 3 become large compared to those of classes 1 and 4. As a result, items of classes 1 and 4 are matched (almost) always immediately, and unmatched items belong to either class 2 or class 3 (but not both at the same time). In the proof of Proposition 38iic, this intuition is formalized by taking the limit of (46) as $\beta \rightarrow +\infty$, which yields $p_{\emptyset} \rightarrow 0$, $p_2 \rightarrow 2\bar{\lambda}_2$, $p_3 \rightarrow 2\bar{\lambda}_3$, and $p_{1,4} \rightarrow 0$. In this regime, the greediness of the policy allows no degree of flexibility in choosing the class of the item to which an incoming item is matched, so the matching rates are unique.

► **Remark 11.** *Some quantities in Proposition 38 are functions of the matching rate $\beta = \mu_{2,3}$, but this dependency is kept implicit to simplify notation. In particular, the intervals $\Pi_{\geq 0}$ and $\Pi_{> 0}$ do not depend on β , but the interval $\Pi_{\mathcal{G}}$ and the coordinates α_+ and α_- do.*

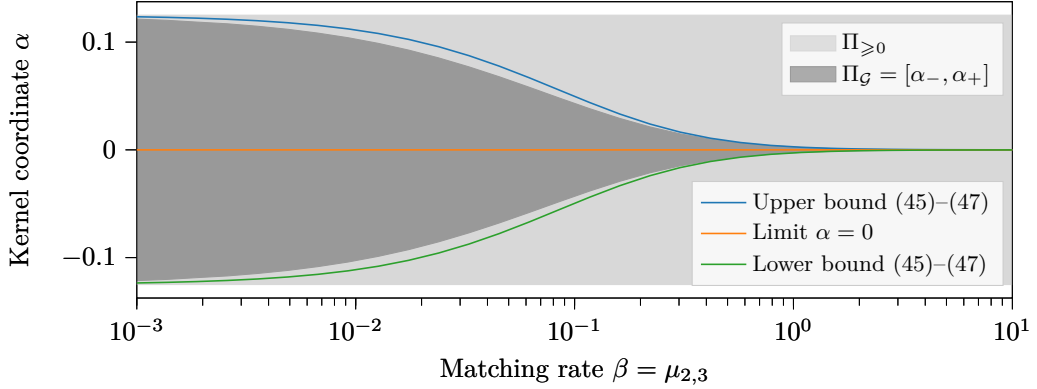
Numerical results.

To illustrate Proposition 38, Figure 25 shows a symmetric example with $\lambda_1 = \bar{\lambda}_2 = \bar{\lambda}_3 = \lambda_4 = \frac{1}{4}$. The figure compares $\Pi_{\geq 0}$ and $\Pi_{\mathcal{G}}$ with the bounds (45)–(47) (converted in the kernel coordinates) and the limit $\alpha = 0$. Each point forming the shape of $\Pi_{\mathcal{G}} = [\alpha_-, \alpha_+]$ is obtained by running a simulation consisting of 10^{10} steps (as specified in Section 7). As announced by Proposition 38, $\Pi_{\mathcal{G}}$ becomes reduced to a single point $\alpha = 0$ when $\beta \rightarrow +\infty$. We also notice that the bounds (45)–(47) are not tight when β is small (in the sense that the difference between α_+ and the upper-bound is no longer negligible compared to the difference between α_+ and the boundary $2 \min(\lambda_1\bar{\lambda}_3, \bar{\lambda}_2, \lambda_4)$ of $\Pi_{\geq 0}$), while at the same time α_+ and α_- become arbitrarily close to the borders of $\Pi_{\geq 0}$ when β tends to zero (which does not contradict the fact that $\Pi_{\mathcal{G}}$ is a strict subset of $\Pi_{\geq 0}$ as long as $\beta > 0$). The gap between $\Pi_{\mathcal{G}}$ and the bounds (45)–(47) comes from the fact that, to obtain the lower bounds for p_1 and p_4 in (47), we neglected the case where there are both class-1 and class-4 unmatched items, and this case is not negligible when β is small.

F.2.3 Intuition of Conjecture 25 (fish matching problem) and discussion.

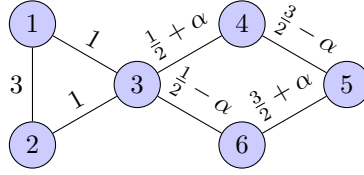
We consider a matching problem, which we call the Fish problem, where we conjecture that $\Pi_{\mathcal{G}} = \Pi_{> 0}$. The Fish matching problem is defined as follows:

$$\begin{aligned}
 V &= \{1, 2, 3, 4, 5, 6\}, \\
 E &= \{\{1, 2\}, \{1, 3\}, \{2, 3\}, \{3, 4\}, \{3, 6\}, \{4, 5\}, \{5, 6\}\}, \\
 \lambda &= (4, 4, 3, 2, 3, 2), \\
 \mathcal{B} &= \{(0, 0, 0, 1, -1, -1, 1)\}, \\
 \mu^\circ &= (3, 1, 1, \frac{1}{2}, \frac{1}{2}, \frac{3}{2}, \frac{3}{2}), \\
 \mu &= (3, 1, 1, \frac{1}{2} + \alpha, \frac{1}{2} - \alpha, \frac{3}{2} - \alpha, \frac{3}{2} + \alpha), \quad \alpha \in \mathbb{R}.
 \end{aligned} \tag{60}$$



■ **Figure 25** Evolution of $\Pi_{\mathcal{G}}$ as a function of β in the diamond problem (43) with $\lambda_1 = \bar{\lambda}_2 = \bar{\lambda}_3 = \lambda_4 = \frac{1}{4}$. For each β , $\Pi_{\mathcal{G}}$ is estimated by simulating the model (D, λ, Φ_+) and leveraging the symmetry of the problem. $\Pi_{\geq 0}$ and other bounds are displayed for comparison. All results are expressed in kernel coordinates.

The general solution μ given in (60) is shown in Figure 12. We can verify by a direct inspection that this matching problem is stabilizable and that $\Pi_{\geq 0} = [-\frac{1}{2}, \frac{1}{2}]$ in kernel coordinates. The kernel coordinate α acts like a slider that is positive (resp. negative) if matches along edges $\{3, 4\}$ and $\{5, 6\}$ are more (resp. less) frequent than matches along edges $\{3, 6\}$ and $\{4, 5\}$.

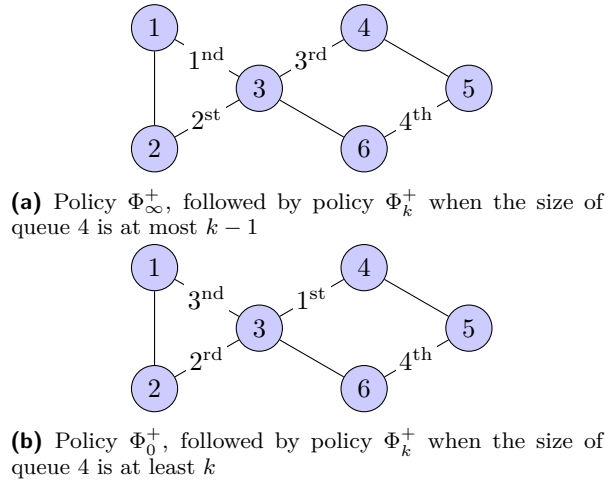


■ **Figure 12 (repeated)** Generic solution to (CE) in the Fish matching problem of (60) with $\lambda = (4, 4, 3, 2, 3, 2)$.

To support Conjecture 25, we build two families of stable greedy policies, denoted by $(\Phi_k^+)_{k \in \mathbb{N}}$ and $(\Phi_k^-)_{k \in \mathbb{N}}$, such that $\lim_{k \rightarrow +\infty} \alpha(\Phi_k^+) = \frac{1}{2}$ and $\lim_{k \rightarrow +\infty} \alpha(\Phi_k^-) = -\frac{1}{2}$. The conclusion then follows from the convexity of the set $\Pi_{\mathcal{G}}$ (Proposition 21). We focus on the family $(\Phi_k^+)_{k \in \mathbb{N}}$, as the family $(\Phi_k^-)_{k \in \mathbb{N}}$ is symmetrical (in the sense that it suffices to exchange classes 4 and 6).

The family $(\Phi_k^+)_{k \in \mathbb{N}}$ is defined as follows. Let Φ_∞^+ denote the HRF policy where edges have the following decreasing reward order: $\{1, 3\}$, $\{2, 3\}$, $\{3, 4\}$, $\{5, 6\}$, followed by all other edges in an arbitrary order. Let Φ_0^+ denote the HRF policy where edges have the following decreasing reward order: $\{3, 4\}$, $\{2, 3\}$, $\{1, 3\}$, $\{5, 6\}$, followed by all other edges in an arbitrary order. The important point is that both policies prioritize edges $\{3, 4\}$ and $\{5, 6\}$ over edges $\{3, 6\}$ and $\{4, 5\}$ (with the hope that this lead to a high α), but that Φ_∞^+ gives higher priority to the “tail” of the fish, while Φ_0^+ gives higher priority to the “trunk”. Now, for each $k \in \mathbb{N}$, Φ_k^+ is the deterministic size-based policy that follows Φ_∞^+ when the queue size of class 4 is at most $k - 1$ and Φ_0^+ when the queue size of class 4 is at least k (that is, $\Phi_k^+(q, i) = \Phi_\infty^+(q, i)$ if $q_4 \leq k - 1$ and $\Phi_k^+(q, i) = \Phi_0^+(q, i)$ if $q_4 \geq k$).

The rationale behind this definition is as follows. According to (60), α is maximal (i.e., equal to $\frac{1}{2}$) when $\mu_{3,4}$ is equal to 1 and $\mu_{3,6}$ is equal to 0. If we allow *non-greedy* policies,



■ **Figure 26** HRF (greedy) policies considered in the Fish matching problem of (60).

$\alpha = \frac{1}{2}$ can be achieved by merely applying the edge-filtering variant of ML (or any maximally-stable policy) on the bijective subgraph of G obtained by eliminating edge $\{3, 6\}$, that is, by *never* performing a match between classes 3 and 6. As we will see, the family $(\Phi_k^+)_{k \in \mathbb{N}}$ of (stable greedy) policies emulates this (non-greedy) edge-filtering policy by favoring edge $\{3, 4\}$ over edge $\{3, 6\}$ while making the probability that $q_4 = 0$ arbitrary small as k increases. Roughly speaking, mixing the policies Φ_∞^+ and Φ_0^+ allows us to control q_4 : Φ_∞^+ is not stable and makes q_4 go to infinity, while Φ_0^+ is stable and reduces q_4 when it is large. All in all, Φ_k^+ keeps the value of q_4 around k , so that the probability that $q_4 = 0$ is low when k is large. In the limit, q_4 is always positive, so that the class-3 items that are not matched with classes 1 or 2 are drained by class 4 (while class 6 is matched only with class 5), so that $\mu_{3,4}(\Phi_k^+)$ tends to 1 and $\mu_{3,6}(\Phi_k^+)$ tends to 0 as $k \rightarrow \infty$.

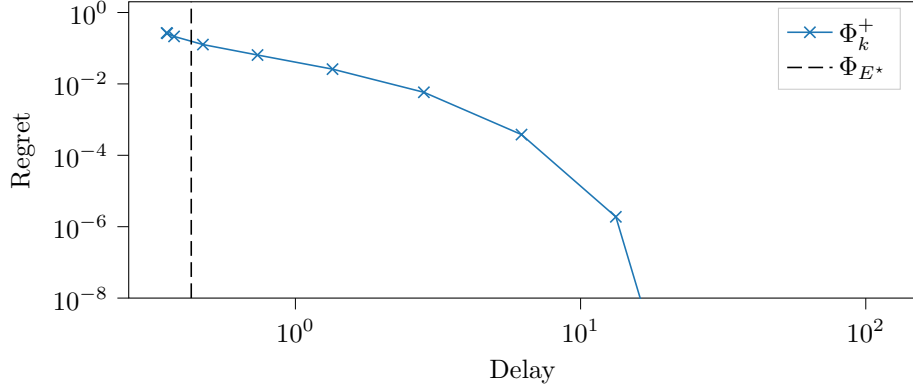
We expect that a rigorous proof that $\mu_{3,6}(\Phi_k^+)$ tends to 0 as $k \rightarrow +\infty$ will involve the following steps, some of which may require further investigation or analysis:

1. For each $k \in \mathbb{N}$, the matching model (G, λ, Φ_k^+) is stable. Stability can be proved by applying a fluid-limit argument that generalizes the framework of [19]²². Therefore, for each $k \in \mathbb{N}$, we can consider a random vector $Q_k = (Q_{k,1}, Q_{k,2}, \dots, Q_{k,n})$ distributed like the vector of queue sizes in the matching model (G, λ, Φ_k^+) in stationary regime, and we know that the vector $\mu(\Phi_k^+)$ is well-defined.
2. For each $k \in \mathbb{N}$, by definition of the policy Φ_k^+ , we have

$$\mu(\Phi_k^+)_{3,6} = \lambda_3 P(Q_{k,4} = 0, Q_{k,6} > 0) + \lambda_6 P(Q_{k,3} > 0, Q_{k,5} = 0) \leq (\lambda_3 + \lambda_6) P(Q_{k,4} = 0),$$

where the inequality arises because the events $(Q_{k,4} = 0, Q_{k,6} > 0)$ and $(Q_{k,3} > 0, Q_{k,5} = 0)$ are both included into the event $(Q_{k,4} = 0)$ (in the latter case, this is because the greediness of the policy prevents the system from containing unmatched items of classes 3 and 4 at the same time). Consequently, to prove that $\mu(\Phi_k^+)_{3,6} \rightarrow 0$, it suffices to prove that $P(Q_{k,4} = 0) \rightarrow 0$ as $k \rightarrow +\infty$. This argument rigorously supports the intuition that, if class 4 becomes “unstable”, then this class “drains” all class-3 items that are not matched with classes 1 and 2.

²² Formalizing this generalization is outside the scope of this paper and could be the topic of a future work.



■ **Figure 27** Approaching the vertex $\alpha = 1/2$ in the Fish graph from Figure 12 with greedy policies ($r = (2, 2, 2, 1, -1, 0, 1)$).

3. To prove that $P(Q_{k,4} = 0) \rightarrow 0$ as $k \rightarrow +\infty$, we believe that we can reason as follows:
- We can prove that, for each $k \in \mathbb{N}$, the conditional average rate at which class 3 is matched with either class 4 or class 6, *given that* $Q_{k,4} \leq k - 1$, is upperbounded by

$$\lambda_3 \left(1 + \frac{\lambda_1}{\lambda_2 + \lambda_3 - \lambda_1} + \frac{\lambda_2}{\lambda_1 + \lambda_3 - \lambda_2} \right)^{-1} = \frac{9}{11}.$$

This upper bound follows by coupling the fish problem with a “worst-case” matching problem in which there is an infinite backlog of items of classes 4 and 6 to be matched with class 3. Roughly speaking, this worst-case matching problem can be obtained by taking $\lambda_4 = \lambda_6 = +\infty$ in the fish problem; then, focusing on nodes 1, 2, and 3, the system behaves like a paw graph with an unstable pendant node obtained by merging classes 4 and 6.

- Using this upper bound, we prove that at least one class among classes 4 and 6 becomes unstable, in the sense that $\lim_{k \rightarrow +\infty} P(Q_{k,4} = 0) = 0$ or $\lim_{k \rightarrow +\infty} P(Q_{k,6} = 0) = 0$. Since we can verify otherwise that class 6 does not become unstable, it follows that $\lim_{k \rightarrow +\infty} P(Q_{k,4} = 0) = 0$.

The complete proof is left as an open question for future works. ◀

Numerical results.

The performance of Φ_k^+ is shown in Figure 27. Since the vertex at $\alpha = 1/2$ is bijective, we include the delay of the reward-optimal policy Φ_{E^*} for comparison. The figure corroborates Conjecture 25, demonstrating that the regret of Φ_k^+ converges to zero. However, as mentioned earlier, this reduction in regret comes at the cost of increasing the size of the class-4 queue, thus increasing the delay of the policy.