



HAL
open science

Stochastic dynamic matching: A mixed graph-theory and linear-algebra approach

Céline Comte, Fabien Mathieu, Ana Bušić

► **To cite this version:**

Céline Comte, Fabien Mathieu, Ana Bušić. Stochastic dynamic matching: A mixed graph-theory and linear-algebra approach. 2022. hal-03502084v3

HAL Id: hal-03502084

<https://hal.science/hal-03502084v3>

Preprint submitted on 25 Feb 2022 (v3), last revised 25 Jun 2023 (v4)

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.

Stochastic dynamic matching: A mixed graph-theory and linear-algebra approach

Céline Comte¹ ✉

Eindhoven University of Technology, Eindhoven, The Netherlands

Fabien Mathieu ✉

LINCS, Paris, France

Ana Bušić ✉

Inria, Paris, France

DI ENS, CNRS, PSL University, Paris, France

Abstract

Stochastic dynamic matching problems have recently drawn attention in the stochastic-modeling community due to their numerous applications, ranging from supply-chain management to kidney exchange programs. In this paper, we consider a matching problem in which items of different classes arrive according to independent Poisson processes. Unmatched items are stored in a queue, and compatibility constraints are described by a simple graph on the classes, so that two items can be matched if their classes are neighbors in the graph. We analyze the efficiency of matching policies, not only in terms of system stability, but also in terms of matching rates between different classes.

Our results rely on the observation that, under any stable policy, the matching rates satisfy a conservation equation that equates the arrival and departure rates of each item class. Our main contributions are threefold. We first introduce a mapping between the dimension of the solution set of this conservation equation, the structure of the compatibility graph, and the existence of a stable policy. In particular, this allows us to derive a necessary and sufficient stability condition that is verifiable in polynomial time. Secondly, we describe the convex polytope of non-negative solutions of the conservation equation. When this polytope is reduced to a single point, we give a closed-form expression of the solution; in general, we characterize the vertices of this polytope using again the graph structure. Lastly, we study the parts of the polytope that can be achieved by a stable policy. We show that greedy policies are limited to the interior of the polytope, with a strict inclusion in general. In contrast, non-greedy policies can reach any point of the interior of this polytope, and also reach the boundary of the polytope depending on a simple condition on the vertices.

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

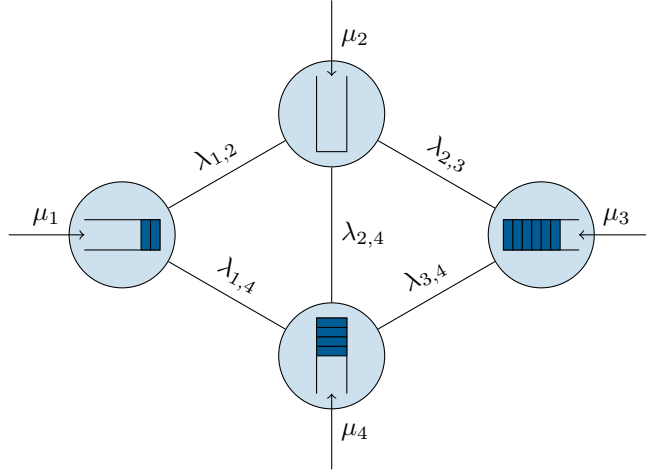
Keywords and phrases stochastic dynamic matching, graph theory, linear algebra, stability, matching rates, conservation equation

1 Introduction

Stochastic dynamic matching problems, in which items arrive at random instants to be matched with other items, have recently drawn a lot of attention in the stochastic-modeling community. These challenging control problems are indeed 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 two categories, called demand and supply, and the objective is to maximize a certain long-term performance criteria by appropriately matching demand items with supply items.

¹ Corresponding author.

In this paper, we consider the following dynamic matching problem. 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 the matching policy decides which matches are performed and when. All in all, a stochastic matching model is described by a triplet (G, μ, Φ) , where $G = (V, E)$ is the compatibility graph, $\mu = (\mu_1, \mu_2, \dots, \mu_n)$ is the vector of per-class arrival rates, and Φ is the matching policy. In Figure 1 for instance, there are four item classes numbered from 1 to 4; classes 2 and 4 are compatible with all classes, while classes 1 and 3 are only compatible with classes 2 and 4.



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

We propose a unified approach to study two closely-related performance criteria, namely the stability and the matching rates along edges. Formally, a matching model (G, μ, Φ) is said to be stable if the associated continuous-time Markov chain is positive recurrent. Assuming that the matching model (G, μ, Φ) is stable, the matching rate λ_k along an edge $k \in E$ with endpoints $i, j \in V$ is the rate at which class- i items and class- j items are matched.

1.1 Contributions

The following observation is powerful despite its simplicity, and it is fundamental to all our results. Under any stable policy, the arrival rate of class- i items is equal to the departure rate of these items, which in turn is equal to the sum of the matching rates along the edges that are incident to node i . In other words, the matching rates satisfy the following *conservation* equation:

$$\sum_{k \in E_i} \lambda_k = \mu_i, \quad i \in V,$$

where $E_i \subseteq E$ is the set of edges that are incident to node i . In matrix form, this equation rewrites $A\lambda = \mu$, where A is the incidence matrix of the compatibility graph G . The solution set of this conservation equation is related to the structure of the graph G via the linear application $y \in \mathbb{R}^m \mapsto Ay \in \mathbb{R}^n$, where n is the number of nodes (or classes), and m is the number of edges. We say that the graph G is surjective (resp. injective, bijective) if the linear application $y \in \mathbb{R}^m \mapsto Ay \in \mathbb{R}^n$ is surjective (resp. injective, bijective), and we give simple

equivalent conditions in terms of the graph structure (Definitions 2.3–2.6 and Proposition 2.7). Our main contributions are threefold, and all rely on these definitions.

We first prove that there exists a direct relation between the possible solutions of the conservation equation, the structure of the compatibility graph, and the existence of a stable policy. More specifically, we prove that a compatibility graph G is stabilizable (in the sense that there exist a vector μ and a policy Φ such that the matching model (G, μ, Φ) is stable) if and only if the graph G is surjective (Proposition 3.1). We then prove that a matching problem (G, μ) is stabilizable (in the sense that there exists a policy Φ such that the matching model (G, μ, Φ) is stable) if and only if the conservation equation has a solution with positive components (Proposition 3.2). In particular, this allows us to verify stabilizability in a time that is polynomial in the number of classes and edges.

We next describe the affine space of solutions of the conservation equation and the convex polytope of solutions with non-negative components. When this convex polytope is reduced to a single point, we derive a closed-form expression for the solution (Proposition 4.1). When this convex polytope is not reduced to a single point, we characterize its vertices, again using the graph structure. We prove in particular that a non-negative solution of the conservation equation is a vertex of the convex polytope if and only if the subgraph restricted to the support of this vector is injective (Proposition 5.5).

Lastly, we investigate the parts of the polytope that can be achieved by an actual stable policy. We first focus on greedy policies, i.e. policies that never postpone a feasible match. We show that greedy policies are limited to the interior of the polytope and that in general, the inclusion is strict (Propositions 6.2, 6.4, and 6.5 and Corollary 6.6). In contrast, non-greedy policies can reach any point of the interior of this polytope (Proposition 6.13 and Corollary 6.14). They can also reach part or all the boundary of this polytope, depending on conditions that are again expressed in terms of the bijectivity of subgraphs of G (Proposition 6.11 and Corollary 6.12).

1.2 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 will be described in details in Section 2.1 [5, 6, 11, 14, 20, 23, 25]. Among these works, the following are particularly relevant because directly related to our results on stability. The paper [23] is the earliest work on this matching model. It derives several necessary and sufficient stability conditions that are instrumental in several of our results, in particular Propositions 3.1 and 3.2. This work also proves that the *match-the-longest* policy is maximally stable (in the sense that always leads to stability whenever the matching problem (G, μ) is stabilizable), a result that is also applied in Proposition 3.2. The papers [14, 25] focus on the *first-come-first-matched* policy. In particular, [25] proves that the first-come-first-matched policy is maximally stable, and [14] provides a new sufficient stability condition we prove to be also necessary in Proposition 3.2.

Other variants of the model were studied recently, and an interesting future work would consist of generalizing our results to these variants. In particular, the paper [20] consider item abandonments, the paper [6] considers graphs with loops, and the papers [19, 26, 27] allow matches between several items by replacing the graph with a hypergraph.

The recent paper [5] is perhaps the closest to ours, and we provide a detailed discussion to highlight the relation with our paper. The equivalence of statements (ii) and (iv) in [5, Theorem 1] is synonym to the equivalence of statements 2 and 3 in Proposition 3.2. Our proof is significantly shorter because it relies more heavily on existing results. [5, Theorem 4] is a special case of 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). Several other formulas derived in [5, Section 7] are special cases of the formulas derived in Proposition 4.1. The model in [5] is slightly more general because it consider graphs with loops, that is, an edge can have identical endpoints, but this paper does not adopt the mixed graph-theory and linear-algebra approach that supports most of our results.

Bipartite stochastic matching

The first example of a stochastic matching model in the literature, which predated the model that we consider, is the bipartite matching model introduced in [12] and studied in [1, 2, 3, 8, 9, 10, 15]. In this model, the compatibility graph is bipartite, with two parts that correspond to supply and demand items, respectively. This bipartite model differs from ours by its arrival process: time is slotted and, during each time slot, one demand item *and* one supply item arrive. Several works have made contributions about stability [2, 8] and matching rates [2], and obtained results similar to those derived in the literature on our model. 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 [22]. Although the questions raised in static and dynamic matching are often different, the conservation equation that we obtain is reminiscent of several results in 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 $\lambda_k \in \{0, 1\}$ for each edge $k \in E$ that maximize $\sum_{k \in E} \lambda_k$ while satisfying the conservation equation with $\mu_i = 1$ 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 [22, Section 7.2]. Therefore, the fractional matching polytope defined in [22, 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 polytope ².

1.3 Outline

The remainder of the paper is organized as follows. Section 2 gives a formal definition of the model and explains our approach. In particular, we introduce the conservation equation and define the notions of surjective, injective, and bijective graph that will be instrumental in the rest of the paper. In Section 3, we give necessary and sufficient stability conditions in terms of the solutions of the conservation equation. Sections 4 and 5 characterize the solution set of the conservation equation and give a closed-form expression of the solution

² The fractional matching polytope is actually defined using non-strict inequalities rather than equalities. However, one can verify that both convex polytopes have the same non-zero vertices.

when it is unique. Lastly, in Section 6, we study the set of matching rates that are effectively achievable under a stable matching policy.

2 Model and approach

We now provide a detailed description of our model and introduce the concepts of injective, surjective, and bijective compatibility graphs.

2.1 Stochastic dynamic matching

Consider a stochastic dynamic matching system in which items arrive at random times to be matched with other items. Each incoming item may be matched with any unmatched item of a compatible class; in this case, both items disappear immediately. Unmatched items are gathered in a waiting queue. In this paper, such a stochastic dynamic matching system will be described by a triplet (G, μ, Φ) , where G is the compatibility graph, μ is the vector of arrival rates, and Φ is the matching policy. We now review each component in details. Notation is summarized in Table 1, and an example is shown in Figure 1.

2.1.1 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 loop). We let n denote the number of nodes and m the number of edges. The set of nodes is denoted by $V = \{v_1, v_2, \dots, v_n\}$, and each node corresponds to a class in the matching model. When there is no ambiguity, for ease of notation, we will refer to a class v_i using its index i . 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 there is an edge with endpoints i and j in the graph. When there is no ambiguity, for ease of notation, we will refer to an edge $e_k \in E$ with endpoints $i, j \in V$ using either its index k or its set of endpoints $\{i, j\}$. In Figure 1 for instance, there are four item classes numbered from 1 to 4. Classes 2 and 4 are compatible with all classes, but classes 1 and 3 are only compatible with classes 2 and 4. The absence of loop means that an item of a given class cannot be matched with other items of the same class.

2.1.2 Vector of arrival rates

Class- i items arrive according to an independent Poisson process with rate $\mu_i > 0$, for each $i \in V$. The vector of arrival rates is denoted by $\mu = (\mu_1, \mu_2, \dots, \mu_n) \in \mathbb{R}_{>0}^n$. Scaling all components of μ by the same positive constant is equivalent to changing the time unit, so we can renormalize μ without changing the system dynamics. In this paper, we will often use the *unit* normalization in which $\sum_{i \in V} \mu_i = 1$. The couple (G, μ) is called a *matching problem*.

2.1.3 Matching policy

Given a matching problem (G, μ) , the policy specifies, upon each item arrival, which matches are performed depending on the system state and the class of the incoming item. The nature of the information that the state needs to retain in order to obtain a continuous-time Markov chain depends on the policy that we consider. To overcome this difficulty, we assume that the state descriptor is an integral part of the definition of a policy.

Formally, a matching policy Φ is a tuple (\mathcal{C}, π) , where \mathcal{C} is a countably infinite state space and π is a function that defines transitions. More specifically, for each $c \in \mathcal{C}$ and $i \in V$, $\pi(\cdot|c, i) : \mathcal{C} \rightarrow \mathbb{R}_{\geq 0}$ is a probability distribution on \mathcal{C} , so that $\pi(d|c, i)$ is the probability that the system jumps from state c to state d upon the arrival of a class- i item. We assume that the matching policy is *adapted* to the matching problem (G, μ) , in the sense that only items with compatible classes are matched, and matches occur only upon item arrivals. In the rest of the paper, we will focus exclusively on matching policies that satisfy Assumption 1 below.

► **Assumption 1.** *We assume that the matching policy Φ satisfies the following assumptions:*

1. *The evolution of the system state defines a continuous-time irreducible Markov chain with state space \mathcal{C} .*
2. *The system state contains enough information to calculate the sizes of the waiting queues of the system, i.e. the number of unmatched items of each class. In other words, there exists a function $h : \mathcal{C} \rightarrow \mathbb{N}^n$ such that, for each $c \in \mathcal{C}$ and $i \in V$, the i -th component of the vector $h(c)$ is the number of unmatched class- i items in state c .*
3. *The state space \mathcal{C} contains a state \emptyset such that $h(\emptyset) = 0$, and we have $h(c) \neq 0$ for each $c \in \mathcal{C} \setminus \{\emptyset\}$. We call \emptyset the empty state.*

Statement 1 in Assumption 1 is standard [8, 23]. Statements 2 and 3 guarantee that the intuitive notion of system stability is indeed captured by the positive recurrence of the Markov chain $\{X(t), t \geq 0\}$, as formalized in Definition 2.1 below. These two statements will be applied in the proofs of Propositions 6.4, 6.5, and A.1 and Corollary 6.6. These proofs remain valid if there are multiple empty sets, as long as the preimage $h^{-1}(0) = \{c \in \mathcal{C} : h(c) = 0\}$ of the vector $0 \in \mathbb{N}^n$ is a finite set. Statement 3 will also be applied in the proof of Proposition 6.1. Not all policies have an empty state; for example, this statement rules out policies that match an incoming item only when there are already at least two unmatched items (assuming the initial state is not empty).

We call $(h(c))_i$, the number of unmatched class- i items in state c , the *queue size of i in c* (c is omitted if it is clear in the context). Similarly, we say that *queue i is empty* (resp. *non-empty*) to express that $(h(c))_i = 0$ (resp. $(h(c))_i > 0$).

We now introduce some examples of policies that will be used in the paper.

Greedy policies

A policy is said to be *greedy* if an incoming item is matched as soon as it arrives if possible, that is, when there is an unmatched item that is compatible. Equivalently, a policy is greedy if the set of unmatched items classes under this policy is always an independent set of the compatibility graph. Here are two examples of deterministic greedy policies:

- **Match-the-longest:** an incoming item is matched immediately with an item of the compatible class that has the longest queue, if any. The system state $q = (q_1, q_2, \dots, q_n)$ gives the queue size of each class, and its state space is $\mathcal{C} = \{q \in \mathbb{N}^n : q_i q_j = 0 \text{ if } j \in V_i\}$, where $V_i \subset V$ is the set of nodes adjacent to node i , for each $i \in V$. The transition probabilities are as follows:
 - if there is $j \in V_i$ such that $q_j > 0$, then $\pi(q - e_{j^*} | q, i) = 1$, where $j^* = \operatorname{argmax}_{j \in V_i} (q_j, j)$ (we use the class index to tie-break queues of same size) and e_{j^*} is the n -dimensional vector with 1 in component j^* and 0 elsewhere;
 - otherwise, $\pi(q + e_i | q, i) = 1$.

As a state of *match-the-longest* is exactly the sizes of the queues, its function h is the identity ($h(q) = q$).

- **First-come-first-matched:** an incoming item is matched immediately with the compatible item that has been waiting the longest, if any. The system state $c = (c_1, c_2, \dots, c_p)$ gives the sequence of unmatched item classes, ordered by arrival times, so that c_1 is the class of the oldest unmatched item. The state space $\mathcal{C} \subseteq V^*$ is made of all finite sequences that do not contain items from compatible classes. In other words, $\mathcal{C} = \bigcup_{\mathcal{I} \in \mathbb{I}} \mathcal{I}^*$, where \mathbb{I} is the set of independent sets of the graph G . The transition probabilities are as follows:
 - if there exists $r \in \{1, 2, \dots, p\}$ such that $c_r \in V_i$, $\pi(c'|c, i) = 1$ with $c' = (c_1, \dots, c_{r^*-1}, c_{r^*+1}, \dots, c_p)$, where $r^* = \min\{r \in \{1, 2, \dots, p\} : c_r \in V_i\}$;
 - otherwise, $\pi(c'|c, i) = 1$ with $c' = (c_1, c_2, \dots, c_p, i)$.

The function h associated to a *first-come-first-matched* policy is just a counter function: $(h(c))_i = |\{j \in c : j = i\}|$.

Note that an empty state \emptyset exists for every greedy policy as long as the compatibility graph G has no isolated node: from any state with unmatched items, an empty state can be reached with an appropriate sequence of incoming items.

Filtering policies

A policy is said to be *filtering* if there exists $E' \subsetneq E$ such that for any state $c \in \mathcal{C}$ and arrival node $i \in V$, the probability to choose $e \notin E'$ when the state is c and an item of class i arrives is 0. Informally, a filtering policy operates on a subgraph $G' = (V, E')$ of G . The edges from E' (resp. from $E \setminus E'$) are called the *allowed* edges (resp. the *forbidden* edges) of the policy.

A policy is said to be *semi-filtering* with a threshold k if there exists $E' \subsetneq E$ such that for any state $c \in \mathcal{C}$ such that $\max_{j \in V} (h(c))_j < k$ and arrival node $i \in V$, the probability to choose $e \notin E'$ when the state is c and an item of class i arrives is 0. Informally, a semi-filtering policy acts as a filtering policy *as long as the queue sizes are not too big*.

A greedy policy like *match-the-longest* can be turned into a filtering or semi-filtering policy by making it operate on a subgraph G' (up to a threshold k for the semi-filtering case). In general, the resulting policy is not greedy (on G), but it still admits an empty state \emptyset as long as the subgraph G' has no isolated node.

Filtering and semi-filtering versions of *match-the-longest* will be used in Section 6.

Stability

In the remainder, we will identify the *matching model* (G, μ, Φ) with the continuous-time Markov chain $\{X(t), t \geq 0\}$. This allows us to define the notions of *stability*, *stabilizability*, and *maximal stability* in Definition 2.1 below.

► **Definition 2.1** (Stability, Stabilizability, and Maximal stability).

1. A matching model (G, μ, Φ) is called *stable* if the associated Markov process $\{X(t), t \geq 0\}$ is positive recurrent.
2. A compatibility graph G is called *stabilizable* if there exist a vector $\mu \in \mathbb{R}_{>0}^n$ of arrival rates and a matching policy Φ such that the matching model (G, μ, Φ) is stable.
3. A matching problem (G, μ) is called *stabilizable* if there exists a matching policy Φ such that the matching model (G, μ, Φ) is stable.
4. A *generic policy* is a function Φ that associates, to each compatibility graph G , a matching policy $\Phi(G)$ adapted to G . A generic policy Φ is called *maximally stable* if the matching model $(G, \mu, \Phi(G))$ is stable whenever (G, μ) is stabilizable.

With a slight abuse of language, we also use the term *generic policy* to describe the matching policy $\Phi(G)$ for any compatibility graph G . The greedy policies *match-the-longest* and *first-come-first-matched* introduced above are obviously generic. They were proved to be maximally stable in [23, 25].

► **Remark 2.2.** On a broader note, our definition of a matching policy is intended to be general, but we still impose several limitations for ease of notation. For instance, our assumption that items can only be matched upon arrivals rules out policies that perform matches in batches at fixed time intervals. We believe that all our results can be extended to other policies if necessary on a case-by-cases basis.

2.1.4 Vector of matching rates and conservation equation

If the matching model (G, μ, Φ) is stable, the *matching rate* λ_k along edge $e_k = \{i, j\}$ is defined as the long-run average number of matches between a class- i item and a class- j item per time unit. Let $\lambda(G, \mu, \phi) = (\lambda_1, \dots, \lambda_m) \in \mathbb{R}_{\geq 0}^m$ denote the vector of matching rates associated with (G, μ, ϕ) . For ease of notation, in the rest of the paper, we will use $\lambda(\Phi)$ or λ , instead of $\lambda(G, \mu, \Phi)$, if the underlying matching problem or model is clear in the context. These matching rates satisfy the following *conservation* equation:

$$\sum_{k \in E_i} \lambda_k = \mu_i, \quad i \in V, \quad (1)$$

where $E_i \subseteq E$ is the set of edges that are incident to node i , for each $i \in V$. In matrix form, this conservation equation writes

$$A\lambda = \mu, \quad (2)$$

where the $n \times m$ matrix $A = (a_{i,k})_{i \in V, k \in E}$ is the *incidence* matrix of the graph G , defined by $a_{i,k} = 1$ if edge k is incident to node i and $a_{i,k} = 0$ otherwise. The majority of the results presented here are deeply connected to the conservation equation (2). In particular, the last part of the paper focuses on the solutions of (2) and considers, for a given stabilizable problem (G, μ) , the following sets:

$$\begin{aligned} \Lambda &= \{y \in \mathbb{R}^m : Ay = \mu\} \text{ (studied in Section 5.1),} \\ \Lambda_{\geq 0} &= \{y \in \mathbb{R}_{\geq 0}^m : Ay = \mu\} \text{ (studied in Section 5.2),} \\ \Lambda_{> 0} &= \{y \in \mathbb{R}_{> 0}^m : Ay = \mu\} \text{ (introduced in Section 6),} \\ \Lambda_G &= \{\lambda(\Phi) : \Phi \text{ is a stable greedy policy}\} \text{ (studied in Section 6.1),} \\ \Lambda_{\mathcal{P}} &= \{\lambda(\Phi) : \Phi \text{ is a stable policy}\} \text{ (studied in Section 6.2).} \end{aligned}$$

Note that, while it is not the main focus of the present work, long-term matching rates can usually be defined even if the matching model (G, μ, Φ) is unstable. In that case, the sum of matching rates along the edges incident to a node are at most equal to the arrival rates, that is, $A\lambda \leq \mu$. Lemma 6.10 shows an example of an unstable policy that admits a limit matching rate.

2.1.5 Numerical evaluation

To support the results presented in this paper, especially the ones from Section 6, simulations were performed using the Python package *Stochastic Matching* [24]. Unless otherwise stated, a matching model is evaluated by simulating 10^{10} arrivals.

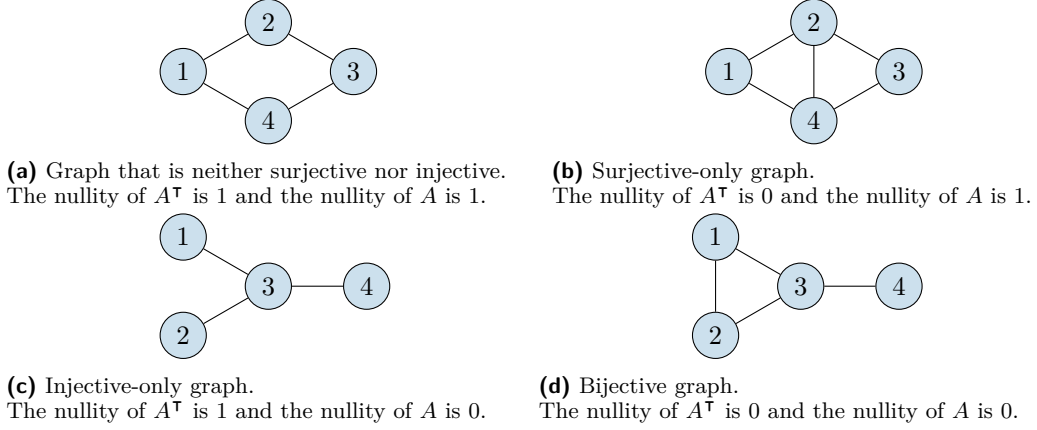
General notation	
$\mathbb{N}, \mathbb{R}, \mathbb{R}_{\geq 0}, \mathbb{R}_{> 0}$	Sets of non-negative integers, real numbers, non-negative real numbers, positive real numbers.
$\geq, \leq, >, <$	Component-wise comparison in \mathbb{R}^n .
$ \mathcal{A} $	Cardinality of the set \mathcal{A} .
Graph notation	
$G = (V, E)$	Simple graph G with $ V = n$ vertices and $ E = m$ edges.
v_i	Vertex indexed by i (denoted i if there is not ambiguity).
$e_{i,j}, \{i, j\},$ or e_k	Edge between vertices i and j , indexed by k .
\mathbb{I}	Family of independent sets of the graph G .
$V_i \subseteq V$	Set of neighbors of node v_i in the graph G .
$V(\mathcal{I}) = \bigcup_{i \in \mathcal{I}} V_i$	Set of neighbors of the vertices indexed by the independent set $\mathcal{I} \in \mathbb{I}$.
$d_{i,j}$	Distance between nodes i and j .
$d_{i,k} = \min(d_{i,j}, d_{i,j'})$	Distance between node i and edge k with endpoints j and j' .
K_ℓ	Complete graph of size $\ell \geq 3$.
C_ℓ	Cycle of size $\ell \geq 3$.
P_ℓ	Path of length $\ell \geq 0$.
$KP_{\ell,r,p}$	Kayak paddle: two cycles C_ℓ and C_r attached by a path P_p .
Matching notation	
$\mu = (\mu_i)_{1 \leq i \leq n}$	Vector of arrival rates of the item classes.
Φ	A matching policy.
$\lambda = (\lambda_k)_{1 \leq k \leq m} = (\lambda_{i,j})_{\{i,j\} \in E}$	Vector of matching rates along the edges.
$\Lambda_{\mathcal{P}}$	Set of matching rates achieved by stable policies.
$\Lambda_{\mathcal{G}}$	Set of matching rates achieved by stable greedy policies.
Linear-algebra notation	
$x = (x_1, x_2, \dots, x_n)$	A vector in \mathbb{R}^n . All vectors in \mathbb{R}^n are column vectors.
$y = (y_1, y_2, \dots, y_m)$	A vector in \mathbb{R}^m . All vectors in \mathbb{R}^m are column vectors.
$A = (a_{i,k})_{i \in V, k \in E}$	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.
$\mathcal{B} = (b_1, \dots, b_d)$	Basis of the right kernel of the matrix A if G is surjective.
$\Lambda = \{y \in \mathbb{R}^m : Ay = \mu\}$	Affine space of the solutions of the conservation equation (2).
$\Lambda_{\geq 0} = \{y \in \mathbb{R}_{\geq 0}^m : Ay = \mu\}$	Polytope of non-negative solutions of (2).
$\Lambda_{> 0} = \{y \in \mathbb{R}_{> 0}^m : Ay = \mu\}$	Set of positive solutions of (2).

■ **Table 1** Table of notation

2.2 Surjectivity, injectivity, and bijectivity

Definitions 2.3–2.6 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 defined by its incidence matrix A is surjective (resp. injective, bijective). Interestingly, we find equivalent conditions in terms of the graph structure. As we will see later, these notions are fundamental to study the stability of stochastic matching

models and the associated matching rates. In particular, we will see in Section 3 that a compatibility graph G is stabilizable if and only if G is surjective in the sense of Definition 2.3. Later, in Sections 3 and 4, we will see that the matching rates in a matching problem (G, μ) are independent of the matching policy Φ (as long as the model is stable) if and only if G is bijective in the sense of Definition 2.5. Examples are shown in Figure 2.



■ **Figure 2** Examples of surjective and injective graphs.

► **Definition 2.3** (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:

1. The function $y \in \mathbb{R}^m \mapsto Ay \in \mathbb{R}^n$ is surjective.
2. The equation $Ay = \mu$ of unknown $y \in \mathbb{R}^m$ has at least one solution for each $\mu \in \mathbb{R}^n$.
3. The left kernel of the matrix A is trivial.
4. Each connected component of the graph G is non-bipartite.

Proof. The equivalence of conditions 1, 2, and 3 is a well-known result in linear algebra. We prove that conditions 3 and 4 are equivalent. This proof is adapted from [16, Lemma 2.2.3].

Let a_1, a_2, \dots, a_n denote the rows of the matrix A , so that $a_i = (a_{i,k})_{k \in \{1, \dots, m\}}$, where $a_{i,k} = 1$ if node i is an endpoint of edge k , and $a_{i,k} = 0$ otherwise. 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

$$x_1 a_1 + x_2 a_2 + \dots + x_n a_n = 0.$$

For each $k \in \{1, \dots, m\}$, the k -th component of this 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 $x_{i_p} = (-1)^p x_{i_1}$ for each $p \in \{1, 2, \dots, k\}$.

First assume that each connected component of G is non-bipartite, and let V' denote the set of nodes in a given connected component. Since this component is non-bipartite, there exists a cycle $i_1, i_2, \dots, i_k, i_{p+1} = i_1$ consisting of an odd number p of nodes. The previous remark implies that $x_{i_1} = 0$, which in turn implies that $x_i = 0$ for each $i \in V'$.

On the contrary, if there exists a connected component of G that is bipartite with parts V_+ and V_- , then we build a non-zero vector by choosing $x_i = 1$ for each $i \in V_+$, $x_i = -1$ for each $i \in V_-$, and $x_i = 0$ for each $i \in V \setminus (V_+ \cup V_-)$. ◀

► **Definition 2.4** (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:

1. The function $y \in \mathbb{R}^m \mapsto Ay \in \mathbb{R}^n$ is injective.
2. The equation $Ay = \mu$ of unknown $y \in \mathbb{R}^m$ has at most one solution for each $\mu \in \mathbb{R}^n$.
3. The right kernel of the matrix A is trivial.
4. 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.

Proof. The equivalence of conditions 1, 2, and 3 is a well-known result in linear algebra. We now prove that conditions 3 and 4 are equivalent.

Let us assume for now that the graph G is connected. We first remark that the proof of Definition 2.3 shows that the nullity of A^\top is 0 if G is non-bipartite and 1 if G is bipartite. We therefore distinguish two cases:

- If G is non-bipartite, 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, the nullity of A^\top is 1, and we conclude similarly 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 3 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 4.

If the graph G is not connected, we can rewrite the matrix 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. ◀

► **Definition 2.5** (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:

1. The function $y \in \mathbb{R}^m \mapsto Ay \in \mathbb{R}^n$ is bijective.
2. The equation $Ay = \mu$ of unknown $y \in \mathbb{R}^m$ has exactly one solution for each $\mu \in \mathbb{R}^n$.
3. The matrix A is invertible.
4. Each connected component of the graph G contains one cycle and this cycle is odd.

Proof. The function $y \in \mathbb{R}^m \mapsto Ay \in \mathbb{R}^n$ is bijective if and only if it is both surjective and injective. Hence, the equivalence of conditions 1 to 4 follows directly from Definitions 2.3 and 2.4. ◀

► **Definition 2.6** (Surjective-only graph and 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 2.7.** Consider an undirected graph $G = (V, E)$ with n nodes and m edges.

1. If G is surjective, then $n \leq m$.
2. If G is injective, then $n \geq m$.
3. If G is bijective, then $n = m$.
4. If G is surjective, then G is also injective if and only if $n = m$.
5. If G is injective, then G is also surjective if and only if $n = m$.

Proof. These statements are again well-known results in linear algebra. ◀

3 Stability conditions

This section gives necessary and sufficient conditions under which a compatibility graph G or a matching problem (G, μ) is *stabilizable* in the sense of Definition 2.1.

3.1 Stabilizable 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 3.1.** *Let G be an undirected graph. The following conditions are equivalent:*

1. *The graph G is stabilizable.*
2. *The graph G is surjective.*

Proof. Equivalence between condition 1 in Proposition 3.1 and condition 4 in Definition 2.3 has been proved in [23, Theorem 1]. ◀

Unless stated otherwise, in the rest of the paper, we assume that the graph G is surjective.

The equivalence between condition 1 in Proposition 3.1 and condition 4 in Definition 2.3 was already proved in [23] in the context of stochastic matching models. However, to the best of our knowledge, the equivalence between condition 1 in Proposition 3.1 and the other definitions of surjectivity introduced in Definition 2.3 has not been considered in the literature on stochastic matching models yet. As we will see later, this new characterization of the stabilizability of a graph G will be useful to analyze the matching rates.

3.2 Stabilizable arrival rates

We now turn to the stabilizability of a matching problem (G, μ) . As recalled in Section 2.1, two examples of greedy policies that stabilize the model whenever this matching problem is stabilizable are *match-the-longest* [23] and *first-come-first-matched* [25]. Proposition 3.2 below provides necessary and sufficient conditions for the matching problem (G, μ) to be stabilizable; condition 2 was already derived in [23], but condition 3 is new.

► **Proposition 3.2.** *Consider an undirected surjective graph G and a vector $\mu \in \mathbb{R}_{>0}^n$. The following conditions are equivalent:*

1. *The matching problem (G, μ) is stabilizable.*
2. *For each $\mathcal{I} \in \mathbb{I}$, we have $\sum_{i \in \mathcal{I}} \mu_i < \sum_{i \in V(\mathcal{I})} \mu_i$.*
3. *The conservation equation (2) has a solution $\lambda \in \mathbb{R}_{>0}^m$ (i.e., with all components positive).*

Proof. Equivalence of conditions 1 and 2 follows from Lemma 1, Proposition 2, and Theorem 2 in [23]. We now prove that conditions 2 and 3 are equivalent. Condition 2 implies condition 3 because: (i) according to [23], under condition 2, (G, μ, Φ) is stable if Φ is the *match-the-longest* policy, and (ii) the associated vector λ of matching rates satisfies condition 3 by ergodicity. That condition 3 implies condition 2 was proved in [14, Lemma 12]. ◀

One might imagine that the time complexity to verify condition 2 in Proposition 3.2 is exponential in the number n of classes in general, as the number of independent sets itself is exponential in n . Yet it was proved in [23, Proposition 1] that there exists an $O(n^3)$ -time algorithm to verify this condition. Unfortunately, this verification is indirect in the sense that it requires constructing a second graph called the bipartite double cover of G . From this perspective, condition 3 gives a more direct way of verifying whether a graph G is stabilizable. We make a case disjunction, depending on whether the graph G (which we have assumed to be surjective) is surjective-only or bijective.

► Remark 3.3. As observed in [14, Lemma 12], if the graph G is surjective, one can always find an particular vector $\mu \in \mathbb{R}_{\geq 0}^n$ such that the matching problem (G, μ) is stabilizable by choosing $\mu = Ay$ for some $y \in \mathbb{R}_{> 0}^m$. A simple example is $y = (\beta, \dots, \beta)$ for some $\beta > 0$, which corresponds to having the components of μ proportional to the degree of each node.

Checking stability on bijective graphs

If the graph G is bijective, then the matrix A is invertible and the conservation equation (2) has a unique solution $A^{-1}\mu$. This implies that the matching problem (G, μ) is stabilizable if and only if all components of $A^{-1}\mu$ are positive. The special case of bijective graphs will be investigated in details in Section 4, including a direct expression of $A^{-1}\mu$.

Checking stability on surjective-only graphs

If the compatibility graph G is surjective-only, the conservation equation (2) has multiple solutions. To determine if one of these solutions is positive, it suffices to solve a linear optimization problem that maximizes the smallest component of a solution of (2). In block-matrix notation, this linear optimization problem can be written as:

$$\begin{aligned} & \underset{z \in \mathbb{R}^{m+1}}{\text{Maximize}} && \begin{bmatrix} 0_{1 \times m} & 1 \end{bmatrix} z, \\ & \text{Subject to} && \begin{bmatrix} A & 0_{n \times 1} \end{bmatrix} z = \mu, \\ & && \begin{bmatrix} I_{m \times m} & -1_{m \times 1} \end{bmatrix} z \geq 0_{m \times 1}, \end{aligned} \tag{3}$$

where, for clarity, we let $0_{p \times q}$ denote the $p \times q$ zero matrix, $1_{p \times q}$ the $p \times q$ all-ones matrix, $I_{p \times p}$ the p -dimensional identity matrix. Here, the first m components of the vector z are the components of a vector $y \in \mathbb{R}^m$ that satisfies (2), and the last component of z is a lower bound of the components of this vector y . The equality constraint means that y satisfies (2), and the inequality constraint means that the last component of z is less than or equal to its other components. The value to maximize is the last component of the vector z .

The linear optimization problem (3) has a solution with positive components if and only if the conservation equation (2) has a solution with positive components. According to Proposition 3.2, this is equivalent to saying that the matching problem (G, μ) is stabilizable. Therefore, to verify if a matching problem (G, μ) is stabilizable, it suffices to find a solution of the linear optimization problem (3) and to check if all its components are positive.

Observe that the optimization problem (3) always has solutions with finite components. Indeed, the set of vectors that satisfy the constraints of (3) contains at least one valid solution with real-valued components (this is again a consequence of the surjectivity of G). We just need to consider an arbitrary solution y of the conservation equation (2) (see Section 5.1.2 for a concrete example using the Moore-Penrose inverse) and to let $z_y = (y_1, y_2, \dots, y_m, \min(y))$. Any solution better than z_y has all its components lower-bounded by $\min(y)$ and upper-bounded by $\max(\mu) - \min(0, (n-1)\min(y))$. The latter bound is obtained by observing that, if edge k is incident to node i and if $(y'_1, y'_2, \dots, y'_m, x')$ is a solution of (3) such that $x' \geq \min(y)$, then $y'_k = \mu_i - \sum_{\ell \in E_i \setminus k} y'_\ell$ by (2). We then use the inequalities $\mu_i \leq \max(\mu)$ and $\sum_{\ell \in E_i \setminus k} y'_\ell \geq \min(0, (n-1)x') \geq \min(0, (n-1)\min(y))$. Therefore, the solutions better than z_y belong to a compact set of \mathbb{R}^{m+1} , which ensures the existence of an optimal solution with finite components.

The optimization problem (3) 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 [21].

If z is a solution of (3), we call the corresponding $y \in \mathbb{R}^m$ a *maximin* solution of the conservation equation Equation (2).

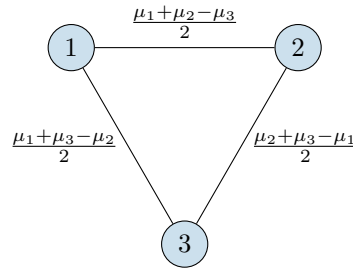
3.3 Early examples

We now illustrate Propositions 3.1 and 3.2 on a few toy examples that will also introduce useful notions for Sections 4–6.

3.3.1 Bijective graphs

We first consider a compatibility graph G that is both surjective and injective. According to Definition 2.5, the conservation equation (2) has a unique solution for each vector $\mu \in \mathbb{R}^n$ of arrival rates. Proposition 3.2 implies that the components of this solution are positive if and only if the matching problem (G, μ) is stabilizable. By Remark 3.3, one can always exhibit a vector $\mu \in \mathbb{R}_{>0}^n$ of arrival rates that satisfies this condition.

► **Example 3.4 (Triangle).** If the graph G is a triangle graph \mathcal{C}_3 and the vector $\mu = (\mu_1, \mu_2, \mu_3)$ is given, the solution of the conservation equation (2) is unique and showed in Figure 3. According to condition 3 in Proposition 3.2, (G, μ) is stabilizable if and only if all components of this solution are positive. This is indeed equivalent to condition 2 in Proposition 3.2, which reads $\mu_1 < \mu_2 + \mu_3$, $\mu_2 < \mu_1 + \mu_3$, and $\mu_3 < \mu_1 + \mu_2$. Note that an alternate way to express this condition consists of saying that μ_1 , μ_2 , and μ_3 are the lengths of the sides of a non-degenerate triangle. Under these conditions, the matching model (G, μ, Φ) is stable under the unique greedy policy Φ (this will be shown in Proposition 6.4).



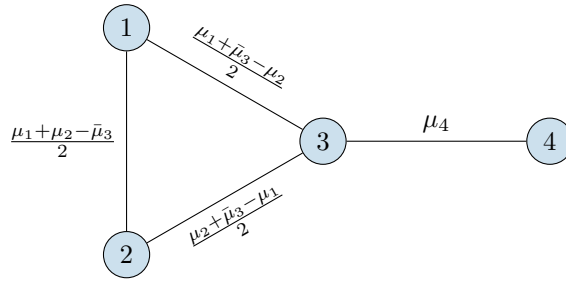
■ **Figure 3** Matching rates in the triangle graph \mathcal{C}_3 .

► **Example 3.5 (Paw graph).** If G is a paw graph, the solution of the conservation equation (2) is again unique and showed in Figure 4. $\bar{\mu}_3 = \mu_3 - \mu_4$ represents the remaining rate of class 3 after the needs of class 4 have been deduced. After this subtraction, the matching rates along edges $\{1, 2\}$, $\{1, 3\}$, and $\{2, 3\}$ are as defined in the triangle graph of Figure 3.

Note that, if positive matching rates guarantee the existence of stable greedy policies like *match-the-longest*, some greedy policies can be unstable. Lemma 6.9 will give a “recipe” that can be used to build an unstable greedy policy on a stabilizable matching problem (G, μ) with a paw graph G .

3.3.2 Bipartite graph (that is neither injective nor surjective)

► **Example 3.6 (Square graph).** Figure 5 shows a square graph $G = \mathcal{C}_4$. This graph is not surjective because it is bipartite between $\{1, 3\}$ (odd component) and $\{2, 4\}$ (even component). Therefore, according to Proposition 3.1, this graph is not stabilizable. Yet, given a vector



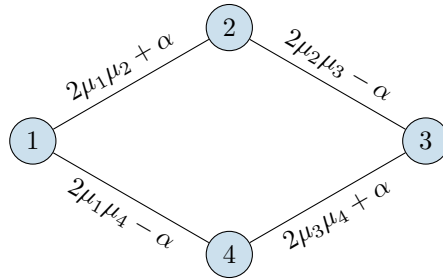
■ **Figure 4** Matching rates in the paw graph. $\bar{\mu}_3 = \mu_4 - \mu_3$ denotes the residual rate that class 3 can provide to classes 1 and 2.

$\mu = (\mu_1, \mu_2, \mu_3, \mu_4)$ of arrival rates, the conservation equation (2) may still have a solution with positive components. This does not contradict Proposition 3.2, as the three statements in this proposition are equivalent only if the graph G is surjective. Assuming unit normalization, the conservation equation (2) has a solution if and only if

$$\mu_1 + \mu_3 = \mu_2 + \mu_4 = \frac{1}{2}. \quad (4)$$

If (4) is not satisfied, the difference between the numbers of unmatched items from the odd and even components evolves like a biased random walk on the integer number line $\{\dots, -2, -1, 0, 1, 2, \dots\}$. This implies that the underlying Markov process is transient, as the number of unmatched items in the component with the highest arrival rate grows linearly with time. On the other hand, if (4) is satisfied, then the random walk is unbiased, but the system is still unstable because the corresponding Markov chain is null recurrent. (Existing studies of matching in bipartite graphs usually solve this issue by coupling arrivals in both components [2, 8, 12] or by assuming that items have a finite patience time [20].)

If (4) is satisfied, the solutions of the conservation equation (2) can be described with a parameter α as shown in Figure 5. The positive solutions correspond to values of α such that $-2 \min(\mu_1\mu_2, \mu_3\mu_4) < \alpha < 2 \min(\mu_2\mu_3, \mu_1\mu_4)$.



■ **Figure 5** Matching rates in the square graph C_4 (not stabilizable), with the normalization $\mu_1 + \mu_3 = \bar{\mu}_2 + \bar{\mu}_4 = \frac{1}{2}$.

3.3.3 Surjective-only graphs

We finally consider compatibility graphs G that are surjective but not injective. In other words, the graph G is stabilizable and the conservation equation (2) has an infinite number of solutions. Whether or not these solutions are achievable by a matching policy will be discussed in Section 6.

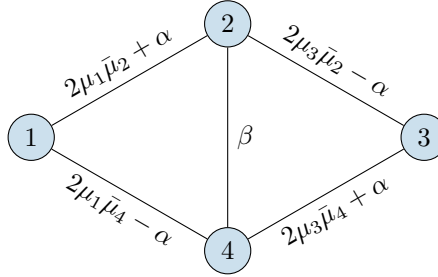
► **Example 3.7** (Diamond (double-fan) graph). Figure 6 shows a diamond graph, that is, a square graph with an additional edge between nodes 2 and 4. Compared to Example 3.6, this additional edge makes the graph non-bipartite, and therefore surjective, so that the graph is stabilizable according to Proposition 3.1. For ease of computation, we assume that the vector $\mu = (\mu_1, \mu_2, \mu_3, \mu_4)$ of arrival rates is normalized so that $\mu_1 + \mu_3 = \frac{1}{2}$. According to condition 2 in Proposition 3.2, the matching problem (G, μ) is stabilizable if and only if

$$\mu_2 < \mu_1 + \mu_3 + \mu_4, \quad \mu_4 < \mu_1 + \mu_2 + \mu_3, \quad \mu_1 + \mu_3 < \mu_2 + \mu_4 \quad (5)$$

With $\beta = \frac{1}{2}(\mu_2 + \mu_4 - \mu_1 - \mu_3) = \frac{1}{2}(\mu_2 + \mu_4) - \frac{1}{4}$, $\bar{\mu}_2 = \mu_2 - \beta$, and $\bar{\mu}_4 = \mu_4 - \beta$, these conditions rewrite:

$$\bar{\mu}_4 > 0, \quad \bar{\mu}_2 > 0, \quad \beta > 0. \quad (6)$$

If these inequalities are satisfied, the general solution of the conservation equation (2) can be described with a parameter α as shown in Figure 6. In particular, the positive solutions correspond to values of α such that $-2 \min(\mu_1 \bar{\mu}_2, \mu_3 \bar{\mu}_4) < \alpha < 2 \min(\bar{\mu}_2 \mu_3, \mu_1 \bar{\mu}_4)$.



■ **Figure 6** Matching rates in the diamond graph, with the normalization $\mu_1 + \mu_3 = \frac{1}{2}$. $2\beta = \mu_2 + \mu_4 - \mu_1 - \mu_3 = \mu_2 + \mu_4 - \frac{1}{2}$ is the difference between the arrival rates of the even and odd components. $\bar{\mu}_2 = \mu_2 - \beta$ and $\bar{\mu}_4 = \mu_4 - \beta$ represent the residual rates that classes 2 and 4 can provide to classes 1 and 3.

Intuitively, compared to the square graph, stabilizable matching problems (G, μ) have a positive difference of 2β between the arrival rates of the even and odd components. This difference is absorbed by the central edge $\{2, 4\}$, which has matching rate β . After subtracting β from μ_2 and μ_4 , the solutions of the conservation equation (2) are exactly the same as in the square graph of Example 3.6.

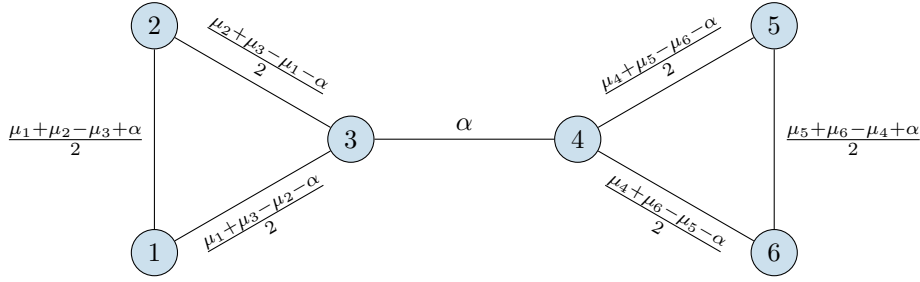
Like Example 3.4 and unlike Example 3.5, the matching model (G, μ, Φ) is stable for every greedy policy Φ provided that (6) is satisfied (this will be shown in Corollary A.2).

► **Example 3.8** (Kayak paddle graph). Figure 7 shows a kayak paddle $G = KP_{3,3,1}$, consisting of two triangles linked by an edge. According to Proposition 3.2, the matching problem (G, μ) is stabilizable if and only if there exists $\alpha > 0$ such that $(\mu_1, \mu_2, \mu_3 - \alpha)$ and $(\mu_4 - \alpha, \mu_5, \mu_6)$ are the vectors of arrival rates of two stabilizable triangle graphs \mathcal{C}_3 .

The solutions of the conservation equation (2) can be described by varying α as shown in Figure 7. Assuming that the matching problem (G, μ) is stabilizable, the solutions of the conservation equation (2) with positive components correspond to the values of α such that

$$0 < \alpha < \min(\mu_3 - |\mu_2 - \mu_1|, \mu_4 - |\mu_5 - \mu_6|).$$

Intuitively, solutions with positive components have a positive matching rate α along edge $\{3, 4\}$. After subtracting this rate from μ_3 and μ_4 , the subgraphs restricted to nodes 1, 2, and 3 and to nodes 4, 5, and 6, respectively, behave exactly like the triangle of Figure 3.



■ **Figure 7** Matching rates in the kayak paddle $KP_{3,3,1}$.

Like Example 3.5 and unlike Examples 3.4 and 3.7, the fact that (G, μ) is stabilizable does not guarantee the stability of any greedy policy.

4 Matching rates in bijective graphs

In the remainder, we will consider exclusively matching problems (G, μ) that are stabilizable. According to Proposition 3.1 and Definition 2.3, this implies that the graph G is surjective, or equivalently, that each connected component of G is non-bipartite. According to statement 4 in Proposition 2.7 and to Proposition 3.2, there are only two possible cases:

1. If $n = m$, the graph G is also bijective. The conservation equation (2) has a unique solution given by $\lambda = A^{-1}\mu$. This solution, which has positive components, gives the matching rates achieved by any stable policy. Each connected component of the graph G is a unicyclic graph, and its unique cycle is odd.
2. If $n < m$, the graph G is surjective-only. The conservation equation (2) has an infinite number of solutions, one of which has positive components. Each connected component of the graph G is non-bipartite, and at least one of these connected components contains an even cycle or a pair of odd cycles.

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

In Proposition 4.1 below, we give a simpler expression for the unique solution $\lambda = A^{-1}\mu$ of the conservation equation (2) in terms of the arrival rate vector μ , under the assumption that the graph G is bijective. We assume without loss of generality that the graph G is connected, as otherwise we can consider each connected component independently. Compared to the expression $\lambda = A^{-1}\mu$, the advantage of Proposition 4.1 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 4.2 and 4.3.

► **Proposition 4.1.** *Consider a matching problem (G, μ) with a compatibility graph $G = (V, E)$ that is connected and bijective, and consider an edge $k \in E$.*

1. *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 $V_k \subset V$ denotes the set of nodes that belong to the tree (including one of the endpoints of edge k), then the matching rate along edge k is given by*

$$\lambda_k = \sum_{i \in V_k} (-1)^{d_{i,k}} \mu_i. \quad (7)$$

2. If edge k belongs to the (unique odd) cycle of the graph G , then the matching rate along edge k is given by

$$\lambda_k = \frac{1}{2} \left(\sum_{i \in V} (-1)^{d_{i,k}} \mu_i \right), \quad (8)$$

where $d_{i,k}$ is the distance between node i and edge k , that is, the minimum distance between node i and an endpoint of edge k .

Proof. We first prove (7) 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 V_k ; the rooted tree associated with k is obtained by designating the corresponding endpoint of edge k as the root. We now prove (7) by induction on the height this rooted tree. Equation (7) is true if the depth 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 (1) to node i yields $\lambda_k = \mu_i$, which is consistent with (7). Now consider an edge k so that the associated rooted tree has height $h \geq 1$. By applying (1) to the root i of this associated rooted tree, we obtain

$$\lambda_k = \mu_i - \sum_{\ell \in E_i \setminus \{k\}} \lambda_\ell.$$

The induction hypothesis guarantees that (7) is satisfied for every $\ell \in E_i \setminus \{k\}$ (as the height of the associated rooted tree is at most $h - 1$). 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 $j \in V_\ell$ and that $V_k = \{i\} \cup (\bigcup_{\ell \in E_i \setminus \{k\}} V_\ell)$ (all sets being disjoint).

We now prove (8) for each edge k that belongs to the cycle. Since the graph G is unicyclic, deleting edge k from G yields a (connected) tree, which can be seen as 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

$$\sum_{i \in V_+} \mu_i - \sum_{i \in V_-} \mu_i = \sum_{i \in V_+} \sum_{\ell \in E_i} \lambda_\ell - \sum_{i \in V_-} \sum_{\ell \in E_i} \lambda_\ell = 2\lambda_k.$$

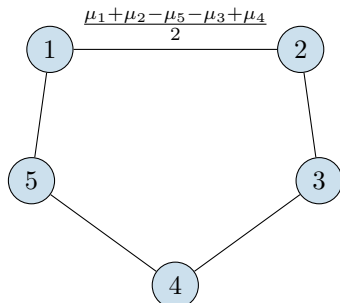
The first equality follows from (1). The second equality holds because each edge $\ell \in E \setminus \{k\}$ has one endpoint in V_+ and another in V_- , so that λ_ℓ appears exactly once in the first nested sum and once in the second; on the contrary, since both endpoints of edge k belong to V_+ , λ_k appears twice in the first nested sum and zero times in the second. Equation (8) follows by observing that $d_{i,k}$ is even if and only if $i \in V_+$. ◀

We remark that the influence of the arrival rate of a node on an edge matching rate 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 as an endpoint of that edge.

► **Example 4.2** (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 8, a direct application of statement 1 in Proposition 4.1 yields

$$\lambda_{1,2} = \frac{1}{2}(\mu_1 + \mu_2 - \mu_3 + \mu_4 - \mu_5).$$

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 $\{1, 2\}$.

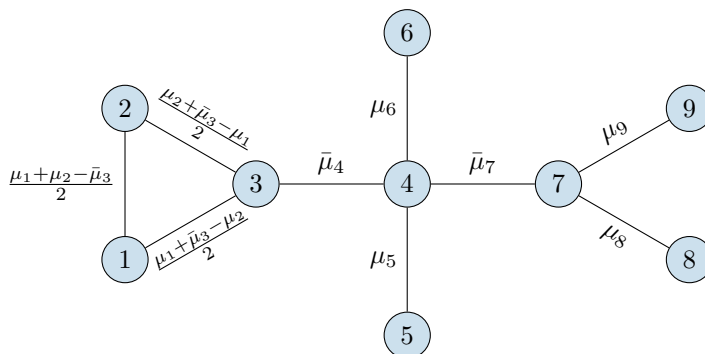


■ **Figure 8** Matching rates in the pentagon graph C_5 . Only rate $\lambda_{1,2}$ is shown for ease of display. The other rates are deduced by permutation.

► **Example 4.3** (Lying puppet). We now consider the graph of Figure 9. Edges $\{1, 2\}$, $\{1, 3\}$, and $\{2, 3\}$ belong to the cycle, and the other edges do not. According to Proposition 4.1, we have

$$\lambda_{1,2} = \frac{\mu_1 + \mu_2 - \bar{\mu}_3}{2}, \quad \lambda_{1,3} = \frac{\mu_1 - \mu_2 + \bar{\mu}_3}{2}, \quad \lambda_{2,3} = \frac{-\mu_1 + \mu_2 + \bar{\mu}_3}{2},$$

where $\bar{\mu}_3 = \mu_3 - \lambda_{3,4}$, and



■ **Figure 9** Matching rates in a “lying puppet” graph with $n = 9$ nodes and $m = 9$ edges. The differences $\bar{\mu}_7 = \mu_7 - \mu_8 - \mu_9$, $\bar{\mu}_4 = \mu_4 - \mu_5 - \mu_6 - \bar{\mu}_7$, and $\bar{\mu}_3 = \mu_3 - \bar{\mu}_4$ are the residual rates that classes 7, 4, and 3 provide to their neighbors of lower index.

$$\begin{aligned} \lambda_{4,5} &= \mu_5, & \lambda_{4,6} &= \mu_6, & \lambda_{7,8} &= \mu_8, \\ \lambda_{7,9} &= \mu_9, & \lambda_{4,7} &= \mu_7 - \lambda_{7,8} - \lambda_{7,9}, & \lambda_{3,4} &= \mu_4 - \lambda_{4,5} - \lambda_{4,6} - \lambda_{4,7}. \end{aligned}$$

This second set of equations can be obtained either by a direct application of (7) or by applying (1) recursively from the leaves. Indeed, applying (1) to nodes 5, 6, 8, and 9 gives directly the values of $\lambda_{4,5}$, $\lambda_{4,6}$, $\lambda_{7,8}$, and $\lambda_{7,9}$, then applying (1) to node 7 gives the value of $\lambda_{4,7}$, and finally applying (1) to node 4 gives the value of $\lambda_{3,4}$. The values of $\lambda_{1,2}$, $\lambda_{1,3}$, and $\lambda_{2,3}$ are similar to Example 3.5, where the arrival rate μ_3 is again replaced with the effective arrival rate $\bar{\mu}_3$ from the point of view of classes 1 and 2.

5 Solution of the conservation equation in surjective-only graphs

Consider a stabilizable matching problem (G, μ) with a surjective-only compatibility graph G . According to Definitions 2.3, 2.4, and 2.6, each connected component of the graph G is non-bipartite, and at least one of these connected components contains an even cycle or a pair of odd cycles. This means that the conservation equation (2) has an infinite number of solutions, and that at least one of these solutions has positive components. This section describes these solutions. Section 5.1 characterizes the affine space of all real-valued solutions of the conservation equation (2), while Section 5.2 focuses on the convex polytope made of the solutions with non-negative components. Whether or not these solutions are achievable by a matching policy will be discussed in Section 6.

Note that the results stated in this section are also applicable to bijective graphs but they are of little interest in that case: the solution is unique and Section 4 already gives a closed-form expression for it. This is why we focus on surjective-only graphs.

5.1 Affine space of real-valued solutions

Let Λ denote the set of solutions of the conservation equation (2), with positive, zero, or negative components, that is

$$\Lambda = \{y \in \mathbb{R}^m : Ay = \mu\}. \quad (9)$$

In Section 5.1.1 we observe that Λ is an affine space of dimension $d = n - m$ that can be described as a translation of the kernel of the incidence matrix by a solution of the conservation equation (2). Section 5.1.2 gives possible solutions that can be used. 5.1.3 gives an algorithm to construct a basis for the kernel of the incidence matrix directly from the graph.

5.1.1 Edge basis, kernel basis

The following proposition characterizes the set of solutions of the conservation equation (2) using the incidence matrix.

► **Proposition 5.1.** *Consider a stabilizable matching problem (G, μ) with a surjective-only compatibility graph G . Let A denote the incidence matrix of G . The solution set Λ of the conservation equation (2) is the affine space obtained by translating the kernel of the matrix A by a particular solution y° of the conservation equation (2), that is,*

$$\Lambda = \{y^\circ + y : y \in \ker(A)\}. \quad (10)$$

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

Proof. That the set Λ is of the form (10) is a well-known result in linear algebra. Definition 2.3 about surjectivity implies that $\text{rank}(A) = n$, and we conclude from the rank-nullity theorem that the nullity of A is $d = n - m$. The affine space Λ has the same dimension according to (10). ◀

Equation (10) tells that, up to translation, Λ is only defined by the structure of the graph G . The vector of arrival rates μ only impacts the translation y° .

Thanks to Proposition 5.1, given a particular solution y° of the conservation equation (2), and a basis $\mathcal{B} = (b_1, b_2, \dots, b_d)$ of $\ker(A)$, we can rewrite the affine space Λ as

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

Said differently, we can define the following affine isomorphism between the coordinate space \mathbb{R}^d and the d -dimensional affine space Λ :

$$\alpha = (\alpha_1, \alpha_2, \dots, \alpha_d) \in \mathbb{R}^d \mapsto y = y^\circ + \alpha_1 b_1 + \alpha_2 b_2 + \dots + \alpha_d b_d \in \Lambda. \quad (11)$$

Therefore, for given y° and \mathcal{B} , there are two natural basis to represent vectors in Λ :

- **Edge basis:** A vector of Λ is described by a vector $y = (y_1, y_2, \dots, y_m) \in \mathbb{R}^m$, where y_k represents a potential matching rate along edge k , for each $k \in E$.
- **Kernel basis:** A vector of Λ is described by a vector $\alpha = (\alpha_1, \alpha_2, \dots, \alpha_d) \in \mathbb{R}^d$, where $d = m - n$ is the dimension of the affine space Λ .

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

- A vector of Λ with coordinates α in kernel basis has coordinates $y^\circ + B\alpha$ in edge basis;
- A vector of Λ with coordinates y in edge basis has coordinates $B^+(y - y^\circ)$ in kernel basis, where B^+ is the pseudo-inverse of B .

As \mathcal{B} is a basis, the columns of B are independent, so that the pseudo-inverse B^+ has the following simple expression:

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

where the $d \times d$ matrix $B^\top B$ is invertible because $\ker(B^\top B) = \ker(B) = \{0\}$.

In practice, which basis we are actually using will be made clear by our choice of letters (either y or α). Both basis have their benefits. The edge basis, by definition, gives directly the potential matching rates. The kernel basis allows to work in lower dimension (d instead of m) and to ignore the conservation equation (2), which is implicitly enforced.

For graphs that have a low kernel dimension d , it is convenient to mix both approaches and to represent a generic vector of Λ , i.e. a generic solution of the conservation equation (2), in the form $y^\circ + B\alpha$. For instance, the solutions on Examples 3.7 and 3.8 are actually displayed in Figures 6 and 7 using that 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.

We now focus on y° (the chosen *origin* for Λ) and \mathcal{B} .

5.1.2 Particular solution

A standard way to simultaneously find a particular solution y° and characterize $\ker(A)$ consists of using the pseudoinverse (or Moore-Penrose inverse) of the matrix A . Definition 2.3 about surjectivity implies that the rows of A are linearly independent, so that the pseudoinverse A^+ of A has the following simple expression:

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

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 y^\perp and the kernel $\ker(A)$ as follows:

$$y^\perp = A^+ \mu, \quad \ker(A) = \{(I_{m \times m} - A^+ A)y : y \in \mathbb{R}^m\},$$

where $I_{m \times m}$ is the m -dimensional identity matrix. The vector y^\perp is the *least-squares solution* of the conservation equation (2), and it is orthogonal to $\ker(A)$. In general, the components of this solution are not non-negative even if non-negative solutions exist. For example, if G

is the diamond graph of Example 3.7, then the matching problem (G, μ) with $\mu = (4, 5, 1, 2)$ is stabilizable, but the solution given by the pseudoinverse is $y^\perp = (\frac{11}{4}, \frac{5}{4}, 1, \frac{5}{4}, -\frac{1}{4})$.

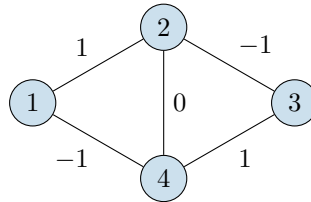
If one prefers a positive solution for y° , one alternate possibility is to use a *maximin* solution by solving the linear optimization problem Equation (3). Remind that to ensure the existence of a maximin solution we used the fact that at least one solution exists, which justifies the introduction of the solution based on the Moore-Penrose inverse.

As stated above, the Moore-Penrose inverse also provides an implicit characterization of $\ker(A)$. However, this characterization is not very practical as it relies on a projection from \mathbb{R}^m to $\ker(A)$. We now give a more direct characterization by building a basis for $\ker(A)$ based on the structure of the compatibility graph G .

5.1.3 Basis of the kernel of the incidence matrix

Recall that a vector $y \in \mathbb{R}^m$ belongs to $\ker(A)$ if and only if $Ay = 0$, which reads $\sum_{k \in E_i} y_k = 0$ for each $i \in V$. In other words, a vector $y \in \mathbb{R}^m$ belongs to $\ker(A)$ if and only if the sum of the components of y associated with the edges incident to the same node is zero. Using this observation, we first give examples of subgraphs that are supports 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, defines 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 3.7 for instance, if edges are numbered in lexicographical order, then $y = (1, -1, -1, 0, 1)$ is a vector of the unidimensional kernel, associated with the even cycle 1–2–3–4 (see Figure 10). Intuitively, even cycles can be used to move weights between “odd” and “even” edges without modifying the value of the product Ay . Actually, in this particular example, the only way to increase the matching rate along edges $\{1, 2\}$ and $\{3, 4\}$ is to reduce the matching rate along edges $\{2, 3\}$ and $\{1, 4\}$, and conversely.

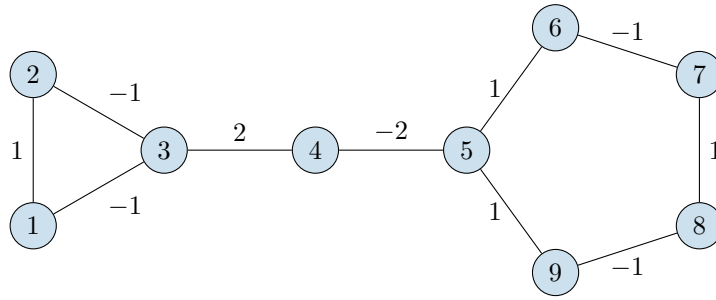


■ **Figure 10** Vector of the kernel space of the diamond graph.

Apart from even cycles, other structures of interest are kayak paddles $KP_{\ell,r,p}$ in which the lengths ℓ and r of both cycles are odd. These 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 11 shows such an assignment for $KP_{3,5,2}$.

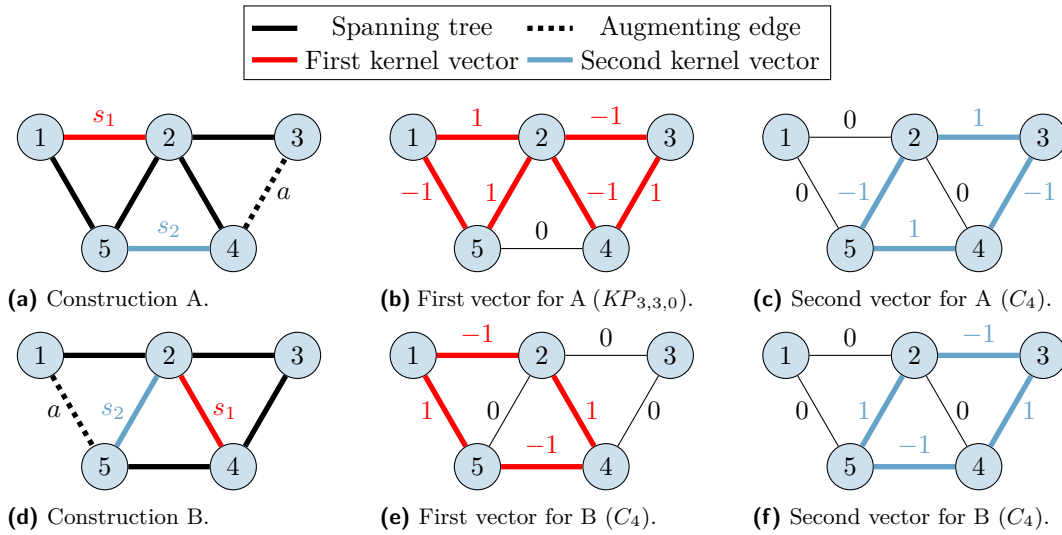
Surprisingly, for any surjective graph G , one can build a basis of $\ker(A)$ using only subgraphs of G that are even cycles and kayak paddles. This is what Algorithm 1 does. It finds $m - n$ distinct edges and associates to each one either an even-cycle base vector or a kayak-paddle base vector. We 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 via zero padding.

Figures 12 and 13 show possible runs of Algorithm 1 on the triamond and codomino graphs, both of which have a two-dimensional kernel. Note that the basis is not unique and depends on our initial choice of the spanning tree \mathcal{T} and the augmenting edge a (see lines



■ **Figure 11** Vector of the kernel space of the kayak paddle $KP_{3,5,2}$.

2 and 3 in Algorithm 1). We now verify that Algorithm 1 terminates and yields the desired result.



■ **Figure 12** Two possible constructions of a kernel basis for the triamond graph.

► **Proposition 5.2.** *Algorithm 1 terminates and returns a basis of the kernel of the incidence matrix A of the compatibility graph G .*

Proof. See Appendix B. ◀

Importantly, given an edge $k \in E$, all solutions of the conservation equation (2) 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 say that edge k belongs neither to an even cycle nor to a kayak paddle. In the diamond graph of Example 3.7 for instance, the edge $\{2, 4\}$ is the only one that does not belong to the even cycle $1-2-3-4$, and it is also the only one with a fixed value β . In general, if an edge $k \in E$ satisfies this unicity condition, then the matching rate along edge k in a stable matching model (G, μ, Φ) is independent of the policy Φ . Note that there is no general relation between the number of edges with uniquely-defined matching rates and the dimensionality d of the affine space Λ .

Data: A connected surjective-only graph $G = (V, E)$
Result: A basis \mathcal{B} of the kernel of the incidence matrix A of G

```

1  $\mathcal{B} \leftarrow \emptyset$ 
2  $\mathcal{T} \leftarrow$  the set of edges of a spanning tree of  $G$ 
3  $a \leftarrow$  an edge in  $E \setminus \mathcal{T}$  such that  $\mathcal{T} \cup \{a\}$  contains an odd cycle
4 for  $s \in E \setminus (\mathcal{T} \cup \{a\})$  do
5    $b \leftarrow 0_{m \times 1}$ 
6   if  $\mathcal{T} \cup \{a, s\}$  contains an even cycle  $C_\ell$  then
7      $c_1, \dots, c_\ell \leftarrow$  edges of  $C_\ell$ , numbered from an arbitrary starting point
8     for  $d \in \{1, \dots, \ell\}$  do
9        $k \leftarrow$  index of  $c_d$  in  $E$ 
10       $b_k = (-1)^d$ 
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 cycle  $C_\ell$  to the kayak central path
14      $v_j \leftarrow$  node connecting the kayak cycle  $C_r$  to the kayak central path
15      $c_1, \dots, c_\ell \leftarrow$  edges of  $C_\ell$ , numbered starting from node  $v_i$ 
16     for  $d \in \{1, \dots, \ell\}$  do
17        $k \leftarrow$  index of  $c_d$  in  $E$ 
18        $b_k = (-1)^d$ 
19      $c_1, \dots, c_p \leftarrow$  edges of the central path, numbered starting from node  $v_i$ 
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      $c_1, \dots, c_r \leftarrow$  edges of  $C_r$ , numbered starting from node  $v_j$ 
24     for  $d \in \{1, \dots, r\}$  do
25        $k \leftarrow$  index of  $c_d$  in  $E$ 
26        $b_k = (-1)^{d+p+1}$ 
27    $\mathcal{B} \leftarrow \mathcal{B} \cup \{b\}$ 

```

■ **Algorithm 1** Construction of a basis of the kernel of the incidence matrix A of the graph G . This algorithm was initially introduced in [18, Section 3] to build a basis of the eigenspace associated with the eigenvalue -2 of the adjacency matrix C of a line graph L .

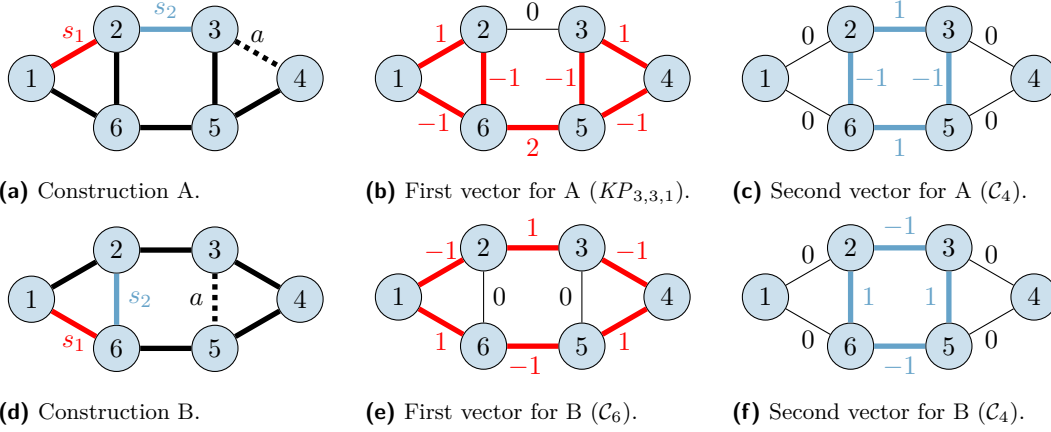
5.2 Convex polytope of positive solutions

We now consider the set $\Lambda_{\geq 0}$ of solutions of the conservation equation (2) that have non-negative components, that is,

$$\Lambda_{\geq 0} = \Lambda \cap \mathbb{R}_{\geq 0}^m = \{y \in \mathbb{R}^m : Ay = \mu, y \geq 0\}. \quad (12)$$

The set $\Lambda_{\geq 0}$ 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. The set $\Lambda_{\geq 0}$ is neither empty nor degenerated to a dimension lower than d because the matching problem (G, μ) is assumed to be stabilizable, which means that Λ contains a point with positive components (i.e. in the interior of the positive orthant). It is bounded because each $y \in \Lambda_{\geq 0}$ satisfies $0 \leq y_k \leq \min_{i \in V_k} (\mu_i)$ for each $k \in E$.

Equation (12) describes $\Lambda_{\geq 0}$ in the edge basis. As $\Lambda_{\geq 0}$ is a subset of Λ , we can also express its elements in the kernel basis introduced in Section 5.1.1. In the kernel basis, $\Lambda_{\geq 0}$



■ **Figure 13** Two possible constructions of a kernel basis for the codomino graph.

is defined by the vectors whose coordinates belong to

$$\Pi_{\geq 0} = \{\alpha \in \mathbb{R}^d : y^\circ + B\alpha \geq 0\}. \quad (13)$$

As Equations (12) and (13) basically represent the same polytope up to the change-of-basis formulas introduced in Section 5.1.1, in the remainder, we will use the same letter $\Lambda_{\geq 0}$ to describe both sets; the implicit basis we are actually using will be made clear by our choice of letters (y for the edge basis or α for the kernel basis).

5.2.1 Vertices of the convex polytope

Vertices can be informally defined as the *corners* or extreme points of a convex polytope. Definition 5.3 gives a formal definition of a vertex (as well as those of a face and a facet, which will be useful later).

► **Definition 5.3** (Vertices and facets, adapted from [28]). *A face of a convex polytope Π is an intersection of Π with a hyperplane such that the polytope Π is entirely contained in one of the two halfspaces determined by the hyperplane. A vertex of Π is a face of dimension 0, while a facet of Π is a face of dimension $d - 1$. A vertex can also be defined as follows: a vector $y \in \Pi$ is a vertex of Π if, and only if, it cannot be written as a convex combination of points in $\Pi \setminus \{y\}$.*

The vertices of $\Lambda_{\geq 0}$ are important if one wants to optimize some linear function of the matching rate, as stated by Proposition 5.4. This kind of optimization occurs in many problems related to dynamic matching systems (see for example [26] and references within).

► **Proposition 5.4.** *Let $w = (w_1, \dots, w_m) \in \mathbb{R}^m$ be a reward function associated to the edges of the compatibility graph. Consider the problem of finding the non-negative solutions of the conservation equation (2) that maximizes the reward:*

$$F = \{y \in \Lambda_{\geq 0} : w^\top y = \max_{z \in \Lambda_{\geq 0}} w^\top z\}. \quad (14)$$

F is a non-empty face of $\Lambda_{\geq 0}$. In particular, there exists a vertex $y \in \Lambda_{\geq 0}$ that maximizes the reward (i.e. $y \in F$).

Proof. This is a standard convex optimization result. The fact that $\Lambda_{\geq 0}$ is bounded guarantees the existence of a maximum r_{\max} amongst the rewards inside $\Lambda_{\geq 0}$. F , which is the intersection of the hyperplane $\{y \in \mathbb{R}^m : w^\top y = r_{\max}\}$ and $\Lambda_{\geq 0}$, is hence a non-empty face of $\Lambda_{\geq 0}$. The fact that any non-empty face contains a vertex is a consequence of the lattice structure of the faces of a polytope. \blacktriangleleft

The interest of Proposition 5.4 is the following: if for any vertex v of $\Lambda_{\geq 0}$ one can provide a stable policy Φ whose matching rate is v , or at least arbitrarily close to v , then we can optimize with a stable policy any linear reward function (or at least be arbitrarily close to the optimal). Finding such policies will be the main focus of Section 6.2.

Proposition 5.5 below gives a simple yet powerful characterization of vertices. The proof of this proposition is borrowed from [13].

► **Proposition 5.5.** *Consider a vector $y \in \Lambda_{\geq 0}$. Let $E_y = \{k \in E : y_k > 0\}$ denote the support of the vector y and $G_y = (V, E_y)$ the subgraph of G obtained by deleting the edges that are not in E_y . The following statements are equivalent:*

1. *The vector y is a vertex of $\Lambda_{\geq 0}$.*
2. *The graph G_y is injective.*

Proof. Let A_y denote the incidence matrix of G_y . By Definition 5.3, if y is not a vertex, there exist $z_1, z_2 \in \Lambda_{\geq 0} \setminus \{y\}$ and $0 < \theta < 1$ such that $y = \theta z_1 + (1 - \theta)z_2$. The components of the vectors z_1 and z_2 are non-negative, so this equality implies that their supports are included into the support of the vector y . In particular, if y' and z'_1 denote the restrictions of y and z_1 to components in E_y , respectively, we obtain $A_y y' = A_y y = \mu = A_y z'_1 = A_y z'_1$ with $y' \neq z'_1$, which means that $G_y = (V, E_y)$ is not injective.

Conversely, if G_y is not injective, there exists a non-zero vector z' in $\mathbb{R}^{|E_y|}$ such that $A_y z' = 0$. If we embed 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 y . This implies that there exists $\varepsilon > 0$ such that both $y - \varepsilon z$ and $y + \varepsilon z$ belong to $\Lambda_{\geq 0}$. The convex combination $y = \frac{1}{2}(y - \varepsilon z) + \frac{1}{2}(y + \varepsilon z)$ proves that the vector y is not a vertex of $\Lambda_{\geq 0}$. \blacktriangleleft

Examples of vertices are shown in Figures 14–16 and will be discussed in details later. For now, it is sufficient to observe that, since the subgraph restricted to the support of a vertex is injective, this subgraph is either bijective (as in Figures 14d–14h and 16c–16f) or injective-only (as in Figures 15d–15f and 16g).

Corollary 5.6 below characterizes the stabilizability of (G_y, μ) for a vertex y . This will be useful in Section 6.2, where the construction of a stable policy on G to achieve a vertex y will depend on the bijectivity of G_y .

► **Corollary 5.6.** *Consider a vertex y of $\Lambda_{\geq 0}$, and define E_y and G_y as in Proposition 5.5. We distinguish two cases, depending on the value of $p = |E_y|$:*

1. *If $p = n$, i.e. exactly d components of y are null, then the subgraph G_y is bijective and the matching problem (G_y, μ) is stabilizable.*
2. *If $p < n$, i.e. strictly more than d components of y are null, then the subgraph G_y is injective-only and the matching problem (G_y, μ) is not stabilizable.*

Proof. Let A_y denote the incidence matrix of G_y . We know from Proposition 5.5 that G_y is injective. In particular, the restriction y' of the vector y to its positive components is the only solution of the conservation equation $A_y z = \mu$, of unknown $z \in \mathbb{R}^p$. It also follows from Proposition 2.7 that $p \leq n$, and that the subgraph G_y is bijective if and only if $p = n$. We now consider the two cases separately:

1. If $p = n$, then G_y is bijective. Proposition 3.2 implies that the matching problem (G_y, μ) is stabilizable, as G is surjective and y' is a solution of the conservation equation $A_y z = \mu$ with positive components.
2. If $p < n$, then G_y is injective-only. Proposition 3.1 implies that the matching problem (G_y, μ) is not stabilizable. ◀

► **Remark 5.7.** Again consider a vertex y of $\Lambda_{\geq 0}$, and define G_y and A_y as in Proposition 5.5. According to Definitions 2.3, 2.4, and 2.6, the subgraph G_y is injective if and only if each connected component of G_y is either a tree or a unicyclic graph with an odd cycle. For each connected component of G_y that is a tree, and therefore a bipartite graph with parts V_+ and V_- , the existence of vertex y implies that

$$\sum_{i \in V_+} \mu_i = \sum_{i \in V_-} \mu_i. \quad (15)$$

Equation (15) follows by summing (1) over the nodes in V_+ on the one hand, summing (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 μ belongs to the image of A_y if and only if μ satisfies (15) for each tree connected component of G_y . Note that this condition is empty if G_y is bijective, as each connected component of G_y then contains an odd cycle.

Conversely, one can wonder which injective surgraph G' of G defines a vertex of $\Lambda_{\geq 0}$. Satisfying (15) for each tree connected component of G' only guarantees the existence of a (unique) solution $z \in \mathbb{R}^p$ to the conservation equation $A_y z = \mu$. If each component of z is positive, then we indeed obtain a corner of $\Lambda_{\geq 0}$ by embedding z in \mathbb{R}^m with zero padding; otherwise, G' does not define a vertex of $\Lambda_{\geq 0}$.

5.2.2 Bijective vertices and facets of the convex polytope

By a slight abuse of notation, we will say that a vertex $y \in \Lambda_{\geq 0}$ is bijective (resp. injective-only) to express that G_y is bijective (resp. injective-only). As stated above, the bijectivity of vertices plays an important role in the study of the matching rates that can be achieved by stable policies, which will be the matter of Section 6. We now detail the relationship between the bijectivity of vertices and the inequalities that define $\Lambda_{\geq 0}$.

Following Corollary 5.6, the bijectivity of a vertex is determined by the number of its components that are positive in edge coordinates. Recall that the d -dimensional polytope $\Lambda_{\geq 0}$ is actually characterized by the m inequalities $y_k \geq 0$ 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 5.8 below. For example, by looking more closely at the general solution obtained for the diamond graph in Figure 6, we conclude that:

- The inequality $y_{2,4} \geq 0$ is satisfied trivially by every vector $y \in \Lambda$, as we have $y_{2,4} = \beta > 0$. Therefore, this inequality does not define a facet.
- If $\mu_1 \bar{\mu}_2 < \mu_3 \bar{\mu}_4$, the inequality $y_{1,2} \geq 0$ supersedes the inequality $y_{3,4} \geq 0$, and conversely. If $\mu_1 \bar{\mu}_2 = \mu_3 \bar{\mu}_4$, these two inequalities are equivalent. In both cases, the inequalities $y_{1,2} \geq 0$ and $y_{3,4} \geq 0$ lead to a single facet.
- If $\mu_1 \bar{\mu}_4 < \mu_3 \bar{\mu}_2$, the inequality $y_{1,4} \geq 0$ supersedes the inequality $y_{2,3} \geq 0$, and conversely. If $\mu_1 \bar{\mu}_4 = \mu_3 \bar{\mu}_2$, these two inequalities are equivalent. In both cases, the inequalities $y_{1,4} \geq 0$ and $y_{2,3} \geq 0$ lead to a single facet.

All in all, the 1-dimensional convex polytope $\Lambda_{\geq 0}$ associated with the diamond graph of Example 3.7 has two facets, even if it is defined by five inequalities. Definition 5.8 below will help us relate these notions to the number of zero components of the vertices of the convex polytope $\Lambda_{\geq 0}$.

► **Definition 5.8** (Adapted from [4, 28]). *Let $k \in E$.*

1. *The inequality $y_k \geq 0$ is said to be tight if there exists a vector $y \in \Lambda_{\geq 0}$ such that $y_k = 0$, in which case we also say that this inequality is tight for the vector y .*
2. *The inequality $y_k \geq 0$ is said to be redundant if removing this inequality does not change the polytope $\Lambda_{\geq 0}$, in the sense that*

$$\Lambda_{\geq 0} = \{y \in \mathbb{R}^m : Ay = \mu, y_\ell \geq 0 \text{ for each } \ell \in E \setminus \{k\}\}.$$

Otherwise, this inequality is called irredundant.

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

Importantly, the number of positive components of a vertex y (considered in Corollary 5.6) is the number of inequalities that are not tight for this vertex. More generally, Definition 5.8 has the following intuitive interpretation. An inequality is tight if the convex polytope $\Lambda_{\geq 0}$ 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 $\Lambda_{\geq 0}$. The matching problem (G, μ) is essential if each tight inequality defines a distinct facet of the convex polytope $\Lambda_{\geq 0}$. 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 5.12, if the matching problem (G, μ) is essential and the polytope $\Lambda_{\geq 0}$ is simple, then every vertex satisfies exactly d (tight) inequalities as equalities, which means that this vertex has d zero components, and therefore $n = m - d$ positive components, so that this vertex is bijective.

All these notions are illustrated in Examples 5.9–5.11 below, which show in particular that a matching problem (G, μ) may be essential even if the polytope $\Lambda_{\geq 0}$ is not simple, and conversely. Consistently with Example 3.7 above, these examples use a kernel basis to verify effortlessly whether an inequality is tight and/or irredundant.

► **Example 5.9** (Essential matching problem). Figure 14 considers a codomino graph with the vector of arrival rates $\mu = (4, 5, 3, 2, 3, 5)$. A particular solution of the conservation equation (2) is $y^\circ = (2, 2, 1, 2, 1, 1, 1, 1) \in \mathbb{R}^8$, 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 13. The generic solution of the conservation equation (2) is shown in Figure 14b.

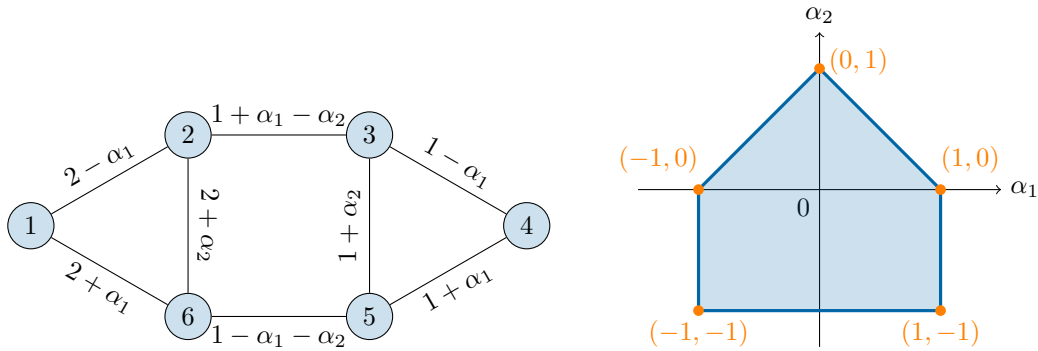
The inequalities are listed in Figure 14a. The 2-dimensional polytope $\Lambda_{\geq 0}$, shown in Figure 14c in kernel basis, is characterized by five tight inequalities which are also irredundant:

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

The matching problem (G, μ) is essential. In kernel basis, the vertices of the convex polytope $\Lambda_{\geq 0}$ are $(0, 1)$, $(-1, 0)$, $(1, 0)$, $(-1, -1)$, and $(1, -1)$, and we can verify on Figure 14c that each vertex belongs to exactly 2 facets. Therefore, like all 2-dimensional polytopes, $\Lambda_{\geq 0}$ is simple. All in all, each vertex of $\Lambda_{\geq 0}$ has 2 zero components and 6 positive components in edge coordinates, so that this vertex is bijective. These vertices are represented in edge basis in Figures 14d–14h.

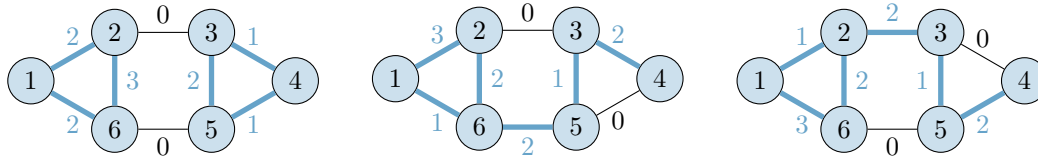
Edge basis	Kernel basis	Tight?	Irredundant?
$y_{1,2} \geq 0$	$\alpha_1 \leq 2$	✗	✗
$y_{1,6} \geq 0$	$\alpha_1 \geq -2$	✗	✗
$y_{2,3} \geq 0$	$\alpha_1 - \alpha_2 \geq -1$	✓	✓
$y_{2,6} \geq 0$	$\alpha_2 \geq -2$	✗	✗
$y_{3,4} \geq 0$	$\alpha_1 \leq 1$	✓	✓
$y_{3,5} \geq 0$	$\alpha_2 \geq -1$	✓	✓
$y_{4,5}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

(a) Inequalities.



(b) Generic solution of the conservation equation (2).

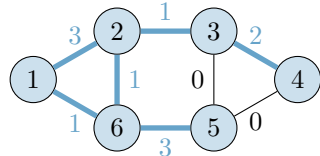
(c) Polytope $\Lambda_{\geq 0}$ in kernel basis.



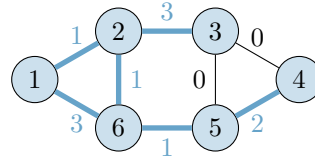
(d) Edge coordinates of (0, 1).

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

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



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



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

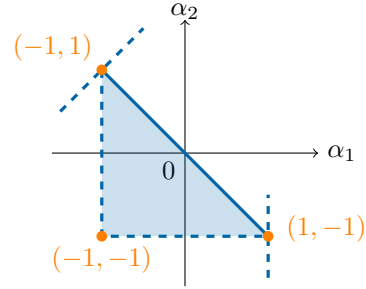
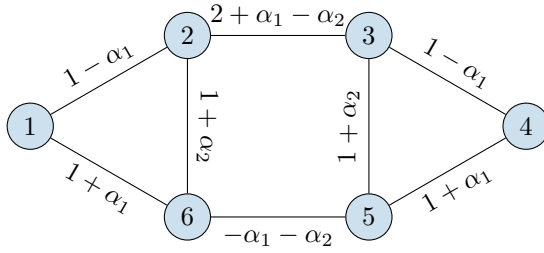
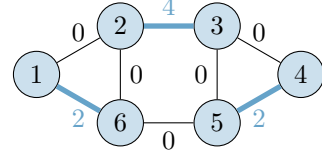
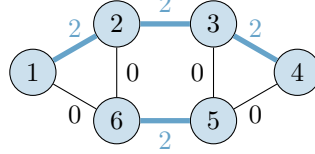
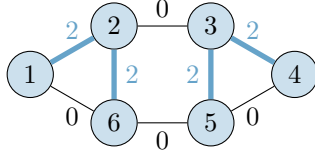
■ **Figure 14** An essential matching problem (G, μ) with a simple polytope $\Lambda_{\geq 0}$. The vector of arrival rates is $\mu = (4, 5, 3, 2, 3, 5) \in \mathbb{R}^6$, a particular solution of the conservation equation (2) is $y^\circ = (2, 2, 1, 2, 1, 1, 1, 1) \in \mathbb{R}^8$, and the chosen base vectors for $\ker(A)$ are $b_1 = (-1, 1, 1, 0, -1, 0, 1, -1)$ and $b_2 = (0, 0, -1, 1, 0, 1, 0, -1)$.

► **Example 5.10 (Non-essential matching problem).** Figure 15 shows the same codomino graph as in Example 5.9, with the same basis of $\ker(A)$, but with the vector of arrival rates $\mu = (2, 4, 4, 2, 2, 2)$. A particular solution of the conservation equation (2) is $y^\circ = (1, 1, 2, 1, 1, 1, 1, 0)$, and the general solution is shown in Figure 15b.

The inequalities are listed in Figure 15a. The 2-dimensional convex polytope $\Lambda_{\geq 0}$ is shown in kernel basis in Figure 15c. All inequalities are tight, but only one is irredundant, so we conclude that the matching problem (G, μ) is not essential, even if the polytope $\Lambda_{\geq 0}$

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

(a) Inequalities

(b) Generic solution of the conservation equation (2). (c) Polytope $\Lambda_{\geq 0}$ 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 $(1, -1)$.

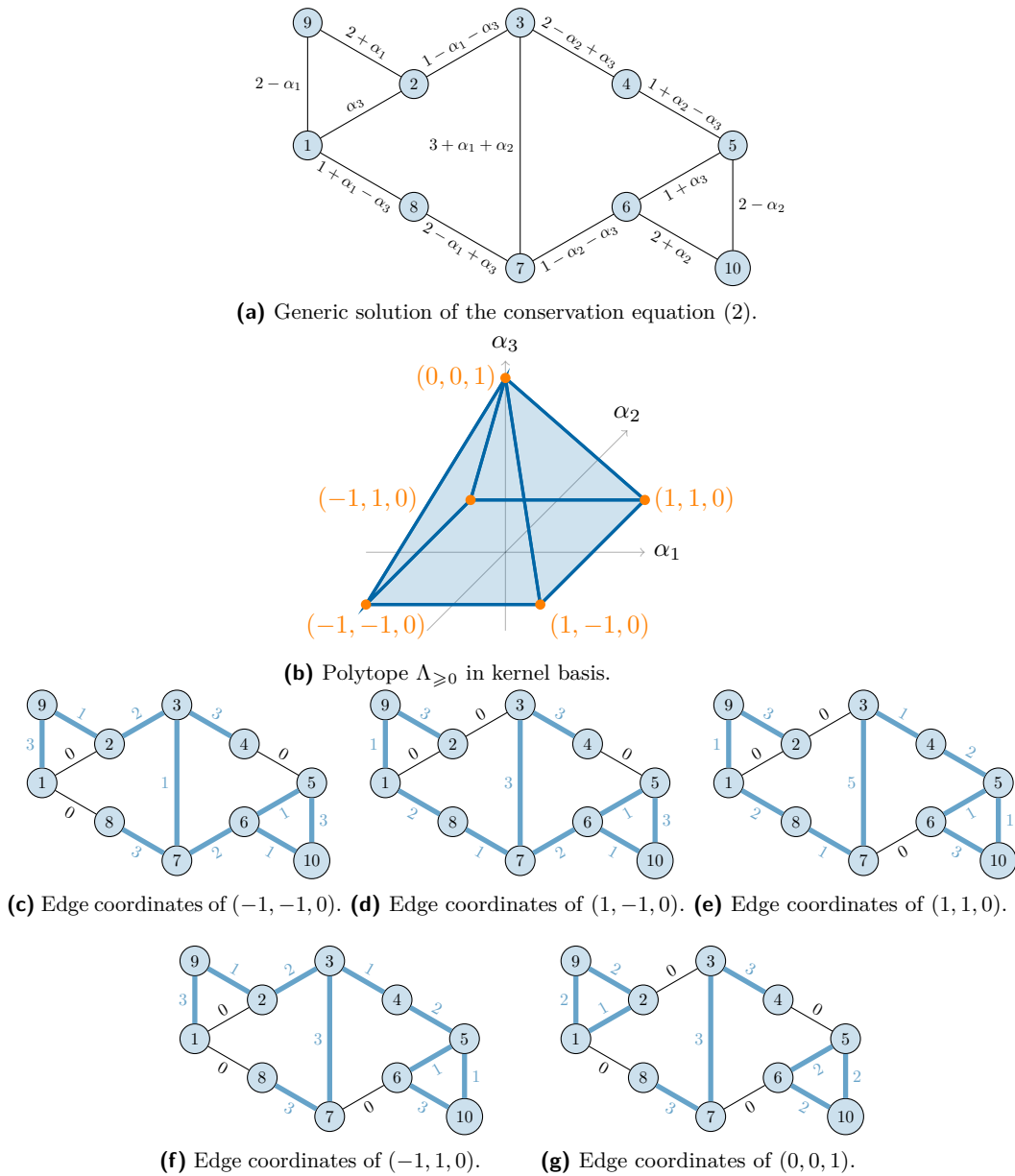
■ **Figure 15** Non-essential matching problem (G, μ) with a simple polytope $\Lambda_{\geq 0}$. The vector of arrival rates is $\mu = (2, 4, 4, 2, 2, 2) \in \mathbb{R}^6$, a particular solution of the conservation equation (2) is $y^\circ = (1, 1, 2, 1, 1, 1, 0) \in \mathbb{R}^8$, and the chosen base vectors for $\ker(A)$ are $b_1 = (-1, 1, 1, 0, -1, 0, 1, -1)$ and $b_2 = (0, 0, -1, 1, 0, 1, 0, -1)$.

is still simple. Correspondingly, even if each vertex belongs to exactly two facets, they all have more than two zero components, 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 15f). This vertex has five zero components in edge coordinates (and only three positive components) even if this vertex belongs to only two facets.

► **Example 5.11 (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 16a. The arrival rate is $\mu = (3, 3, 6, 3, 4, 4, 6, 3, 4, 4) \in \mathbb{R}^{10}$. The particular solution and kernel basis are shown on the edges. The set $\Lambda_{\geq 0}$, shown in Figure 16b 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),



■ **Figure 16** Essential matching problem (G, μ) with a non-simple polytope $\Lambda_{\geq 0}$. The arrival rate is $\mu = (3, 3, 6, 3, 4, 4, 6, 3, 4, 4) \in \mathbb{R}^{10}$. A particular solution and the chosen base vectors for $\ker(A)$ are implicitly shown on the edges of Figure 16a.

so we conclude that the matching problem (G, μ) 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 16c–16g. The polytope $\Lambda_{\geq 0}$ 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 16g that this vertex has four zero components and only nine positive components in edge basis; the subgraph defined by the support of this vertex is injective-only.

In light of the examples above, we can give the following characterization of the bijective

vertices of $\Lambda_{\geq 0}$.

► **Proposition 5.12.** *Let y be a vertex of $\Lambda_{\geq 0}$. The following statements are equivalent:*

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

Proof. We first remark that the number of null components of an element $y \in \Lambda_{\geq 0}$ are by definition the number of inequalities that are tight for y . It is in particular at least the number of facets that intersect y , with equality if, and only if, none of the inequalities tight for y is redundant.

If y is a vertex, y belongs to at least d facets. It is bijective if, and only if, d of its components are null, which in view of the remark above is equivalent to say that y belongs to exactly d facets of $\Lambda_{\geq 0}$ and none of the inequalities tight for y is redundant.

As for the last statement, it follows directly from Definition 5.8. ◀

6 Matching rates in surjective-only graphs

Section 5 describes $\Lambda_{\geq 0}$, the set of non-negative solutions of the conservation equation (2). In this section, we investigate which of these solutions may, or may not, be achieved by a (stable) policy. There are many cases where this question matters:

- As stated in Proposition 5.4, if an edge-dependent reward is earned each time a match is performed, the long-term reward is maximized by favoring edges with the maximal reward and, more precisely, by achieving a vector on a face of the polytope $\Lambda_{\geq 0}$.
- In *chained matching*, the matches performed in the stable matching model (G, μ, Φ) form the (non-Poisson) arrival process of another matching model, with a compatibility graph $G' = (E, E')$ (the edges of G are the nodes of G'). In particular, the matching rate vector λ of the first model is the vector of arrival rates in the second model. Being able to control the vector λ can therefore help stabilize the second model.

Unless otherwise stated, we consider a stabilizable matching problem (G, μ) with a surjective-only compatibility graph G , so that the set $\Lambda_{\geq 0}$ of non-negative solutions of the conservation equation (2) is non-trivial.

Given a policy Φ that stabilizes the matching problem (G, μ) , we let $\lambda(\Phi)$ denote the vector of matching rates in the matching model (G, μ, Φ) . We consider the set of matching rates achieved by stable policies (resp. of stable greedy policies), defined as follows:

$$\Lambda_{\mathcal{P}} = \{\lambda(\Phi) : \Phi \text{ is a stable policy of the matching model } (G, \mu)\},$$

$$\Lambda_{\mathcal{G}} = \{\lambda(\Phi) : \Phi \text{ is a stable greedy policy of the matching model } (G, \mu)\}.$$

Section 6.1 focuses on the set $\Lambda_{\mathcal{G}}$. We show in particular that $\Lambda_{\mathcal{G}} \subseteq \Lambda_{> 0}$, where $\Lambda_{> 0}$ is the (non-empty) set of solutions of the conservation equation (2) with positive components. We also show that, in many cases, the inclusion is strict. Informally, this result shows that greedy policies are not very good at navigating inside $\Lambda_{\geq 0}$. In contrast, we show in Section 6.2 that filtering policies can be used to reach some faces of $\Lambda_{\geq 0}$. In particular, if all vertices of $\Lambda_{\geq 0}$ are bijective, we have $\Lambda_{\mathcal{P}} = \Lambda_{\geq 0}$. Finally, in Section 6.2, we also use semi-filtering policies to show that $\Lambda_{> 0} \subseteq \Lambda_{\mathcal{P}}$.

The following result will be useful throughout this section.

► **Proposition 6.1.** *The sets $\Lambda_{\mathcal{P}}$ and $\Lambda_{\mathcal{G}}$ are convex.*

Proof. We first show the convexity of $\Lambda_{\mathcal{P}}$. Let $\Phi_1 = (\mathcal{C}_1, \pi_1)$ and $\Phi_2 = (\mathcal{C}_2, \pi_2)$ be two stable policies and $0 < \beta < 1$ be a linear coefficient. To build a stable policy Φ_β such that $\lambda(\Phi_\beta) = \beta\lambda(\Phi_1) + (1 - \beta)\lambda(\Phi_2)$, we combine Φ_1 and Φ_2 in proper proportions.

Let $\emptyset_1 \in \mathcal{C}_1$ (resp. $\emptyset_2 \in \mathcal{C}_2$) denote the empty state of Φ_1 (resp. Φ_2), as defined in Section 2.1. Let T_1 (resp. T_2) be the mean return time to the empty state \emptyset_1 (resp. \emptyset_2) in the matching model (G, μ, Φ_1) (resp. (G, μ, Φ_2)). We construct the policy Φ_β on the state space $\mathcal{C}_1 \times \mathcal{C}_2$ with empty state $(\emptyset_1, \emptyset_2)$. On the arrival of an item of class $i \in V$:

- If the current state is (c_1, \emptyset_2) with $c_1 \in \mathcal{C}_1 \setminus \{\emptyset_1\}$, then we apply the policy Φ_1 , that is, we choose a new state $c'_1 \in \mathcal{C}_1$ according to $\pi_1(\cdot|c_1, i)$ and move to the state (c'_1, \emptyset_2) .
- If the current state is (\emptyset_1, c_2) with $c_2 \in \mathcal{C}_2 \setminus \{\emptyset_2\}$, then we apply the policy Φ_2 , that is, we choose a new state $c'_2 \in \mathcal{C}_2$ according to $\pi_2(\cdot|c_2, i)$ and move to the state (\emptyset_1, c'_2) .
- If the current state is $(\emptyset_1, \emptyset_2)$, then we apply Φ_1 with probability $p_1 = \frac{\beta T_2}{\beta T_2 + (1 - \beta) T_1}$ and Φ_2 otherwise. In other words, the next state is (c_1, \emptyset_2) with probability p_1 and (\emptyset_1, c_2) with probability $1 - p_1$, where c_1 is chosen according to $\pi_1(\cdot|\emptyset_1, i)$ and c_2 is chosen according to $\pi_2(\cdot|\emptyset_2, i)$.

In essence, every time the system is in state $(\emptyset_1, \emptyset_2)$, the policy Φ_β selects either the policy Φ_1 or the policy Φ_2 and follows this policy until the next passage in the empty state $(\emptyset_1, \emptyset_2)$. Importantly, if we only consider the time intervals when the policy Φ_1 (resp. Φ_2) is applied, the evolution is exactly the same as under policy Φ_1 (resp. Φ_2). In particular, the long-term matching rate observed on these intervals is $\lambda(\Phi_1)$ (resp. $\lambda(\Phi_2)$). This guarantees that the evolution of the state (c_1, c_2) under the policy Φ_β defines an irreducible continuous-time Markov chain with state space $\mathcal{C}_1 \times \mathcal{C}_2$, and that the matching model (G, μ, Φ_β) is stable.

By construction, the long-run fraction of time that Φ_1 (resp. Φ_2) is chosen is proportional to βT_2 (resp. $(1 - \beta) T_1$). This implies that the fractions of decisions taken according to Φ_1 and Φ_2 are proportional to β and $(1 - \beta)$, respectively. As the matching rate when Φ_β behaves like Φ_1 (resp. Φ_2) is $\lambda(\Phi_1)$ (resp. $\lambda(\Phi_2)$), we conclude that $\lambda(\Phi_\beta) = \beta\lambda(\Phi_1) + (1 - \beta)\lambda(\Phi_2)$.

This proves that $\Lambda_{\mathcal{P}}$ is convex. To prove the convexity of $\Lambda_{\mathcal{G}}$, it suffices to observe that, if the policies Φ_1 and Φ_2 are greedy, so is the policy Φ_β . ◀

6.1 Greedy policies

Greedy policies are appealing candidates when one wants to choose a policy for a matching problem (G, μ) : they are usually simple to implement and some of them, like *match-the-longest* or *first-come-first-matched*, are maximally stable on stable graphs.

However, this section will show that they are not very flexible in terms of achievable matching rates. Although we do not provide a universal tight description of $\Lambda_{\mathcal{G}}$ (which we believe to be a difficult task), we do provide several examples that indicate that $\Lambda_{\mathcal{G}}$ is generally (but not always) reduced relative to $\Lambda_{\geq 0}$.

For a start, as stated by Proposition 6.2, greedy policies cannot reach the boundaries of the convex polytope $\Lambda_{\geq 0}$.

► **Proposition 6.2.** *If the compatibility graph G is surjective-only and the matching problem (G, μ) is stabilizable, then the set $\Lambda_{\mathcal{G}}$ is non-empty and $\Lambda_{\mathcal{G}} \subseteq \Lambda_{> 0}$.*

Proof. The set $\Lambda_{\mathcal{G}}$ is non-empty because the greedy policies *match-the-longest* [23] and *first-come-first-matched* [25] are known to be stable.

We now prove that $\Lambda_{\mathcal{G}} \subseteq \Lambda_{> 0}$. Consider a stable greedy policy Φ and let $\lambda = (\lambda_k : k \in E)$ denote the corresponding vector of matching rates. 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 , a class- i item arrives, and a class- j

item arrives. Let p_\emptyset denote the stationary probability that the matching model (G, μ, Φ) is in the empty state \emptyset . We now that $p_\emptyset > 0$ because this matching model is stable, and the previous remark implies that $\lambda_k \geq p_\emptyset \mu_i \mu_j / \sum_{\ell \in V} \mu_\ell > 0$. Since this is true for each edge $e_k \in E$, we conclude that $\lambda \in \Lambda_{>0}$. ◀

Proposition 6.2 has a consequence on the capacity of greedy policies to maximize some reward function.

► **Corollary 6.3.** *Consider the problem of maximizing an edge-dependent reward defined by a vector w as stated in Proposition 5.4. Let $r_{\max} = \max_{y \in \Lambda_{\geq 0}} w^\top y$ be the optimal reward. One of the two exclusive statements below is true:*

1. All vectors of $\Lambda_{\geq 0}$ are optimal, i.e. $\forall y \in \Lambda_{\geq 0}, w^\top y = r_{\max}$.
2. All stable greedy policies are suboptimal, i.e. $\forall \lambda \in \Lambda_G, w^\top \lambda < r_{\max}$.

Proof. We know from Proposition 5.4 that the set of $y \in \Lambda_{\geq 0}$ that maximize the reward is a non-empty face F of $\Lambda_{\geq 0}$. If $F = \Lambda_{\geq 0}$, we are obviously in the first case (all $y \in \Lambda_{\geq 0}$ are optimal). One can verify that this corresponds to situations where w is orthogonal to $\ker(A)$ (for example if all components of w are identical). Otherwise, because of the lattice structure of the faces of a polytope, F belongs to a facet of $\Lambda_{\geq 0}$, which means that there is at least one edge that is null for all vectors of F . In particular, as all components of the matching rates of a stable greedy policy are positive, no greedy policy is optimal. ◀

To further support the intuition that greedy policies are not very good at navigating the set $\Lambda_{\geq 0}$, Sections 6.1.1 and 6.1.2 study two examples where Λ_G is a strict subset of $\Lambda_{>0}$. However, this is not a universal result, as Section 6.1.3 gives a (well-chosen) counter-example where $\Lambda_G = \Lambda_{>0}$.

Section 6.2 will show how the introduction of (semi-)filtering policies drastically improves things and allows us to reach $\Lambda_{>0}$ and some of the faces of $\Lambda_{\geq 0}$.

6.1.1 Complete graph

Proposition 6.4 below essentially states that greedy policies do not allow any degree of freedom in the case of complete graphs, in the sense that all stable greedy policies lead to the same dynamics and the same vector of matching rates. In contrast, the polytope $\Lambda_{\geq 0}$ has dimension $m - n = \frac{n(n-3)}{2}$, where n is the number of classes and $m = \binom{n}{2} = \frac{n(n-1)}{2}$ is the number of possible matches.

► **Proposition 6.4.** *Consider the complete graph K_n with $n \geq 3$.*

1. The matching problem (K_n, μ) is stabilizable if and only if $\mu_i < \frac{\bar{\mu}}{2}$ for each $i \in V$, where $\bar{\mu} = \sum_{i \in V} \mu_i$ is the total arrival rate.
2. All greedy policies are equivalent in the sense that the evolution of $h(c)$ (the vector counting the number of unmatched items of each class) defines a continuous-time Markov chain with the same transition diagram and state space $\mathcal{N} \subseteq \mathbb{N}$, given by

$$\mathcal{N} = \{0\} \cup \left(\bigcup_{i \in V} \{x\delta_i, x \in \mathbb{N}^*\} \right),$$

where, for each $i \in V$, δ_i is the n -dimensional vector with one in component i and zero elsewhere, and \mathbb{N}^* is the set of positive integers. We let Φ_G denote the greedy policy with minimal state space $\mathcal{C} = \mathcal{N}$.

3. If the matching problem (K_n, μ) is stabilizable, then the matching model $(K_n, \mu, \Phi_{\mathcal{G}})$ is stable, and the associated matching rates are given by

$$\lambda_k = \mu_i p_j + \mu_j p_i, \quad e_k = \{i, j\} \in E, \quad (16)$$

where p_i is the stationary probability that the queue i is non-empty, given by

$$p_i = \frac{\frac{\mu_i}{\bar{\mu} - 2\mu_i}}{1 + \sum_{\ell=1}^n \frac{\mu_\ell}{\bar{\mu} - 2\mu_\ell}}, \quad i \in V. \quad (17)$$

In particular, we have $\Lambda_{\mathcal{G}} \subsetneq \Lambda_{>0}$ for each $n \geq 4$.

Proof. Statement 1 is a direct consequence of Proposition 3.2.

We now prove statement 2. Since the compatibility graph is complete, two items can be matched if and only if they belong to different classes. As a result, a greedy policy can only reach states in which at most one queue is non-empty. Moreover, under a greedy policy, if a class- i item arrives and finds a non-empty queue $j \neq i$, the incoming class- i item is matched with a class- j item, and the queue size of j is decreased by 1. Otherwise, the queue size of i is increased by 1. This is sufficient to prove statement 2.

The proof of statement 3 builds upon the previous observation. The natural greedy policy $\Phi_{\mathcal{G}}$ is an instance of the match-the-longest policy, which is known to be maximally stable [23]. Equation (16) follows from the PASTA property and the observation that, for each $i, j \in V$ with $i \neq j$, a match between classes i and j happens in one of the following two cases:

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

To prove (17), it suffices to observe that the Markov chain defined in statement 1 has a very specific structure: for each $i \in V$, the restriction of the transition diagram of this Markov chain to the states where all queues but i are empty defines a birth-and-death process with birth rate μ_i (a class- i item arrives) and death rate $\bar{\mu} - \mu_i$ (an item of a class $j \neq i$ arrives and is matched with a class- i item). If p_{\emptyset} denotes the stationary probability of the empty state, it follows that

$$p_i = p_{\emptyset} \sum_{c \geq 1} \left(\frac{\mu_i}{\bar{\mu} - \mu_i} \right)^c = p_{\emptyset} \frac{\mu_i}{\bar{\mu} - 2\mu_i}, \quad i \in V.$$

The value of p_{\emptyset} follows from the normalizing equation $p_{\emptyset} + \sum_{i \in V} p_i = 1$. ◀

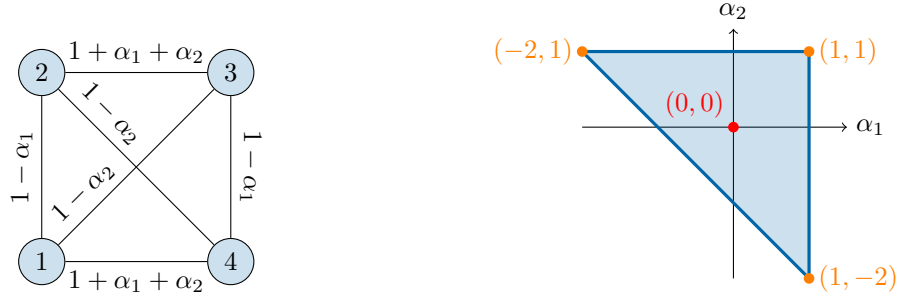
Figure 17 illustrates this result on a complete graph K_4 in which all arrival rates are equal to 3. In kernel basis, the polytope $\Lambda_{\geq 0}$ is defined by the inequalities $\alpha_1 \leq 1$, $\alpha_2 \leq 1$, and $\alpha_1 + \alpha_2 \geq -1$, i.e. it is a triangle of vertices $(-2, 1)$, $(1, -2)$, and $(1, 1)$. Yet, only the solution $\alpha = (0, 0)$ can be achieved by a greedy policy.

6.1.2 Diamond graph

Proposition 6.5 and Corollary 6.6 below focus on the diamond graph. We show that the set $\Lambda_{\mathcal{G}}$ of vectors reachable by a greedy policy, while not reduced to a single vector like for a complete graph, is still a strict subset of the set $\Lambda_{>0}$.

► **Proposition 6.5.** *Consider the matching problem (G, μ) of Example 3.7, with the normalization $\mu_1 + \mu_3 = \bar{\mu}_2 + \bar{\mu}_4 = \frac{1}{2}$. We define the following values:*

$$\begin{aligned} q_{1,3} &= \frac{1}{4\beta}, & q_1 &= \frac{\mu_1}{\mu_3 + 2\beta}, & q_3 &= \frac{\mu_3}{\mu_1 + 2\beta}, \\ q_2 &= \frac{\bar{\mu}_2 + \beta}{2\bar{\mu}_4}, & q_4 &= \frac{\bar{\mu}_4 + \beta}{2\bar{\mu}_2}, & p_{\emptyset} &= \frac{1}{1 + q_{1,3} + q_2 + q_4}. \end{aligned}$$



(a) Generic solution of the conservation equation (2). (b) $\Lambda_{\geq 0}$ in kernel basis. $\Lambda_{\mathcal{G}}$ is the red point.

■ **Figure 17** Matching problem (K_4, μ) with $\mu = (3, 3, 3, 3)$.

The matching problem (G, μ) is stable for all greedy policies. Moreover, the matching rates under any greedy policy verify:

$$\begin{aligned} \lambda_{2,4} = \beta, \quad \lambda_{1,2} > p_{\emptyset} \mu_1 q_2 + p_{\emptyset} \mu_2 q_1, \quad \lambda_{1,4} > p_{\emptyset} \mu_1 q_4 + p_{\emptyset} \mu_4 q_1, \\ \lambda_{3,2} > p_{\emptyset} \mu_3 q_2 + p_{\emptyset} \mu_2 q_3, \quad \lambda_{3,4} > p_{\emptyset} \mu_3 q_4 + p_{\emptyset} \mu_4 q_3. \end{aligned} \quad (18)$$

In particular, if we let β grow to infinity without changing the values of $\mu_1, \bar{\mu}_2, \mu_3,$ and $\bar{\mu}_4,$ we have

$$\begin{aligned} \lim_{\beta \rightarrow +\infty} \lambda_{1,2} = 2\mu_1 \bar{\mu}_2, \quad \lim_{\beta \rightarrow +\infty} \lambda_{1,4} = 2\mu_1 \bar{\mu}_4, \\ \lim_{\beta \rightarrow +\infty} \lambda_{3,2} = 2\mu_3 \bar{\mu}_2, \quad \lim_{\beta \rightarrow +\infty} \lambda_{3,4} = 2\mu_3 \bar{\mu}_4. \end{aligned} \quad (19)$$

Proof. While for the complete graph, a greedy policy has no decision to take, here a greedy policy makes decisions on two occasions:

- a class-2 item arrives while queues 1 and 3 are non-empty;
- a class-4 item arrives while queues 1 and 3 are non-empty.

In all other situations, the decision of the policy is automatic. In particular, the evolution of the queue size of 2, the queue size of 4, and the sum of queue sizes of 1 and 3 (i.e. the number of unmatched items that belong to class 1 or 3, without distinction) are independent of the greedy policy considered. We will leverage this to prove (18). Let p_{\emptyset} denote the stationary probability of the empty state. We can partition the reachable states of the system into three $M/M/1$ queues sharing the same empty state:

- The states where all queues but 2 are empty behave like a $M/M/1$ queue with load $\rho_2 = \frac{\bar{\mu}_2 + \beta}{1 + \beta - \bar{\mu}_2}$. In particular, the probability that queue 2 is non-empty is $p_{\emptyset} \frac{\rho_2}{1 - \rho_2} = p_{\emptyset} q_2$.
- The states where all queues but 4 are empty behave like a $M/M/1$ queue with load $\rho_4 = \frac{\bar{\mu}_4 + \beta}{1 + \beta - \bar{\mu}_4}$. In particular, the probability that queue 4 is not empty is $p_{\emptyset} \frac{\rho_4}{1 - \rho_4} = p_{\emptyset} q_4$.
- The states where queues 2 and 4 are empty, partitioned by the sum of the queue sizes of 1 and 3, behave like a $M/M/1$ queue with load $\rho_{1,3} = \frac{1}{1 + 4\beta}$. In particular, the probability that the queue 1 or the queue 3 is non-empty is $p_{\emptyset} \frac{\rho_{1,3}}{1 - \rho_{1,3}} = p_{\emptyset} q_{1,3}$.

It follows that the probability of the empty state is $p_{\emptyset} = \frac{1}{1 + q_2 + q_4 + q_{1,3}}$.

Now consider the matches between classes 1 and 2. There are two cases where these matches occur:

- A class-1 item arrives while 2 is not empty. This happens with intensity $\mu_1 p_{\emptyset} q_2$.
- A class-2 item arrives and is matched with a class-1 item. The exact intensity depends on the policy but a strict lower bound is $p_{\emptyset} q_1 \mu_2$ ($p_{\emptyset} q_1$ is a strict lower bound of the

probability that 1 is the only non-empty class; it comes from discarding the cases where 1 is not empty after one remaining item 3 has been removed; the remaining cases behave like a $M/M/1$ queue with load $\frac{\mu_1}{1/2+2\beta}$.

In other words, $\lambda_{1,2} > p_{\emptyset}\mu_1q_2 + p_{\emptyset}\mu_2q_1$. The lower bounds for $\lambda_{1,4}$, $\lambda_{3,2}$, and $\lambda_{3,4}$ are derived the same way. The equality $\lambda_{2,4} = \beta$ has already been observed in Example 3.7. It comes from the fact that the edge $\{2, 4\}$ does not belong to $\ker(A)$ and has the same matching rate in all stable policies.

Lastly, we need to prove Equation (19). First observe that $\lim_{\beta \rightarrow +\infty} p_{\emptyset}\mu_1q_2 = 2\mu_1\bar{\mu}_2$ and $\lim_{\beta \rightarrow +\infty} p_{\emptyset}\mu_1q_4 = 2\mu_1\bar{\mu}_4$, while $\lim_{\beta \rightarrow +\infty} p_{\emptyset}\mu_2q_1 = \lim_{\beta \rightarrow +\infty} p_{\emptyset}\mu_4q_1 = 0$. Hence if we add the two first inequalities in Equation (18) and take the limit we get

$$\lim_{\beta \rightarrow +\infty} \lambda_{1,2} + \lambda_{1,4} \geq 2\mu_1\bar{\mu}_2 + 2\mu_1\bar{\mu}_4 = \mu_1.$$

By conservation principle we always have $\lambda_{1,2} + \lambda_{1,4} = \mu_1$, which implies the two first lines in (19). The two last lines are obtained the same way by considering $\lambda_{3,2}$ and $\lambda_{3,4}$. ◀

Propositions 6.4 and 6.5 show that there are matching problems (G, μ) where no greedy policy can approach a border of $\Lambda_{\geq 0}$ beyond a certain point. This suggests in particular that greedy policies are not always adapted to optimize cost functions of the matching rates.

One interpretation of Equation (19) is as follows: as the matching rate β between classes 2 and 4 grows, all other things being equal, the probability that queue 1 or queue 3 is non-empty goes to 0 as the only dominant cases are: queue 2 is non-empty ($p_2 \approx 2\bar{\mu}_2$); queue 4 is not empty ($p_4 \approx 2\bar{\mu}_4$). This means that the probability that one particular policy has an actual decision to make goes to 0: all greedy policies tend to behave the same way.

The following characterizes the vertices of $\Lambda_{\mathcal{G}}$.

► **Corollary 6.6.** *Consider the matching problem (G, μ) of Example 3.7, again with the normalization $\mu_1 + \mu_3 = \bar{\mu}_2 + \bar{\mu}_4 = \frac{1}{2}$. In kernel basis, we have*

$$\Lambda_{\geq 0} = [-2 \min(\mu_1\bar{\mu}_2, \mu_3\bar{\mu}_4), 2 \min(\bar{\mu}_2\mu_3, \mu_1\bar{\mu}_4)].$$

Let Φ_+ and Φ_- denote the greedy policies that give priority to edges $\{1, 2\}$ and $\{3, 4\}$ and to edges $\{2, 3\}$ and $\{1, 4\}$, respectively. If we let α_+ (resp. α_-) denote, in kernel basis, the matching rates under policy Φ_+ (resp. Φ_-), we have:

$$-2 \min(\mu_1\bar{\mu}_2, \mu_3\bar{\mu}_4) < \alpha_- \leq \alpha_+ < 2 \min(\bar{\mu}_2\mu_3, \mu_1\bar{\mu}_4),$$

$$\Lambda_{\mathcal{G}} = [\alpha_-, \alpha_+] \subsetneq \Lambda_{> 0} \text{ (in kernel basis),}$$

$$\lim_{\beta \rightarrow +\infty} \alpha_- = \lim_{\beta \rightarrow +\infty} \alpha_+ = 0$$

Proof. We only need to prove that $\Lambda_{\mathcal{G}} = [\alpha_-, \alpha_+]$, as the rest of the corollary is a rephrasing of Proposition 6.5 in kernel basis.

We first prove that the policy Φ_+ maximizes the matching rates along edges $\{1, 2\}$ and $\{3, 4\}$ among all stable greedy policies. We consider a greedy policy Φ and couple the matching models (G, μ, Φ) and (G, μ, Φ_+) to have the same arrival processes. The key is to notice that the queue size of 2, the queue size of 4, and the sum of the queue sizes of 1 and 3 have the exact same evolution (cf proof of Proposition 6.5) in both policies. To monitor the different behaviors, we can just observe the difference d between the queue size of 1 in Φ_+ and the queue size of 1 in Φ . Upon arrival of a class- i item, consider the different possibilities of evolution for d :

- d has the same value before and after the arrival: that means that Φ_+ and Φ have taken the same matching decision.
- Otherwise, one of the policy matched an item from class 1 while the other matched an item from class 3. In particular the arriving item is of class 2 or 4. We then distinguish four subcases:
 1. d was non-negative and increased. That means that Φ_+ matched an item from class 3 while Φ matched an item from class 1. The queue 1 in Φ_+ was non-empty (because of the sign of d), so the arrival item was of class 4 (otherwise Φ^+ would have matched an item from class 1). We can conclude that Φ_+ matched a pair $\{3, 4\}$ while Φ matched a pair $\{1, 4\}$.
 2. d was positive and decreased.
 3. d was non-positive and decreased. That means that Φ_+ matched an item from class 1 while Φ matched an item from class 3. The queue 3 in Φ_+ was non-empty (because of the sign of d), so the arrival item was of class 2 (otherwise Φ^+ would have matched an item from class 3). We can conclude that Φ_+ matched a pair $\{1, 2\}$ while Φ matched a pair $\{3, 2\}$.
 4. d was negative and increased.

Assuming that we started from the empty state, we can conclude that the number of matches along edges $\{1, 2\}$ and $\{3, 4\}$ is always greater or equal in Φ_+ than in Φ : the only situations where Φ selects $\{1, 2\}$ or $\{3, 4\}$ and Φ_+ does not necessarily correspond to case 2 or case 4 above. These cases can be associated in a one-to-one basis with a past case 1 or a past case 3, where Φ_+ selects $\{1, 2\}$ or $\{3, 4\}$ and Φ does not. This shows that, among all greedy policy, Φ_+ maximizes α . Using a similar reasoning with Φ_- , it follows that in kernel basis the vector of matching rates associated with any stable greedy policy belongs to the interval $[\alpha_-, \alpha_+]$. The conclusion follows by applying Proposition 6.1. ◀

To illustrate Corollary 6.6, we consider in Figure 18 a symmetric example where $\mu_1 = \bar{\mu}_2 = \mu_3 = \bar{\mu}_4 = \frac{1}{4}$. The figure compares $\Lambda_{\geq 0}$ and $\Lambda_{\mathcal{G}}$ with the bounds (18) and limits (19) (converted in kernel basis). We observe, as announced by Corollary 6.6, that $\Lambda_{\mathcal{G}}$ converges to 0 when β grows. We also notice that the bounds from Equation (18) are not tight when β is small, but that Φ_+ and Φ_- seem to be able to be as close as we want to the borders of $\Lambda_{\geq 0}$ when β is small enough (this does not contradict the fact that for any given β , $\Lambda_{\mathcal{G}}$ is strictly inside $\Lambda_{\geq 0}$). The reason for the gap is that to get Equation (18), we discarded some cases where both queues 1 and 3 are non-empty (cf the second case of the proof for $\lambda_{1,2}$): these situations become not negligible when β is small.

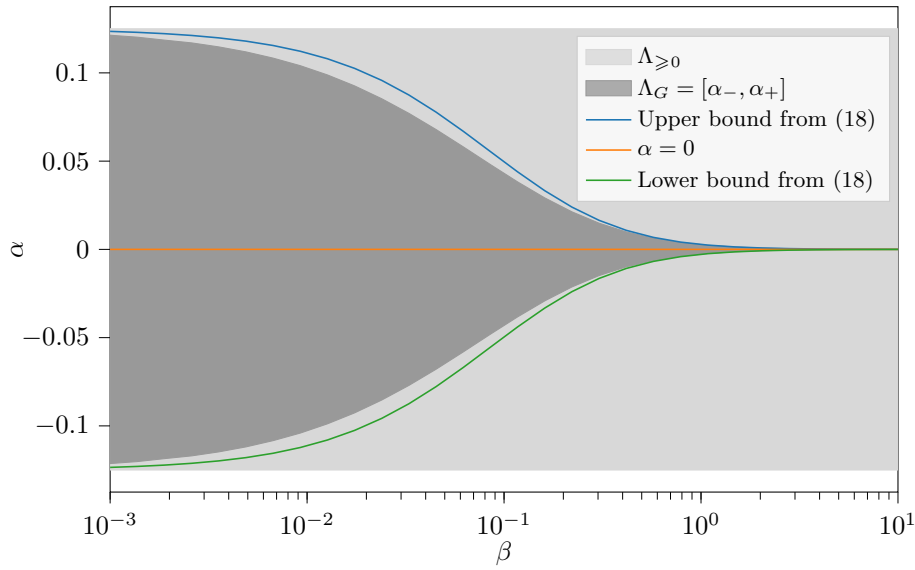
6.1.3 Fish graph

In view of the examples above, one can start to understand how the constraint of greediness makes difficult to adjust the matching rate of a greedy policy. In most stabilizable surjective-only matching problems, one should expect $\Lambda_{\mathcal{G}} \subsetneq \Lambda_{>0}$. However, as shown by Proposition 6.7 and Corollary 6.8, there is at least one counter-example of a stabilizable surjective-only matching model where $\Lambda_{\mathcal{G}} = \Lambda_{>0}$.

► **Proposition 6.7.** *Consider the matching problem (G, μ) depicted Figure 19. (G, μ) is stabilizable and $\Lambda_{\geq 0} = [0, 1]$ (in kernel basis).*

For $k \geq 0$, let Φ_k be the greedy matching policy defined by:

- *if the queue size of 4 is less than k , choose by decreasing order of priority edges $\{1, 3\}$, $\{2, 3\}$, $\{3, 4\}$, $\{5, 6\}$, or one of the other edges;*



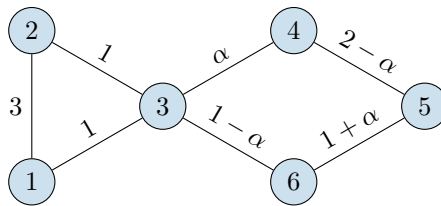
■ **Figure 18** Evolution of Λ_G as a function of β for the diamond graph with $\mu_1 = \bar{\mu}_2 = \mu_3 = \bar{\mu}_4 = \frac{1}{4}$. For each β , Λ_G is estimated by a simulation of Φ_+ , leveraging the symmetry of the matching problem. $\Lambda_{\geq 0}$ and other bounds are displayed for comparison. All results are expressed in kernel basis.

- otherwise choose by decreasing order of priority edges $\{3, 4\}$, $\{5, 6\}$, $\{1, 3\}$, $\{2, 3\}$, or one of the other edges.

We have:

- (G, μ, Φ_k) is stable for all $k \geq 0$;
- if $\alpha_+(k)$ denotes the matching rate of Φ_k in kernel basis, $\lim_{k \rightarrow \infty} \alpha_+(k) = 1$.

► **Corollary 6.8.** For the matching problem (G, μ) described in Figure 19, we have $\Lambda_G =]0, 1[= \Lambda_{>0}$.



■ **Figure 19** Generic solution of the conservation equation (2) on a fish graph with arrival rates $\mu = (4, 4, 3, 2, 3, 2)$.

The proof of 6.7 requires three main steps:

1. We first need to show that using priorities, one can limit the matching rates between class-3 items and class-4 or class-6 items;
2. Using this limitation, we build an *unstable* greedy policy $\Phi_{+\infty}$ for which the matching rate at edge $\{3, 6\}$ is null;
3. We then show that the family of *stable* greedy policies Φ_k that can be arbitrarily close to $\Phi_{+\infty}$ in the sense that the matching rate at edge $\{3, 6\}$ can be arbitrarily close to zero.

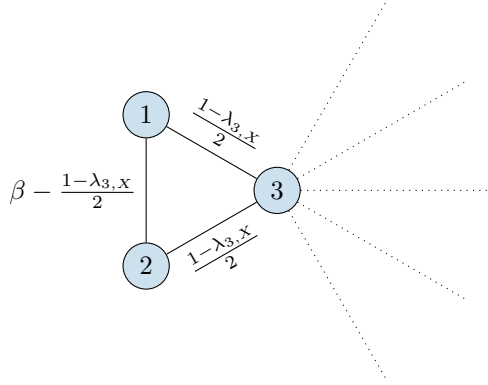
Notice that Φ_k is uniquely defined: all decisions that may occur in Figure 19 are uniquely determined. The fact that (G, μ) is stabilizable is a direct consequence of the existence of

positive solutions (e.g. $\alpha = 1/2$). The fact that $\Lambda_{\geq 0} = [0, 1]$ can be immediately deduced from the generic solution of the conservation equation (2) in Figure 19.

The first step for proving Proposition 6.7 is Lemma 6.9 below: the matching rate between class 3 and (class 4 or class 6) is limited when the priority of $\{1, 3\}$ and $(2, 3)$ is high.

► **Lemma 6.9.** *Let (G, μ) be a matching problem that contains the triangle subgraph depicted in Figure 20 (the triangle is only connected to the rest of G through node 3). Let Φ be a greedy policy where vertex 3 always prefer 1 or 2 over any other edge (the behavior of Φ for other decisions can be arbitrary). If $\beta > 1/2$, we have:*

- *The process of the queue sizes at 1, 2, and 3 is positive recurrent;*
- *The total matching rate $\lambda_{3,X}$ between 3 and nodes other than 1 and 2 is limited (i.e. upper-bounded) by $\frac{1}{1+2\beta}$.*



■ **Figure 20** Triangle sub-graph, with arrival rates $\mu = (\beta, \beta, 1, \dots)$. Node 3 is the unique bridge between the triangle and the rest of the graph. $\lambda_{3,X} = \mu_3 - \lambda_{2,3} - \lambda_{1,3}$ denotes the sum of matching rates between 3 and the rest of the graph.

Proof. Consider some $\beta > 1/2$. We first observe that the queue size of 3 is stable as it is controlled by $2\beta > 1$ (nodes 1 and 2 are not linked to anyone but each other and 3). To prove the stability of the queues 1 and 2, we must show that $\{1, 3\}$ and $\{2, 3\}$ have positive rates, which is the case if $\lambda_{3,X} < 1$. To do that, we associate to each arrival sequence of items 1, 2, and 3 the following virtual policy Ψ : the items are matched as in Φ , with the assumption that 3 always has a non-empty neighbor in the rest of the graph. Within Ψ , the rest of the graph acts like a “black hole” so whenever an item 3 arrives and 1 and 2 are empty, it is matched with some node from the rest of the graph. We can easily check that the number of matches $\{3, X\}$ made under Φ is always at most the number of matches $\{3, X\}$ under Ψ . Under Ψ , 3 is always empty (all arrivals are instantly matched) and we can partition the state of queues in 1, 2, and 3 into two $M/M/1$ queues sharing the same empty state, each one having load $\rho := \frac{\beta}{1+\beta}$. It follows that the probability of the empty state is

$$p_{\emptyset} = \frac{1}{1 + 2\frac{\rho}{1-\rho}} = \frac{1}{1 + 2\beta}.$$

The matching rate of $\{3, X\}$ in Ψ is exactly p_{\emptyset} (arrival of a class-3 item while queues 1 and 2 are empty), so the matching rate $\lambda_{3,X}$ in the original policy Φ checks

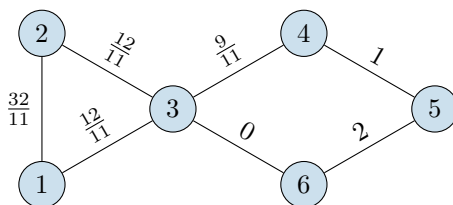
$$\lambda_{3,X} \leq \frac{1}{1 + 2\beta} < 1.$$

In particular, the queues 1 and 2 are stable. ◀

The second step for proving Proposition 6.7 is Lemma 6.10 below, which consists of producing a greedy policy that is unstable with respect to node 4 but stable for the other nodes.

► **Lemma 6.10.** *In the matching (G, μ) depicted Figure 21, let Φ_∞ be the greedy matching policy defined by: choose by decreasing order of priority edges $\{1, 3\}$, $\{2, 3\}$, $\{3, 4\}$, $\{5, 6\}$, or one of the other edges. We have:*

- (G, μ, Φ_∞) is unstable;
- The process defined by the queue sizes of all nodes but 4 is positive recurrent;
- The matching rates achieved by the system are the ones from Figure 21 (the conservation equation (2) does not hold).



■ **Figure 21** Matching rates under the unstable policy $\Phi_{+\infty}$ ($\mu = (4, 4, 3, 2, 3, 2)$). The unstable node 4 does not satisfy the conservation equation (2), as its total matching rate is $\frac{20}{11} < 2$.

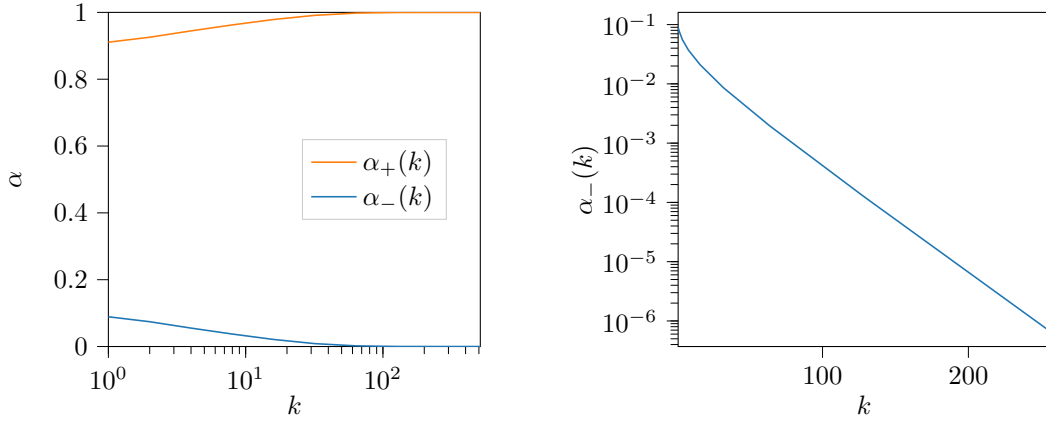
Proof. We first apply Lemma 6.9 on vertices 1 to 3, using a scaling of 3 and $\beta = 4/3$. It shows that 1, 2, and 3 are stable and that $\lambda_{3,X}$ is at most $3 \frac{1}{1+2\frac{4}{3}} = \frac{9}{11}$.

We now focus on the even cycle 3–4–5–6. The rate from classes 4, and 6, $\mu_4 + \mu_6 = 4$ needs to be absorbed by items from classes 3 and 5 but the rate of class 5 is 3 and the rate that class 3 can allocate to the cycle is at most $\frac{9}{11} < 1$. We deduce that the sum of the queue sizes of 4 or 6 is transient. However, 5 always chooses class 6 if it can and $\mu_5 > \mu_6$ so the queue size 6 is stable. This means that queue size of 4 is the unique transient process and that the queue size 5 is stable.

In fact, as the queue size of 4 is non-empty with probability 1, the queue sizes of 3 and 5 are empty with probability 1. It also means that Φ_∞ behaves like the virtual policy Ψ (cf proof of Lemma 6.9) for nodes 1, 2, and 3. This gives $\lambda_{1,2}$, $\lambda_{1,3}$, $\lambda_{2,3}$, and $\lambda_{3,X}$. Because of the priority rule, $\lambda_{3,4} = \lambda_{3,X}$ and $\lambda_{3,6} = 0$. The last matching rates are obtained by using the conservation principle on the (stable) nodes 6 and 5. ◀

We now perform the last step and prove Proposition 6.7.

Proof of Proposition 6.7. We first need to prove that Φ_k is stable. The queue size of 4 is positive recurrent: as soon as it is greater than k , all arrival from 3 are matched to class-4 items so the departure rate becomes greater than the arrival rate ($\mu_3 = 3 > \mu_4 = 2$). Stability of other vertices ensues. Call $p_\emptyset(k)$ the probability that 4 is empty under Φ_k . As Φ_k behaves for 4 like the transient process of Φ_∞ up to size k , we have $\lim_{k \rightarrow \infty} p_\emptyset(k) = 0$. A match $\{3, 6\}$ can only occur when an item 3 or 6 arrives while vertex 4 is empty so we have $0 \leq 1 - \alpha_+(k) \leq 5p_\emptyset(k)$, which concludes the proof. ◀

(a) Evolution of $\alpha_-(k)$ and $\alpha_+(k)$.(b) Evolution of $\alpha_-(k)$ in logarithmic scale.

■ **Figure 22** Evolution of $\alpha_-(k)$ and $\alpha_+(k)$ as a function of the threshold k , as defined in Proposition 6.7 and Corollary 6.8 and proofs. All results are expressed in kernel basis.

Proof of Corollary 6.8. We proved that there is a family of greedy policies such that their matching rate can be arbitrarily close to 1 (in kernel basis). We just need to exhibit another family such that the matching rate, in kernel basis, can be arbitrarily close to 0 and use the convexity of Λ_G to conclude. For $k \geq 0$, let Φ_{-k} be the greedy matching policy defined by:

- if there are less than k items of type 4 in the system, choose by decreasing order of priority edges $\{1, 3\}$, $\{2, 3\}$, $\{3, 6\}$, $\{4, 5\}$, or one of the other edges;
- otherwise choose by decreasing order of priority edges $\{3, 6\}$, $\{4, 5\}$, $\{1, 3\}$, $\{2, 3\}$, or one of the other edges.

Using exactly the same proof than for Proposition 6.7 (Φ_{-k} is Φ_k after switching the labels 4 and 6), we get that if $\alpha_-(k)$ denotes the matching rate of Φ_{-k} in kernel basis, $\lim_{k \rightarrow \infty} \alpha_-(k) = 0$. ◀

Figure 22 shows $\alpha_-(k)$ and $\alpha_+(k)$, obtained by simulation. We can observe that they converge to 0 and 1 respectively, validating Corollary 6.8. Moreover the convergence seems linear, i.e. like ρ^k for some $0 < \rho < 1$, as hinted by Figure 22b.

6.2 Arbitrary policies

We now investigate $\Lambda_{\mathcal{P}} \subseteq \Lambda_{\geq 0}$, the subset of matching rates that can be achieved by an arbitrary stable policy. As $\Lambda_{\mathcal{P}}$ is convex, we first focus on its relation with the vertices of $\Lambda_{\geq 0}$, which is given by Proposition 6.11.

► **Proposition 6.11.** *Let y be vertex of $\Lambda_{\geq 0}$.*

- *If y is bijective, $y \in \Lambda_{\mathcal{P}}$;*
- *If y is injective-only, $y \notin \Lambda_{\mathcal{P}}$.*

Proof. Let $G_y = (V, E_y)$ be the subgraph of G associated to y . The first part is straightforward by considering a *match-the-longest* policy with a filter on E_y , denoted $\Phi(y)$. (G_y, μ) is stabilizable and y is its unique matching rate. $\Phi(y)$ behaves like a greedy *match-the-longest* policy on G_y , so $(G, \mu, \Phi(y))$ is stable and its matching rate is y . The second part is obtained by noticing that if y is injective-only, G_y admits at least a bipartite component (more specifically a tree). The arrival drift between the two parts of the bipartite component makes impossible the existence of a stable policy. ◀

Proposition 6.11 essentially states that bijective vertices are easy to reach, while injective-only vertices are not due to a “bipartite curse”. This gives us a first partial characterization of $\Lambda_{\mathcal{P}}$.

► **Corollary 6.12.** *If a face F of $\Lambda_{\geq 0}$ contains only bijective vertices, $F \in \Lambda_{\mathcal{P}}$. In particular, $\Lambda_{\mathcal{P}} = \Lambda_{\geq 0}$ if and only if $\Lambda_{\geq 0}$ is simple and (G, μ) is essential.*

Proof. We just use the Proposition 6.1 (convexity of $\Lambda_{\mathcal{P}}$) and Proposition 6.11. The last statement comes from Proposition 5.12. ◀

To go further, we propose to introduce a family of stable policies that is able to arbitrarily approach a vertex of $\Lambda_{\geq 0}$ even if it is injective-only.

► **Proposition 6.13.** *Let y be a vertex of $\Lambda_{\geq 0}$. For $k \geq 0$, consider the following semi-filtering policy, denoted $\Phi_k(y)$:*

- *If the size of the longest queue is less than k , apply a filtering match-the-longest policy on $G_y = (V, E_y)$;*
- *Otherwise, apply a greedy match-the-longest policy.*

$\Phi_k(y)$ is stable and $\lim_{k \rightarrow \infty} \lambda(\Phi_k(y)) = y$.

Sketch of proof. In essence, the technique is similar to the one used for Proposition 6.7: we take an unstable policy that achieves the desired goal and make it stable by reverting to a stable policy when the queue sizes become too large. If the threshold is high enough, most of the matchings will be made under the unstable policy.

We present here a sketch of proof for the hard case where G_y is a tree (like in Example 3.7 when $\mu_1 \bar{\mu}_2 = \mu_3 \bar{\mu}_4$). When G_y has multiple connected components, bijective or injective-only, the proof needs to be adapted accordingly.

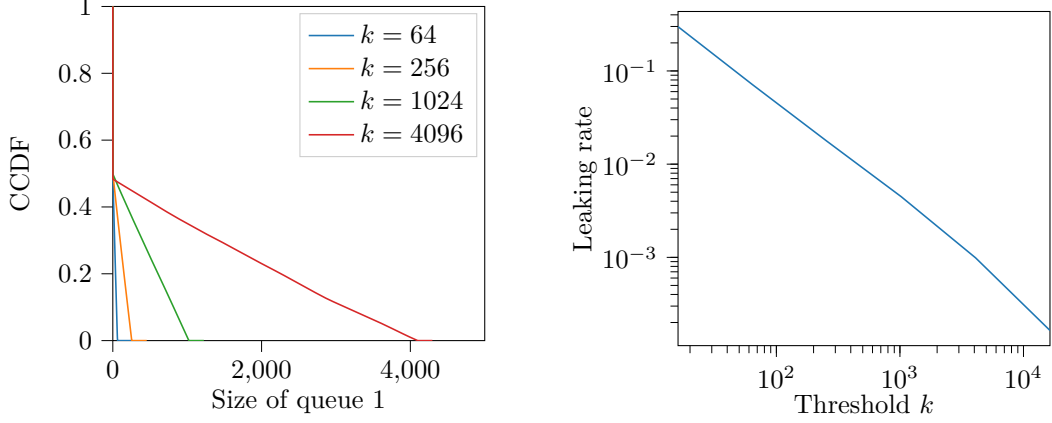
The stability of $\Phi_k(y)$ comes from the fact that apart from a finite set of states (the states where the queue sizes are less than k), $\Phi_k(y)$ behaves like a stable policy.

Let p_ℓ be the probability that the longest queue of the system has size ℓ . If we look at $\ell \geq k$, the stability induces a negative drift, which means that we have $\sum_{\ell \geq k} p_\ell \leq C p_k$ for a constant C that does not depend on k . Conversely, for $\ell \leq k$, the size of the longest queue is mostly controlled by the drift of an unbiased random walk between the two parts of the bipartite graph, which means $p_\ell \geq c p_k$ for another constant c . By combining the two, we get that the probability that the longest queue of the system has a size greater than k tend to 0 when k goes to infinity. As matchings outside E_y only occur when the longest queue is greater than k , we conclude that the matching rate of an edge outside E_y goes to 0, which by continuity of the conservation principle and injectivity of G_y means that $\lim_{k \rightarrow \infty} \lambda(\Phi_k(y)) = y$. ◀

One drawback of Proposition 6.13 is that the average size of the longest queue will tend to grow with k when the vertex is injective-only. In other words, there exists a trade-off between approaching an injective-only vertex and the minimization of the waiting time. This issue is somehow similar to the threshold-based greedy policies we introduce for Proposition 6.7, where the vertices of $\Lambda_{\geq 0}$ could be asymptotically reached by making the average queue size of a given node grow.

Note that in [26], a family of policies is introduced to optimize a reward function on edges. Their goal is similar to ours, with two main differences: first they aim at optimizing a reward function on matching rates but do not provide any description of the limit rate, while we show that the limit rate is a vertex of $\Lambda_{\geq 0}$; second their family is a linear combination of a reward-based policy and a virtual queue policy, parameterized by a balance factor β ,

while we rely on semi-filtering greedy policies with a threshold cut k . Yet, they also notice in their solution that the system has an average queue size that grows in $1/\beta$, which is the equivalent of our k .



(a) Distribution of the queue size of one node for several thresholds.

(b) Total matching rate along forbidden edges as a function of the threshold ($16 \leq k \leq 16384$).

■ **Figure 23** Evaluation of a semi-filtering policy applied to the matching problem described in Example 5.10. The target vertex is the one from Figure 15f.

To illustrate 6.13, we evaluated $\Phi_k(y)$ on the codomino matching problem from Example 5.10. Figure 23 presents the results for the vertex y depicted in Figure 15f. y is injective-only: its support is three non-adjacent edges ($\{1, 6\}, \{2, 3\}, \{4, 5\}$), i.e. three connected components that are all bipartite.

Figure 23a shows the complementary cumulative distribution function (CCDF) for the size of queue 1 (the other nodes have a similar distribution) for different value of k . We observe the following:

- Half of time, the queue is empty. It can be related to the cases where the unbiased drift between the number of class-1 arrivals and the number of class-6 arrivals favors class 6, so the queue 6 is non-empty and items from class 1 are matched on arrival.
- The rest of time, the size of queue 1 seems to be uniformly distributed between 0 and k .
- The cases where the queue size is greater than k seem negligible.

These observations are consistent with the proof of Proposition 6.13: most of the time, a semi-filtering policy is in its filtering regime, and the evolution of queues within this regime looks like an unbiased random walk capped by k .

Figure 23b shows the convergence of $\lambda(\Phi_k(y))$ to y by measuring as a function of k the *leaking rate*, i.e. the total matching rates observed along the forbidden edges ($\{1, 2\}, \{2, 6\}, \{3, 4\}, \{3, 5\}, \{5, 6\}$). The convergence seems to be logarithmic, i.e. like k^{-1} . This is in line with the fact that the probability to have a queue size greater than k , e.g. to disable the filtering, is in k^{-1} .

► **Corollary 6.14.** *Any positive solution of the conservation equation (2) can be obtained by a stable policy. In other words,*

$$\Lambda_{>0} \subseteq \Lambda_{\mathcal{P}} \subseteq \Lambda_{\geq 0}.$$

Proof. Obviously $\Lambda_{\mathcal{P}} \subseteq \Lambda_{\geq 0}$ because any stable policy must respect the conservation principle. From Proposition 6.13, we know that any vertex of $\Lambda_{\geq 0}$ is part of the closure of $\Lambda_{\mathcal{P}}$, so by

convexity of $\Lambda_{\mathcal{P}}$ this closure is $\Lambda_{\geq 0}$. In particular, $\Lambda_{\mathcal{P}}$ contains the interior of $\Lambda_{\geq 0}$, which is $\Lambda_{>0}$. ◀

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, and Amy R. Ward. Matching Impatient and Heterogeneous Demand and Supply. *arXiv:2102.02710 [cs, eess, math]*, December 2021. arXiv: 2102.02710. URL: <http://arxiv.org/abs/2102.02710>.
- 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. *arXiv:2111.13231 [math]*, November 2021. arXiv: 2111.13231. URL: <http://arxiv.org/abs/2111.13231>.
- 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 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.
- 8 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.
- 9 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>.
- 10 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.
- 11 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>.
- 12 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.
- 13 Céline Comte and Jaap Storm. Global and local stability in parallel service systems with redundancy. To appear.
- 14 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.
- 15 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.
- 16 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.

- 17 Reinhard Diestel. *Graph Theory*. Springer Publishing Company, Incorporated, 5th edition, August 2017.
- 18 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.
- 19 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.
- 20 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>.
- 21 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.
- 22 László Lovász and M. D. Plummer. *Matching Theory*. American Mathematical Soc., 2009.
- 23 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.
- 24 Fabien Mathieu. Stochastic Matching. URL: https://balouf.github.io/stochastic_matching/index.html.
- 25 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.
- 26 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.
- 27 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.
- 28 Günter M. Ziegler. *Lectures on polytopes*. Springer-Verlag, New York, 1995. URL: http://www.worldcat.org/search?qt=worldcat_org_all&q=9780387943657.

A Minimal stability region for greedy matching policies

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 [8, Proposition 5.1] for non-bipartite matching models.

► **Proposition A.1.** *Consider a matching problem (G, μ) with a connected graph G . If*

$$\sum_{i \in V(\mathcal{I})} \mu_i > \frac{1}{2} \sum_{i \in V} \mu_i, \quad \mathcal{I} \in \mathbb{I}, \quad (20)$$

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

Proof. Consider a greedy matching policy Φ and its continuous-time Markov chain $\{C(t), t \geq 0\}$, with state space \mathcal{C} , and let $(C_k)_{k \in \mathbb{N}}$ denote the discrete time chain obtained by uniformization with rate 1. We consider the Lyapunov function $L : \mathcal{C} \rightarrow \mathbb{R}$ defined by $L(c) = \|h(c)\|_1$, that is, the number of unmatched items in state c . For each $k \in \mathbb{N}$ and $c \in \mathcal{C}$, we have

$$\mathbb{E}(L(C_{k+1}) \mid C_k = c) - L(c) = \sum_{i \in V \setminus V(\mathcal{I})} \mu_i - \sum_{i \in V(\mathcal{I})} \mu_i = - \left(\sum_{i \in V(\mathcal{I})} \mu_i - \sum_{i \in V \setminus V(\mathcal{I})} \mu_i \right)$$

where $\mathcal{I} = \{i \in \mathcal{I} : h(c)_i > 0\}$ is the set of unmatched customer classes in state c . Importantly, if $L(c) > 0$ (that is, $c \neq \emptyset$), then \mathcal{I} is an independent set of the compatibility graph G because the policy Φ is greedy. It follows that, for each $c \neq \emptyset$, we have

$$\mathbb{E}(L(C_{k+1}) \mid C_k = c) - L(c) \leq -\varepsilon, \quad \text{with } \varepsilon = \min_{\mathcal{I} \in \mathbb{I}} \left(\sum_{i \in V(\mathcal{I})} \mu_i - \sum_{i \in V \setminus V(\mathcal{I})} \mu_i \right).$$

Equation (20) implies that $\varepsilon > 0$. Using the Lyapunov-Foster theorem [7, Theorem 1.1], we conclude that the matching model (G, μ, Φ) is stable. ◀

As one out expect, any matching problem (G, μ) that satisfies (A.1) is stabilizable in the sense of Definition 2.1. Indeed, (A.1) implies condition 2 in Proposition 3.2 because $\mathcal{I} \subseteq V \setminus V(\mathcal{I})$ for each $\mathcal{I} \in \mathbb{I}$. Corollary A.2 below shows that, conversely, whether a stabilizable matching problem satisfies (A.1) depends on the structure of the graph G : statements 1 and 2 exhibit compatibility graphs G such that (A.1) is satisfied whenever the matching problem (G, μ) is stabilizable, while statements 3 and 4 exhibit stabilizable compatibility graphs G for which (A.1) is never satisfied.

► **Corollary A.2.** *Consider a matching problem (G, μ) .*

1. *If G is a complete graph K_n with $n \geq 3$ (that is, a graph with diameter 1), then (20) is satisfied, and therefore the matching model (G, μ, Φ) is stable for each greedy policy Φ .*
2. *If G is the diamond graph of Example 3.7, then (20) is satisfied, and therefore the matching model (G, μ, Φ) is stable for each greedy policy Φ .*
3. *If G has diameter greater than or equal to 3, then (20) is not satisfied.*
4. *If G contains a leaf (that is, a node with degree 1), then (20) is not satisfied.*

Proof. We prove the four statements one after another.

1. The independent sets of a complete graph K_n are the singletons. Therefore, we verify that equation (20) and condition 2 in Proposition 3.2 are both equivalent to $\mu_i < \frac{1}{2} \sum_{i \in V} \mu_i$, which proves statement 1.
2. Statement 2 follows by observing that (5) and (20) are equivalent for the diamond graph.
3. Assume that G contains two nodes i and j that are at distance 3 or more, so that the sets V_i and V_j are disjoint. Therefore, applying (5) to both $\{i\}$ and $\{j\}$ and summing the inequalities yields $\sum_{i \in U} \mu_{i'} > \sum_{i' \in V} \mu_{i'}$, with $U = V_i \cup V_j \subseteq V$. This is a contradiction, so (5) cannot be satisfied by both $\{i\}$ and $\{j\}$. This proves statement 3.
4. If the graph G contains a leaf, that is, a node i with a single neighbor j , then again applying (5) to both $\{i\}$ and $\{j\}$ and summing the inequalities yields $\sum_{i \in U} \mu_{i'} > \sum_{i' \in V} \mu_{i'}$, where $U = V_j \cup \{j\} \subseteq V$, which is again a contradiction. ◀

B Proof of Proposition 5.2

The algorithm is mainly based on the notion of *cycle space* of a graph, whose main notions we recall (cf [17, Section 1.9] for details).

A *spanning subgraph* of a graph $G = (V, E)$ is a subgraph $G' = (V, E')$ with $E' \subseteq E$ the same set of nodes. A subgraph is *Eulerian* if every vertex has an even degree. In particular, if E' are the edges of a cycle of G , then (V, E') is Eulerian. The *cycle space* of G is the vector space made of all Eulerian spanning subgraphs of G , using symmetric difference for addition and the two-element field for scalar multiplication. Its dimension is $m - n + 1$.

Proof that Algorithm 1 terminates. We first prove the existence of edge a defined on line 3 of Algorithm 1. By definition of a spanning tree, $\mathcal{T} \cup \{a\}$ contains a unique cycle for each $a \in E \setminus \mathcal{T}$. The $m - n + 1$ cycles thus obtained are independent (each one contains a distinct a) so they form a basis of the cycle space of G . Since the linear combination of even cycles cannot produce an odd cycle and G contains at least one (it is non-bipartite), one of the basis vector must be an odd cycle.

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 this is an even cycle included into $\mathcal{T} \cup \{a, s\}$, and we are therefore in case (i).
- If C_ℓ is odd, and C_r and C_ℓ have at least one edge in common, 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). ◀

Proof that Algorithm 1 returns the correct result. We finally prove that the vectors v generated (i) belong to the kernel of A , (ii) are independent, and (iii) there are $m - n$ of them. Statement (iii) is straightforward because the cardinality of $E \setminus (\mathcal{T} \cup \{a\})$ is $m - n$. To verify (i), it suffices to observe that $Av = 0$, e.g. that for each node the total of its edges weighted with v is 0:

- Any node not adjacent to the support of v has only null edges;
- Any cycle node apart from v_i and v_j has non-zero weights $(-1, 1)$;
- Any path node apart from v_i and v_j has non-zero weights $(-2, 2)$;
- If $p > 0$, v_i has non-zero weights $(-1, -1, 2)$; v_j has non-zero weights $(-1, -1, 2)$ if p is odd or $(1, 1, -2)$ if p is even;
- If $p = 0$, v_i is v_j . It has non-zero weights $(-1, -1, 1, 1)$.

Lastly, to verify (ii), we observe that, for each $s \in E \setminus (\mathcal{T} \cup \{a\})$, s belongs to the cycle or kayak paddle used, so the vector constructed from edge s is the only one with a non-zero value in s . ◀