On Structural Rank and Resilience of Sparsity Patterns

Mohamed-Ali Belabbas, Xudong Chen, and Daniel Zelazo, *Senior Member, IEEE*

***Abstract*—A sparsity pattern in $\mathbb{R}^{n\times m}$, for $m \geq n$, is a vector subspace of matrices admitting a basis consisting of canonical basis vectors in $\mathbb{R}^{n\times m}$. We represent a sparsity pattern by a matrix with $0/\star$-entries, where $\star$-entries are arbitrary real numbers and 0-entries are equal to 0. We say that a sparsity pattern has full structural rank if the maximal rank of matrices contained in it is n. In this article, we investigate the degree of resilience of patterns with full structural rank: We address questions, such as how many $\star$-entries can be removed without decreasing the structural rank and, reciprocally, how many $\star$-entries one needs to add so as to increase the said degree of resilience to reach a target. Our approach goes by translating these questions into max-flow problems on appropriately defined bipartite graphs. Based on these translations, we provide algorithms that solve the problems in polynomial time.**

***Index Terms*—Graph theory, matchings, max-flows, passivity, sparsity patterns.**

I. Introduction

THE development of network-enabled systems [1], [2], [3], [4] is creating new opportunities for integrating theretofore disconnected systems. These systems, however, come with new challenges associated with their secure and resilient operation in the face of network-level faults or even malicious actors intentionally aiming to disrupt their functionality. An inherent challenge in problems related to the resilience of these systems against faults or attacks stems from their combinatorial nature, which is induced by the network interconnections. Indeed, the removal or addition of a communication link in a network is a binary operation and does not fit well within the framework of robust control theory. Closely related to the network resilience problem is the study of structural properties of dynamical systems. The structure is often described by graphs, where the sparsity pattern of the system parameters indicates the presence or absence of edges in an associated graph. The so-called *structural system theory* aims to determine whether controllable or stable dynamics can be sustained by a given system structure, which is described via a sparsity patterns for the system matrices [5], [6], [7], [8], [9], [10], [11], [12].

In this article, we address a novel resilience problem for structural system theory. A basic object in that domain is a sparsity pattern, i.e., a vector space of matrices admitting a basis comprised only of canonical basis vectors. We represent them as matrices with $0/\star$-entries, where $\star$ denote arbitrary real entries. The starting point of our analysis is to determine whether a given sparsity pattern contains an open set of matrices of *full rank*. Necessary and sufficient conditions for this requirement to hold are in fact well known and can easily be described using a graph machinery (see Lemma 3). The core problems we address in this article go beyond that. We consider the *resilience* of the full-rank property of these sparsity patterns—here, resilience refers to the property of a sparsity pattern being full-rank after the removal of $\star$-entries (which can be viewed as attacks on communication links). The list of specific problems will be presented in Section I-B.

The application areas of this work encompass problems related to structural stability of linear systems [13], structural controllability of linear ensemble systems [8], and passivation of networked systems [14]. For the sake of illustration, we elaborate on the last application domain in Section I-A.

Outside of the control theory literature, problems seeking to understand the structural rank of sparsity patterns have also been addressed in the mathematical literature. In particular, we mention the *minimum rank* problem, which aims to determine the minimum rank of real symmetric matrices in a sparsity pattern (where the $\star$-entries *have to* be nonzero); see [15] and [16], and the references therein for a comprehensive survey on this subject. A typical approach to the minimal rank problem involves analyzing a corresponding *inverse problem*, which is trying to identify a graph structure from the spectrum of a

Manuscript received 15 June 2022; accepted 22 September 2022. Date of publication 4 October 2022; date of current version 28 July 2023. The work of M.-A. Belabbas and X. Chen was supported in part by NSF under Grant ECCS-1809076 and Grant ECCS-1809315 and in part by the Air Force Office of Scientific Research under Grant FA9550-20-1-0076 and Grant FA9550-20-1-0333. The work of D. Zelazo was supported by the United States–Israel Binational Science Foundation under Grant 2017658 and Grant 2285/20 from the Israel Science Foundation. Recommended by Associate Editor M. K. Camlibel. *(Mohamed-Ali Belabbas and Xudong Chen contributed equally to this work.) (Corresponding author: Xudong Chen.)*

Mohamed-Ali Belabbas is with the Department of Electrical and Computer Engineering and the Coordinated Science Laboratory, University of Illinois Urbana-Champaign, Urbana, IL 61801 USA (e-mail: belabbas@illinois.edu).

Xudong Chen is with the Department of Electrical, Computer, and Energy Engineering, University of Colorado Boulder, Boulder, CO 80309 USA (e-mail: xudong.chen@colorado.edu).

Daniel Zelazo is with the Faculty of Aerospace Engineering, Technion-Israel Institute of Technology, Haifa 3200003, Israel (e-mail: dzelazo@technion.ac.il).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TAC.2022.3212013.

Digital Object Identifier 10.1109/TAC.2022.3212013

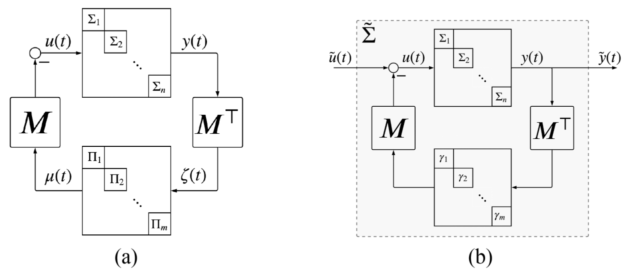


Fig. 1. General network system and a network passivation approach. (a) Block-diagram of the network system (Σ, Π, M). (b) Passivation of the system Σ over the interconnection *M*.

matrix [17]. Some other relevant works include the rank reduction of the adjacency matrix of a directed graph (digraph) by vertex and/or edge deletions [18].

A. Application: Resilience for Network Passivation

To illustrate the importance of the sparsity patterns for network systems and their influence on network robustness and resilience, we will look at a general network systems architecture. This section is, thus, meant to provide a system theoretic motivation for the problems mentioned above, but the remainder of this article does not rely on the notions introduced here.

Consider an ensemble of n agents and m controllers that may exchange state information over a network represented by a matrix $M \in \mathbb{R}^{n\times m}$. For ease of exposition, we let the entries of M be either 0 or 1. In this sense, when $M_{ij} = 1$, it means that controller j has access to state information from agent i. The matrix M can, therefore, also be associated with a graph $G = (V, E)$ with $|V| = n + m$ nodes and edge-set E describing the sparsity pattern of M.

For this setup, we assume that the agents and the controllers are associated with the dynamical systems $\Sigma_i : u_i \mapsto y_i$ for $i = 1, \dots, n$ and $\Pi_j : \zeta_j \mapsto \mu_j$, for $j = 1, \dots, m$. Here, we assume that the agent dynamics and controllers are SISO systems (i.e., $u_i, y_i, \zeta_j, \mu_j \in \mathbb{R}$). The loop is closed by taking $\zeta(t) = M^\top y(t)$ and $u(t) = -M\mu(t)$. This interconnection structure is motivated by the association of each controller with a set of agents. Thus, controller j receives a linear combination of the outputs of adjacent agents (the adjacency relation is encoded in the jth column of M) and distributes its control output back to the same set of agents. We denote such systems by the triplet (Σ, Π, M), as shown in Fig. 1(a). Note that if the matrices M are taken to be the *incidence matrix* of a graph G, then the system (Σ, Π, M) describes the well-known diffusively coupled networks [19], [20], [21].

The stability of the interconnection in Fig. 1(a) can be guaranteed by the (output-strict) passivity of the systems Σ_i and passivity of the controllers Π_j [21], [22]. In many applications, however, it may not be possible to guarantee the passivity of the agents Σ_i. This corresponds to some or all of the agents possessing a negative passivity index; see [23] and [24] for more details on this notion. Nevertheless, it is still desirable to be able to interconnect these so-called passive-short systems with each other to achieve group coordination tasks. In this direction, there have been recent works that aim to passify these agents over the network itself [14], [25], [26]. This architecture can be seen in Fig. 1(b), where the gains γ_i are chosen to ensure that the system from external input $\tilde{u}$ to output $\tilde{y} = y$ is passive. If this can be achieved, then it can be shown that the network interconnection $(\tilde{\Sigma}, \Pi, M)$ is stable, where $\tilde{\Sigma}$ maps $\tilde{u}$ to output $\tilde{y}$ [14]. The conditions for which this is possible were explored in [14]. The main result can be extended to the general network structure M so we state it as follows without a proof.

Lemma 1: Let $R = \mathrm{diag}(\rho_1, \dots, \rho_n)$ be a diagonal matrix containing the passivity index ρ_i of each agent Σ_i, and assume that $\rho_i < 0$ for at least one agent. If $R + M\,\mathrm{diag}(\gamma)M^\top$ is positive semidefinite, then $\tilde{\Sigma}$, mapping $\tilde{u}(t)$ to $\tilde{y}(t)$ as in Fig. 1(b), is passive with respect to any steady-state input–output pair. Moreover, if $R + M\,\mathrm{diag}(\gamma)M^\top$ is positive-definite, then $\tilde{\Sigma}$ is output-strict passive. Furthermore, there exist scalars γ_i, for $i = 1, \dots, m$, such that $R + M\,\mathrm{diag}(\gamma)M^\top > 0$ if and only if $x^\top Rx > 0$ for any nonzero $x \in \mathrm{Ker}(M^\top)$.

This result shows that for a given network matrix M, it may not even be possible to guarantee a network passivation scheme that ensures output-strict passivity of $\tilde{\Sigma}$. At the same time, it hints that for a given set of passivity indices ρ_i, a change to the network matrix M may allow for output-strict passivation. This result also shows that for a *full-rank* matrix $MM^\top$, it will always be possible to find a single gain γ such that $R + \gamma MM^\top > 0$.

With this setup, we can now motivate the study of the *structural rank* of the interconnection matrix M. For a matrix M with a given sparsity pattern, how many of its entries can be removed, corresponding to compromising the network connection between an agent and controller, before the matrix loses rank. In the case where the network is being used to also passify the agents, this loss of rank may lead to the loss of passivity of $\tilde{\Sigma}$, thereby destroying the convergence guarantees of the network system $(\tilde{\Sigma}, \Pi, M)$. To illustrate this, we present a brief example.

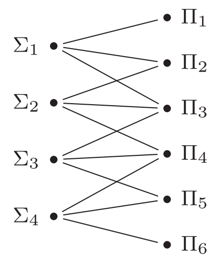
Example 1: We consider an ensemble of $n = 4$ identical but unstable plants, with dynamics of each agent described by the SISO transfer function $\Sigma_i(s) = (s + 0.5)/(s - 1)$ for $i \in \{1, \dots, 4\}$. It can be verified that the agents are output passive-short with $\rho = -2$.[1] The agents are to be controlled according to the architecture in Fig. 1(a) with the network matrix M, as illustrated in Fig. 2. Since M is full rank, according to Lemma 1, the ensemble can be passified (and stabilized) over the network using the architecture in Fig. 1(b) and with gain $\gamma > 3.4142$ (found by using, for example, semidefinite programming [14]).

Consider now a scenario where an attacker successfully disables controllers Π_1 and Π_6 (corresponding to nulling columns 1 and 6 in M). Even under such an attack, the matrix M maintains full column rank and can still be passified, now with a gain of $\gamma > 13.7082$. On the other hand, if, in addition, the connection between Σ_3 and Π_3 is severed (i.e., changing entry M_{13} to 0), then M loses rank and it is no longer possible to passify the system over the network. Consequently, the architecture of

[1]The passivity index can be computed, for example, in MATLAB using the command `getPassiveIndex`.

$$\begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 \end{pmatrix}$$

(a)



(b)

Fig. 2. (a) Network matrix M in Example 1, and (b) its graph representation. Each column of M represents a controller Π_e while each row corresponds to a system Σ_i.

Fig. 1(a) can not be used to control the ensemble and the attacker was successful in disabling the system. ■

The above example illustrates the importance of structural rank for networked systems. Applications that can rely on this analysis include plug-and-play control for networked systems [27], [28], [29].

B. Problem Formulation and Contributions

In this section, we formulate the core problems addressed in this article. We start by introducing the notions of sparsity patterns and their rank.

A *sparsity pattern* $\mathcal{S}(n, m)$ in $\mathbb{R}^{n\times m}$ (or simply $\mathcal{S}$ if (n, m) is clear from the context) is a vector subspace that admits a basis consisting only of matrices E_{ij}'s, i.e., matrices with 1 on the ijth entry and 0, elsewhere. Such a vector space is, thus, fully determined by the pairs (i, j), which indicate the entries of matrices in $\mathcal{S}$ that are not always zero. We denote by $E(\mathcal{S})$ the collection of all such pairs, hence $\dim \mathcal{S} = |E(\mathcal{S})|$. We refer to the entries of $\mathcal{S}(n, m)$ indexed by $E(\mathcal{S})$ as $\star$-entries, and the other entries as 0-entries.

Definition 1 (Rank of sparsity pattern): The rank of a sparsity pattern $\mathcal{S}$, denoted by $\mathrm{rk}\, \mathcal{S}$, is the maximal value of the ranks of matrices in $\mathcal{S}$:

$$\mathrm{rk}\, \mathcal{S} := \max_{A\in\mathcal{S}} (\mathrm{rk}\ A).$$

It should be clear that $\mathrm{rk}\, \mathcal{S}(n, m) \leq \min\{n, m\}$. Returning to the example of Section I-A, we are interested in finding sparse matrices M such that $MM^\top$ is full rank (i.e., rank n). Thus, we assume in the sequel that $m \geq n$.

The set of sparsity patterns of the same parameters (n, m) admits a natural partial order.

Definition 2 (Partial order on sparsity patterns): Given patterns $\mathcal{S}(n, m)$ and $\mathcal{S}'(n, m)$, we write $\mathcal{S}' \succeq \mathcal{S}$ if $E(\mathcal{S}') \supseteq E(\mathcal{S})$ and $\mathcal{S}' \succ \mathcal{S}$ if $E(\mathcal{S}') \supsetneq E(\mathcal{S})$.

We now precisely define the notions of resilience studied in this article.

Definition 3a (Resilience): Given positive integers n and m with $m \geq n$, a sparsity pattern $\mathcal{S}(n, m)$ of rank n is *exactly k-resilient*, for $0 \leq k \leq |E(\mathcal{S})|$, if the following hold.

1) All patterns $\mathcal{S}' \preceq \mathcal{S}$ with $|E(\mathcal{S}')| \geq |E(\mathcal{S})| - k$ are of rank n.
2) There exists an $\mathcal{S}' \prec \mathcal{S}$ with $|E(\mathcal{S}')| = |E(\mathcal{S})| - k - 1$ whose rank is less than n.

We denote by $\mathrm{rsl}\, \mathcal{S}$ the degree of resilience of $\mathcal{S}$.

In general, finding the degree of resilience of a zero-pattern is a difficult problem. We introduce in the following a slightly stronger notion of resilience, termed *strong resilience*, which will allow us to develop fast algorithms to obtain bounds on the degree of resilience.

To motivate this definition, observe that a sparsity pattern $\mathcal{S}$ is exactly 0-resilient if its rank is n and there exists a pattern $\mathcal{S}' \prec \mathcal{S}$ whose rank is strictly less than n. Thus, when expressing $\mathcal{S}$ as a direct sum of sparsity patterns, it is clear that if *any* of the summand is of rank n, then so is $\mathcal{S}$. Following this fact, we now have the following definition.

Definition 3b (Strong resilience): Given positive integers n and m with $m \geq n$, a sparsity pattern $\mathcal{S}(n, m)$ of rank n is *exactly strongly k-resilient*, for $k \geq 0$, if it contains a *direct sum*[2] of $(k + 1)$ but not $(k + 2)$, sparsity patterns each of which is 0-resilient. We denote by $\text{s-rsl}\, \mathcal{S}$ the degree of strong resilience of $\mathcal{S}$.

On occasion, we will deal with sparsity patterns $\mathcal{S}(n, m)$ that are not full rank, i.e., $\mathrm{rk}\, \mathcal{S}(n, m) < n$. By convention, we set

$$\text{(s-)rsl}\, \mathcal{S}(n, m) := -1 \text{ if } \mathrm{rk}\, \mathcal{S}(n, m) < n.$$

Throughout this article, we shall always consider the *exact* degree of (strong) resilience of a sparsity pattern. Thus, for convenience, we will omit "exact" in the sequel if there is no confusion.

By the arguments outlined before Definition 3b, if a sparsity pattern is strongly k-resilient, then it is at least k-resilient. However, the converse is not true: there exist k-resilient patterns that cannot be expressed as a direct sum of $(k + 1)$ patterns which are 0-resilient (an example is given in Fig. 4). Nevertheless, we will show in Corollary 2 (see Section III-A) that the gap between the two notions does not have any impact on the minimal dimensions of patterns meeting either definition. Specifically, if d_k is the *minimal dimension* of a k-resilient pattern (provided that it exists)

$$d_k := \min_{\mathcal{S}\,:\,\mathrm{rsl}\, \mathcal{S}=k} |E(\mathcal{S})|,$$

then there exists a pattern of dimension d_k which is *strongly k-resilient*.

Standing from the perspective of a system designer, we pose the following questions.

P1: Given a sparsity pattern $\mathcal{S}$, what is its degree of (strong-) resilience?

P2: Given a sparsity pattern $\mathcal{S}$, what is the least number of $\star$-entries one should add to obtain a degree of (strong-) resilience k^*? This problem can be expressed as follows:

$$\min |E(\mathcal{S}^*)| \text{ s.t. } \mathcal{S}^* \succeq \mathcal{S} \text{ with } \text{(s-)rsl}\, \mathcal{S}^* = k^*.$$

P3: Given a sparsity pattern $\mathcal{S}$, what is the largest degree of (strong-)resilience we can achieve by adding p $\star$-entries?

[2]The *direct* sum $\mathcal{S}' \oplus \mathcal{S}''$, for $\mathcal{S},' \mathcal{S}'' \subseteq \mathcal{S}$, is well defined only if $\mathcal{S}' \cap \mathcal{S}'' = \{0\}$.

This problem can be expressed as follows:

$$\max\,(\text{s-})\text{rsl}\,\mathcal{S}^* \text{ s.t. } \mathcal{S}^* \succeq \mathcal{S} \text{ with } |E(\mathcal{S}^*)| = |E(\mathcal{S})| + p.$$

We pose the above questions for both resilience and strong resilience. Focusing on strong resilience will lead to polynomial-time algorithms that solve these questions exactly, thereby providing bounds for resilience. The main results are formulated in Theorems 3, 8, and 9, respectively.

Outline of proofs: The first step in our analysis is to assign a bipartite graph to a sparsity pattern and to relate the full-rank property, and its resilience, to the existence of matchings in this graph. This is done in Section II. We then proceed toward the first result, Theorem 1, in which relying on a result of König [30] to characterize the bipartite graphs corresponds to strongly k-resilient patterns. This is done in Section III-A. Relying on Theorem 1, we then translate the three problems formulated above into problems about *max-flows* over graphs. In more details, we first create several variations on the bipartite graphs associated with a sparsity pattern by adding source and target nodes, turning undirected edges into directed ones, and appropriately assigning edge- and node-capacities. We then introduce several max-flow problems defined on these modified bipartite graphs and, moreover, prove that integral solutions to these max-flows problems provide solutions to the original problems P1–P3. Finally, we demonstrate that these max-flow problems can be solved using standard algorithms in polynomial time.

II. Bipartite Graphs, Matchings, and Resilience

A. Background on Graph Theory and Flows

We introduce the necessary background and notations about graph theory and related flow problems. In this article, we will be concerned with bipartite graphs, i.e., graphs which admit a partition of their node set into two disjoint components with the property that nodes in the same components share no edge.

Denote by $G(n,m) = (V_\alpha \cup V_\beta, E)$ an undirected bipartite graph on $(n+m)$ nodes: by convention, there are n *left nodes* denoted by $\alpha_1,\ldots,\alpha_n$ and m *right nodes* denoted by $\beta_1,\ldots,\beta_m$. On occasion, we will write G by omitting the arguments (n,m) if it is clear from the context. Each edge of $G(n,m)$ connects a left node with a right node. An edge in $G(n,m)$ is thus denoted by (α_i,β_j). We say that $G' = (V,' E')$ is a subgraph of G if $V' \subseteq V_\alpha \cup V_\beta$ and $E' \subseteq E$. Given a node α in G, we denote by $\deg(\alpha, G')$ the degree of α relative to G', defined as the number of edges in E' incident to α (equivalently, the number of neighbors of α in G'). We will also consider *directed* bipartite graphs in the following; we denote the *directed* edge from α_i to β_j by $\alpha_i\beta_j$.

Recall that a *matching* in the graph $G(n,m)$ is a set of edges so that no two distinct edges are incident to the same node. For $n=m$, a *perfect matching* P in $G(n,n)$ is a set of n edges such that each node of $G(n,n)$ is incident to exactly one of these n edges. For the general case $m \neq n$, we introduce the following definition.

Definition 4 (Left-perfect matchings): A *left-perfect matching* in a bipartite graph $G(n,m) = (V_\alpha \cup V_\beta, E)$, with $m \geq n$, is a set of n edges in E so that no two distinct edges are incident to the same node.

Equivalently, $G(n,m)$, for $n \leq m$, admits a left-perfect matching if there exist n distinct right nodes $\beta_{i_1},\ldots,\beta_{i_n}$ such that the subgraph $G'(n,n)$ induced by the left nodes V_α and $\{\beta_{i_1},\ldots,\beta_{i_n}\}$ has a perfect matching. We say that two matchings P_1 and P_2 of G are *disjoint* if $P_1 \cap P_2 = \varnothing$.

Let $\vec{G} = (V,\vec{E})$ be an arbitrary digraph with two special nodes $s,t \in V$, termed the *source* and *target* nodes, respectively. The source node has no incoming edges and the target has no outgoing edges. A *capacity* on $\vec{G}$ is a function $c : \vec{E} \to \mathbb{R}_{\geq 0}$. Given the capacity, a *flow* on $\vec{G}$ is a function $f : \vec{E} \to \mathbb{R}_{\geq 0}$ such that:

1) $f(e) \leq c(e)$ for all $e \in \vec{E}$;
2) the following balance condition is satisfied at all nodes $v \in V - \{s,t\}$:

$$\sum_{u:uv\in\vec{E}} f(uv) = \sum_{w:vw\in\vec{E}} f(vw). \tag{1}$$

The *value* of the flow f is defined as

$$|f| := \sum_{v:sv\in\vec{E}} f(sv) = \sum_{v:vt\in\vec{E}} f(vt). \tag{2}$$

We denote by F_c the set of all flows on $\vec{G}$ with capacity function c. The celebrated *max-flow problem* [31] is the optimization problem formulated as follows:

$$\max_{f\in F_c} |f|.$$

It is well known that finding a solution f^* to the above optimization problem can be done in polynomial time using, e.g., the Ford–Fulkerson algorithm [32]. Note that a solution to the max-flow problem is not necessarily integer-valued, i.e., there may exist edges e such that $f^*(e)$ are not integers, even if c is integer-valued. However, if c is integer-valued (which will be the case in this article), then the output of the Ford–Fulkerson algorithm initialized at an integer-valued flow is integer-valued as well and, thus, provides an *integer-valued* maximum flow [31]. This statement is referred to as the *integrality theorem*.

A fundamental result in the study of max-flow problems is the *max-flow min-cut theorem*, which we briefly describe here. To this end, we recall the definition of a *cut* in the digraph $\vec{G} = (V,\vec{E})$ with the capacity function c: An *s-t* cut (S,T) in $\vec{G}$ is a partition of the node set of $\vec{G}$ into two disjoint sets $S \ni s$ and $T \ni t$. We denote the set of all *s-t* cuts in $\vec{G}$ as $\mathcal{C}$. For a given cut $(S,T) \in \mathcal{C}$, we let

$$\vec{E}_{(S,T)} := \{v_iv_j \in \vec{E} \mid v_i \in S, v_j \in T\}.$$

Then, the *capacity* of the cut (S,T) is defined as

$$c(S,T) := \sum_{e\in\vec{E}_{(S,T)}} c(e).$$

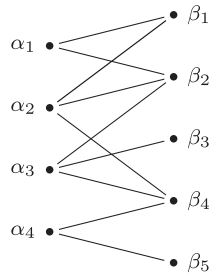
The min-cut problem is then formulated as follows:

$$\min_{(S,T)\in\mathcal{C}} c(S,T).$$

The max-flow min-cut Theorem [33] says the following.

$$\begin{pmatrix} \star & \star & 0 & 0 & 0 \\ \star & \star & 0 & \star & 0 \\ 0 & \star & \star & \star & 0 \\ 0 & 0 & 0 & \star & \star \end{pmatrix}$$

(a)



(b)

Fig. 3. We illustrate the correspondence between (a) sparsity pattern and (b) bipartite graph. The left and right nodes are labeled as α_i and β_j, respectively. A star in the ijth entry corresponds to the edge (α_i, β_j) in the bipartite graph.

Lemma 2: Given a digraph $\vec{G}$ with source s and target t and capacity function c, let F_c be the set of corresponding flow maps on $\vec{G}$ and $\mathcal{C}$ the set of s-t cuts in $\vec{G}$. Then,

$$\max_{f\in F_c} |f| = \min_{(S,T)\in\mathcal{C}} c(S,T).$$

B. Graph Theoretic View on (Strong) Resilience

To proceed, we establish some standard connections between graph theoretic concepts and the pattern properties introduced here. First, to a given sparsity pattern $\mathcal{S}(n,m)$, we can assign the bipartite graph $G(n,m) = (V_\alpha \cup V_\beta, E)$ on $(n+m)$ nodes with edge set E given by the rule: the ijth entry of $\mathcal{S}$ is a $\star$ if and only if (α_i, β_j) is an edge in E. See Fig. 3 for an illustration.

Since this representation of sparsity patterns as bipartite graphs is one-to-one, we also write (s-)$\operatorname{rsl} G$ to refer to the degree of (strong) resilience of the corresponding pattern $\mathcal{S}$. We now relate (s-)$\operatorname{rsl} G$ to perfect matchings of G. The following result is standard, and we include a proof in the Appendix for completeness.

Lemma 3: A sparsity pattern $\mathcal{S}(n,m)$ is of rank n if and only if its associated bipartite graph $G(n,m)$ admits a left-perfect matching.

As an immediate consequence of the above lemma, we can characterize k-resilient bipartite graphs as follows.

Lemma 4: A bipartite graph $G = (V_\alpha \cup V_\beta, E)$ is k-resilient if and only if the following hold.

1) For *any* subset $E' \subset E$ with $|E'| = k$, $G' = (V_\alpha \cup V_\beta, E - E')$ contains a left-perfect matching.
2) There exists a subset E' with $|E'| = k+1$ such that $G' = (V_\alpha \cup V_\beta, E - E')$ does not contain a left-perfect matching.

We can also characterize strongly k-resilient bipartite graphs using perfect matchings.

Lemma 5: A bipartite graph G is strongly k-resilient if and only if it has exactly $(k+1)$ *disjoint* left-perfect matchings.

Proof: We first show that if G has exactly $(k+1)$ disjoint left-perfect matchings, it is strongly k-resilient. Denote by $P_1, \ldots, P_{k+1}$ the disjoint left-perfect matchings in G. By Lemma 3, the graph induced by each left-perfect matching in G corresponds to a 0-resilient subpattern of $\mathcal{S}$. Furthermore, since the $(k+1)$ left-perfect matchings are disjoint, the sparsity pattern corresponding to their union is the direct sum of the subpatterns corresponding to the P_i. It then follows from Definition 3b that G is strongly k-resilient.

We now show that if G is strongly k-resilient, then it has exactly $(k+1)$ disjoint left-perfect matchings. First, note that G cannot have more than $(k+1)$ disjoint left-perfect matchings because otherwise, by the above argument, G is at least strongly $(k+1)$-resilient. It remains to show that G has at least $(k+1)$ disjoint left-perfect matchings. By definition of strong resilience, $\mathcal{S}$ contains $(k+1)$ subpatterns $\mathcal{S}_1, \ldots, \mathcal{S}_{k+1}$ that are 0-resilient and $\mathcal{S}_i \cap \mathcal{S}_j = \{0\}$ for $i \neq j$. Owing to the correspondence between sparse patterns and bipartite graphs, each subpattern corresponds a subgraph of G. Denote these subgraphs by $G_1, \ldots, G_{k+1}$. Since each pattern is 0-resilient, by Lemma 3, each G_i contains at least one left-perfect matching P_i. Since $\mathcal{S}_i \cap \mathcal{S}_j = \{0\}$ for $i \neq j$, it follows that G_i and G_j are edgewise disjoint, and hence, P_i and P_j are disjoint as well. We have, thus, shown that G has at least $(k+1)$ disjoint left-perfect matchings. ∎

III. Main Results

A. On k- and Strong k-Resilience

From Lemma 5, we know that a strongly k-resilient pattern is associated with a bipartite graph that contains exactly $(k+1)$ disjoint left-perfect matchings. To better understand strong resilience, we characterize graphs that can be obtained as unions of disjoint left-perfect matchings.

Theorem 1: A bipartite graph $G(n,m)$, for $m \geq n$, is a union of k, for $1 \leq k \leq m$, disjoint left-perfect matchings if and only if the following hold.

1) The degree of each left node is exactly k.
2) The degree of each right node is less than or equal to k.

Note that the degree of each right node of $G(n,m)$ is at most n, so for $k \geq n$, item 2) of Theorem 1 holds trivially. It is not too hard to see that the bipartite graphs characterized by Theorem 1 exist for every $k = 1, \ldots, m$.

Proof: We first establish the necessity of the two items. The necessity of item (1) is obvious. For item 2), assume, to the contrary, that there is at least one node in V_β, say v_{β_j}, with degree larger than k. Since each node of V_β is incident to at *most one* edge in a left-perfect matching, after removing the k disjoint perfect matchings of $G(n,m)$, v_{β_j} will have degree strictly larger than 0, and thus, $G(n,m)$ is not a union of k left-perfect matchings.

We next establish the sufficiency of the two items. The proof relies on the use of König's Line Coloring Theorem [34, Th. 1.4.18], which can be equivalently stated as follows: Let $G = (V_\alpha \cup V_\beta, E)$ be an arbitrary bipartite graph, and $\Delta(G)$ be the maximal degree of G, i.e., $\Delta(G) := \max_{v\in V_\alpha\cup V_\beta} \deg(v)$. Further, let $\chi(G)$ be the minimal number ℓ of *disjoint* matchings $P_1, \ldots, P_\ell$ in G such that $E = \cup_{i=1}^{\ell} P_i$. Then, $\chi(G) = \Delta(G)$. Applying König's Line Coloring Theorem to $G(n,m)$ as in the theorem statement, we obtain that E is a union of k disjoint

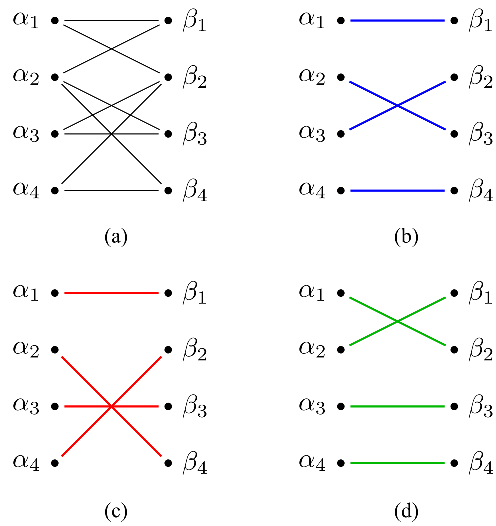


Fig. 4. (a) Graph contains three distinct (but not pairwise disjoint) perfect matchings, depicted in (b)–(d). The intersection of these three perfect matchings is the empty set, i.e., there is no common edge to all these matchings. Hence, the graph in (a) is at least 1-resilient. Moreover, since $\deg(\alpha_i) = \deg(\beta_i) = 2$ for $i = 1, 3, 4$, the graph in (a) is exactly 1-resilient. However, since the pairwise intersections of the matchings are not empty, it is not *strongly* 1-resilient.

matchings $P_1, \ldots, P_k$. In order to show that these matchings are all left-perfect matchings, it suffices to show that they are all of cardinality n. Indeed, since $n \leq m$, any matching of cardinality n is necessarily left-perfect. Note that $|E| = \sum_{i=1}^{k} |P_i|$ and, by the hypothesis on $G(n,m)$, $|E| = kn$. Finally, since G is bipartite, the cardinality of *any* matching in G cannot exceed n. We conclude that all matchings $P_1, \ldots, P_k$ have cardinality n and are, thus, left-perfect matchings. ■

The following result is a corollary of Theorem 1.

Corollary 2: The following two statements hold.

1) For any given $k = 1, \ldots, m-1$, the minimal number of edges needed for $G(n,m)$, with $m \geq n$, to be k-resilient (or strongly k-resilient) is $(k+1)n$.
2) Given a pair of positive integers (n,m) with $m \geq n$, the maximal degree of resilience (or strong resilience) of a bipartite graph $G(n,m)$ is $(m-1)$.

Proof: We first establish the fact that if a bipartite graph $G(n,m)$ is k-resilient, then it has at least $(k+1)n$ edges. To see this, recall that by Lemma 3, $G(n,m)$ is k-resilient if, after removing k edges, the remaining graph still admits a left-perfect matching. Hence, the degree of each left node has to be at least $(k+1)$ because, otherwise, such node can be disconnected from the others by the removal of the edges incident to it. Since $G(n,m)$ is bipartite, this proves the claim. Item 1 is then an immediate consequence of the above fact and Theorem 1.

We now prove item 2. To consider maximal degree of (strong) resilience, it suffices to let $G(n,m)$ be the complete bipartite graph (owing to the monotonicity of resilience with respect to adding edges). In this case, we show that the degree of (strong) resilience of $G(n,m)$ is $(m-1)$. On the one hand, the degree of every left node is m. From the fact established at the beginning of the proof, we have that $G(n,m)$ is at most $(m-1)$-resilient. On the other hand, by Theorem 1, $G(n,m)$ is a union of m disjoint left-perfect matchings. Thus, by Lemma 5, $G(n,m)$ is strongly $(m-1)$-resilient. ■

Corollary 2 says that k- and strongly k-resilience require the same minimal number of edges, and that the maximal degrees of resilience and of strong resilience one can achieve for a given (n,m) are also the same. Nevertheless, they are distinct notions: strong k-resilience is strictly stronger than k-resilience. We provide an example in Fig. 4 where a graph that is 1-resilient but strongly 0-resilient is depicted.

In the sequel, we will mostly focus on *strong* resilience. The main reason for this is the characterization provided by Theorem 1, which we can leverage to obtain provable solutions to problems P1–P3. An equivalent characterization for resilience appears harder to obtain. While the previous example shows that resilience can be strictly weaker than strong resilience, Corollary 2 shows that the two notions are interchangeable when the number of edges used (which can be viewed as resources deployed by the designer) is to be minimized.

B. Solution to Problem P1

In this section, we show how to determine the degree of strong resilience of a bipartite graph $G(n,m)$ for $n \leq m$, i.e., we provide a solution to Problem P1. The solution is constructive, in the sense that we also exhibit a set of edges, which is a union of disjoint left-perfect matchings and can be obtained in polynomial time. This is done by translating the problem into a max-flow problem and appealing to Theorem 1.

We start with the following definition, which takes a bipartite graph G and a nonnegative integer ℓ and produces a directed version of G, denoted by $\bar{G}$, and a capacity function defined on the edge set of $\bar{G}$:

Definition 5: Let $G(n,m) = (V_\alpha \cup V_\beta, E)$ be a bipartite graph and $\ell \geq 0$ be an integer. Define the digraph $\bar{G}(n,m) = (\bar{V}, \bar{E})$ and the capacity function $\bar{c}_\ell : \bar{E} \to \mathbb{Z}_{\geq 0}$ as follows:

1) Add two new nodes to G, denoted by s and t:
$$\bar{V} := V_\alpha \cup V_\beta \cup \{s, t\}.$$
2) Create the edge set $\bar{E}$ as a union of $\bar{E}_0$ and $\bar{E}_1$ where
$$\bar{E}_0 := \{s\alpha_i, \beta_j t \mid \alpha_i \in V_\alpha, \beta_j \in V_\beta\},$$
$$\bar{E}_1 := \{\alpha_i \beta_j \mid (\alpha_i, \beta_j) \in E\}.$$
3) Define $\bar{c}_\ell$ as follows:
$$\bar{c}_\ell(e) := \begin{cases} \ell & \text{if } e \in \bar{E}_0, \\ 1 & \text{if } e \in \bar{E}_1. \end{cases} \tag{3}$$

The two new nodes s and t added in step 1 are the source and the target of $\bar{G}$, respectively. The value of ℓ will be problem-dependent and specified as follows. We illustrate the definition in Fig. 5.

Denote by F_ℓ the set of integer-valued flow maps on $\bar{G}$ with respect to $\bar{c}_\ell$. When $\ell = 0$, F_ℓ is the singleton $\{f\}$, where $f(e) = 0$ for all $e \in \bar{E}$.

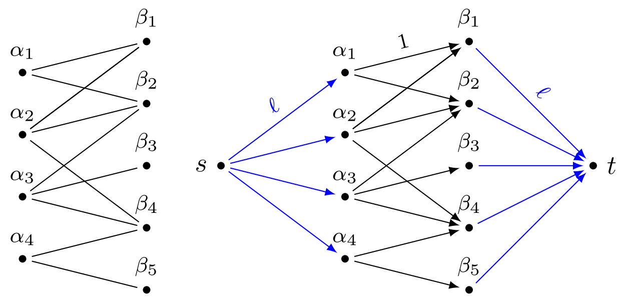


Fig. 5. Given a bipartite graph $G(4,5)$ in (a), we plot the digraph $\bar{G}(4,5)$ in (b). The edge set $\bar{E}$ of $\bar{G}(4,5)$ is partitioned into two subsets $\bar{E}_0$ and $\bar{E}_1$ depicted in blue and black, respectively. The capacity of each blue (resp. black) edge is ℓ (resp. 1).

Given a flow $f \in F_\ell$, we define the subgraph of the original bipartite graph G *induced by* f as follows:

$$G_f := (V_\alpha \cup V_\beta, E_f) \text{ with}$$
$$E_f := \{(\alpha_i, \beta_j) \in E \mid f(\alpha_i\beta_j) \neq 0\}. \quad (4)$$

In words, we select only edges of G whose directed versions in $\bar{G}$ are used by the flow f.

Recall that for a flow $f \in F_\ell$, its value $|f|$ is given by (2). We need the following definition.

Definition 6 (Saturated flows): Given the digraph $\bar{G}(n,m)$ and a nonnegative integer ℓ, we say that a flow $f \in F_\ell$ on $\bar{G}(n,m)$ is *saturated* if $|f| = n\ell$. We denote by $\bar{F}_\ell$ the set of saturated flows on $\bar{G}(n,m)$.

Note that a saturated flow is necessarily a max-flow because, by Definition 5, the value of a flow f cannot exceed $n\ell$. Further, note that by the integrality theorem (see Section II-A), if a real-valued flow f with $|f| = n\ell$ exists, then $\bar{F}_\ell$ is nonempty. The following lemma shows that saturated flows are in correspondence with disjoint left-perfect matchings.

Lemma 6: Let $\ell \geq 1$ and $f \in F_\ell$. Then, $f \in \bar{F}_\ell$ if and only if G_f is a union of ℓ disjoint left-perfect matchings.

Proof: Assuming that $f \in F_\ell$ and $G_f = (V_\alpha \cup V_\beta, E_f)$ is a disjoint union of ℓ left-perfect matchings, we show that $f \in \bar{F}_\ell$. It should be clear that $\deg(\alpha_i; G_f) = \ell$ for each $\alpha_i \in V_\alpha$. Hence, there exist $1 \leq i_1, \ldots, i_\ell \leq \ell$ so that $(\alpha_i, \beta_{i_j}) \in E_f$, for $1 \leq i \leq n$ and $1 \leq j \leq \ell$. From the definition of G_f and the fact that f is integer-valued, we have that $f(\alpha_i\beta_{i_j}) \geq 1$. On the one hand, using (1) and (2), we obtain that

$$|f| = \sum_{i=1}^{n} f(s\alpha_i) = \sum_{(\alpha_i,\beta_{i_j}) \in E_f} f(\alpha_i\beta_{i_j}) \geq n\ell.$$

On the other hand, since $f \in F_\ell$, $|f| \leq n\ell$. Thus, we must have that $|f| = n\ell$ and, hence, $f \in \bar{F}_\ell$.

Reciprocally, assuming that $f \in \bar{F}_\ell$, we show that G_f is a disjoint union of ℓ left-perfect matchings. To this end, we claim that the degree of each left node in G_f is exactly ℓ and the degree of each right node in G_f is less than or equal to ℓ. If this holds, then the result is an immediate consequence of Theorem 1. We now prove the claim. For the left nodes, because f is saturated, $n\ell = |f| = \sum_{i=1}^{n} f(s\alpha_i)$. By the definition of the capacity function (3), $f(s\alpha_i) \leq \ell$. It follows that $f(s\alpha_i) = \ell$ for all $i = 1, \ldots, n$. Next, by the balance condition and the definition of G_f in (4), we have that

$$f(s\alpha_i) = \sum_{j:(\alpha_i,\beta_j)\in E} f(\alpha_i\beta_j) = \sum_{j:(\alpha_i,\beta_j)\in E_f} f(\alpha_i\beta_j).$$

Further, by the capacity function (3) and the fact that f is integer-valued, we have that $f(\alpha_i\beta_j) = 1$ for $(\alpha_i, \beta_j) \in E_f$, and thus, there are exactly n edges in E_f incident to α_i. Finally, for the right nodes, one can apply similar arguments: first, from the capacity function, we have that $f(\beta_j t) \leq \ell$; then, the balance condition $f(\beta_j t) = \sum_{i:(\alpha_i,\beta_j)\in E_f} f(\alpha_i\beta_j)$ implies that $\deg(\beta_j; G_f) \leq \ell$. ∎

With the above preliminaries, we now provide a solution to Problem P1.

Theorem 3: Let $G(n,m)$ be a bipartite graph with $m \geq n$. Let $\bar{G}(n,m)$ be the digraph from Definition 5 and $\bar{F}_\ell$ be given as in Definition 6. Let $\ell^* := \max\{\ell \geq 0 \mid \bar{F}_\ell \neq \varnothing\}$. Then, $0 \leq \ell^* \leq m$ and the following hold.

1) For any $\ell \in \{0, \ldots, \ell^*\}$, $\bar{F}_\ell \neq \varnothing$. For any $f \in \bar{F}_\ell$, the bipartite graph $G_f(n,m)$ given in (4) is a union of ℓ disjoint left-perfect matchings.
2) The degree of strong resilience of $G(n,m)$ is $(\ell^* - 1)$.

Proof: Recall that when $\ell = 0$, $\bar{F}_0$ is a singleton, so $\ell^* \geq 0$. We next show that $\ell^* \leq m$. Suppose to the contrary that $\ell^* > m$; then, by Lemma 6, G contains at least $(m+1)$ disjoint left-perfect matchings, which contradicts item 2) of Corollary 2 saying that the degree of strong-resilience of $G(n,m)$ is at most $(m-1)$ (and, hence, $G(n,m)$ can contain at most m disjoint left-perfect matchings). We now establish the two conditions of the theorem.

Proof of item 1): Because $\bar{F}_{\ell^*}$ is nonempty, by Lemma 6, G contains ℓ^* disjoint left-perfect matchings, denoted by $P_1, \ldots, P_{\ell^*}$. Let $G' = (V_\alpha \cup V_\beta, E')$ be the subgraph of G induced by $P_1, \ldots, P_\ell$. We use G' to define a flow f on $\bar{G}$ as follows:

$$f(e) := \begin{cases} \ell & \text{if } e = s\alpha_i \text{ for } \alpha_i \in V_\alpha, \\ \deg(\beta_j; G') & \text{if } e = \beta_j t, \\ 1 & \text{if } e = \alpha_i\beta_j \text{ for } (\alpha_i, \beta_j) \in E', \\ 0 & \text{otherwise.} \end{cases}$$

It should be clear that $f \in \bar{F}_\ell$. Using again Lemma 6, we have that for any $f \in \bar{F}_\ell$, G_f is a union of ℓ disjoint left-perfect matchings.

Proof of item 2): First, we consider the case $\ell^* = 0$ and show that G does not have a left-perfect matching (i.e., $\text{s-rsl}\, G = -1$). Suppose, to the contrary, that there exists a left-perfect matching P in G; then, consider the flow f on $\bar{G}$ defined as follows:

$$f(e) := \begin{cases} 0 & \text{if } e = \alpha_i\beta_j \text{ with } (\alpha_i, \beta_j) \notin P, \\ 1 & \text{otherwise.} \end{cases}$$

It is not hard to see that f is a flow on $\bar{G}$ with respect to $\bar{c}_1$ (i.e., $\ell = 1$). By construction, f is a saturated flow in $\bar{F}_1$, which is a

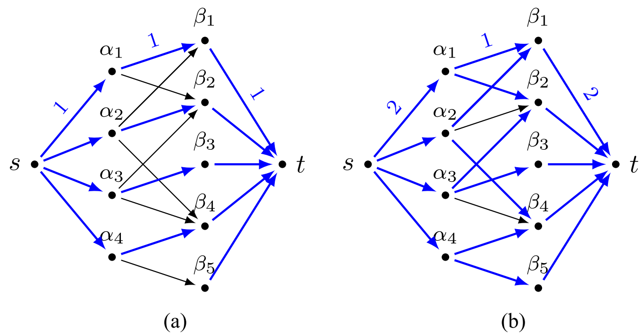


Fig. 6. We show saturated flows f_ℓ on the graph $\bar{G}(4,5)$ as depicted in Fig. 5 for (a) $\ell = 1$ and (b) $\ell = 2$. The edges with nonzero values under f_ℓ are highlighted in blue. In the case $\ell = 2$, the two disjoint left-perfect matchings are $\{(\alpha_1,\beta_1),(\alpha_2,\beta_4),(\alpha_3,\beta_2),(\alpha_4,\beta_5)\}$ and $\{(\alpha_1,\beta_2),(\alpha_2,\beta_1),(\alpha_3,\beta_3),(\alpha_4,\beta_4)\}$. Note, in particular, that the perfect matching of case $\ell = 1$ is not part of the disjoint matchings for $\ell = 2$.

contradiction. For the other case where $\ell^* \geq 1$, the item follows from the definition of ℓ^* and Lemma 6. ■

Example 2: Consider the bipartite graph $G(4,5)$ given in Fig. 5. We run, e.g., the Ford–Fulkerson algorithm for the weighted digraph $\bar{G}(4,5)$ with $\ell = 1, 2$. We show in Fig. 6 the corresponding saturated flows, which implies that $\bar{F}_\ell \neq \varnothing$ for $\ell = 1, 2$. Also, note that $\bar{F}_3 = \varnothing$ because the degrees of nodes α_1 and α_4 in $G(4,5)$ are both 2. Using Theorem 3, we have that $G(4,5)$ is strongly 1-resilient. ■

Theorem 3 provides an algorithmic solution, of polynomial-time complexity, to P1, i.e., to determine $\text{s-rsl}\, G(n,m)$. The algorithm is as follows: Start by setting $\ell := m$, and repeat the following procedure.

1) Construct the digraph $\bar{G}(n,m)$ and the capacity function $\bar{c}_\ell$. The complexity is $O(m+n)$.
2) Run the Ford–Fulkerson algorithm on $\bar{G}(n,m)$ initialized at the zero flow and denote its output by f. The complexity is $O(n^2 m\ell)$ [35]. If $|f| = n\ell$, then return $\text{s-rsl}\, G(n,m) = \ell - 1$. The algorithm is over.
3) If $|f| < n\ell$ and if $\ell \geq 2$, decrease the value of ℓ by 1 and return to step 1. Otherwise, return $\text{s-rsl}\, G(n,m) = -1$ and the algorithm is over.

C. Minimal Number of Edges to Increase $\text{s-rsl}\, G$

In this section, we address the following simple question: Given a graph $G(n,m)$ which is a union of k disjoint left-perfect matchings with $k \leq m-1$, how many edges need to be added to this graph to obtain a bipartite graph $G^*(n,m)$ which is a union of $(k+1)$ disjoint left-perfect matchings? Understanding this problem provides a solution to Problems P2 and, partially, P3 for the special case where G is a union of disjoint left-perfect matchings. The advantage of the solution proposed here, when compared to the algorithms provided in the next section for solving general cases, is that it allows to establish an analytical bound on the number of edges needed to increase the degree of strong resilience.

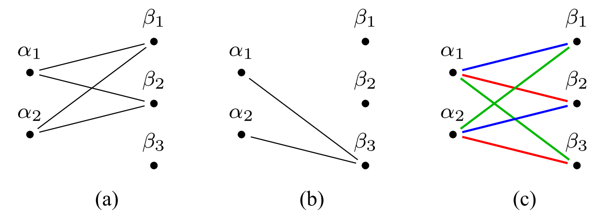


Fig. 7. (a) We depict a graph $G(2,3)$ which is the union of two disjoint left-perfect matchings $P_1 = \{(\alpha_1,\beta_1),(\alpha_2,\beta_2)\}$ and $P_2 = \{(\alpha_1,\beta_2),(\alpha_2,\beta_1)\}$. (b) We depict the complementary graph $G^c(2,3)$; it does not contain a left-perfect matching. (c) We plot the three disjoint left-perfect matchings in the complete bipartite graph $K(2,3) = G(2,3) \cup G^c(2,3)$.

To proceed, we introduce the natural notion of the *graph complement*. Given the *complete bipartite graph* $K = (V_\alpha \cup V_\beta, E_K)$ and the bipartite graph $G = (V_\alpha \cup V_\beta, E)$, we denote by G^c the complement of G (in K); more precisely

$$G^c := (V_\alpha \cup V_\beta, E_K - E).$$

Special case $m = n$: We have the following result.

Lemma 7: Let $G(n,n)$ be a union of k disjoint perfect matchings with $k \leq n$. Let $G^c(n,n)$ be the complement of $G(n,n)$. Then, $G^c(n,n)$ is a union of $(n-k)$ disjoint perfect matchings.

Proof: Since $m = n$, by Theorem 1, the degree of every node in $G(n,n)$ is k. It then follows that the degree of each node in $G^c(n,n)$ is $(n-k)$. Using Theorem 1 again, we conclude that $G^c(n,n)$ is a union of $(n-k)$ disjoint perfect matchings. ■

It should be clear that adding *any* perfect matching of $G^c(n,n)$ to $G(n,n)$ yields a graph $G^*(n,n)$, which is a union of $(k+1)$ disjoint perfect matchings. However, such a fact cannot be extended to the case $m > n$, as seen in the following example.

Example 3: To see this, consider the graph $G(2,3)$ in Fig. 7, which depicts a simple case for which $m > n$. Here, $G(2,3)$ is the union of two disjoint left-perfect matchings. It is easy to see that $G^c(2,3)$ does not contain a left-perfect matching. Nevertheless, adding $G^c(2,3)$ to $G(2,3)$ still yields the graph $K(2,3)$, which is a disjoint union of three left-perfect matchings. The key difference between this case and the one with $n = m$ is that in the latter case, one can always produce a graph $G^*(n,n)$, which is composed of the all of the existing k disjoint perfect matchings of $G(n,n)$ and an additional disjoint perfect matching. In this example $G(2,3)$, the three disjoint left-perfect matchings of $G^*(2,3)$ do not contain all of the left-perfect matchings that were used to express $G(2,3)$ as a disjoint union of perfect matchings. Generally speaking, this fact precludes the use of simple inductive arguments that rely on adding n edges while keeping the k disjoint perfect matchings that made $G(n,m)$. ■

General case $m \geq n$: We establish the following result, the proof of which will be *constructive*.

Theorem 4: Let $G(n,m)$, with $n \leq m$, be a union of k disjoint left-perfect matchings for $k < m$. Then, one can add ℓn edges, for $1 \leq \ell \leq m-k$, to $G(n,m)$ such that the resulting graph $G^*(n,m)$ is a union of $(k+\ell)$ disjoint left-perfect matchings.

The next result is then an immediate consequence of Theorem 4.

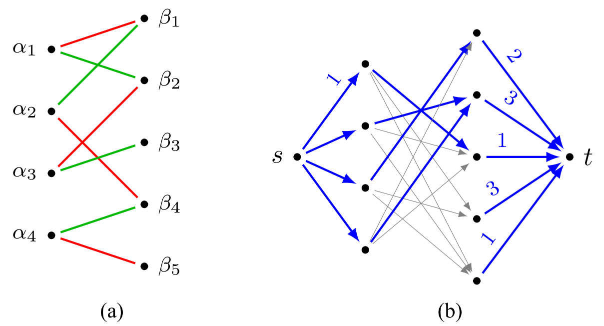


Fig. 8. Bipartite graph $G(4,5)$ in (a) is a union of two disjoint left-perfect matchings, highlighted in red and green. The weighted digraph in (b) is $\hat{G}^c(4,5)$ for $k=3$. Correspondingly, by Definition 7, the capacity of an edge $\beta_j t$ is given by $4-\deg(\beta_j;G)$, whereas the capacity of each remaining edge is 1. We then run the Ford–Fulkerson algorithm and highlight (in blue) a solution f to the max-flow problem (6). By Prop. 7, the capacity of any such solution is given by $|f|=n=4$.

Corollary 5: Given a strongly k-resilient $G(n,m)$, with $k<m$, and given a budget of p additional edges, one can select p edges $\{e_1,\ldots,e_p\}$ out of $G^c(n,m)$ such that the new graph $G(n,m)\cup\{e_1,\ldots,e_p\}$ is at least strongly $(k+\lfloor\frac{p}{n}\rfloor)$-resilient.

The remainder of this section is devoted to the proof of Theorem 4. It suffices to prove the theorem for the case $\ell=1$; one can then iteratively apply this case to prove the general result. The proof has two parts: The first part relates the feasibility of the addition problem (i.e., the problem of adding n edges to $G(n,m)$ to form a union of $(k+1)$ disjoint left-perfect matchings) to a max-flow problem; this is akin to what was done in Section III-B. Here, we define a max-flow problem whose capacity function allows us to decide whether the addition problem is feasible. Then, in the second part, relying on the max-flow min-cut theorem, we compute explicitly the maximal capacity by computing the corresponding minimal cut.

Max-flow formulation: We start by constructing another directed version of the bipartite graph $\hat{G}(n,m)$ with an appropriate capacity function. The solution of a newly defined max-flow problem on this graph will yield the edges needed to increase the resilience.

Definition 7: Given a bipartite graph $G=(V_\alpha\cup V_\beta,E)$ and an integer k, define the digraph $\hat{G}^c=(\hat{V},\hat{E})$ and the capacity function $\hat{c}_k:\hat{E}\to\mathbb{Z}_{\geq 0}$ as follows.

1) Add two new nodes to G, denoted by s and t:

$$\hat{V}:=V_\alpha\cup V_\beta\cup\{s,t\}.$$

2) Create the edge set $\hat{E}$ as a union of $\hat{E}_0$ and $\hat{E}_1$ where

$$\hat{E}_0:=\{s\alpha_i\mid\alpha_i\in V_\alpha\}\cup\{\alpha_i\beta_j\mid(\alpha_i,\beta_j)\notin E\},$$

$$\hat{E}_1:=\{\beta_j t\mid\beta_j\in V_\beta\}.$$

3) If $e\in\hat{E}_0$, then $\hat{c}_k(e):=1$; if $e=\beta_j t\in\hat{E}_1$, then

$$\hat{c}_k(e):=\max\{0,(k+1)-\deg(\beta_j;G)\}. \quad (5)$$

We illustrate the definition in Fig. 8.

Let $\hat{F}_k$ be the set of integer-valued flow maps on $\hat{G}^c(n,m)$. We will now relate the max-flow problem on $\hat{G}^c(n,m)$:

$$\max_{f\in\hat{F}_k}|f| \quad (6)$$

to Theorem 4. As mentioned above, requiring an integer solution is not constraining; it suffices to use the Ford–Fulkerson algorithm.

Proposition 6: Let $G(n,m)$ be a union of k disjoint left-perfect matchings, for $0\leq k<m$, and $\hat{G}^c(n,m)$ be given in Definition 7. Let f be a solution to problem (6). If $|f|=n$, then there exist n edges $\{e_1,\ldots,e_n\}\in G^c(n,m)$ so that $G^*(n,m):=G(n,m)\cup\{e_1,\ldots,e_n\}$ is a union of $(k+1)$ disjoint left-perfect matchings.

Proof: Given the flow f on $\hat{G}^c$, we let G^c_f be the subgraph of G^c induced by f as defined in (4).

Because $|f|=n$ and because the capacity assigned to the edges $s\alpha_i$, for $\alpha_i\in V_\alpha$, is 1, the inflow at every node α_i is also 1. Also, since the capacities of edges of type $\alpha_i\beta_j$ are 1, we have that there are exactly n edges of this type for which f is nonzero, and thus, there are exactly n edges in G^c_f. By construction, these edges are incident to n *distinct* left nodes (as otherwise, it implies that an edge of type $s\alpha_i$ has a flow above its capacity of 1). Denote by $\{e_1,\ldots,e_n\}$ this set of edges in G^c_f.

We show that adding this set of edges to G yields a G^*, which is a union of $(k+1)$ disjoint left-perfect matchings. We do so by verifying that G^* satisfies the two items in Theorem 1.

1) $\deg(\alpha_i;G^*)=k+1$, for all $\alpha_i\in V_\alpha$. This holds because of the following three facts. First, by assumption, $\deg(\alpha_i,G)=k$. Next, note that G and G^c_f have disjoint sets of edges. Finally, the edges $e_1,\ldots,e_n$ in G^c_f are incident to n distinct left nodes.
2) $\deg(\beta_j;G^*)\leq k+1$ for all $\beta_j\in V_\beta$. This holds because of the following three facts. First, by assumption, $\deg(\beta_j;G)\leq k$ for $\beta_j\in V_\beta$. Second, recalling that the capacities of the edges in $\hat{E}_1$ are given in (5), we have that for each right node β_j,

$$\deg(\beta_j;G^c_f)\leq(k+1)-\deg(\beta_j;G).$$

Finally, because G^* is the disjoint union of G and G^c_f, $\deg(\beta_j;G^*)=\deg(\beta_j;G)+\deg(\beta_j;G^c_f)\leq(k+1)$.

We have, thus, shown that the two items of Theorem 1 are satisfied by G^*. ■

Equipped with the above Proposition, Theorem 4 is easily seen to be equivalent to the following result.

Proposition 7: Let $G(n,m)$ be a union of k disjoint left-perfect matchings, for $0\leq k<m$, and $\hat{G}^c(n,m)$ be as in Definition 7. Let f be a solution to the max-flow problem (6). Then, $|f|=n$.

Proof: To prove the result, we rely on the use of the max-flow min-cut theorem (see Lemma 2), which applied here reduces the problem to showing that for every cut (S,T) in $\hat{G}^c$, its capacity $c(S,T)\geq n$ and, furthermore, this lower bound is realizable.

For a given cut (S,T) in $\hat{G}^c$, we let $S_\alpha:=S\cap V_\alpha$ and $T_\alpha:=T\cap V_\alpha$ be the sets of left nodes contained in S and T, respectively. Similarly, we define $S_\beta:=S\cap V_\beta$ and $T_\beta:=T\cap V_\beta$.

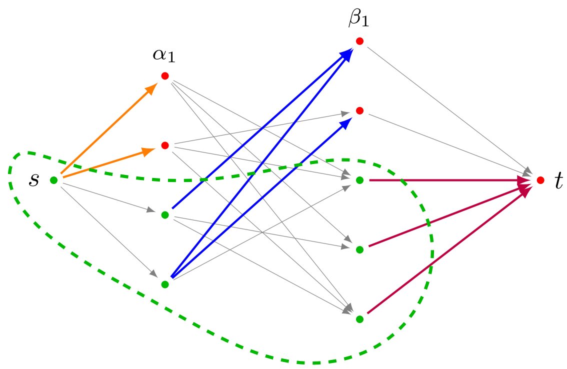


Fig. 9. We illustrate the three terms defined in (8). In this digraph, we let the cut (S,T) be such that the nodes depicted in green (resp. red) are nodes in S (resp. T). The set S is circled by the dashed green line. Then, the term $c(s,T_\alpha)$ is the sum of the capacities of the edges depicted in orange, the term $c(S_\alpha,T_\beta)$ is the sum of the capacities of the edges depicted in blue, and the term $c(S_\beta,t)$ is the sum of the capacities of the edges depicted in purple. The capacity of the cut (S,T) is easily seen to be sum of these three terms.

Let $p := |T_\alpha|$ and $q := |S_\beta|$. For every such cut, we can write its capacity into the sum of three terms

$$c(S,T) = c(s,T_\alpha) + c(S_\alpha,T_\beta) + c(S_\beta,t) \tag{7}$$

where the three terms are given by

$$\begin{cases} c(s,T_\alpha) & := & \sum_{\alpha_i\in T_\alpha} c(s\alpha_i), \\ c(S_\alpha,T_\beta) & := & \sum_{\alpha_i\in S_\alpha,\beta_j\in T_\beta} c(\alpha_i\beta_j), \\ c(S_\beta,t) & := & \sum_{\beta_j\in S_\beta} c(\beta_j t). \end{cases} \tag{8}$$

We evaluate these three terms (also, see Fig. 9 for an illustration) as follows.

First term $c(s,T_\alpha)$: Note that by item 3) of Definition 7, $\hat{c}_k(s\alpha_i)=1$, for $\alpha_i\in T_\alpha$, so

$$c(s,T_\alpha) = p. \tag{9}$$

Second term $c(S_\alpha,T_\beta)$: We first establish the following inequality:

$$c(S_\alpha,T_\beta) \geq \sum_{\alpha_i\in S_\alpha} \deg(\alpha_i;G^c) - \sum_{\beta_j\in S_\beta} \deg(\beta_j;G^c). \tag{10}$$

To see it holds, first note that the total number of outgoing edges incident to the nodes $\alpha_i\in S_\alpha$ is exactly given by $\sum_{\alpha_i\in S_\alpha}\deg(\alpha_i;G^c)$. Every such outgoing edge is necessarily incident to either a node in S_β or a node in T_β. Furthermore, the number of incoming edges incident to nodes $\beta_j\in S_\beta$ is given by $\sum_{\beta_j\in S_\beta}\deg(\beta_j;G^c)$. Similarly, every such incoming edge can be incident to either a node in S_α or a node in T_α. It then follows that the number of edges incident to both S_α and T_β in $\hat{G}^c$ is bounded below by the expression on the right-hand side of (10). Because $\hat{c}_k(\alpha_i\beta_j)=1$, the inequality (10) holds.

Now, we evaluate the two sums on the right-hand side of (10). For the first sum, since G is a union of k disjoint left-perfect matchings, we have that $\deg(\alpha_i;G^c)=(m-k)$ for all $i=1,\ldots,n$. Further, since $|S_\alpha| = n-|T_\alpha| = n-p$,

$$\sum_{\alpha_i\in S_\alpha} \deg(\alpha_i;G^c) = (m-k)(n-p). \tag{11}$$

For the second sum, since the degree of each node β_j in G^c is $n-\deg(\beta_j;G)$ and since $|S_\beta|=q$,

$$\sum_{\beta_j\in S_\beta} \deg(\beta_j;G^c) = qn - \sum_{\beta_j\in S_\beta} \deg(\beta_j;G). \tag{12}$$

Plugging (11) and (12) into (10), we obtain that

$$c(S_\alpha,T_\beta) \geq (m-k)(n-p) - qn + \sum_{\beta_j\in S_\beta} \deg(\beta_j;G). \tag{13}$$

Third term $c(S_\beta,t)$: From (5) and the fact that $|S_\beta|=q$,

$$c(S_\beta,t) = (k+1)q - \sum_{\beta_j\in S_\beta} \deg(\beta_j;G). \tag{14}$$

We now use the facts just established to show that $c(S,T)\geq n$. Specifically, we use (7)–(9) and (13)–(14) to obtain that

$$\begin{aligned} c(S,T) &\geq p + (m-k)(n-p) - qn + (k+1)q \\ &\geq (p+q) + (m-k)(n-p) - (n-k)q \\ &\geq (p+q) + (m-k)(n-p-q) \\ &\geq (p+q) + (n-p-q) \\ &\geq n. \end{aligned}$$

To obtain the second line from the first, we simply rearrange terms. To obtain the third line from the second, we use the fact that $m\geq n$, and using furthermore the assumption that $m>k$, we obtain the fourth line from the third.

Finally, note that if we let $S:=\{s\}$ and $T:=\hat{V}-\{s\}$, then $c(S,T)=n$. ∎

At the end of this section, we conclude that given a graph $G(n,m)$, which is a union of k disjoint left-perfect matchings, and $1\leq\ell\leq m-k$, one can obtain a $G^*(n,m)\succ G(n,m)$, which is a union of $(k+\ell)$ disjoint left-perfect matchings using the following algorithm in polynomial-time: Start by setting $\ell'=0$ and $G'=G$; while $\ell'<\ell$, repeat the following steps.

1) Construct $\hat{G}'^c$ and $\hat{c}_{k+\ell'}$ given by Definition 7. The complexity is $O(m+n)$.
2) Run the Ford–Fulkerson algorithm on $\hat{G}'^c$ and denote by f the output. The complexity is $O(n^2(m-k))$.
3) Update G' to be the union of the current G' and G^c_f (note that G' and G^c_f are edgewise disjoint) and increase ℓ' by 1. The complexity is $O(n)$.

D. Solutions to Problems P2 and P3

In this section, we let $G(n,m)$ be an arbitrary bipartite graph, with $m\geq n$ as above, and $G^c(n,m)$ be its complement in the complete bipartite graph $K(n,m)$.

Recall that for Problem P2, we aim to find a set of edges in G^c of least cardinality which, when added to G, yields a graph which is (strongly) k-resilient, and for Problem P3, given a budget of p edges and a graph G, we aim to maximize the degree of (strong) resilience by optimally choosing these p additional edges.

We provide complete solutions to the two problems for *strong* resilience in the following, together with a polynomial-time algorithm that fulfills the respective goals.

Fair matchings and fair b-matchings: One of the major hurdles in adding edges to G to increase the number of left-perfect matchings is that one has the option to use edges that already exist in G to create said additional matchings. The use of these existing edges should of course be prioritized as much as possible over the addition of new edges. We can recast this problem by considering the embedding of G into the complete bipartite graph $K = (V_\alpha, V_\beta, E_K)$. This embedding allows us to view both Problems P2 and P3, which are dual to each other, as the problem of selecting edges in K to obtain a desired number of disjoint left-perfect matchings *while maximizing* the use of edges that belong to G. Moreover, this point of view will allow us to appeal to algorithms that obtain such matchings and, thus, solve the above-mentioned problem in polynomial time.

To proceed, we rely on the notion of fair matching and, more specifically, *fair b-matching* in a bipartite graph. Such matchings are described in relation to the following additional structures on a graph.

1) A *capacity function* μ at the *nodes*, which is a positive-integer valued function $\mu : V_\alpha \cup V_\beta \to \mathbb{Z}_{\geq 0}$ which provides an upper bound on the degrees of the nodes in a b-matching.
2) A *priority order* for the possible neighbors of each node. Assuming that there are r different priorities, we label them as $1, \ldots, r$. The priority order indicates which edges of G are preferred to appear in the matching.

For our purpose, we only need to consider a particular class of fair b-matching problems: 1) Elements of that class are defined on the complete bipartite graph K; 2) the capacity functions μ are constant functions with value equal to $(k^* + 1)$, where k^* is the target degree of strong-resilience; and 3) the priority order has $r = 2$ classes, and is induced by G in the sense that a node α_i (resp. β_j) prefers β_j (resp. α_i) if (α_i, β_j) is an edge in G. We refer the reader to [36] and [37] for a general introduction to b-matchings.

Formally, we introduce the following definition of b-matching and fair b-matching considered in this article.

Definition 8 (b-matching and fair b-matching): Let $K = (V_\alpha \cup V_\beta, E_K)$ be the complete bipartite graph and $G = (V_\alpha \cup V_\beta, E)$ be a subgraph of K. A *b-matching* is a subset $P \subseteq E_K$ for which each vertex $v \in V_\alpha \cup V_\beta$ is incident to at most $(k^* + 1)$ edges of P. A *fair b-matching* is a b-matching of *maximal cardinality* so that $|P \cap E|$ is maximized.

We make the following observation.

Lemma 8: The subset P is a b-matching of maximal cardinality if and only if it is a disjoint union of $(k^* + 1)$ disjoint left-perfect matchings.

Proof: First, it should be clear that if P is a b-matching, then by the capacity condition in Definition 8, $|P| \leq (k^* + 1)n$. Next, let P be an arbitrary union of $(k^* + 1)$ disjoint left-perfect matchings. Then, P satisfies the capacity condition and $|P| = (k^* + 1)n$. Thus, such a P is a b-matching of maximal cardinality.

Now, let P be a b-matching of maximal cardinality and $G = (V_\alpha \cup V_\beta, P)$. Suppose that P is *not* a union of $(k^* + 1)$ disjoint left-perfect matchings; then, by Theorem 1 and the capacity condition in Definition 8, there exists at least one node $\alpha_i \in V_\alpha$ such that

$$\deg(\alpha_i; G) < k^* + 1. \tag{15}$$

To see this, note that a graph G induced by a b-matching always satisfies item 2) of Theorem 1; hence, if G is not a union of $(k^* + 1)$ disjoint left-perfect matchings, then item 1) cannot be met, which implies that $\deg(\alpha_i; G) < k^* + 1$ for some α_i. On the one hand, as a consequence of (15), the cardinality of P is strictly less than $(k^* + 1)n$. On the other hand, by the arguments at the beginning of the proof, if we let P' be an arbitrary union of $(k^* + 1)$ disjoint left-perfect matchings, then P' is a b-matching with $|P'| = (k^* + 1)n > |P|$, which is a contradiction. ■

If $P^* \subseteq E_K$ is a b-matching of maximal cardinality, a *fair* b-matching can be obtained by first finding all b-matchings of cardinality $|P^*|$ and, amongst those, selecting one which maximizes $|P^* \cap E|$. It is known that finding a fair b-matching in $K(n, m)$ can be done in polynomial time. To be more precise, if we let $N := m + n$ be the number of nodes of $K(n, m)$ and $M := mn$ be the number of edges in $K(n, m)$, there exist algorithms solving fair b-matching problems in $O(NM \log(N^2/M) \log(N))$ time, using $O(M)$ space [37].

Solution to Problem P2 for strong resilience: We now reduce Problem P2 to the fair b-matching problem. Let k^* be the target degree of strong resilience. If $\text{s-rsl}\, G \geq k^*$, then no additional edge is needed and we are done. Otherwise, we have the following result.

Theorem 8: Let $G(n, m) = (V_\alpha \cup V_\beta, E)$ be a bipartite graph with $m \geq n$ and $\text{s-rsl}\, G < k^*$ with $0 \leq k^* \leq (m - 1)$. Let P^* be a solution to the fair b-matching problem of Definition 8. Then, the following hold.

1) The graph $G^*(n, m) := (V_\alpha \cup V_\beta, E \cup P^*)$ is strongly k^*-resilient.
2) The minimal number of edges out of $G^c(n, m)$ one needs to add to $G(n, m)$ to obtain a strongly k^*-resilient graph $G^*(n, m)$ is given by

$$\delta^* := |P^*| - |P^* \cap E|. \tag{16}$$

Note that for a given graph $G(n, m)$, δ^* depends only on the number k^* (in particular, it does *not* depend on the choice of P^* from Definition 8). If necessary, we will write explicitly $\delta^*(k)$ to indicate such dependence.

By item 1, we have that $G^* \succ G$; by item 2, G^* contains the least number of additional edges so as to be strongly k^*-resilient. Thus, Problem P2 is indeed solved for strong k^*-resilience.

Proof of Theorem 8: We establish the two items as follows.

Proof of item 1: We show that G^* contains exactly $(k^* + 1)$ disjoint left-perfect matchings. By Lemma 8, P^* is a union of $(k^* + 1)$ disjoint left-perfect matchings. Since the edge set of G^* contains P^*, G^* contains at least $(k^* + 1)$ disjoint left-perfect matchings. Now, suppose, to the contrary, that G^* contains (exactly) $\bar{k}$ disjoint left-perfect matchings, with $\bar{k} \geq (k^* + 2)$; then, we let $\{P_i\}_{i=1}^{\bar{k}}$ be a set of such matchings. Let $\rho_i := |P_i - E|$, i.e., ρ_i is the number of edges in P_i but not in E. It follows that $|E \cap P^*| = |P^*| - \sum_{i=1}^{\bar{k}} \rho_i$. Relabel the P_i, if necessary, so that $\rho_1 \geq \cdots \geq \rho_{\bar{k}}$. Then, ρ_1 has to be positive, since otherwise $\rho_i = 0$ for all $i = 1, \ldots, \bar{k}$, which implies that all these $\bar{k}$ matchings P_i are contained in E, contradicting the assumption that $\text{s-rsl}\, G < k^*$ (the same arguments imply that ρ_2 has to be positive as well). Now, let $P := \cup_{i=2}^{k^*+2} P_i$. Then, by Lemma 8, P is a b-matching and $|P| = |P^*|$. Moreover, $|E \cap P| = |P| - \sum_{i=2}^{k^*+2} \rho_i > |E \cap P^*|$, which contradicts the assumption that P^* is a *fair* b-matching.

Proof of item 2: Let $G^* = (V_\alpha \cup V_\beta, P^*)$ be induced by an arbitrary *fair* b-matching P^*. Then, the cardinality $|P^* \cap E|$ is maximized over all b-matchings P of maximal cardinality and, thus, $|P| - |P \cap E|$ is minimized. This proves item 2 and completes the proof. ■

Solution to Problem P3 for strong resilience: Since Problem P3 is dual to Problem P2, it can similarly be solved via a reduction to the fair b-matching problem. Precisely, we have the following result.

Theorem 9: Let $G(n,m) = (V_\alpha \cup V_\beta, E)$ be a bipartite graph with $m \geq n$ and p be a positive integer. Then, the solution to the following optimization problem (Problem P3 for strong resilience):

$$\max \text{s-rsl}\, G^*(n,m) = (V_\alpha \cup V_\beta, E^*)$$
$$\text{s.t. } G^*(n,m) \succeq G(n,m) \text{ and } |E^*| - |E| = p$$

is given by

$$\max\{k \mid \delta^*(k) \leq p\},$$

where $\delta^*(k)$ is defined in (16).

Proof: It is an immediate consequence of Theorem 8: On the one hand, for any k with $\delta^*(k) \leq p$, one can always add p edges out of G^c to G so that the resulting graph G^* is at least strongly k-resilient. On the other hand, it is clear from the definition of $\delta^*(k)$ that it is infeasible to obtain a graph with strong k-resilience by adding fewer than $\delta^*(k)$ edges to G.

IV. Conclusion

We have addressed in this article the resilience of the structural rank of sparsity patterns. The first step in our approach to solve the problems was to recast them as problems posed for bipartite graphs. We then provided a characterization of bipartite graphs corresponding to sparsity patterns of full rank (see Theorem 1). Based on this characterization, we provided provably correct polynomial-time algorithms to solve three problems dealing with strong resilience of the pattern: Given a sparsity pattern, 1) what is its degree of strong resilience, i.e., how many $\star$-entries can be removed without affecting the structural rank; 2) what is the minimal number of $\star$-entries one needs to add to a pattern so as to reach a target degree of strong resilience; and 3) given that one can add p $\star$-entries to a sparsity-pattern, where to place these entries so as to maximize the degree of strong resilience. As shown in Fig. 4, strong resilience *strictly* implies resilience. The problem of computing the exact degrees of resilience for sparsity patterns will be addressed in the future work.

Appendix
Proof of Lemma 3

Proof: To prove the result, we first introduce a few preliminaries. Given a digraph $\vec{G} = (V, \vec{E})$ on n nodes, we say that $\vec{G}$ admits a *Hamiltonian decomposition* [7] if there is a subgraph $\vec{G}' = (V, \vec{E}')$, with $\vec{E}' \subseteq \vec{E}$, such that $\vec{G}'$ is a *disjoint* union of cycles. To a sparsity pattern $\mathcal{S}(n,n)$, we can associate a digraph $\vec{G} = (V, \vec{E})$ on n nodes $\gamma_1, \ldots, \gamma_n$ as follows: $\gamma_i\gamma_j \in \vec{E}$ if the pair (i,j) belongs to $E(\mathcal{S}(n,n))$. It is well known that $\mathcal{S}(n,n)$ admits a matrix of full rank if and only if $\vec{G}$ admits a Hamiltonian decomposition. Let $G(n,n)$ be the bipartite graph associated with the same sparsity pattern $\mathcal{S}(n,n)$. It is also well known that $G(n,n)$ has a perfect matching if and only if the digraph $\vec{G}$ admits a Hamiltonian decomposition (see [13] for a simple account of this fact).

With the above preliminaries, we now return to establish Lemma 3. First, note that the rank $\mathcal{S}(n,m)$ is n if and only if there exist n columns so that the subpattern induced by these columns is of full rank; precisely, there exists $1 \leq j_1 < \cdots < j_n \leq m$ so that the sparsity pattern $\mathcal{S}'(n,n)$ defined by the index set

$$E(\mathcal{S}') = \{(i, j_k) \in E(\mathcal{S}) \mid 1 \leq k \leq n\}$$

is of rank n. Owing to the preliminaries above, $\mathcal{S}'(n,n)$ is of full rank if and only if the associated digraph $\vec{G}'$ on n nodes admits a Hamiltonian decomposition. Furthermore, the existence of this Hamiltonian decomposition implies that the bipartite graph $G'(n,n)$ corresponding to $\mathcal{S}'(n,n)$ contains a perfect matching.

From the definition of $\mathcal{S}'$, it is not hard to see that $G'(n,n)$ can be realized as a subgraph of $G(n,m)$; more precisely, $G'(n,n)$ is the subgraph of $G(n,m)$ induced by the nodes $\alpha_i \in V_\alpha$ and nodes $\beta_{j_1}, \ldots, \beta_{j_n}$. Thus, a perfect matching in $G'(n,n)$ is mapped using the above inclusion to a left-perfect matching in $G(n,m)$. We, thus, conclude that the rank of $\mathcal{S}(n,m)$ is n if and only if G has a left-perfect matching. ■

References

[1] C. Savaglio, M. Ganzha, M. Paprzycki, C. Badica, M. Ivanovic, and G. Fortino, "Agent-based Internet of things: State-of-the-art and research challenges," *Future Gener. Comput. Syst.*, vol. 102, pp. 1038–1053, 2020.

[2] T. Sanislav, S. Zeadally, G. Mois, and H. Fouchal, "Multi-agent architecture for reliable cyber-physical systems (CPS)," in *Proc. IEEE Symp. Comput. Commun.*, 2017, pp. 170–175.

[3] L. Elefteriadou et al., "Enabling transportation networks with automated vehicles: From individual vehicle motion control to networked fleet management," in *Road Vehicle Automation 7*, G. Meyer and S. Beiker, Eds. Cham, Switzerland: Springer, 2020, pp. 49–62.

[4] S. Zhao and D. Zelazo, "Bearing rigidity theory and its applications for control and estimation of network systems: Life beyond distance rigidity," *IEEE Control Syst. Mag.*, vol. 39, no. 2, pp. 66–83, Apr. 2019.

[5] C.-T. Lin, "Structural controllability," *IEEE Trans. Autom. Control*, vol. AC-19, no. 3, pp. 201–208, Jun. 1974.

[6] J.-M. Dion, C. Commault, and J. Van der Woude, "Generic properties and control of linear structured systems: A survey," *Automatica*, vol. 39, no. 7, pp. 1125–1144, 2003.

[7] M.-A. Belabbas, "Sparse stable systems," *Syst. Control Lett.*, vol. 62, no. 10, pp. 981–987, 2013.

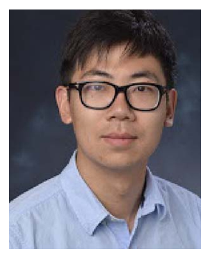
[8] X. Chen, "Sparse linear ensemble systems and structural controllability," *IEEE Trans. Autom. Control*, vol. 67, no. 7, pp. 3337–3348, Jul. 2022.

[9] B. Gharesifard and X. Chen, "Structural averaged controllability of linear ensemble systems," *IEEE Contr. Syst. Lett.*, vol. 6, pp. 518–523, 2022.

[10] A. Kirkoryan and M.-A. Belabbas, "Decentralized stabilization with symmetric topologies," in *Proc. IEEE 53rd Conf. Decis. Control*, 2014, pp. 1347–1352.
[11] Z. Yuan, C. Zhao, Z. Di, W.-X. Wang, and Y.-C. Lai, "Exact controllability of complex networks," *Nature Commun.*, vol. 4, no. 1, 2013, Art. no. 2447.
[12] M. Doostmohammadian and U. A. Khan, "Minimal sufficient conditions for structural observability/controllability of composite networks via Kronecker product," *IEEE Trans. Signal Inf. Process. Netw.*, vol. 6, pp. 78–87, 2020.
[13] M.-A. Belabbas, "Algorithms for sparse stable systems," in *Proc. IEEE 52th Conf. Decis. Control*, 2013, pp. 3457–3462.
[14] M. Sharf and D. Zelazo, "Network feedback passivation of passivity-short multi-agent systems," *IEEE Contr. Syst. Lett.*, vol. 3, no. 3, pp. 607–612, Jul. 2019.
[15] S. Fallat and L. Hogben, "The minimum rank of symmetric matrices described by a graph: A survey," *Linear Algebra Appl.*, vol. 426, pp. 558–582, Oct. 2007.
[16] L. Hogben, "Minimum rank problems," *Linear Algebra Appl.*, vol. 432, no. 8, pp. 1961–1974, 2010.
[17] B. Gutkin and U. Smilansky, "Can one hear the shape of a graph?," *J. Phys. A, Math. Gen.*, vol. 34, pp. 6061–6068, Jul. 2001.
[18] S. M. Meesum and S. Saurabh, "Rank reduction of directed graphs by vertex and edge deletions," in *LATIN: Theoretical Informatics*, E. Kranakis, G. Navarro, and E. Chávez, Eds. Berlin, Germany: Springer, 2016, pp. 619–633.
[19] F. Dörfler and F. Bullo, "Synchronization in complex networks of phase oscillators: A survey," *Automatica*, vol. 50, pp. 1539–1564, 2014.
[20] J. K. Hale, "Diffusive coupling, dissipation, and synchronization," *J. Dyn. Differ. Equ.*, vol. 9, pp. 1–52, Jan. 1997.
[21] M. Arcak, "Passivity as a design tool for group coordination," *IEEE Trans. Autom. Control*, vol. 52, no. 8, pp. 1380–1390, Aug. 2007.
[22] M. Bürger, D. Zelazo, and F. Allgöwer, "Duality and network theory in passivity-based cooperative control," *Automatica*, vol. 50, no. 8, pp. 2051–2061, 2014.
[23] F. Zhu, M. Xia, and P. J. Antsaklis, "Passivity analysis and passivation of feedback systems using passivity indices," in *Proc. Amer. Control Conf.*, 2014, pp. 1833–1838.
[24] M. Sharf, A. Jain, and D. Zelazo, "A geometric method for passivation and cooperative control of equilibrium-independent passivity-short systems," *IEEE Trans. Autom. Control*, vol. 66, no. 12, pp. 5877–5892, Dec. 2021.
[25] M. W. S. Atman, T. Hatanaka, Z. Qu, N. Chopra, J. Yamauchi, and M. Fujita, "Motion synchronization for semi-autonomous robotic swarm with a passivity-short human operator," *Int. J. Intell. Robot. Appl.*, vol. 2, pp. 235–251, Jun. 2018.
[26] C. Kojima and T. Namerikawa, "Passivity-short in networked nonlinear systems," in *Proc. SICE Int. Symp. Control Syst.*, 2019, pp. 450–454.
[27] F. Strehle, A. J. Malan, S. Krebs, and S. Hohmann, "A port-Hamiltonian approach to plug-and-play voltage and frequency control in islanded inverter-based ac microgrids," in *Proc. IEEE Conf. Decis. Control*, 2019, pp. 4648–4655.
[28] Z. Qu and M. A. Simaan, "Modularized design for cooperative control and plug-and-play operation of networked heterogeneous systems," *Automatica*, vol. 50, no. 9, pp. 2405–2414, 2014.
[29] S. Bodenburg and J. Lunze, "Plug-and-play control of interconnected systems with a changing number of subsystems," in *Proc. Eur. Control Conf.*, 2015, pp. 3520–3527.
[30] D. König, "Über graphen und ihre anwendung auf determinantentheorie und mengenlehre," *Mathematische Annalen*, vol. 77, no. 4, pp. 453–465, 1916.
[31] R. T. Rockafellar, *Network Flows and Monotropic Optimization*. Belmont, MA, USA: Athena Scientific, 1998.
[32] L. R. Ford Jr and D. R. Fulkerson, *Flows in Networks*, vol. 54. Princeton, NJ, USA: Princeton Univ. Press, 2015.
[33] A. Brualdi Richard, *Introductory Combinatorics.* London, U.K.: Pearson Educ., 2010.
[34] L. Lovász and M. D. Plummer, *Matching Theory*, vol. 367. Providence, RI, USA: Amer. Math. Soc., 2009.
[35] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*. Cambridge, MA, USA: MIT Press, 2009.
[36] R. P. Anstee, "A polynomial algorithm for b-matchings: An alternative approach," *Inf. Process. Lett.*, vol. 24, no. 3, pp. 153–157, 1987.
[37] C.-C. Huang, T. Kavitha, K. Mehlhorn, and D. Michail, "Fair matchings and related problems," *Algorithmica*, vol. 74, no. 3, pp. 1184–1203, 2016.

Mohamed-Ali Belabbas received the undergraduate degree from École Centrale Paris, Paris, France, and Université Catholique de Louvain, Louvain, Belgium, in 2001, and the Ph.D. degree in applied mathematics from Harvard University, Cambridge, MA, USA, in 2006.

He is currently an Associate Professor with the Electrical and Computer Engineering Department, University of Illinois, Urbana-Champaign, IL, USA, and the Coordinated Science Laboratory. His research interests include networked control system, stochastic control, and geometric control theory.



Xudong Chen received the B.S. degree in electronic engineering from Tsinghua University, Beijing, China, in 2009, and the Ph.D. degree in electrical engineering from Harvard University, Cambridge, MA, USA, in 2014.

He is an Assistant Professor with the Department of Electrical, Computer, and Energy Engineering, University of Colorado Boulder, Boulder, CO, USA. Prior to that, he was a Postdoctoral Fellow with the Coordinated Science Laboratory, University of Illinois, Urbana-Champaign. His current research interests are in the area of control theory, stochastic processes, optimization, game theory, and their applications in modeling, analysis, control, and estimation of large-scale complex systems.

Dr. Chen was a recipient of the 2020 Air Force Young Investigator Program Award, the 2021 NSF Career Award, and the 2021 Donald P. Eckman Award.

Daniel Zelazo (Senior Member, IEEE) received the B.Sc. and M.Eng. degrees in electrical engineering and computer science from the Massachusetts Institute of Technology, Cambridge, MA, USA, in 1999 and 2001, respectively, and the Ph.D. degree in aeronautics and astronautics from the University of Washington, Seattle, WA, USA, in 2009.

From 2010 to 2012, he was a Postdoctoral Research Associate and Lecturer with the Institute for Systems Theory and Automatic Control, University of Stuttgart, Germany. He is an Associate Professor of aerospace engineering with the Technion-Israel Institute of Technology, Haifa, Israel. He is currently an Associate Editor for the *IEEE Control Systems Letters* and Subject Editor for the *International Journal of Robust and Nonlinear Control.* His research interests include topics related to multiagent systems.