Pattern Matching in Protein-Protein Interaction Graphs

Gaëlle Brevier-Giberti, Roméo Rizzi, Stéphane Vialette

Pattern Matching in Protein-Protein Interaction Graphs

Gaëlle Brevier

Laboratoire G-SCOPE
46 avenue Félix Viallet, 38031 Grenoble Cedex - France

Romeo Rizzi

Dipartimento di Matematica ed Informatica (DIMI),
Università di Udine, Via delle Scienze 208, I-33100 Udine, Italy

Stéphane Vialette

IGM-LabInfo, CNRS UMR 8049, Univ. Paris-Est,
5 Bd Descartes 77454 Marne-la-Vallée, France

Abstract

In the context of comparative analysis of protein-protein interaction graphs, we use a graph-based formalism to detect the preservation of a given protein complex (pattern graph) in the protein-protein interaction graph (target graph) of another species with respect to (w.r.t.) orthologous proteins. We give an efficient exponential-time randomized algorithm in case the occurrence of the pattern graph in the target graph is required to be exact. For approximate occurrences, we prove a tight inapproximability results and give four approximation algorithms that deal with bounded degree graphs, small ortholog numbers, linear forests and very simple yet hard instances, respectively.

Email addresses: Gaelle.Giberti@imag.fr (Gaëlle Brevier), Romeo.Rizzi@dimi.uniud.it (Romeo Rizzi), vialette@univ-mlv.fr (Stéphane Vialette).

1 Introduction

High-throughput analysis makes possible the study of protein-protein interactions at a genome-wise scale [GB+02,HG+02,UG+00], and comparative analysis tries to determine the extent to which protein networks are conserved among species. Indeed, mounting evidence suggests that proteins that function together in a pathway or a structural complex are likely to evolve in a correlated fashion, and, during evolution, all such functionally linked proteins tend to be either preserved or eliminated in a new species [PMT+99].

Protein interactions identified on a genome-wide scale are commonly visualized as protein interaction graphs, where proteins are vertices and interactions are edges [TSU04]. Experimentally derived interaction networks can be extremely complex, so that it is a challenging problem to extract biological functions or pathways from them. However, biological systems are hierarchically organized into functional modules. Several methods have been proposed for identifying functional modules in protein-protein interaction graphs. As observed in [PLEO04], cluster analysis is an obvious choice of methodology for the extraction of functional modules from protein interaction networks. Comparative analysis of protein-protein interaction graphs aims at finding complexes that are common to different species. Kelley *et al.* [KSK+03] developed the program PathBlast, which aligns two protein-protein interaction graphs combining topology and sequence similarity. Sharan *et al.* [SIK+04] studied the conservation of complexes (they focused on dense, clique-like interaction patterns) that are conserved in *Saccharomyces cerevisae* and *Helicobacter pylori*, and found 11 significantly conserved complexes (several of these complexes match very well with prior experimental knowledge on complexes in yeast only). They actually recasted the problem of searching for conserved complexes as a problem of searching for heavy subgraphs in an edge- and node-weighted graph, whose vertices are orthologous protein pairs. A promising computational framework for alignment and comparison of more than one protein network together with a three-way alignment of the protein-protein interaction networks of *Caenorhabditis elegans*, *Drosophila melanogaster* and *Saccharomyces cerevisae* is presented in [SSK+04].

In [FLV04], this pattern matching problem was stated as the problem of finding an occurrence of a pattern graph G in a target graphs H w.r.t lists constraints (referred hereafter as the Exact-(ρ,σ)-Matching problem): to each vertex u of G is associated a lists $\mathcal{L}(u)$ of vertices in H and the occurrence of G in H is required to be an injective graph homomorphism ϕ from G to H such that $\phi(u) \in \mathcal{L}(u)$ for each vertex u in G. The two parameters ρ and σ denote the maximum size of a list of G and the maximum number of occurrences of a vertex of H among the lists of G, respectively ; Roughly speaking, the rationale of this approach is as follows. First, graph homomorphism only preserves

adjacency, and hence can deal with interaction datasets that are missing many true protein interactions. Second, injectivity is required in order to establish a bijective relationship between proteins in the complex and proteins in the occurrence. Finally, graph homomorphism with respect to orthologous links can be easily recasted as list homomorphism: a list of putative orthologs is associated to each protein (vertex) of the complex, and each such protein can only be mapped by the homomorphism to a protein occurring in its list. In the context of comparative analysis of protein-protein interaction graphs, drastic restrictions were imposed on the size of the lists. Some (classical and parameterized) hardness results together with several heuristics for the Exact-(ρ, σ)-Matching problem were presented in [FLV04]. These results were improved in [FRV05]. Of particular importance in the context of computational biology, we investigated in [FRV05] the problem of finding approximate occurrences (the Max-(ρ, σ)-Matching problem), *i.e.*, the injective mapping of G to H were no longer required to be a graph homomorphism but to match as many edges as possible.

Aiming at presenting accurate computational models, we combine here state-of-the art approaches to identifying orthologs (genes in different species that originate from a single gene in the last common ancestor of these species) for transferring functional information between genes in different organisms with a high degree of reliability [TKL97,RSS01] and the above mentioned line of research by considering additional structural constraints on the lists: for each distinct vertices u and v of G, either $\mathcal{L}(u) = \mathcal{L}(v)$ or $\mathcal{L}(u) \cap \mathcal{L}(v) = \emptyset$. The obtained problem is modeled by replacing lists by colors: to all vertices of G and H is associated a color and a vertex of G can only be mapped to a vertex of H with the same color.

This paper is organized as follows. We briefly discuss in Section 2 basic notations and definitions that we will use throughout the paper. In Section 3 we give a randomized algorithm for finding an injective mapping w.r.t to the colorings that matches all the edges of the pattern graph. We prove in Section 4 that the problem of finding an injective mapping w.r.t to the colorings that matches as many edges of the pattern graph as possible is hard to approximate even if both the pattern graph and the target graph are linear forests or trees. Section 5 is devoted to approximation with a focus on three restricted but still hard cases: (i) the pattern graph or the target graph has bounded degree, (ii) the number of occurrences of each color in the target graph is considered to be small, (iii) both the pattern graph and the target graphs are linear forests and (iv) each color occurs two times in both G and H. Section 6 concludes our word and propose future directions of research.

2 Preliminaries

We assume readers have basic knowledge about graph theory [Die00] and we shall thus use most conventional terms of graph theory without defining them (we only recall basic notations). Let G be a graph. We write $\mathbf{V}(G)$ for the set of vertices and $\mathbf{E}(G)$ for the set of edges. Also, we write $\mathbf{n}(G)$ for $|\mathbf{V}(G)|$ and $\mathbf{m}(G)$ for $|\mathbf{E}(G)|$. The maximum degree $\Delta(G)$ of a graph G is the largest degree over all vertices. A graph is called a *linear forest* if every component is a path. Let G be a graph together with a coloring $\lambda : \mathbf{V}(G) \to \mathcal{C}$ of its vertices. For any color $c_i \in \mathcal{C}$, we denote by $\mathcal{C}_G(c_i)$ the set of vertices of G that are colored with color c_i, *i.e.*, $\mathcal{C}_G(c_i) = \{u \in \mathbf{V}(G) : \lambda(u) = c_i\}$. The *multiplicity* of λ in G, written $\texttt{mult}(G, \lambda)$, is the maximum number of occurrences of a color in G, *i.e.*, $\texttt{mult}(G, \lambda) = \max\{|\mathcal{C}_G(c_i)| : c_i \in \mathcal{C}\}$. Let G and H be two graphs and let $\theta : \mathbf{V}(G) \to \mathbf{V}(H)$ be an injective mapping. The set of edges of G that are preserved in H by θ is written $\mathsf{match}(G, H, \theta)$, *i.e.*, $\mathsf{match}(G, H, \theta) = \{\{u, v\} \in \mathbf{E}(G) : \{\theta(u), \theta(v)\} \in \mathbf{E}(H)\}$. If both G and H are equipped with some colorings $\lambda_G : \mathbf{V}(G) \to \mathcal{C}$ and $\lambda_H : \mathbf{V}(H) \to \mathcal{C}$ of their vertices, a mapping $\theta : \mathbf{V}(G) \to \mathbf{V}(H)$ is said to be *with respect to* (w.r.t.) λ_G and λ_H if $\lambda_G(u) = \lambda_H(\theta(u))$ for every $u \in \mathbf{V}(G)$, *i.e.*, θ is a color preserving mapping. For simplicity, we shall usually abbreviate such a mapping as $\theta : \mathbf{V}(G) \xrightarrow{\lambda_G, \lambda_H} \mathbf{V}(H)$.

We are now in position to formally define the Max–(ρ, σ)–Matching–Colors problem we are interested in.

Max–(ρ, σ)–Matching–Colors

- **Input** : Two graphs G and H together with the coloring mappings $\lambda_G : \mathbf{V}(G) \to \mathcal{C}$, $\texttt{mult}(G, \lambda_G) = \rho$, and $\lambda_H : \mathbf{V}(H) \to \mathcal{C}$, $\texttt{mult}(H, \lambda_H) = \sigma$.
- **Solution** : An injective mapping $\theta : \mathbf{V}(G) \xrightarrow{\lambda_G, \lambda_H} \mathbf{V}(H)$.
- **Measure** : The number of edges of G matched by the injective mapping θ, *i.e.*, $|\mathsf{match}(G, H, \theta)|$.

We designate by Exact–(ρ, σ)–Matching–Colors the extremal problem of finding an injective mapping $\theta : \mathbf{V}(G) \xrightarrow{\lambda_G, \lambda_H} \mathbf{V}(H)$ that matches all the edges of G, *i.e.*, θ is required to be an injective graph homomorphism [FRV05]. Also, we call an instance of both the Max–(ρ, σ)–Matching–Colors and Exact–(ρ, σ)–Matching–Colors problems *colorful* if $\rho = 1$.

Let $\langle G, H, \lambda_G, \lambda_H \rangle$ be an instance of the Max–(ρ, σ)–Matching–Colors. First, a necessary and sufficient condition for an injective mapping to exist is $|\mathcal{C}_G(c_i)| \leq$

$|\mathcal{C}_H(c_i)|$ for each color $c_i \in \mathcal{C}$. Second, an edge $\{u, v\} \in \mathbf{E}(G)$, $\lambda_G(u) = c_u$ and $\lambda_G(v) = c_v$, is called a *bad edge* if there does not exist distinct $u' \in \mathcal{C}_H(c_u)$ and $v' \in \mathcal{C}_H(c_v)$ such that $\{u', v'\} \in \mathbf{E}(H)$. Clearly, if we remove from G its bad edges, this does not affect the optimal solutions for the Max–(ρ, σ)–Matching–Colors problem, since bad edges can never be matched. Notice that we can tell bad edges apart in $O(\sigma^2\, \mathbf{m}(G)) = O(\mathbf{m}(G))$ time, provided σ is a constant. Therefore, throughout the paper, we will consider only trim instances as defined in the following.

Definition 1 (Trim instance) *An instance $\langle G, H, \lambda_G, \lambda_H \rangle$ of the* Max–(ρ, σ)–Matching–Colors *or the* Exact–(ρ, σ)–Matching–Colors *problem is said to be* trim *if the following conditions hold true: (i) for each color $c_i \in \mathcal{C}$, $|\mathcal{C}_G(c_i)| \leq |\mathcal{C}_H(c_i)|$, and (ii) for each edge $\{u_i, u_j\} \in \mathbf{E}(G)$, there exists an edge $\{v_i, v_j\} \in \mathbf{E}(H)$ such that $\lambda_G(u_i) = \lambda_H(v_i)$ and $\lambda_G(u_j) = \lambda_H(v_j)$.*

3 Exact colorful instances

This section is devoted to the Exact–$(1, \sigma)$–Matching–Colors problem. On the one hand, both the Exact–$(1, \sigma)$–Matching–Colors problem for $\Delta(G) \leq 2$ and the Exact–$(\rho, 2)$–Matching–Colors problem are polynomial-time solvable for any constant ρ and σ [FLV04]. On the other hand, the Exact–$(1, 3)$–Matching–Colors problem for $\Delta(G) = 3$ and $\Delta(H) = 4$ is NP-complete [FRV05]. We first observe that the Exact–$(1, \sigma)$–Matching–Colors problem is easily solvable in $\tilde{O}(\sigma^{\mathbf{n}(G)})$ time: the brute-force algorithm tries all possible injective mappings $\theta : \mathbf{V}(G) \xrightarrow{\lambda_G, \lambda_H} \mathbf{V}(H)$ and returns the best one. We give a faster randomized algorithm (referred hereafter as Algorithm Rand-Exact-Matching-Colors) than runs in $\tilde{O}(f(\sigma)^{\mathbf{n}(G)})$ expected time, where $f(\sigma) = \dfrac{4\sigma(2\sigma-2)^3}{4(2\sigma-2)^3 + 27(2\sigma-3)}$. Observe that $f(\sigma) < \sigma$, for $\sigma > 2$. For the sake of illustration, $f(3) < 2.279$, $f(4) < 3.460$ and $f(5) < 4.578$.

We present here a random walk algorithm which is similar to [MU05]. For simplicity, we assume the worst case where each color occurs exactly σ times in graph H. The basic idea is to start with a random injective mapping θ, look at an edge e of G that is not matched by θ, select at random one end-vertex u of e and finally change at random the image of u, *i.e.*, $\theta(u)$. Observe however that, oppositely to satisfiability-like algorithms where changing the assignment of a boolean variable in an unsatisfied clause result in a satisfied clause, the edge e might be here still not matched by the new injective mapping θ. The complete description of the alorithm is given as Algorithm Rand-Exact-Matching-Colors.

Let $\langle G, H, \lambda_G, \lambda_H \rangle$ be an arbitrary instance of the Exact–$(1, \sigma)$–Matching–Colors problem, and suppose that there exists an injective homomorphism

Rand-Exact-Matching-Colors($\langle G, H, \lambda_G, \lambda_H \rangle$)

Input: An instance $\langle G, H, \lambda_G, \lambda_H \rangle$ of the Exact–$(1, \sigma)$–Matching–Colors problem.

Output: An occurrence of G in H, *i.e.*, an injective homomorphism $\theta : \mathbf{V}(G) \xrightarrow{\lambda_G, \lambda_H} \mathbf{V}(H)$ (if such a mapping exists).

Repeat, terminating whether an occurrence of G in H w.r.t λ_G and λ_H is found:

- Let $\theta : \mathbf{V}(G) \xrightarrow{\lambda_G, \lambda_H} \mathbf{V}(H)$ be a random injective.
- **Loop** up to $3\,\mathbf{n}(G)$ times, terminating whether an occurrence of G in H w.r.t λ_G and λ_H is found:
  Choose at random an edge $e \in \mathbf{E}(G)$ that is not matched by θ, choose at random one vertex $u \in e$ and change at random the value of $\theta(u)$ w.r.t λ_G and λ_H.

$\theta_{\mathbf{opt}} : \mathbf{V}(G) \xrightarrow{\lambda_G, \lambda_H} \mathbf{V}(H)$, *i.e.*, $\langle G, H, \lambda_G, \lambda_H \rangle$ is a YES instance. Without loss of generality we may assume that, for each color $c_i \in \mathcal{C}$, exactly σ vertices of H are colored with color c_i (and hence H has $\sigma|\mathcal{C}|$ vertices). Fix any injective mapping $\theta : \mathbf{V}(G) \xrightarrow{\lambda_G, \lambda_H} \mathbf{V}(H)$ and let $\theta_i : \mathbf{V}(G) \xrightarrow{\lambda_G, \lambda_H} \mathbf{V}(H)$ be the injective mapping after the i-th step of the inner loop of Algorithm Rand-Exact-Matching-Colors. Let X_i be the number of vertices $u \in \mathbf{V}(G)$ such that $\theta_i(u) = \theta_{\mathbf{opt}}(u)$. If $X_i = \mathbf{n}(G)$, Algorithm Rand-Exact-Matching-Colors terminates with an injective homomorphism. Clearly, the algorithm could terminate before $X_i = \mathbf{n}(G)$ by finding a different injective homomorphism, but for our analysis the worst case is that the algorithm only stops when $X_i = \mathbf{n}(G)$.

Suppose $1 \leq X_i \leq \mathbf{n}(G) - 1$. At each step, we choose an edge $e = \{u, v\} \in \mathbf{E}(G)$ that is not matched. Since $\langle G, H, \lambda_G, \lambda_H \rangle$ is a YES instance, θ_i and $\theta_{\mathbf{opt}}$ disagree on at least one of u and v. Suppose first that θ_i and $\theta_{\mathbf{opt}}$ disagree on exactly one of u and v. Then, the probability of increasing the number of agreements between $\theta_{\mathbf{opt}}$ and θ_{i+1} is $(2\sigma - 2)^{-1}$, the probability of decreasing the number of agreements between $\theta_{\mathbf{opt}}$ and θ_{i+1} is $(\sigma - 1)(2\sigma - 2)^{-1}$ and the probability of obtaining the same number of agreements between $\theta_{\mathbf{opt}}$ and θ_{i+1} is $(\sigma - 2)(2\sigma - 2)^{-1}$. Suppose now that θ_i and $\theta_{\mathbf{opt}}$ disagree on both u and v. Then, the probability of increasing the number of agreements between $\theta_{\mathbf{opt}}$ and θ_{i+1} is $2(2\sigma - 2)^{-1}$, the probability of decreasing the number of agreements between $\theta_{\mathbf{opt}}$ and θ_{i+1} is 0 (θ_i and $\theta_{\mathbf{opt}}$ indeed already both disagree on both vertices) and the probability of obtaining the same number of agreements between $\theta_{\mathbf{opt}}$ and θ_{i+1} is $(2\sigma - 4)(2\sigma - 2)^{-1}$. In the light of the above probabilities, let us thus consider the pessimistic stochastic process

$Y = (Y_1, Y_2, \ldots)$ defined as follows:

$$\mathsf{Pr}[Y_{i+1} = j+1 | Y_i = j] \geq \frac{1}{2\sigma - 2},$$
$$\mathsf{Pr}[Y_{i+1} = j-1 | Y_i = j] \leq \frac{2\sigma - 3}{2\sigma - 2}.$$

This stochastic process is best understood by using the same metaphor as in [MU05]: consider a particle moving on the integer line, with probability $(2\sigma - 1)^{-1}$ of moving up by one and probability $(2\sigma - 3)(2\sigma - 2)^{-1}$ of moving down by one. Observe that in the pessimistic stochastic process Y the particle never stays in place whereas the probability of obtaining the same number of agreements is non-zero in Algorithm Rand-Exact-Matching-Colors. Let r_j be the probability of exactly k "*moves down*", and $j + k$ "*moves up*" in a sequence of $2k + j$ moves. We have $r_j \geq \left(\frac{2\sigma - 3}{2\sigma - 2}\right)^k \left(\frac{1}{2\sigma - 2}\right)^{j+k}$. Now, let q_j be the probability that the algorithm finds an injective homomorphism within $j + 2k \leq 3\,\mathbf{n}(G)$ steps, starting from a random injective mapping $\theta : \mathbf{V}(G) \xrightarrow{\lambda_G, \lambda_H} \mathbf{V}(H)$.

Lemma 2 $q_j \geq \frac{\sqrt{3}}{8\sqrt{\pi j}} \left(\frac{27(2\sigma - 3)}{4(2\sigma - 2)^3}\right)^j$.

PROOF. We first observe that r_j is a lower bound for q_j, and hence

$$q_j \geq \max_{0 \leq k \leq j} \binom{j + 2k}{k} \left(\frac{2\sigma - 3}{2\sigma - 2}\right)^k \left(\frac{1}{2\sigma - 2}\right)^{j+k}.$$

If we restrict ourselves to $k = j$, we obtain

$$q_j \geq \binom{3j}{j} \left(\frac{2\sigma - 3}{2\sigma - 2}\right)^j \left(\frac{1}{2\sigma - 2}\right)^{2j} = \binom{3j}{j} \left(\frac{2\sigma - 3}{(2\sigma - 2)^3}\right)^j.$$

Combining this with standard Stirling's approximation yields

$$q_j \geq \frac{\sqrt{3}}{8\sqrt{\pi j}} \left(\frac{27}{4}\right)^j \left(\frac{2\sigma - 3}{(2\sigma - 2)^3}\right)^j = \frac{\sqrt{3}}{8\sqrt{\pi j}} \left(\frac{27(2\sigma - 3)}{4(2\sigma - 2)^3}\right)^j$$

and the lemma is proved. □

Let p_j be the probability that a random injective mapping $\theta : \mathbf{V}(G) \xrightarrow{\lambda_G, \lambda_H} \mathbf{V}(H)$ has j disagreements with $\theta_{\mathbf{opt}}$. We now derive a lower bound for q, the probability that the process finds an occurrence of G in H w.r.t λ_G and λ_H in $3\,\mathbf{n}(G)$ steps starting from a random injective mapping.

Lemma 3 $q \geq \dfrac{\sqrt{3}}{8\sqrt{\pi\ \mathbf{n}(G)}} \left(\dfrac{4(2\sigma-2)^3 + 27(2\sigma-3)}{4\sigma(2\sigma-2)^3} \right)^{\mathbf{n}(G)}$.

PROOF. By definition, we have $q \geq \sum_{j=0}^{\mathbf{n}(G)} q_j\ p_j \geq \dfrac{1}{\sigma^{\mathbf{n}(G)}} + \sum_{j=1}^{\mathbf{n}(G)} q_j\ p_j$. Combining this with Lemma 2, we obtain

$$\begin{aligned}
q &\geq \frac{1}{\sigma^{\mathbf{n}(G)}} + \sum_{j=1}^{\mathbf{n}(G)} \binom{n}{j} \frac{1}{\sigma^{\mathbf{n}(G)}} \frac{\sqrt{3}}{8\sqrt{\pi j}} \left(\frac{27(2\sigma-3)}{4(2\sigma-2)^3} \right)^j \\
&\geq \frac{\sqrt{3}}{8\sqrt{\pi\ \mathbf{n}(G)}} \frac{1}{\sigma^{\mathbf{n}(G)}} \sum_{j=0}^{\mathbf{n}(G)} \binom{n}{j} \left(\frac{27(2\sigma-3)}{4(2\sigma-2)^3} \right)^j (1)^{n-j} \\
&= \frac{\sqrt{3}}{8\sqrt{\pi\ \mathbf{n}(G)}} \frac{1}{\sigma^{\mathbf{n}(G)}} \left(1 + \frac{27(2\sigma-3)}{4(2\sigma-2)^3} \right)^{\mathbf{n}(G)} \\
&= \frac{\sqrt{3}}{8\sqrt{\pi\ \mathbf{n}(G)}} \left(\frac{4(2\sigma-2)^3 + 27(2\sigma-3)}{4\sigma(2\sigma-2)^3} \right)^{\mathbf{n}(G)}.
\end{aligned}$$

□

Therefore, if we assume that there exists an injective mapping $\theta : \mathbf{V}(G) \xrightarrow{\lambda_G, \lambda_H} \mathbf{V}(H)$, the number of random injective mappings the process tries before finding an occurrence of G in H is a geometric random variable with parameter q. Hence, the expected of random injective mappings tried is q^{-1}, and for each injective mapping the algorithm uses at most $3\,\mathbf{n}(G)$ steps. Therefore, the expected number of steps until a solution is found is bounded by $O(\mathbf{n}(G)^{3/2}\, f(\sigma)^{\mathbf{n}(G)})$. We have thus proved the following.

Proposition 4 *Algorithm* Rand-Exact-Matching-Colors *returns an injective homomorphism* $\theta : \mathbf{V}(G) \xrightarrow{\lambda_G, \lambda_H} \mathbf{V}(H)$ *(if such a mapping exists) in* $\tilde{O}(f(\sigma)^{\mathbf{n}(G)})$ *time, where* $f(\sigma) = \dfrac{4\sigma(2\sigma-2)^3}{4(2\sigma-2)^3 + 27(2\sigma-3)}$.

4 Hardness results

Recall that the Max–$(1,2)$–Matching–Colors problem for bipartite graphs G and H with $\Delta(G) = 3$ and $\Delta(H) = 2$ (resp. with $\Delta(G) = 6$ and $\Delta(H) = 5$) is APX-hard and is not approximable within ratio 1.0005 (resp. 1.0014), unless P = NP [FRV05]. Therefore, there is a natural interest to investigate

the complexity issues of the Max–(ρ,σ)–Matching–Colors problem for restricted graph classes. Unfortunately, as we shall prove here, the Max–$(3,3)$–Matching–Colors (resp. Max–$(2,2)$–Matching–Colors) problem is APX-hard even if both G and H are linear forests (resp. trees with maximum degree 3).

Proposition 5 *The* Max–$(3,3)$–Matching–Colors *problem is* APX*-hard even if both G and H are linear forests.*

PROOF. We propose a L-reduction from the Max-2-Sat-3 problem (Given a set X of variables and a boolean formula ϕ in conjunctive normal form where clause consists in at most 2 literals and each variable appears in at most 3 clauses, find a truth assignment for X that satisfies as many clauses as possible) which is known to be APX-complete [PY91]. We assume that each negated literal and each positive literal occurs at most twice, since otherwise a self-reduction would trivially apply. Let ϕ be an arbitrary input for the Max-2-Sat-3 problem. Let $X = \{x_1, \ldots, x_n\}$ denote the set of variables and $C = \{c_1, \ldots, c_m\}$ denote the set of clauses. For each $j = 1, 2, \ldots, m$ and each $\ell = 1, 2$, we write $c_j[\ell]$ for the ℓ-th literal of the clause c_j. For each variable $x_i \in X$, we let $\mathsf{nb\_occ}(x_i)$ stands for the number of occurrences of variable x_i in ϕ (counting together both positive and negative occurrences); we may clearly assume here that $2 \leq \mathsf{nb\_occ}(x_i) \leq 3$ for each $x_i \in X$ (a self-reduction would trivially apply in case $\mathsf{nb\_occ}(x_i) = 1$). We now describe how to construct the corresponding instance of the Max–$(2,2)$–Matching–Colors problem.

Let us start by considering the associated graph G. We introduce two vertices $x_i^G[1]$ and $x_i^G[2]$ for each variable $x_i \in X$, and one vertex c_j^G for each clause $c_j \in C$. For each $x_i \in X$, we introduce the edge $\{x_i^G[1], x_i^G[2]\}$. Also, for each clause c_j and each $\ell \in \{1, 2\}$, we introduce the vertex $t_i^G[k]$ (resp. $f_i^G[k]$) if the ℓ-th literal of clause c_i is the k-th occurrence of positive (resp. negative) literal x_i (resp. $\overline{x_i}$) together with the edge $\{c_j^G, t_i^G[k]\}$ (resp. $\{c_j^G, f_i^G[k]\}$).

We now turn to defining the corresponding graph H. For each variable $x_i \in X$, we introduce four vertices $t_i^H[1]$, $t_i^H[2]$, $f_i^H[1]$ and $f_i^H[2]$, and the two edges $\{t_i^H[1], t_i^H[2]\}$ and $\{f_i^H[1], f_i^H[2]\}$. For each clause c_j and each $\ell \in \{1, 2\}$, we introduce the vertex $c_j^H[\ell]$ and the edge $\{c_j^H[\ell], t_i^H[k]\}$ (resp. $\{c_j^H[\ell], f_i^H[k]\}$) if the ℓ-th literal of clause c_i is the k-th occurrence of positive (resp. negative) literal x_i (resp. $\overline{x_i}$). Also, for each clause c_j we introduce one isolated vertex $c_i^H[3]$, and for each $x_i \in X$ we introduce two isolated vertices $y_i^H[1]$ and $y_i^H[2]$.

Let $\mathcal{C} = \{\mathbf{x_i}[\ell] : 1 \leq i \leq n \wedge 1 \leq \ell \leq 2\} \cup \{\mathbf{c_j} : 1 \leq j \leq m\}$ be our set of colors. Define the mapping $\lambda_G : \mathbf{V}(G) \to \mathcal{C}$ by $\lambda_G(x_i^G[1]) = \mathbf{x_i}[\mathbf{1}]$ for all $x_i^G[1] \in \mathbf{V}(G)$, $\lambda_G(x_i^G[2]) = \mathbf{x_i}[\mathbf{2}]$ for all $x_i^G[2] \in \mathbf{V}(G)$, $\lambda_G(t_i^G[\ell]) = \mathbf{x_i}[\mathbf{k}]$ for all $t_i^G[k] \in \mathbf{V}(G)$, $\lambda_G(f_i^G[\ell]) = \mathbf{x_i}[\mathbf{k}]$ for all $f_i^G[k] \in \mathbf{V}(G)$, and $\lambda_G(c_i^G) = \mathbf{c_j}$ for

all $c_j^G \in \mathbf{V}(G)$. Also, define the mapping $\lambda_H : \mathbf{V}(H) \to \mathcal{C}$ by $\lambda_H(t_i^H[k]) = \mathbf{x_i}[\mathbf{k}]$ for all $t_i^H[k] \in \mathbf{V}(H)$, $\lambda_H(f_i^H[k]) = \mathbf{x_i}[\mathbf{k}]$ for all $f_i^H[k] \in \mathbf{V}(H)$, $\lambda_H(c_j^H[k]) = \mathbf{c_j}$ for all $c_j^H[k] \in \mathbf{V}(H)$, and $\lambda_H(y_i^H[k]) = \mathbf{x_i}[\mathbf{k}]$ for all $y_i^H[k] \in \mathbf{V}(H)$. It is easily seen that both G and H are linear forests in which each color occurs at most three times. An illustration of this construction is given in Figure 1.

Let f be a truth assignment for X. Write $\mathcal{A}(\phi, f)$ the number of clauses of ϕ that are satisfied by f. Define an injective mapping $\theta : \mathbf{V}(G) \xrightarrow{\lambda_G, \lambda_H} \mathbf{V}(H)$ as follows. For each $x_i \in X$, if $f(x_i) = \texttt{true}$ (resp. $f(x_i) = \texttt{false}$) then $\theta(x_i^G[k]) = f_i^H[k]$ (resp. $\theta(x_i^G[k]) = t_i^H[k]$). Now, let c_j be any clause of ϕ and let $z_i^G[k]$ and $z_{i'}^G[k']$, $z \in \{t, f\}$, be the two vertices connected to vertex c_j^G. Suppose first that the clause c_j is satisfied by its ℓ-th literal. If $\ell = 1$, then $\theta(c_j^G) = c_j^H[\ell]$, $\theta(z_i^G[k]) = z_i^H[k]$ and $\theta(z_{i'}^G[k']) = y_{i'}^H[k']$. Otherwise, if $\ell = 2$, then $\theta(c_j^G) = c_j^H[\ell]$, $\theta(z_i^G[k]) = y_i^H[k]$ and $\theta(z_{i'}^G[k']) = z_{i'}^H[k']$. Finally, if the clause c_j is not satisfied by f, set $\theta(c_j^G) = c_j^H[3]$, $\theta(z_i^G[k]) = y_i^H[k]$ and $\theta(z_{i'}^G[k']) = y_{i'}^H[k']$. The reader is invited to check that the injective mapping θ preserves $n + \mathcal{A}(\phi, f)$ edges.

Conversely, let $\theta : \mathbf{V}(G) \xrightarrow{\lambda_G, \lambda_H} \mathbf{V}(H)$ be an injective mapping. Write $\mathcal{A}(\theta)$ the number of edges that are preserved by θ. Define Θ to be the set of all injective mappings $\theta' : \mathbf{V}(G) \xrightarrow{\lambda_G, \lambda_H} \mathbf{V}(H)$ that preserved at least $\mathcal{A}(\theta)$ edges. For each $\theta' \in \Theta$, define $S(\theta') = \{i : \{t_i^H[1], t_i^H[2]\}$ or $\{f_i^H[1], f_i^H[2]\}$ is matched by $\theta'\}$. We observe that, by construction, any $\theta' \in \Theta$ cannot match both $\{t_i^H[1], t_i^H[2]\}$ and $\{f_i^H[1], f_i^H[2]\}$, and hence $|S(\theta')| \leq n$ for all $\theta' \in \Theta$. We claim that there exists $\theta' \in \Theta$ such that $|S(\theta')| = n$. Let $\theta^* \in \Theta$ be such that $|S(\theta^*)| \geq S(\theta')$ for all $\theta' \in \Theta$. Suppose, for the sake of contradiction, that $|S(\theta^*)| < n$. Then, there exists one i, $1 \leq i \leq n$, such that neither $\{t_i^H[1], t_i^H[2]\}$ nor $\{f_i^H[1], f_i^H[2]\}$ are matched by θ^*. We now make the important observation that, by construction, at least one of the four vertices $t_i^H[1]$, $t_i^H[2]$, $f_i^H[1]$ and $f_i^H[2]$ is not connected to a vertex $c_j^H[\ell]$ (since each variable x_i appears in at most 3 clauses of ϕ). Without loss of generality, assume that $t_i^H[1]$ is not connected to a vertex $c_j^H[\ell]$. Therefore, there exists $\theta' \in \Theta$ such that $|S(\theta')| = n$, and hence for each $1 \leq i \leq n$, exactly one of the two edges $\{t_i^H[1], t_i^H[2]\}$ and $\{f_i^H[1], f_i^H[2]\}$ of H are matched by θ'. As an immediate consequence, $\mathcal{A}(\theta) = n + k$ for some $0 \leq k \leq m$. Indeed, at most one of the two edges incident to vertex c_i^G, Define now a truth assignment f for X as follows: for each $x_i \in X$, $f(x_i) = \texttt{true}$ if and only if the edge $\{f_i^H[1], f_i^H[2]\}$ of H is matched by θ'. According to the above, f satisfies k clauses of ϕ

Consequently, we have $\mathbf{opt}(\theta) \leq n + \mathbf{opt}(\phi) \leq m + \mathbf{opt}(\phi) \leq 2\,\mathbf{opt}(\phi) + \mathbf{opt}(\phi) = 3\,\mathbf{opt}(\phi)$, where the second inequality comes from the fact that we can assume that ϕ contains more clauses than variables and the last inequality is due to the fact that at least half of the clauses of a boolean formula in conjunctive normal form are always satisfiable. Finally, observe that for any

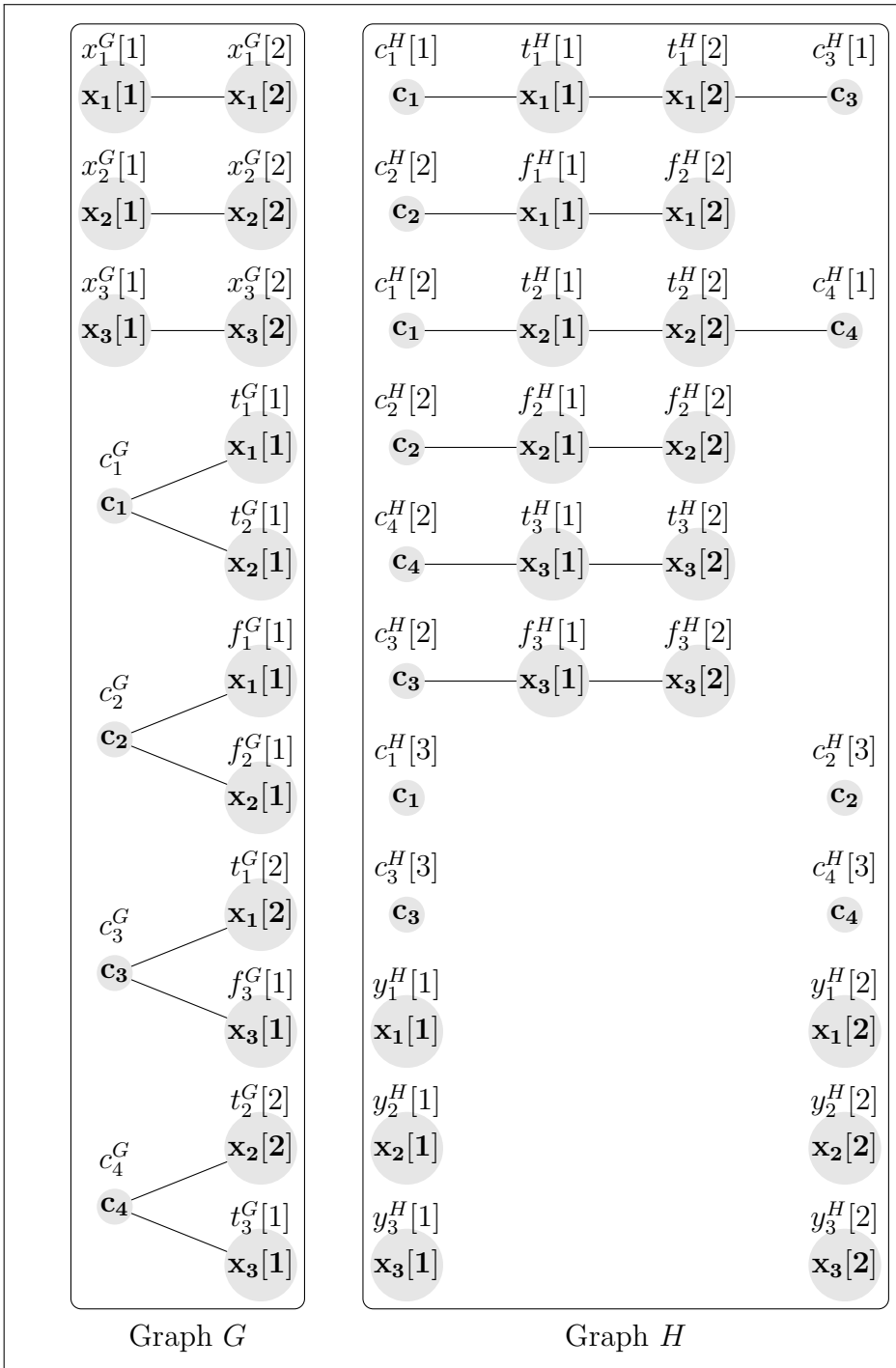

Figure 1. Illustration of the proof of Proposition 5 for the boolean formula $\phi = (x_1 \vee x_2) \wedge (\overline{x_1} \vee \overline{x_2}) \wedge (x_1 \vee \overline{x_3}) \wedge (x_2 \vee x_3)$. For the sake of clarity, the color of each vertex is given in shaded circle form.

injective mapping $\theta : \mathbf{V}(G) \xrightarrow{\lambda_G,\lambda_H} \mathbf{V}(H)$, one can find a truth assignment for ϕ such that $\mathcal{A}(\phi, f) \geq \mathcal{A}(\theta) - n$. This implies that $\mathbf{opt}(\phi) - \mathcal{A}(\phi, f) = \mathbf{opt}(\theta) - n - \mathcal{A}(\phi, f) = \mathbf{opt}(\theta) - \mathcal{A}(\theta)$. It follows that the proposed reduction is, indeed, a L-reduction – with parameters $\alpha = 3$ and $\beta = 1$ – from the Max-2-Sat-3 problem to the Max–$(3,3)$–Matching–Colors problem where both G and H are linear forests. □

It remains open, however, whether the Max–(ρ,σ)–Matching–Colors problem for linear forests G and H is polynomial-time solvable in case $\rho < 3$. The rationale of this question stems from the following proposition.

Proposition 6 *The* Max–$(2,2)$–Matching–Colors *problem is* APX*-hard even if both G and H are trees.*

PROOF. For the sake of clarity, in the reduction described here below, the graph G is actually a forest, but it is very easy to make it into a tree by suitable addition of edges. Furthermore, in the reduction we describe the degree of the graphs G and H is not bounded, but it is easy to modify the reduction so that $\Delta(G), \Delta(H) \leq 3$. These details will be discussed at the end of the proof. The reduction is from the Vertex Cover problem for cubic graphs, which is known to be APX-hard [PY91,AK00]. Assume thus given a cubic graph W of order n and size m. We now describe the associated instance of the Max–$(2,2)$–Matching–Colors problem. For ease of exposition, let us first define $P = \{(u,e) \in \mathbf{V}(W) \times \mathbf{E}(W) : u \in e\}$ to be the set of *pins* of the graph W. The set of colors $\mathcal{C}$ is now defined by $\mathcal{C} = \{\mathbf{c_0}, \mathbf{c'_0}\} \cup \mathcal{C}_V \cup \mathcal{C}_E \cup \mathcal{C}_P$, where $\mathcal{C}_V = \{\mathbf{c_u} : u \in \mathbf{V}(W)\}$, $\mathcal{C}_E = \{\mathbf{c_e} : e \in \mathbf{E}(W)\}$ and $\mathcal{C}_P = \{\mathbf{c_{u,e}} : (u,e) \in P\}$.

Next, let us specify the forest G. The forest G contains a tree T defined by $\mathbf{V}(T) = \{r_0^G\} \cup \{T_u : u \in \mathbf{V}(W)\} \cup \{T_{u,e} : (u,e) \in P\}$ and $\mathbf{E}(T) = \{\{r_0, T_u\} : u \in \mathbf{V}(W)\} \cup \{\{T_u, T_{u,e}\} : (u,e) \in P\}$. The mapping $\lambda_G : \mathbf{V}(T) \to \mathcal{C}$ is defined by $\lambda_G(r_0^G) = \mathbf{c_0}$, $\lambda_G(T_u) = \mathbf{c_u}$, $u \in \mathbf{V}(W)$ and $\lambda_G(T_{u,e}) = \mathbf{c_{u,e}}$, $(u,e) \in P$. Besides the tree T, the forest G contains other m connected components. More precisely, for each edge $e = \{u,v\} \in \mathbf{E}(W)$, the forest G contains a connected component C^e defined by $\mathbf{V}(C^e) = \{C_e^e, C_u^e, C_v^e\}$ and $\mathbf{E}(C^e) = \{\{C_e^e, C_u^e\}, \{C_e^e, C_v^e\}\}$, *i.e.*, the connected component C^e is nothing but a length two path in which the vertex C_e^e is adjacent both to C_u^e and to C_v^e. The mapping $\lambda_G : \mathbf{V}(C^e) \to \mathcal{C}$ is defined by $\lambda_G(C_e^e) = \mathbf{c_e}$, $\lambda_G(C_u^e) = \mathbf{c_{u,e}}$ and $\lambda_G(C_v^e) = \mathbf{c_{v,e}}$.

Finally, let us define the tree H by $\mathbf{V}(H) = \{r_0^H, s_0^H\} \cup \{H_u, H_{\overline{u}} : u \in \mathbf{V}(W)\} \cup \{H_{u,e}, H_{\overline{u},e} : (u,e) \in P\} \cup \{H'_{u,e} : (u,e) \in P\}$ and $\mathbf{E}(H) = \{\{r_0^H, s_0^H\}\} \cup \{\{r_0^H, H_u\} : u \in \mathbf{V}(W)\} \cup \{\{s_0^H, H_{\overline{u}}\} : u \in \mathbf{V}(W)\} \cup \{\{H_{u,e}, H_u\} : (u,e) \in P\} \cup \{\{H_{\overline{u},e}, H_{\overline{u}}\} : (u,e) \in P\} \cup \{\{H_{u,e}, H'_{u,e}\} : (u,e) \in P\}$. The mapping $\lambda_H : \mathbf{V}(H) \to \mathcal{C}$ is defined by $\lambda_H(r_0^H) = \mathbf{c_0}$, $\lambda_H(s_0^H) = \mathbf{c'_0}$, $\lambda_H(H_u) = \lambda_H(H_{\overline{u}}) = \mathbf{c_u}$,

$u \in \mathbf{V}(W)$, $\lambda_H(H_{u,e}) = \lambda_H(H_{\overline{u},e}) = \mathbf{c_{u,e}}$, $(u,e) \in P$, and $\lambda_H(H'_{u,e}) = \mathbf{c_e}$, $(u,e) \in P$. The description of the reduction is complete. The following two claims are the key of the APX-hardness proof.

Claim 7 *Assume W has a vertex cover $X \subseteq \mathbf{V}(W)$, $|X| = k$. Then the instance $\langle G, H, \lambda_G, \lambda_H \rangle$ obtained from the original graph W as outlined above admits a color preserving injective mapping $\theta : \mathbf{V}(G) \xrightarrow{\lambda_G, \lambda_H} \mathbf{V}(H)$ such that $|\mathsf{match}(G, H, \theta)| = 3m + n - k$.*

PROOF. The solution mapping θ is defined as follows. First, we define θ over the vertices of the tree T as follows: (i) $\theta(r_0^R) = r_0^H$, (ii) for each $u \in \mathbf{V}(W)$, $\theta(T_u) = H_{\overline{u}}$ if $u \in X$ and $\theta(T_u) = H_u$ otherwise, and (iii) for each $(u,e) \in P$, $\theta(T_{(u,e)}) = H_{(\overline{u},e)}$ if $u \in X$ and $\theta(T_{(u,e)}) = H_{(u,e)}$ otherwise. Next, for each edge $e = \{u, v\} \in \mathbf{E}(W)$, we define θ over the three vertices of the connected component C^e of G as follows: (iv) if $u \in X$ then $\theta(C^e_e) = H'_{(u,e)}$, otherwise $\theta(C^e_e) = H'_{(v,e)}$, (v) if $u \in X$ then $\theta(C^e_u) = H_{(u,e)}$, otherwise $\theta(C^e_u) = H_{(\overline{u},e)}$, and (vi) if $v \in X$ then $\theta(C^e_v) = H_{(v,e)}$, otherwise $\theta(C^e_v) := H_{(\overline{v},e)}$. It is easy to check out that θ is a color preserving injective mapping that maps precisely $3\,m + n - k$ edges of G into edges of H. $\square$

Claim 8 *Assume the instance $\langle G, H, \lambda_G, \lambda_H \rangle$ obtained from the original graph W as outlined above admits a color preserving injective mapping $\theta : \mathbf{V}(G) \xrightarrow{\lambda_G, \lambda_H} \mathbf{V}(H)$ that maps t edges of G into edges of H. Then, starting from the knowledge of θ, we can find out in polynomial-time a vertex cover $X \subseteq \mathbf{V}(W)$ of the original graph W with $|X| \geq 3\,m + n - t$.*

PROOF. Notice first that for each edge $e = \{u, v\} \in \mathbf{E}(W)$, at most one of the two edges of the component C^e of G is mapped by θ into an edge of H. Indeed, the central vertex of the length two path C^e has color $\mathbf{c_e}$ and the (two) vertices of H having color $\mathbf{c_e}$ are leaves of H.

Let us now call the mapping θ *reasonable* if the following two conditions are met: (i) for each edge $e = \{u, v\} \in \mathbf{E}(W)$, precisely one of the two edges of the component C^e of G is mapped by θ into an edge of H, and (ii) for each $(u,e) \in P$, $\theta(T_{(u,e)}) = H_{(\overline{u},e)}$ if and only if $\theta(T_u) = H_{\overline{u}}$. It is indeed easy to propose a simple pre-processing algorithm which, starting from θ, constructs a reasonable color preserving injective mapping θ' which maps at least as many edges of G as θ does. There is thus no loss of generality in assuming now that θ is reasonable. At this point, we define a subset X of $\mathbf{V}(W)$ as follows. Each vertex $u \in \mathbf{V}(W)$ belongs to X if and only if $\theta(T_u) = H_{\overline{u}}$. Since θ is reasonable, it follows that X is indeed a vertex cover of W. Furthermore, if t is the number of edges of G that θ maps into edges of H, then $|X| = 3m + n - t$. $\square$

To make G into a tree it suffices to add the following edges: the vertex r_0^G is adjacent to the central vertex C_e^e of the component C^e for each $e \in \mathbf{E}(W)$. It is quite clear that this simple modification does not affect the validity of the reduction and of the two claims. It is also very easy yet technical to rely on trees G and H of maximum degree 3 by introducing bifurcation points in the paths from the root T_0 to the vertices T_{v_i} as suitable and paralleling the same transformation also into H. To conclude in showing that the proposed reduction is an L-reduction, one has to observe that the Vertex Cover problem is APX-hard even when restricted to instances where a minimum vertex cover takes at least half of the vertices (as proven *e.g.* by the famous Trotter-NemHauser reduction). □

5 Approximation algorithms

We proved in Section 4 that the Max–$(3,3)$–Matching–Colors problem for linear forests is APX-hard. In the light of this negative result, we first focus here on approximating the Max–(ρ,σ)–Matching–colors problem for bounded degree graphs and give a polynomial-time approximation algorithm that achieves a performance ratio of $2(\Delta_{\min}+1)$, $\Delta_{\min} = \min\{\Delta(G), \Delta(H)\}$, for the Max–$(\rho,\sigma)$–Matching–colors problem. Next, we propose a randomized algorithm with performance ratio 4σ for the Max–(ρ,σ)–Matching–colors problem. and we give an approximation algorithm that achieves a performance ratio of 4 in case both G and H are linear forests. Finally, we prove the Max–$(2,2)$–Matching–Colors problem to be approximable within ratio 1.1442.

5.1 Bounded degree graphs

We first consider bounded degree graphs. Let $\mathcal{C} = \{c_1, c_2, \ldots, c_m\}$ be a set of colors and G be a graph whose vertices are colored with colors taken from $\mathcal{C}$. Also, let $A = [a_{i,j}]$ be a symmetric matrix of order m whose entries are natural integers. Consider the problem, referred hereafter as the Max-Matching-with-Color-Constraints (MMwCC) problem, of finding in G a maximum cardinality matching $\mathcal{M} \subseteq \mathbf{E}(G)$ subject to the constraint that, for $1 \leq i \leq j \leq m$, the number of edges in $\mathcal{M}$ having one end-vertex colored c_i and one end-vertex colored c_j is at most $a_{i,j}$. It is clear that a straightforward greedy algorithm delivers a 2-approximation algorithm for the MMwCC problem.

Lemma 9 *The* MMwCC *problem is approximable within ratio* 2.

Recall that an *edge coloring* of a graph G is *proper* if no two adjacent edges are assigned the same color. A proper edge coloring with k colors is called a proper *k-edge-coloring* and is equivalent to the problem of partitioning the edge set into k matchings. The smallest number of colors needed in a proper edge coloring of a graph G is *the chromatic index* $\chi'(G)$ [Die00]. Vizing's Theorem [Viz64] states that $\chi'(G) \leq \Delta(G)+1$ and that such an edge coloring can be found in polynomial-time.

Proposition 10 *For any ρ and σ, the* Max–(ρ,σ)–Matching–colors *problem is approximable within ratio $2(\Delta_{min}+1)$, where $\Delta_{min} = \min\{\Delta(G), \Delta(H)\}$.*

PROOF. Let $\langle G, H, \lambda_G, \lambda_H\rangle$ be an arbitrary trim instance of the Max–(ρ,σ)–Matching–colors problem. Assume first $\Delta_{\min} = \Delta(H)$. According to Vizing's Theorem, H admits a proper edge coloring with at most $\Delta(H)+1$ colors, say $\{c'_1, c'_2, \ldots, c'_{\Delta(H)+1}\}$. For $1 \leq i \leq \Delta(H)+1$, let H_i be the graph obtained from H by deleting all edges but those colored with color c'_i. Notice that H_i is certainly a matching, and hence by resorting to Lemma 9, we can easily obtain a 2-approximation algorithm for the Max–(ρ,σ)–Matching–colors problem for the new instance $(G, H_i, \mathcal{C}, \lambda_G, \lambda_H)$. Furthermore, returning the best one these $\Delta(H)+1$ mappings yields an approximation algorithm with performance ratio $2(\Delta(H)+1)$. If $\Delta_{\min} = \Delta(G)$, we apply the same above arguments to G to obtain an approximation algorithm with performance ratio $2(\Delta(G)+1)$, which completes the proof. □

5.2 A randomized algorithm

We give here a randomized approximation algorithm which achieves a performance ratio of $4\,\sigma$ for the Max–(ρ,σ)–Matching–colors problem, for any ρ and σ. Let $\mathcal{C}$ be a set of colors and G be a graph whose vertices are colored with colors taken from $\mathcal{C}$. Define a *legal (ℓ_1,ℓ_2)-labeling* of G to be an assignment to labels $\{\ell_1,\ell_2\}$ to the vertices of G such that, for each color $c_i \in \mathcal{C}$, either $\left\lfloor \frac{|\mathcal{C}_G(c_i)|}{2} \right\rfloor$ or $\left\lceil \frac{|\mathcal{C}_G(c_i)|}{2} \right\rceil$ vertices in $\mathcal{C}_G(c_i)$ are labeled ℓ_1. Of particular importance here is the fact that it is easy to choose at random a legal (ℓ_1,ℓ_2)-labeling of G. Define the *cut induced by a legal (ℓ_1,ℓ_2)-labeling* to be the set of edges that have one end-vertex with label ℓ_1 and one end-vertex with label ℓ_2.

Consider now an arbitrary trim instance $\langle G, H, \lambda_G, \lambda_H\rangle$ of the Max–(ρ,σ)–Matching–colors problem and let $\theta_{\mathbf{opt}} : \mathbf{V}(G) \xrightarrow{\lambda_G,\lambda_H} \mathbf{V}(H)$ be an optimal solution. Now, let L be a random legal (ℓ_1,ℓ_2)-labeling of G and $C_L \subseteq \mathbf{E}(G)$ be the cut induced by L. Finally, let $E' = C_L \cap \mathsf{match}(G, H, \theta_{\mathbf{opt}})$. Clearly $\mathsf{Exp}[|E'|] \geq \frac{1}{2}|\mathsf{match}(G, H, \theta_{\mathbf{opt}})|$. Combining this with a weighted bipartite matching algorithm yields the following result.

Proposition 11 *There exists a randomized algorithm for the* Max–(ρ,σ)–Matching–Colors *problem with expected performance ratio* $4\,\sigma$.

PROOF. Let $\theta_{\mathbf{opt}} : \mathbf{V}(G) \xrightarrow{\lambda_G,\lambda_H} \mathbf{V}(H)$ be an optimal solution. Fix any random legal (ℓ_1,ℓ_2)-labeling L of G and let $V_1 \subseteq \mathbf{V}(G)$ (resp. $V_2 \subseteq \mathbf{V}(G)$) the set of vertices having label ℓ_1 (resp. ℓ_2). Assign at random the vertices in V_1 to vertices in $\mathbf{V}(H)$, with respect to λ_G and λ_H. We denote by θ this partial assignment. We claim that the three following conditions hold true for at least $(4\,\sigma)^{-1}$ of the edges $e=\{u,v\}$ in $\mathsf{match}(G,H,\theta_{\mathbf{opt}})$: (i) one of the end-vertex of e, say u, has label ℓ_1 and is correctly assigned, *i.e.*, $\theta_{\mathbf{opt}}(u)=\theta(u)$, (ii) one of the end-vertex of e, say v, has label ℓ_2 (and hence is not yet assigned), and (iii) $\theta_{\mathbf{opt}}(v)$ is still free, *i.e.*, no vertex of G with label ℓ_1 has been assigned to it in the first step. Indeed, for any $c_i \in \mathcal{C}$, since at most $\left\lceil \frac{|\mathcal{C}_G(c_i)|}{2} \right\rceil$ of the vertices in $\mathcal{C}_G(c_i)$ have been randomly assigned to the at least $|\mathcal{C}_H(c_i)|$ vertices of H, then it follows that the probability that a vertex of H is the image of a wrong vertex (according to $\theta_{\mathbf{opt}}$) is at most $\frac{1}{2}$. To complete the proof, we notice that the problem of assigning the vertices of V_2 to the remaining vertices in H in such a way to maximize the number of edges in the cut induced by the labeling L that are matched in H according to θ can be solved to optimality in polynomial-time by a natural reduction to weighted bipartite matching. □

5.3 Linear forests

We proved in Section 4 that the Max–$(3,3)$–Matching–Colors problem is APX-hard even if both G and H are linear forests. Furthermore, according to Proposition 10, the Max–(ρ,σ)–Matching–Colors problem for linear forests is approximable within ratio $2(\Delta_{\min}+1)=6$. We strengthen this result here by giving an algorithm that achieves a performance ratio of 4 for the Max–(ρ,σ)–Matching–Colors problem for linear forests. Interestingly enough, the proof make use of weighted 2-intervals sets. More precisely, our approach is based on the 2-Interval–Pattern problem [Via04,CHL+06]. This problem, initially motivated by RNA secondary structure prediction, asks to find a maximum cardinality subset of a 2-interval set with respect to some prespecified geometric constraints.

We need some additional definitions. A 2-*interval* [TH79,BYHN+02,Via04]. is the union of two disjoint intervals and is denoted by $D=(I,J)$ where I and J are two (closed) intervals such that I is completely to the left of J. Two 2-intervals $D_1=(I_1,J_1)$ and $D_2=(I_2,J_2)$ are *disjoint*, if both 2-intervals share no common point. A 2-interval $D=(I,J)$ is said to be *balanced* if $|I|=|J|$, *i.e.*, both intervals have the same length. By abuse of notation, a set of balanced 2-interval is also said to be balanced. Let $\mathcal{D}$ be a set 2-intervals. If

we associate to each 2-interval $D \in \mathcal{D}$ a weight $\omega(D)$, the weight of $\mathcal{D}$, denoted $\omega(\mathcal{D})$, is defined to be the sum of the weights of all the 2-intervals in $\mathcal{D}$.

Let $\langle G, H, \lambda_G, \lambda_H \rangle$ be a trim instance of the Max–(ρ,σ)–Matching–Colors problem where both G and H are linear forests. Let $P_1^G, P_2^G, \ldots, P_k^G$ (resp. $P_1^H, P_2^H, \ldots, P_\ell^H$) be the collection of all paths of G (resp. H). First, we arrange the paths $P_1^G, P_2^G, \ldots, P_k^G$ and next the paths $P_1^H, P_2^H, \ldots, P_\ell^H$ along an horizontal line, arbitrarily. According to this arrangement, we define the *label* (resp. *reversal label*) of any subpath of a path to be string obtained by concatenating the colors (view as letters) of the vertices of the path reading from left to right (resp. right to left). Second, we construct a corresponding set of weighted 2-intervals $\mathcal{D}[G,H]$ as follows. For each pair (Q_i^G, Q_j^H), where Q_i^G is a subpath of length at least one of a path in $\{P_1^G, P_2^G, \ldots, P_k^G\}$ and Q_j^H is a subpath of length at least one of a path in $\{P_1^H, P_2^H, \ldots, P_\ell^H\}$, Q_G^i and Q_j^H having the same length, if the label of Q_i^G is identical to the label of Q_j^H or to the reversal label of Q_j^H, we add to $\mathcal{D}[G,H]$ a 2-interval whose left interval covers all the vertices (and only those vertices) of the subpath Q_i^G and whose right interval covers all the vertices (and only those vertices) of the subpath Q_j^H. The weight of this 2-interval is merely defined to be the length of the subpath Q_i^G (which also the length of the subpath Q_j^H). Without loss of generality, we may assume that each 2-interval in $\mathcal{D}[G,H]$ is balanced and that two 2-intervals that correspond to two vertex-disjoint pairs of subpaths are disjoint. See Figure 2 for an illustration of this construction.

Lemma 12 *There exists a pairwise disjoint subset $\mathcal{D}' \subseteq \mathcal{D}[G,H]$ of weight $\omega(\mathcal{D}')$ if and only if there exists an injective mapping $\theta : \mathbf{V}(G) \xrightarrow{\lambda_G,\lambda_H} \mathbf{V}(H)$ such that $|\mathsf{match}(G,H,\theta)| \geq \omega(\mathcal{D}')$.*

According to Lemma 12 it is thus enough to focus on finding a maximum weighted subset of $\mathcal{D}[G,H]$ of disjoint 2-intervals, which is exactly the 2-Interval–Pattern problem. In [CHL+06], an algorithm with performance ratio 4 is proposed for finding a subset of disjoint 2-intervals in a balanced 2-intervals set. We have thus proved the following.

Corollary 13 *For any ρ and σ, the* Max–(ρ,σ)–Matching–Colors *problem is approximable within ratio* 4 *in case both G and H are linear forests.*

5.4 Approximating the Max–$(2,2)$–Matching–Colors *problem*

This last part is devoted to the NP-hard Max–$(2,2)$–Matching–Colors problem. We prove this special case to be approximable within ratio 1.1442. The basic idea is to transform any instance of the Max–$(2,2)$–Matching–Colors problem into an instance of the the Max–2–CSP problem. An instance of

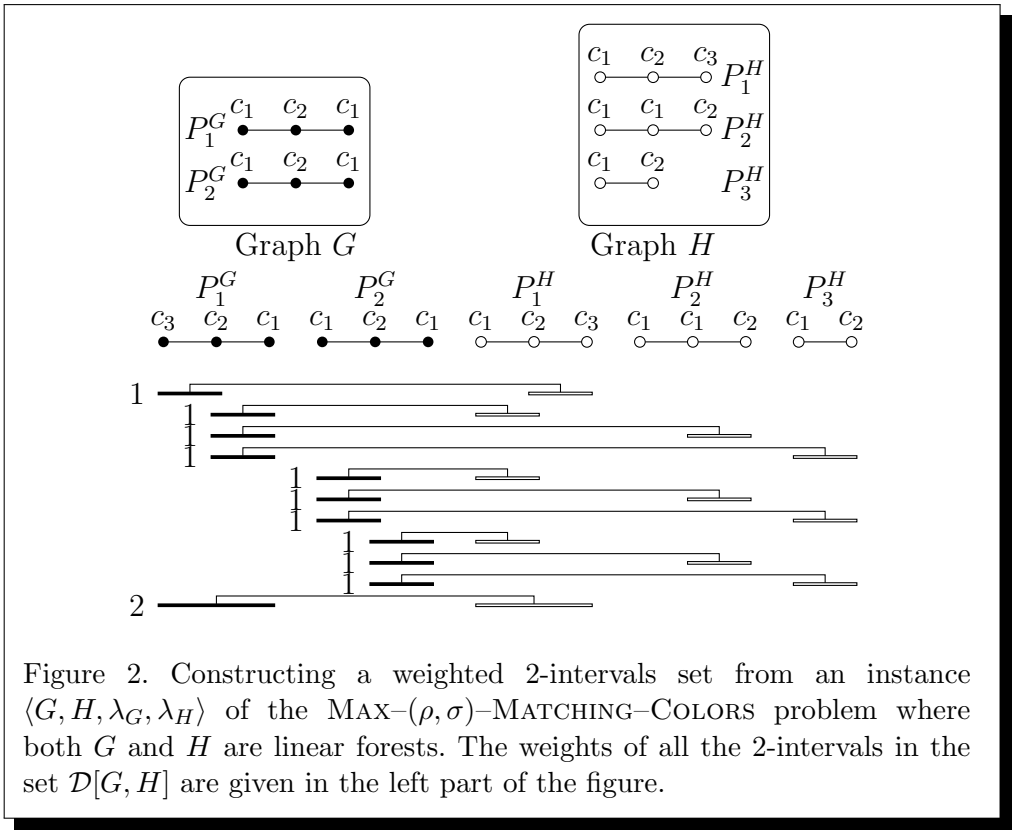

Figure 2. Constructing a weighted 2-intervals set from an instance $\langle G, H, \lambda_G, \lambda_H \rangle$ of the MAX–(ρ, σ)–MATCHING–COLORS problem where both G and H are linear forests. The weights of all the 2-intervals in the set $\mathcal{D}[G, H]$ are given in the left part of the figure.

the (boolean) MAX–CSP problem consists of a set of boolean variables and a collection of constraints which are applied to certain specified subsets of these variables; the goal is to find values for the variables which maximize the number of simultaneously satisfied constraints. For analyzing purposes it is useful to consider restricted subclasses of the MAX–CSP problem. Most importantly, MAX–k–CSP is the maximum constraint satisfaction problem where each constraint depends on at most k variables.

Let $\langle G, H, \lambda_G, \lambda_H \rangle$ be an arbitrary instance of the the MAX–$(2,2)$–MATCHING–COLORS problem. For the sake of simplification, by adding dummy isolated vertices if needed, we assume that each color occurs exactly twice in both G and H. Let X be a set of boolean variables defined by $X = \{x_c : c \in \mathcal{C}\}$, and write $X \cup \overline{X}$ the set of literals over X. Furthermore, define a partial function $f : \mathbf{V}(G) \times \mathbf{V}(H) \to X \cup \overline{X}$ as follows. For each color $c \in \mathcal{C}$, write u and u' the two vertices of G with color c, and v and v' the two vertices of H with color c, and define $f(u, v) = x_c$ and $f(u', v') = \overline{x_c}$, the choice is arbitrary. We now turn to defining the corresponding set of constraints $\mathcal{F}(G, H)$. Let $e = \{u, v\} \in \mathbf{E}(G)$, and write $\lambda_G(u) = c_i$ and $\lambda_G(v) = c_j$. We only need to consider the case where $c_i \neq c_j$ (the case $c_i = c_j$ is indeed trivial if $\rho = 2$ and $\sigma = 2$). Thus, let u' and u'' be the two vertices of H with color c_i, and v' and v'' be the two vertices of H with color c_j. Without loss of generality, assume in addition $f(u, u') = x_{c_i} = x_i$ and $f(v, v') = x_{c_j} = x_j$ (and hence $f(u, u'') = \overline{x_{c_i}} = \overline{x_i}$ and $f(v, v'') = \overline{x_{c_j}} = \overline{x_j}$). We add to $\mathcal{F}(G, H)$ the

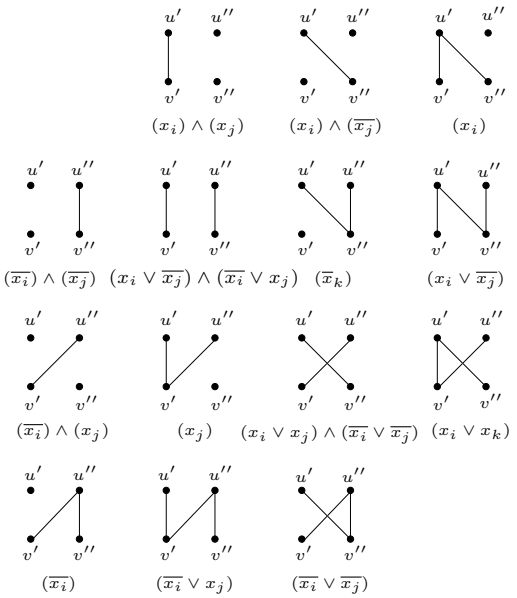

Figure 3. Construction of the constraints in $\mathcal{F}(G, H)$.

constraint f_e defined accordingly to Figure 3. Clearly, there exists an injective mapping $\theta : \mathbf{V}(G) \xrightarrow{\lambda_G, \lambda_H} \mathbf{V}(H)$ that matches k edges of G if and only if k constraints of the constraint set $\mathcal{F}(G, H)$ are simultaneously satisfied. But, each constraint $f_e \in \mathcal{F}(G, H)$ is involved in at most 2 variables, and hence $\mathcal{F}(G, H)$ is an instance of the Max–2–CSP problem. Combining this with the fact that the Max–2–CSP problem is approximable with ratio 1.1442 [LLZ02], we obtain the following proposition.

Proposition 14 *The* Max–$(2, 2)$–Matching–Colors *problem is approximable within ratio* 1.1442*.*

6 Conclusion

In the context of comparative analysis of protein-protein interaction graphs, we considered the problem of finding an occurrence of a given complex in the protein-protein interaction graph of another species. We gave an efficient randomized algorithm in case the mapping is required to be an injective homomorphism. Also, we proved the Max–$(3, 3)$–Matching–Colors problem for linear forests to be APX-hard and we gave an approximation algorithm that achieves a performance ratio of $2(\Delta_{\min} + 1)$, a randomized algorithm with approximation ratio $4\,\sigma$ and a simple approximation algorithm with performance ratio 4 in case both G and H are linear forests.

We mention here some possible directions for future works. First, is it possi-

ble to improve the approximation ratio for bounded degree graphs presented in Proposition 10? Second, due to biological constraints, improving Proposition 11 is of particular interest. In particular, does a deterministic or randomized approximation algorithm with performance ratio σ exist for the Max–(ρ, σ)–Matching–Colors problem?

References

[AK00] P. Alimonti and V. Kann, *Some apx-completeness results for cubic graphs*, Theoretical Computer Science **237** (2000), no. 1-2, 123–134.

[BYHN+02] R. Bar-Yehuda, M.M. Halldorsson, J. Naor, H. Shachnai, and I. Shapira, *Scheduling split intervals*, Proc. 13th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), 2002, pp. 732–741.

[CHL+06] M. Crochemore, D. Hermelin, G. Landau, D. Rawitz, and S. Vialette, *Approximating the 2-interval pattern problem*, Theoretical Computer Science (special issue for Alberto Apostolico) (2006), To appear.

[Die00] R. Diestel, *Graph theory*, second ed., Graduate texts in Mathematics, no. 173, Springer-Verlag, 2000.

[FLV04] I. Fagnot, G. Lelandais, and S. Vialette, *Bounded list injective homomorphism for comparative analysis of protein-protein interaction graphs*, Proc. 1st Algorithms and Computational Methods for Biochemical and Evolutionary Networks (CompBioNets), KCL publications, 2004, pp. 45–70.

[FRV05] G. Fertin, R. Rizzi, and S. Vialette, *Finding exact and maximum occurrences of protein complexes in protein-protein interaction graphs*, Proc. 30th International Symposium on Mathematical Foundations of Computer Science (MFCS), Lecture Notes in Computer Science, vol. 3618, 2005, pp. 328–339.

[GB+02] A.C. Gavin, M. Boshe, et al., *Functional organization of the yeast proteome by systematic analysis of protein complexes*, Nature **414** (2002), no. 6868, 141–147.

[HG+02] Y. Ho, A. Gruhler, et al., *Systematic identification of protein complexes in Saccharomyces cerevisae by mass spectrometry*, Nature **415** (2002), no. 6868, 180–183.

[KSK+03] B.P. Kelley, R. Sharan, R.M. Karp, T. Sittler, D. E. Root, B.R. Stockwell, and T. Ideker, *Conserved pathways within bacteria and yeast as revealed by global protein network alignment*, PNAS **100** (2003), no. 20, 11394–11399.

[LLZ02] M. Lewin, D. Livnat, and U. Zwick, *Improved rounding techniques for the max 2-sat and max di-cut problems*, Proc. 9th International Conference on Integer Programming and Combinatorial Optimization (IPCO), Lecture Notes in Computer Science, vol. 2337, 2002, pp. 67–82.

[MU05] M. Mitzenmacher and E. Upfal, *Probability and computing: Randomized algorithms and probabilistic analysis*, Cambridge University Press, 2005.

[PLEO04] J.B. Pereira-Leal, A.J. Enright, and C.A. Ouzounis, *Detection of functional modules from protein interaction networks*, Proteins **54** (2004), no. 1, 49–57.

[PMT+99] M. Pellegrini, E.M. Marcotte, M.J. Thompson, D. Eisenberg, and T.O. Yeates, *Assigning protein functions by comparative genome analysis: protein phylogenetic profiles*, PNAS **96** (1999), no. 8, 4285–4288.

[PY91] C.H. Papadimitriou and M. Yannakakis, *Optimization, approximation and complexity classes*, J. of Computer and System Sciences **43** (1991), 425–440.

[RSS01] M. Remm, C.E.V. Storm, and E.L.L. Sonnhammer, *Automatic clustering of orhologs and in-paralogs from pairwise species comparisons*, Journal of Molecular Biology **314** (2001), 1041–1052.

[SIK+04] R. Sharan, T. Ideker, B. Kelley, R. Shamir, and R.M. Karp, *Identification of protein complexes by comparative analysis of yeast and bacterial protein interaction data*, Proc. 8th annual international conference on Computational molecular biology (RECOMB), ACM Press, 2004, pp. 282–289.

[SSK+04] R. Sharan, S. Suthram, R.M. Kelley, T. Kuhn, S. McCuin, P. Uetz, T. Sittler, R. Karp, and T. Ideker, *Conserved patterns of protein interaction in multiple species*, PNAS **102** (2004), no. 6, 1974–1979.

[TH79] W.T. Trotter and F. Harary, *On double and multiple interval graphs*, J. Graph Theory **3** (1979), 205–211.

[TKL97] R.L. Tatusov, E.V. Koonin, and D.J. Lipman, *A genomic perspective on protein families*, Science **278** (1997), no. 5338, 631–637.

[TSU04] B. Titz, M. Schlesner, and P. Uetz, *What do we learn from high-throughput protein interaction data?*, Expert Review of Anticancer Therapy **1** (2004), no. 1, 111–121.

[UG+00] P. Uetz, L. Giot, et al., *A comprehensive analysis of protein-protein interactions in Saccharomyces cerevisae*, Nature **403** (2000), no. 6770, 623–627.

[Via04] S. Vialette, *On the computational complexity of* 2*-interval pattern matching*, Theoretical Computer Science **312** (2004), no. 2-3, 223–249.

[Viz64] V.G. Vizing, *On an estimate of the chromatic class of a p-graph*, Diskret. Analiz **3** (1964), 23–30, (in Russian).