



HAL
open science

Finding Occurrences of Protein Complexes in Protein-Protein Interaction Graphs

Guillaume Fertin, Romeo Rizzi, Stéphane Vialette

► **To cite this version:**

Guillaume Fertin, Romeo Rizzi, Stéphane Vialette. Finding Occurrences of Protein Complexes in Protein-Protein Interaction Graphs. *Journal of Discrete Algorithms*, 2009, 7 (1), pp.90-101. <hal-00416394>

HAL Id: hal-00416394

<https://hal.science/hal-00416394v1>

Submitted on 14 Sep 2009

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

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



HAL Authorization

Finding Occurrences of Protein Complexes in Protein-Protein Interaction Graphs ^{★,★★}

Guillaume Fertin ^a, Romeo Rizzi ^b and Stéphane Vialette ^c

^a*Laboratoire d'Informatique de Nantes-Atlantique (LINA), CNRS UMR 6241, Université de Nantes, 2 rue de la Houssinière, 44322 Nantes Cedex 3, France*

`Guillaume.Fertin@univ-nantes.fr`

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

`Romeo.Rizzi@dimi.uniud.it`

^c*IGM-LabInfo, CNRS UMR 8049, Université Paris-Est, 5 Bd Descartes 77454 Marne-la-Vallée, France*

`vialette@univ-mlv.fr`

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 G in the protein-protein interaction graph H of another species with respect to (w.r.t.) orthologous proteins. Two problems are considered: the EXACT- (μ_G, μ_H) -MATCHING problem and the MAX- (μ_G, μ_H) -MATCHING problems, where μ_G (resp. μ_H) denotes in both problems the maximum number of orthologous proteins in H (resp. G) of a protein in G (resp. H). Following [10], the EXACT- (μ_G, μ_H) -MATCHING problem asks for an injective homomorphism of G to H w.r.t. orthologous proteins. The optimization version is called the MAX- (μ_G, μ_H) -MATCHING problem and is concerned with finding an injective mapping of a graph G to a graph H w.r.t. orthologous proteins that matches as many edges of G as possible. For both problems, we essentially focus on bounded degree graphs and extremal small values of parameters μ_G and μ_H .

Key words: Computational biology, computational complexity, approximation algorithm, parameterized complexity, protein-protein interaction graph.

[★] A preliminary version appeared in the Proceedings of the 30th International Symposium on Mathematical Foundations of Computer Science (MFCS 2005), Gdansk, Poland, August 2005. LNCS Vol. 3618, pp 328-339.

^{★★}This work was partially supported by the French-Italian PAI Galileo project number 08484VH.

1 Introduction

High-throughput analysis makes possible the study of protein-protein interactions at a genome-wide scale [13,15,27], 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 [21].

Protein interactions identified on a genome-wide scale are commonly visualized as protein interaction graphs, where proteins are vertices and interactions are edges [26]. 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 [22], 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.* [17] developed the program PathBlast, which aligns two protein-protein interaction graphs combining topology and sequence similarity. Sharan *et al.* [24] studied the conservation of complexes (they focused on dense, clique-like interaction patterns) that are conserved in *Saccharomyces cerevisiae* 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. Pathways detection is considered in [28] and [23]. 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 cerevisiae* is presented in [25] (see also [18] and [16]).

Following the line of research presented in [10], we consider here the problem of finding an occurrence of a given complex in the protein-protein interaction graph of another species. Notice that we do not make any assumption about the topology of the complex, such as clique-like structure. In [10], this is formulated as the problem of searching for a list injective homomorphism, *i.e.*, an injective homomorphism with respect to orthologous links, of the complex (viewed as a graph) to a protein-protein interaction graph. The special case where all lists are required to be either equal or disjoint is considered in [7].

Roughly speaking, the rationale of this 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, we need to impose *drastic restrictions* on the size of the lists. We will make the following important assumption (referred hereafter as the parameters μ_G and μ_H): no protein has an unbounded number of orthologs in the other species, *i.e.*, each list has a constant size (upper bounded by parameter μ_G) and each protein has a constant number of occurrences among the lists (upper bounded by parameter μ_H). We, however, observe that graph homomorphisms would in practice be too strict in detecting protein complex homology. Also, a single protein in one species might be associated with a number of orthologs in another species. Moreover, the scoring function may be far more complex than just counting the number of conserved interactions. The present paper is devoted to analyzing the complexity of this problem (the EXACT- (μ_G, μ_H) -MATCHING problem) together with its natural optimization version (the MAX- (μ_G, μ_H) -MATCHING problem) in case of bounded degree graphs and extremal small values of parameters μ_G and μ_H .

The paper is organized as follows: Section 2 introduces formally the two problems. We prove in Section 3 new tight complexity results for the EXACT- (μ_G, μ_H) -MATCHING problem for bounded degree graphs and introduce the correspondence number of an instance. In Section 4, it is shown that the MAX- (μ_G, μ_H) -MATCHING problem for bounded degree graphs is **APX**-hard. This result is complemented in Section 5 by showing that the MAX- $(\mu_G, 1)$ -MATCHING problem for bounded degree graphs is approximable with constant ratio. Finally, we prove in Section 6 that the MAX- $(\mu_G, 1)$ -MATCHING problem for bounded degree graphs parameterized by the number of matched edges is fixed-parameter tractable.

2 Preliminaries

Let G be a graph. We write $\mathbf{V}(G)$ for the set of vertices and $\mathbf{E}(G)$ for the set of edges, and abbreviate $\#\mathbf{V}(G)$ to $\mathbf{n}(G)$ and $\#\mathbf{E}(G)$ to $\mathbf{m}(G)$. The maximum degree $\Delta(G)$ of a graph G is the largest degree over all vertices. Let G and H be two graphs. For any injective mapping $\theta : \mathbf{V}(G) \rightarrow \mathbf{V}(H)$, let us denote by $\text{match}(G, H, \theta)$ the edges of G that are matched by θ , *i.e.*, $\text{match}(G, H, \theta) = \{\{u, v\} \in \mathbf{E}(G) : \{\theta(u), \theta(v)\} \in \mathbf{E}(H)\}$. An *homomorphism*

of G to H is a mapping $\theta : \mathbf{V}(G) \rightarrow \mathbf{V}(H)$ such that $\{u, v\} \in \mathbf{E}(G)$ implies $\{\theta(u), \theta(v)\} \in \mathbf{E}(H)$. Clearly, an injective mapping θ is an homomorphism of G to H if $\#\text{match}(G, H, \theta) = \mathbf{m}(G)$. Given lists $\mathcal{L}(u) \subseteq \mathbf{V}(H)$, $u \in \mathbf{V}(G)$, a *list homomorphism* of G to H with respect to the lists $\mathcal{L}(u)$, $u \in \mathbf{V}(G)$, is an homomorphism θ with the additional constraint that $\theta(u) \subseteq \mathcal{L}(u)$ for all $u \in \mathbf{V}(G)$. Mappings of G to H with respect to the lists $\mathcal{L}(u)$, $u \in \mathbf{V}(G)$, are defined in a similar way. For simplicity of notation, given lists $\mathcal{L}(u) \subseteq \mathbf{V}(H)$, $u \in \mathbf{V}(G)$, we abbreviate $\{u : v \in \mathcal{L}(u)\}$ to $\mathcal{L}^{-1}(v)$, $v \in \mathbf{V}(H)$. Let G and H be two graphs. Lists $\mathcal{L}(u) \subseteq \mathbf{V}(H)$, $u \in \mathbf{V}(G)$, are called (μ_G, μ_H) -*bounded* if the two following conditions hold true: (1) $\max\{\#\mathcal{L}(u) : u \in \mathbf{V}(G)\} \leq \mu_G$ and (2) $\max\{\#\mathcal{L}^{-1}(v) : v \in \mathbf{V}(H)\} \leq \mu_H$.

We consider here the problem of finding an occurrence of a given complex in the protein-protein interaction graph of another species. Finding an occurrence with respect to orthologous links can easily be reformulated as a list injective homomorphism problem: 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. The problem, called the EXACT- (μ_G, μ_H) -MATCHING problem, is defined formally as follows.

EXACT- (μ_G, μ_H) -MATCHING

- **Input** : Two graphs G and H , and (μ_G, μ_H) -bounded lists $\mathcal{L}(u) \subseteq \mathbf{V}(H)$, $u \in \mathbf{V}(G)$.
- **Question** : Is there an injective list homomorphism of G to H w.r.t. lists $\mathcal{L}(u)$, $u \in \mathbf{V}(G)$?

In the context of comparative analysis of protein-protein interaction graphs, we need to impose strong restrictions on the size of the lists we consider. We thus assume, throughout the paper, that both μ_G and μ_H are constant, *i.e.*, $\mu_G = O(1)$ and $\mu_H = O(1)$.

It is proved in [10] that the EXACT- $(2, \mu_H)$ -MATCHING problem is linear-time solvable for any constant $\mu_H \geq 1$, and that the EXACT- $(3, 1)$ -MATCHING problem is **NP**-complete even if both G and H are bipartite graphs or split graphs. A first contribution in this paper is to complete the determination of the precise border between tractable and intractable cases for the EXACT- (μ_G, μ_H) -MATCHING problem. Moreover, we begin here the analysis of optimization versions of the problem. Indeed, requiring an injective homomorphism, *i.e.*, an injective mapping that preserves *all* edges of G , might result in an over-constrained problem, though it may exist good approximate solutions, *i.e.*, solutions that match many edges of G . This suggests the following

maximization problem for practical applications.

MAX- (μ_G, μ_H) -MATCHING

- **Input** : Two graphs G and H , and (μ_G, μ_H) -bounded lists $\mathcal{L}(u) \subseteq \mathbf{V}(H)$, $u \in \mathbf{V}(G)$.
- **Solution** : An injective mapping $\theta : \mathbf{V}(G) \rightarrow \mathbf{V}(H)$ w.r.t. lists $\mathcal{L}(u)$, $u \in \mathbf{V}(G)$.
- **Measure** : $\#\text{match}(G, H, \theta)$, i.e., $\#\{\{u, v\} \in \mathbf{E}(G) : \{\theta(u), \theta(v)\} \in \mathbf{E}(H)\}$.

Of particular importance here is the fact that θ is no longer required to be a homomorphism in the MAX- (μ_G, μ_H) -MATCHING problem. Furthermore, the present paper mainly focuses on a particular case of the optimization problem, i.e., the MAX- $(\mu_G, 1)$ -MATCHING problem.

Let $\langle G, H, \mathcal{L} \rangle$ be an instance of the MAX- (μ_G, μ_H) -MATCHING. An edge $\{u, v\} \in \mathbf{E}(G)$ is called a *bad edge* if there does not exist distinct $u' \in \mathcal{L}(u)$ and $v' \in \mathcal{L}(v)$ 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- (μ_G, μ_H) -MATCHING problem, since bad edges can never be matched. Notice that we can tell bad edges apart in $O(\mu_G^2 \mathbf{m}(G)) = O(\mathbf{m}(G))$ time, since μ_G is assumed to be a constant. Furthermore, by resorting on classical bipartite matching techniques, we can check in $O(\mathbf{n}(H) + \mathbf{m}(G) \sqrt{\mathbf{n}(G)})$ time whether there exists at least an injective mapping of G to H w.r.t. lists $\mathcal{L}(u)$, $u \in \mathbf{V}(G)$. Moreover, before solving the problem, we can surely remove from H all those nodes u' with $\#\mathcal{L}^{-1}(u') = 0$. Therefore, throughout the paper, we will consider only trim instances as defined in the following.

Definition 1 (Trim instance) *An instance $\langle G, H, \mathcal{L} \rangle$ of the MAX- (μ_G, μ_H) -MATCHING problem is a trim instance provided that (i) there exists an injective mapping of G to H w.r.t. lists $\mathcal{L}(u)$, $u \in \mathbf{V}(G)$, (ii) $\#\mathcal{L}^{-1}(u') > 0$ for all $u' \in \mathbf{V}(H)$ and (iii) G does not contain any bad edges.*

3 Exact matching

This section is devoted to completing the determination of the precise border between tractable and intractable cases for the EXACT- (μ_G, μ_H) -MATCHING problem [10]. Also, we introduce the *correspondence number* of any instance of the EXACT- $(\mu_G, 1)$ -MATCHING problem which aims at separating yes instances from possibly no instances.

3.1 Complexity issues

We begin by giving a straightforward algorithm for the EXACT- $(\mu_G, 1)$ -MATCHING problem in case $\Delta(G) \leq 2$.

Proposition 2 *The EXACT- $(\mu_G, 1)$ -MATCHING problem for $\Delta(G) \leq 2$ is solvable in $O(\mathbf{n}(G))$ time for any constant μ_G .*

PROOF. Since $\mu_H = 1$, there is no loss of generality in assuming that G is a connected graph (for otherwise we can process each connected component independently). Furthermore, since $\Delta(G) = 2$, G is either a path or a cycle.

Let us first suppose that G is a path of length k . Write $\mathbf{V}(G) = \{u_1, u_2, \dots, u_{k+1}\}$ such that $\{u_i, u_{i+1}\} \in \mathbf{E}(G)$ for $1 \leq i \leq k$. For each $v \in \mathcal{L}(u_i)$, $1 \leq i \leq k+1$, define $T(v)$ to be true if and only if there exists an injective homomorphism of $G[\{u_1, u_2, \dots, u_i\}]$ to $H[\cup_{1 \leq j \leq i} \mathcal{L}(u_j)]$ w.r.t. lists $\mathcal{L}(u_j)$, $1 \leq j \leq i$ (where $G[V']$ denotes the subgraph of G induced by the set $V' \subseteq \mathbf{V}(G)$).

Clearly,

$$\begin{aligned} \forall v \in \mathcal{L}(u_1), \quad T(v) &= \text{true} \\ \forall 1 < i \leq k+1, \quad \forall v \in \mathcal{L}(u_i), \quad T(v) &= \bigvee_{\substack{v' \in \mathcal{L}(u_{i-1}) \\ \{v', v\} \in \mathbf{E}(H)}} T(v') \end{aligned}$$

and there is an injective homomorphism of G to H w.r.t. lists $\mathcal{L}(u)$, $u \in \mathbf{V}(G)$, if $\bigvee_{v \in \mathcal{L}(u_k)} T(v) = \text{true}$. This is a $O(\mu_G^2 k) = O(\mathbf{n}(G))$ time dynamic programming algorithm.

Suppose now that G is a cycle of length $k+2$. Write $\mathbf{V}(G) = \{u_1, u_2, \dots, u_{k+1}\}$ such that $\{u_i, u_{i+1}\} \in \mathbf{E}(G)$ for $1 \leq i \leq k$, and $\{u_{k+1}, u_1\} \in \mathbf{E}(G)$. For any $v \in \mathcal{L}(u_1)$ and any $v' \in \mathcal{L}(u_{k+1})$, let us denote by $\Pi(v, v')$ the sub-problem obtained (i) by deleting the edge $\{u_{k+1}, u_1\}$ in G , (ii) by deleting all vertices in $\mathcal{L}(u_1)$ but v , and (iii) by deleting all vertices in $\mathcal{L}(u_{k+1})$ but v' . We have at most μ_G^2 sub-problems, each of them can be solved in $O(\mathbf{n}(G))$ time using the above dynamic programming algorithm. We now observe that there is an injective homomorphism of G to H w.r.t. lists $\mathcal{L}(u)$, $u \in \mathbf{V}(G)$, if for some $v \in \mathcal{L}(u_1)$ and $v' \in \mathcal{L}(u_{k+1})$ with $\{v, v'\} \in \mathbf{E}(H)$, $\Pi(v, v')$ is a positive instance. This is a $O(\mu_G^2 \mathbf{n}(G)) = O(\mathbf{n}(G))$ time algorithm. \square

It follows from the above proposition that the EXACT- $(\mu_G, 1)$ -MATCHING problem for $\Delta(H) \leq 2$ is polynomial-time solvable. Indeed, if $\Delta(G) \geq 3$, the answer is trivially **no** and otherwise, *i.e.*, $\Delta(G) \leq 2$, Proposition 2 applies.

One may however argue that the proposition above is too constrained to be of interest. Unfortunately, despite the simplicity of Proposition 2, the result is quite tight - taking into consideration both $\Delta(G)$ and $\Delta(H)$ - as shown in the two following propositions (recall also that the EXACT-(2, μ_H)-MATCHING problem is polynomial-time solvable for any constant μ_H [10]).

Proposition 3 *The EXACT-(3, 2)-MATCHING problem is NP-complete even if both G and H are bipartite graphs with $\Delta(G) \leq 1$ and $\Delta(H) \leq 2$.*

PROOF. The EXACT-(3, 2)-MATCHING problem is easily seen to be in NP. The reduction is from the 3-SAT problem. We assume the additional restriction that each variable appears in at most 3 of the clauses, counting together both positive and negative occurrences. It is known that the 3-SAT problem is NP-complete even when restricted as above [12]. Notice furthermore that we can always assume that each negated literal and each positive literal occurs at most twice, since otherwise there would be a variable without positive or without negative occurrences, and hence a self-reduction would apply. Assume given an input ϕ to the 3-SAT problem. Let $X = \{x_1, \dots, x_n\}$ denote the set of variables and $C = \{c_1, \dots, c_m\}$ denote the set of clauses. We now describe how to construct the corresponding instance of the EXACT-(3, 2)-MATCHING problem.

To ϕ we associate a bipartite graph, denoted G - which in fact is a matching - as follows. For each variable $x_i \in X$, we introduce two vertices $x_i^G[1]$ and $x_i^G[2]$, and one edge $\{x_i^G[1], x_i^G[2]\}$. For each clause $c_j \in C$, we introduce two vertices $c_j^G[1]$ and $c_j^G[2]$, and one edge $\{c_j^G[1], c_j^G[2]\}$. To ϕ we also associate a second bipartite graph, denoted H , as follows. For each variable $x_i \in X$, we introduce four vertices $x_i^H[T, 1]$, $x_i^H[T, 2]$, $x_i^H[F, 1]$ and $x_i^H[F, 2]$, and the two edges $\{x_i^H[T, 1], x_i^H[T, 2]\}$ and $\{x_i^H[F, 1], x_i^H[F, 2]\}$. For each clause $c_j \in C$, we introduce three vertices $c_j^H[1]$, $c_j^H[2]$ and $c_j^H[3]$, and also three edges defined as follows. For $\ell \in \{1, 2, 3\}$, let \hat{x}_i be the ℓ -th literal of the clause c_j . Assume \hat{x}_i is the p -th positive (or, resp., negative) occurrence of variable x_i , where $p \in \{1, 2\}$. Then we introduce the edge $\{c_j^H[\ell], x_i^H[T, p]\}$ (or, resp., $\{c_j^H[\ell], x_i^H[F, p]\}$). Notice that for each $j \in \{1, 2, \dots, m\}$ and $\ell \in \{1, 2, 3\}$, vertex $c_j^H[\ell]$ has a unique neighbor in H . For ease of exposition, we denote by $N(c_j^H[\ell])$ this unique neighbor. We now turn to describing the associated lists. To each $x_i^G[p] \in \mathbf{V}(G)$, $1 \leq p \leq 2$, we associate the list $\mathcal{L}(x_i^G[p]) = \{x_i^H[T, p], x_i^H[F, p]\}$. To each $c_j^G[2] \in \mathbf{V}(G)$, we associate the list $\mathcal{L}(c_j^G[2]) = \{c_j^H[\ell] : 1 \leq \ell \leq 3\}$. Finally, to each $c_j^G[1] \in \mathbf{V}(G)$, we associate the list $\mathcal{L}(c_j^G[1]) = \{N(c_j^H[\ell]) : 1 \leq \ell \leq 3\}$.

Clearly, $\mu_G = 3$, $\mu_H = 2$, $\Delta(G) = 1$, *i.e.*, G is a matching, and $\Delta(H) = 2$ (H is indeed made of paths of length at most 3). An illustration of the construction is given in Figure 1 for the CNF formula $\phi = (x_1 \vee \bar{x}_2 \vee \bar{x}_3) \wedge (\bar{x}_1 \vee x_2 \vee \bar{x}_3) \wedge$

$(x_1 \vee x_2 \vee x_3)$. We claim that there exists a satisfying truth assignment for ϕ if and only if there exists an injective list homomorphism of G to H w.r.t. lists $\mathcal{L}(u)$, $u \in \mathbf{V}(G)$.

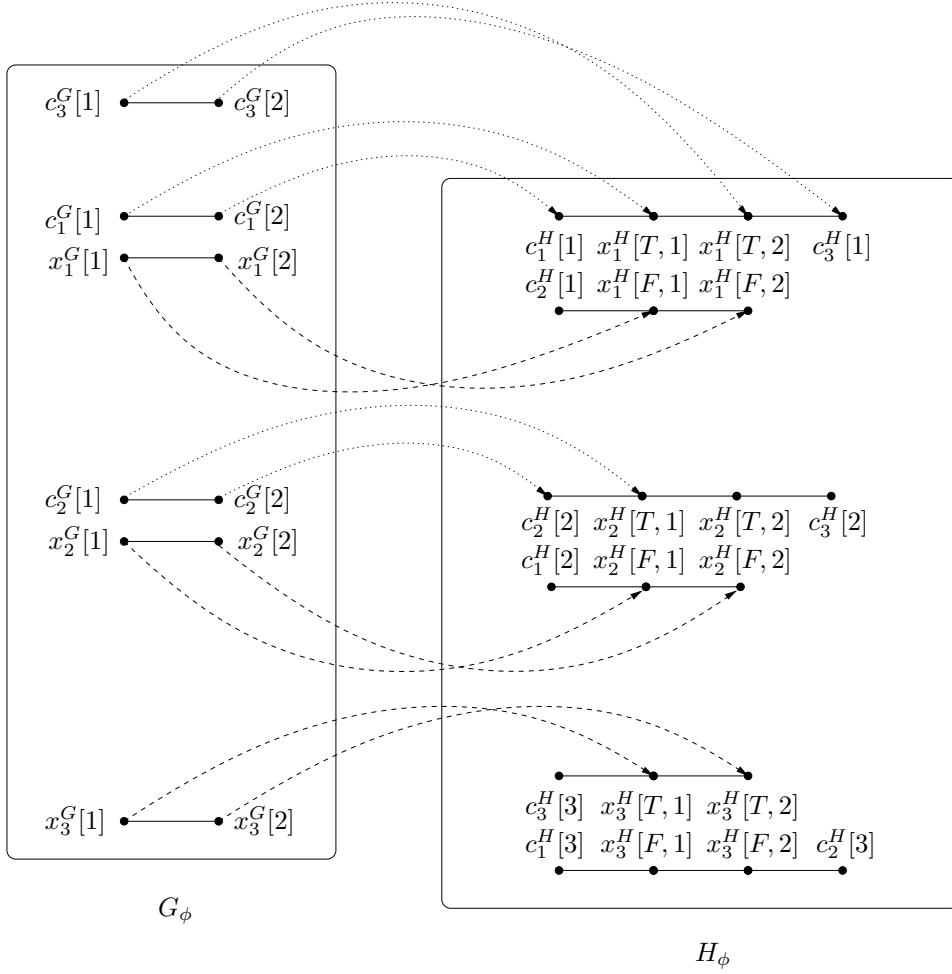


Figure 1. Illustration of the proof of Proposition 3 for the boolean formula $\phi = (x_1 \vee \bar{x}_2 \vee \bar{x}_3) \wedge (\bar{x}_1 \vee x_2 \vee \bar{x}_3) \wedge (x_1 \vee x_2 \vee x_3)$. Both G and H are bipartite graphs, and $\Delta(G) = 1$ and $\Delta(H) = 2$. Shown here is the satisfying truth assignment $f : X \rightarrow \{\text{true}, \text{false}\}$ defined by $f(x_1) = \text{true}$, $f(x_2) = \text{true}$ and $f(x_3) = \text{false}$, together with the injective mapping θ of G to H (denoted here by dashed and dotted lines).

Let $f : X \rightarrow \{\text{true}, \text{false}\}$ be a truth assignment for ϕ that satisfies all clauses. If $f(x_i) = \text{true}$, then define $\theta(x_i^G[1]) = x_i^H[F, 1]$ and $\theta(x_i^G[2]) = x_i^H[F, 2]$, else define $\theta(x_i^G[1]) = x_i^H[T, 1]$ and $\theta(x_i^G[2]) = x_i^H[T, 2]$. For every clause c_j , take an $\ell \in \{1, 2, 3\}$ such that the ℓ -th literal of c_j evaluates to **true** under f , and define $\theta(c_j^G[2]) = c_j^H[\ell]$ and $\theta(c_j^G[1]) = N(c_j^H[\ell])$. It can be easily verified that θ is an injective homomorphism of G to H w.r.t. lists $\mathcal{L}(u)$, $u \in \mathbf{V}(G)$.

Conversely, suppose that there is an injective list homomorphism θ of G to H w.r.t. lists $\mathcal{L}(u)$, $u \in \mathbf{V}(G)$. We first observe that, by construction, we must

have $\theta(x_i^G[1]) = x_i^H[T, 1]$ and $\theta(x_i^G[2]) = x_i^H[T, 2]$, or $\theta(x_i^G[1]) = x_i^H[F, 1]$ and $\theta(x_i^G[2]) = x_i^H[F, 2]$, for all $1 \leq i \leq n$, since $\{x_i^G[1], x_i^G[2]\} \in \mathbf{E}(G)$. Define a truth assignment $f : X \rightarrow \{\text{true}, \text{false}\}$ as follows: If $\theta(x_i^G[1]) = x_i^H[F, 1]$ then $f(x_i) = \text{true}$, else define $f(x_i) = \text{false}$, for all $1 \leq i \leq n$. We claim that f is a satisfying truth assignment for ϕ . Indeed, for any clause c_j , let $\ell \in \{1, 2, 3\}$ be such that $c_j^H[\ell] = \theta(c_j^G[1])$. Clearly, the ℓ -th literal of ϕ evaluates to **true** under the truth assignment f . \square

Proposition 4 *The EXACT-(3, 1)-MATCHING problem is NP-complete even when $\Delta(G) \leq 3$ and $\Delta(H) \leq 4$.*

PROOF. It is well-known that deciding whether a graph G of maximum degree 3 has chromatic number 3 is NP-complete [12]. It follows that, when given a graph G' , $\Delta(G') = 3$, and a subset $E' \subseteq \mathbf{E}(G')$ deciding whether we can assign one of 3 possible colors to each node in $\mathbf{V}(G')$ in such a way that every two adjacent nodes $u, v \in \mathbf{V}(G')$ have the same color if and only if $\{u, v\} \in E'$ is NP-complete even when $\mathbf{E}(G') \setminus E'$ is a matching in G' and the edges in E' form vertex-disjoint paths in G' . Indeed, starting from G , explode each node $v \in \mathbf{V}(G)$ into a path P'_v with as many nodes as the degree of v in G . Put all edges of each path P'_v into E' to force all nodes in P'_v to search for a common color. The matching $\mathbf{E}(G') \setminus E'$ will be (arbitrarily) chosen as to contain an edge with an endpoint in P'_u and the other in P'_v if and only if $\{u, v\}$ is an edge in G .

Now, starting from the pair (G', E') , we show how to construct an “equivalent” instance $\langle G, H, \mathcal{L} \rangle$ of the EXACT-(3, 1)-MATCHING problem with $\Delta(G) = 3$ and $\Delta(H) = 4$. Take $G = G'$ and let H be the graph defined by

$$\begin{aligned} \mathbf{V}(H) &= \bigcup_{u \in \mathbf{V}(G)} \{u_1, u_2, u_3\} \\ \mathbf{E}(H) &= \left(\bigcup_{\{u, v\} \in E'} \{\{u_i, v_i\} : 1 \leq i \leq 3\} \right) \cup \\ &\quad \left(\bigcup_{\{u, v\} \in E(G) \setminus E'} \{\{u_i, v_j\} : 1 \leq i \leq 3 \wedge 1 \leq j \leq 3 \wedge i \neq j\} \right) \end{aligned}$$

Clearly, $\Delta(G) = \Delta(G') = 3$, and $\Delta(H) = 4$ since $E(G) \setminus E'$ is a matching in G' and no vertex of G' is adjacent to more than two edges in E' . Moreover, by taking $\mathcal{L}(u) = \{u_1, u_2, u_3\}$ for each $u \in \mathbf{V}(G)$, it is guaranteed that there exists a mapping $c : \mathbf{V}(G') \rightarrow \{1, 2, 3\}$ with $c(u) = c(v)$ whenever $\{u, v\} \in E'$, and $c(u) \neq c(v)$ whenever $\{u, v\} \in E \setminus E'$ if and only if there exists an injective homomorphism of G to H w.r.t. lists $\mathcal{L}(u)$, $u \in \mathbf{V}(G)$. Hence, the two instances $\langle G', E' \rangle$ and $\langle G, H, \mathcal{L} \rangle$ are equivalent in the sense that solving one solves also the other, since they have the same answer. Notice moreover that $\mathcal{L}(v)$ and

$\mathcal{L}(u)$ are disjoint whenever $u \neq v$, and hence $\#\mathcal{L}^{-1}(u_i) = 1$ for each $u_i \in \mathbf{V}(H)$ under the assumption of trim instance. \square

3.2 The correspondence number

The remainder of this section is devoted to the EXACT- $(\mu_G, 1)$ -MATCHING problem. For each trim instance $\langle G, H, \mathcal{L} \rangle$ of the EXACT- $(\mu_G, 1)$ -MATCHING problem, define the *correspondence number* $C(G, H, \mathcal{L})$ of (G, H, \mathcal{L}) by

$$C(G, H, \mathcal{L}) = \min_{\{u, v\} \in \mathbf{E}(G)} \frac{\#\{\{u', v'\} : u' \in \mathcal{L}(u) \wedge v' \in \mathcal{L}(v) \wedge \{u', v'\} \in \mathbf{E}(H)\}}{\#\mathcal{L}(u) \#\mathcal{L}(v)}.$$

Clearly, $\mu_G^{-2} \leq C(G, H, \mathcal{L}) \leq 1$ (the lower bound comes from the fact that $\langle G, H, \mathcal{L} \rangle$ is a trim instance). Furthermore, if $C(G, H, \mathcal{L}) = 1$, then there exists an injective homomorphism θ of G to H w.r.t. lists $\mathcal{L}(u)$, $u \in \mathbf{V}(G)$; any injective mapping of G to H w.r.t. lists $\mathcal{L}(u)$, $u \in \mathbf{V}(G)$, is indeed a solution. Ideally, one would like to determine a bound c^* as small as possible, $\mu_G^{-2} < c^* < 1$, such that if $C(G, H, \mathcal{L}) > c^*$ then $\langle G, H, \mathcal{L} \rangle$ is a **yes** instance and if $C(G, H, \mathcal{L}) \leq c^*$ then $\langle G, H, \mathcal{L} \rangle$ is possibly a **no** instance. Unfortunately, it is difficult to obtain such a precise bound and we thus focus here on the determination of two bounds c_{low} and c_{up} , $\mu_G^{-2} \leq c_{\text{low}} \leq c_{\text{up}} \leq 1$, such that if $C(G, H, \mathcal{L}) > c_{\text{up}}$ then $\langle G, H, \mathcal{L} \rangle$ is a **yes** instance and if $C(G, H, \mathcal{L}) \leq c_{\text{low}}$ then $\langle G, H, \mathcal{L} \rangle$ is possibly a **no** instance. Of course, the smaller c_{up} and $c_{\text{up}} - c_{\text{low}}$ are, the better our estimation is. We propose here two bounds c_{up} and c_{low} with $c_{\text{up}} - c_{\text{low}} = \frac{1-e^{-1}}{\Delta(G)-1}$.

Proposition 5 *Let $\langle G, H, \mathcal{L} \rangle$ be a trim instance of the EXACT- $(\mu_G, 1)$ -MATCHING problem. If*

$$C(G, H, \mathcal{L}) > \frac{2\Delta(G) - 1 - e^{-1}}{2\Delta(G) - 1}$$

then there exists an injective homomorphism θ of G to H w.r.t. lists $\mathcal{L}(u)$, $u \in \mathbf{V}(G)$.

PROOF. The proof is by direct application of the Lovász local lemma [9]. For each $u \in \mathbf{V}(G)$ with $\mathcal{L}(u) = \{u_1, u_2, \dots, u_q\}$, $q \leq \mu_G$, suppose that $\theta(u)$ is set to u_1, u_2, \dots , or u_q independently and equiprobably. Since $\mu_H = 1$, it follows that θ is an injective mapping from $\mathbf{V}(G)$ to $\mathbf{V}(H)$ w.r.t. lists $\mathcal{L}(u)$, $u \in \mathbf{V}(G)$. Let $\mathcal{E}(\{u, v\})$ denote the event that the edge $\{u, v\} \in \mathbf{E}(G)$ is not matched by the random injective mapping θ . For one,

$$\Pr[\overline{\mathcal{E}(\{u, v\})}] \geq C(G, H, \mathcal{L}) > \frac{2\Delta(G) - 1 - e^{-1}}{2\Delta(G) - 1}$$

and hence

$$\Pr[\mathcal{E}(\{u, v\})] \leq 1 - \frac{2\Delta(G) - 1 - e^{-1}}{2\Delta(G) - 1}.$$

For another, each event $\mathcal{E}(\{u, v\})$ is mutually independent of all other events except for at most $2\Delta(G) - 2$ events since $\mu_H = 1$. Write

$$p = \max_{\{u, v\} \in \mathbf{E}(G)} \Pr[\mathcal{E}(\{u, v\})].$$

Hence,

$$ep(2\Delta(G) - 2 + 1) \leq e\left(1 - \frac{2\Delta(G) - 1 - e^{-1}}{2\Delta(G) - 1}\right)(2\Delta(G) - 1) = 1.$$

According to the Lovász local lemma [4], we now thus obtain

$$\Pr[\overline{\bigcap_{\{u, v\} \in \mathbf{E}(G)} \mathcal{E}(\{u, v\})}] > 0.$$

Therefore, with positive probability, the random injective mapping θ matches all edges of G , and hence there must be an injective homomorphism of G to H w.r.t. lists $\mathcal{L}(u)$, $u \in \mathbf{V}(G)$. \square

According to Proposition 5, if $\Delta(G) = 1$ (resp. $\Delta(G) = 2$ and $\Delta(G) = 3$) and $C(G, H, \mathcal{L}) > 0.633$ (resp. $C(G, H, \mathcal{L}) > 0.878$ and $C(G, H, \mathcal{L}) > 0.927$) then there exists an injective homomorphism θ of G to H w.r.t. lists $\mathcal{L}(u)$.

Proposition 6 *Let $\langle G, H, \mathcal{L} \rangle$ be a trim instance of the EXACT- $(\mu_G, 1)$ -MATCHING problem. If*

$$C(G, H, \mathcal{L}) \leq \frac{\Delta(G) - 1}{\Delta(G)}$$

then an injective homomorphism θ of G to H w.r.t. lists $\mathcal{L}(u)$, $u \in \mathbf{V}(G)$, might not exist.

PROOF. For any $d > 1$, we provide a generic construction of an instance $\langle G, H, \mathcal{L} \rangle$ of the EXACT- $(\mu_G, 1)$ -MATCHING problem with $\Delta(G) = d$ and $C(G, H, \mathcal{L}) \leq \frac{\Delta(G) - 1}{\Delta(G)}$ for which there does not exist an injective homomorphism θ of G to H w.r.t. lists $\mathcal{L}(u)$, $u \in \mathbf{V}(G)$.

Fix an integer $d > 1$. and let $G = K_{1,d}$. Write u_0 the central vertex of G (the vertex with degree d) and u_1, u_2, \dots, u_d the vertices of G of degree 1. We now define a bipartite graph H . The vertices of H are defined by $\mathbf{V}(H) = V \cup W$, $V = \{v_1, v_2, \dots, v_d\}$ and $W = \{w_1, w_2, \dots, w_d\}$. The edges of H are defined by $\mathbf{E}(H) = \{\{u_i, w_j\} : 1 \leq i \leq d \wedge 1 \leq j \leq d \wedge i \neq j\}$. Now the associated

lists \mathcal{L} are:

$$\begin{aligned}\mathcal{L}(u_0) &= \{w_1, w_2, \dots, w_d\} \\ \mathcal{L}(u_i) &= \{v_i\}, \quad i = 1, 2, \dots, d.\end{aligned}$$

Observe that if $d = \Delta(G)$ is a constant, then so is μ_G .

We claim that there does not exist an injective homomorphism θ of G to H w.r.t. lists $\mathcal{L}(u)$, $u \in \mathbf{V}(G)$. Indeed, let θ' be any injective mapping of $\mathbf{V}(G)$ to $\mathbf{V}(H)$ w.r.t lists $\mathcal{L}(u)$, $u \in \mathbf{V}(G)$. Suppose $\theta'(u_0) = w_i$. We now observe that by construction $\{v_i, w_i\}$ is not an edge of H . Then it follows that θ' does not match the edge $\{u_0, u_i\}$ of G , and hence θ' is not an injective homomorphism of G to H w.r.t. lists $\mathcal{L}(u)$, $u \in \mathbf{V}(G)$.

The correspondence number of this generic instance is

$$C(G, H, \mathcal{L}) = \frac{d-1}{d}.$$

But here $d = \Delta(G)$, and hence there exists one instance (G, H, \mathcal{L}) satisfying

$$C(G, H, \mathcal{L}) = \frac{\Delta(G) - 1}{\Delta(G)}$$

for which there does not exist an injective homomorphism θ of G to H w.r.t. lists $\mathcal{L}(u)$, $u \in \mathbf{V}(G)$. \square

Combining Proposition 5 with Proposition 6, we obtain

$$c_{\text{up}} - c_{\text{low}} \leq \frac{2\Delta(G) - 1 - e^{-1}}{2\Delta(G) - 1} - \frac{\Delta(G) - 1}{\Delta(G)} = \frac{1 - e^{-1}}{\Delta(G) - 1} = \frac{0.632}{\Delta(G) - 1}.$$

4 Hardness of the $\text{Max}-(\mu_G, \mu_H)$ -Matching problem

The present and following sections are concerned with the optimization version of the problem. First, it follows from Proposition 3 that the $\text{MAX}-(3, 2)$ -MATCHING problem is **NP**-complete even if both G and H are bipartite graphs with $\Delta(G) \leq 1$ and $\Delta(H) \leq 3$. Moreover, by Proposition 4, we know that the $\text{MAX}-(3, 1)$ -MATCHING problem is **NP**-complete even if $\Delta(G) \leq 3$ and $\Delta(H) \leq 4$. We proved in [11] that the $\text{MAX}-(2, 1)$ -MATCHING problem is **APX**-hard even if both G and H are bipartite graphs with $\Delta(G) \leq 3$ and $\Delta(H) \leq 3$. We strengthen here this result by showing that the $\text{MAX}-(2, 1)$ -MATCHING problem is **APX**-complete (membership to **APX** is in fact deferred to the next section) even if both G and H are bipartite graphs with

$\Delta(G) \leq 3$ and $\Delta(H) \leq 2$. This has to be compared with the EXACT- $(2, \mu_H)$ -MATCHING problem, which is linear-time solvable for any constant μ_H [10].

Proposition 7 *The MAX- $(2, 1)$ -MATCHING 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 $\mathbf{P} = \mathbf{NP}$.*

PROOF. We propose a reduction from the MAX-2-SAT- B problem (each variable appears in at most B clauses, counting together both positive and negative occurrences) which is known to be APX-complete for $B \geq 3$ [20]. 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- B problem. Let $X = \{x_1, \dots, x_n\}$ denote the set of variables and $C = \{c_1, \dots, c_m\}$ denote the set of clauses. We now describe how to construct the corresponding instance of the MAX- $(2, 1)$ -MATCHING problem.

To ϕ we associate a first bipartite graph G defined as follows: we introduce one vertex x_i^G for each variable $x_i \in X$, and one vertex c_j^G for each clause $c_j \in C$. Also, for each $j = 1, 2, \dots, m$ and each $\ell = 1, 2$, we introduce the edge $\{x_i^G, c_j^G\}$ if the ℓ -th literal of clause c_j is a positive or a negative occurrence of variable x_i . To ϕ we also associate a second bipartite graph H defined as follows: we introduce two vertices $x_i^H[T]$ and $x_i^H[F]$ for each variable $x_i \in X$, and two vertices $c_j^H[1]$ and $c_j^H[2]$ for each clause $c_j \in C$. Also, for each $j = 1, 2, \dots, m$ and each $\ell = 1, 2$, we introduce the edge $\{x_i^H[T], c_j^H[\ell]\}$ if the ℓ -th literal of clause c_j is the positive literal x_i or the edge $\{x_i^H[F], c_j^H[\ell]\}$ if the ℓ -th literal of clause c_j is the negative literal \bar{x}_i . We now turn to describing the associated lists. To each $x_i^G \in \mathbf{V}(G)$ we associate the list $\mathcal{L}(x_i^G) = \{x_i^H[T], x_i^H[F]\}$. To each $c_j^G \in \mathbf{V}(G)$, we associate the list $\mathcal{L}(c_j^G) = \{c_j^H[1], c_j^H[2]\}$.

Clearly, $\mu_G = 2$, $\mu_H = 1$, $\Delta(G) = B$, $\Delta(H) = B - 1$, and both G and H are bipartite graphs. We claim that there exists a truth assignment that satisfies k clauses of C if and only if there exists an injective mapping $\theta : \mathbf{V}(G) \rightarrow \mathbf{V}(H)$ w.r.t. lists $\mathcal{L}(u)$, $u \in \mathbf{V}(G)$, such that $\#\text{match}(G, H, \theta) = k$.

Suppose that there exists a truth assignment $f : X \rightarrow \{\mathbf{true}, \mathbf{false}\}$ that satisfies k clauses of C . Consider the injective mapping $\theta : \mathbf{V}(G) \rightarrow \mathbf{V}(H)$ w.r.t. lists $\mathcal{L}(u)$, $u \in \mathbf{V}(G)$, defined as follows:

$$\theta(x_i^G) = \begin{cases} x_i^H[T] & \text{if } f(x_i) = \mathbf{true}, \\ x_i^H[F] & \text{if } f(x_i) = \mathbf{false}, \end{cases}$$

and

$$\theta(c_j^G) = \begin{cases} c_j^H[1] & \text{if clause } c_j \text{ is satisfied by its first literal or is not satisfied,} \\ c_j^H[2] & \text{if clause } c_j \text{ is satisfied by its second literal.} \end{cases}$$

It can be easily verified that $\#\text{match}(G, H, \theta) = k$.

Conversely, suppose that there exists an injective mapping $\theta : \mathbf{V}(G) \rightarrow \mathbf{V}(H)$ w.r.t. lists $\mathcal{L}(u)$, $u \in \mathbf{V}(G)$, such that $\#\text{match}(G, H, \theta) = k$. Define a truth assignment $f : X \rightarrow \{\mathbf{true}, \mathbf{false}\}$ as follows: $f(x_i) = \mathbf{true}$ if $\theta(x_i^G) = x_i^H[T]$ and $f(x_i) = \mathbf{false}$ if $\theta(x_i^G) = x_i^H[F]$. By construction, an edge of G is matched by θ if and only if the corresponding clause is satisfied by the truth assignment f . Then it follows that k clauses of C are satisfied by f .

Inapproximability results for the MAX-(2, 1)-MATCHING problem now follow from [5] where it is proved that the MAX-2-SAT-3 (resp. MAX-2-SAT-6) problem is not approximable within ratio 1.0005 (resp. 1.0014). \square

5 Approximating the Max-($\mu_G, 1$)-Matching problem

We proved in the preceding section that the MAX-(2, 1)-MATCHING problem is **APX**-hard even if both G and H are bipartite graphs with $\Delta(G) \leq 3$ and $\Delta(H) \leq 2$. We show in this section that the MAX-($\mu_G, 1$)-MATCHING problem for bounded degree graphs G belongs to **APX** for any constant μ_G , thereby proving that the MAX-(2, 1)-MATCHING problem is **APX**-complete. In addition, we give a fast randomized algorithm for the MAX-($\mu_G, 1$)-MATCHING problem that achieves a ratio $2\mu_G$ for any constant μ_G .

Recall first that a *matching* in a graph G is a subset of pairwise vertex disjoint edges of G . The *matching number* $\nu(G)$ of G is the size of a largest matching of G . A *linear forest* is a forest, *i.e.*, an acyclic simple graph, in which every connected component is a path. The *linear arboricity* $\text{la}(G)$ of a graph G is the minimum number of linear forests in G , whose union is the set of all edges of G [1] (see also [3]).

Conjecture 8 (The linear arboricity conjecture [1]) *The linear arboricity of every d -regular graph is $\lceil (d+1)/2 \rceil$.*

This conjecture was shown to be asymptotically correct as $d \rightarrow \infty$ [3]. Although the linear arboricity conjecture received a considerable amount of attention, the best general result concerning it is that $\text{la}(G) \leq \lceil 3\Delta(G)/5 \rceil$ for even $\Delta(G)$ and that $\text{la}(G) \leq \lceil (3\Delta(G) + 2)/5 \rceil$ for odd $\Delta(G)$ [4].

Lemma 9 *Let G be a graph. Then, $\nu(G) \geq \frac{\mathbf{m}(G)}{2 \text{la}(G)}$.*

PROOF. A simple cardinality argument shows that there exists a linear forest in G that contains at least $\frac{\mathbf{m}(G)}{\text{la}(G)}$ edges, and hence $\nu(G) \geq \frac{\mathbf{m}(G)}{2 \text{la}(G)}$ since each connected component of a linear forest is a path. \square

Proposition 10 *For any trim instance, the MAX- $(\mu_G, 1)$ -MATCHING problem is approximable within ratio $2 \text{la}(G)$ in $O(\mathbf{n}(G) + \mathbf{m}(G)\sqrt{\mathbf{n}(G)})$ time for any constant $\mu_G \geq 1$.*

PROOF. Let $\langle G, H, \mathcal{L} \rangle$ be a trim instance of the MAX- $(\mu_G, 1)$ -MATCHING problem. Now, let $\mathcal{M} \subseteq \mathbf{E}(G)$ be any maximum matching in G . Consider the mapping $\theta : \mathbf{V}(G) \rightarrow \mathbf{V}(H)$ defined as follows. For each edge $\{u, v\} \in \mathcal{M}$, let $u' \in \mathcal{L}(u)$ and $v' \in \mathcal{L}(v)$ be two vertices of H such that $\{u', v'\} \in \mathbf{E}(H)$ (such vertices exist since the instance is supposed to be trim). We then set $\theta(u) = u'$ and $\theta(v) = v'$. For any vertex $u \in \mathbf{V}(G)$ which is not incident to any edge in \mathcal{M} (in case \mathcal{M} is not a perfect matching), we set $\theta(u) = v$, where v is any vertex in $\mathcal{L}(u)$. Clearly, θ is well-defined and is injective since $\mu_H = 1$.

So, if we let θ be our solution mapping, it is a simple matter to check that $\#\text{match}(G, H, \theta) \geq \#\mathcal{M}$, and hence

$$\frac{\text{opt}(G, H, \mathcal{L})}{\#\text{match}(G, H, \theta)} \leq \frac{\text{opt}(G, H, \mathcal{L})}{\#\mathcal{M}}.$$

Combining this with $\text{opt}(G, H, \mathcal{L}) \leq \mathbf{m}(G)$ and $\#\mathcal{M} = \nu(G) \geq \frac{\mathbf{m}(G)}{2 \text{la}(G)}$ (Lemma 9), we obtain

$$\frac{\text{opt}(G, H, \mathcal{L})}{\#\text{match}(G, H, \theta)} \leq \mathbf{m}(G) \frac{2 \text{la}(G)}{\mathbf{m}(G)} = 2 \text{la}(G)$$

and the approximation ratio is proved. We now turn to proving the time complexity. Finding a maximum matching in G is an $O(\mathbf{m}(G)\sqrt{\mathbf{n}(G)})$ time procedure [19]. Since constructing θ is an $O(\mu_G^2 \nu(G) + \mathbf{n}(G) - 2\nu(G)) = O(\mathbf{m}(G) + \mathbf{n}(G))$ time procedure, the algorithm, as a whole, runs in $O(\mathbf{n}(G) + \mathbf{m}(G)\sqrt{\mathbf{n}(G)})$ time. \square

Corollary 11 *The MAX- $(\mu_G, 1)$ -MATCHING problem is approximable within ratio $2 \lceil 3\Delta(G)/5 \rceil$ for even $\Delta(G)$ and ratio $2 \lceil (3\Delta(G) + 2)/5 \rceil$ for odd $\Delta(G)$, for any $\Delta(H)$ and any constant μ_G .*

PROOF. Combine Proposition 10 with $\text{la}(G) \leq \lceil 3\Delta(G)/5 \rceil$ for even $\Delta(G)$ and $\text{la}(G) \leq \lceil (3\Delta(G) + 2)/5 \rceil$ for odd $\Delta(G)$. \square

Corollary 12 *The MAX-(2, 1)-MATCHING problem is APX-complete even if both G and H are bipartite graphs with $\Delta(G) \leq 3$ and $\Delta(H) \leq 2$.*

Corollary 13 *If the linear arboricity conjecture is true, then the MAX-($\mu_G, 1$)-MATCHING problem is approximable within ratio $\Delta(G) + 1$ if $\Delta(G)$ is odd, and $\Delta(G) + 2$ if $\Delta(G)$ is even, for any $\Delta(H)$ and any constant μ_G .*

PROOF. According to [4], since every graph G is a subgraph of a $\Delta(G)$ -regular graph (which may contain more vertices, as well as more edges than G), the linear arboricity conjecture is equivalent to the statement that the linear arboricity of every graph G is at most $\lceil (\Delta(G) + 1)/2 \rceil$. The result thus follows from Proposition 10. \square

We now turn to giving a fast randomized algorithm for the MAX-($\mu_G, 1$)-MATCHING problem. Using a straightforward application of the *probabilistic method* [4] - a powerful tool for demonstrating the existence of combinatorial objects - we gave in [11] a linear-time randomized μ_G^2 -approximation algorithm. We strengthen this result by giving here a polynomial-time randomized $2\mu_G$ -approximation algorithm for the MAX-($\mu_G, 1$)-MATCHING problem.

Lemma 14 *There is a polynomial-time randomized algorithm that achieves a performance ratio $2\mu_G$ for the MAX-($\mu_G, 1$)-MATCHING problem restricted to trim instances with unbounded degree graphs G and H .*

PROOF. A plain description of our algorithm is as follows. First, let $S \subseteq \mathbf{V}(G)$ be a random subset given by $\Pr[u \in S] = 1/2$, $u \in \mathbf{V}(G)$, these probabilities being mutually independent. Next, the injective mapping $\theta : \mathbf{V}(G) \rightarrow \mathbf{V}(H)$ is computed in two subsequent steps. In the first step, for each $u \in \mathbf{V}(G) - S$, $\theta(u)$ is chosen uniformly at random among the at most μ_G elements in $\mathcal{L}(u)$; it is crucial here to note that $\mu_H = 1$ and hence that $\mathcal{L}(u) \cap \mathcal{L}(v) = \emptyset$ for all $u, v \in \mathbf{V}(G)$. Finally, in the second step, the mapping θ is extended over the whole $\mathbf{V}(G)$ as follows: for each $s \in S$, in any order, let $\theta(s)$ be any node s' in $\mathcal{L}(u)$ maximizing $\#\{\{s, v\} \in \mathbf{E}(G) : v \in \mathbf{V}(G) - S \wedge \{s', \theta(v)\} \in \mathbf{E}(H)\}$. The description of the algorithm is complete.

For the sake of the analysis, let $\theta_{\text{opt}} : \mathbf{V}(G) \rightarrow \mathbf{V}(H)$ be an injective mapping w.r.t. lists $\mathcal{L}(u)$, $u \in \mathbf{V}(G)$, such that $\#\text{match}(G, H, \theta_{\text{opt}}) \geq \#\text{match}(G, H, \theta')$ for all injective mappings $\theta' : \mathbf{V}(G) \rightarrow \mathbf{V}(H)$ w.r.t. lists $\mathcal{L}(u)$, $u \in \mathbf{V}(G)$, and let $E_{\text{opt}} \subseteq \mathbf{E}(G)$ be all the edges in G that are matched by θ_{opt} , *i.e.*,

$$E_{\text{opt}} = \{\{u, u'\} \in \mathbf{E}(G) : \{\theta_{\text{opt}}(u), \theta_{\text{opt}}(u')\} \in \mathbf{E}(H)\}.$$

Call an edge $\{u, u'\} \in \mathbf{E}(G)$ a *cut-edge* if exactly one of u and u' is in S , and let $E_{\text{cut}} = \{\{u, u'\} \in \mathbf{E}(G) : u \in S \wedge u' \in \mathbf{V}(G) - S\}$ be the set of all cut-edges. Furthermore, let $E_{\text{cut}}^* \subseteq E_{\text{cut}}$ be the set of those cut-edges $e \in E_{\text{cut}}$, say $e = \{u, v\}$ with $u \in S$ and $v \notin S$, such that $\theta(v) = \theta_{\text{opt}}(v)$.

For each edge $e \in \mathbf{E}(G)$, let $X(e)$ be the random variable defined by $X(e) = 1$ if and only if $e \in E_{\text{opt}} \cap E_{\text{cut}}^*$. Write

$$X = \sum_{e \in \mathbf{E}(G)} X(e)$$

Also, let

$$Y = \sum_{e \in \mathbf{E}(G)} Y(e)$$

where $Y(e)$ is the indicator variable for $e = \{u, v\} \in \mathbf{E}(G)$ being matched by θ , *i.e.*, $Y(\{u, v\}) = 1$ if and only if $\{\theta(u), \theta(v)\} \in \mathbf{E}(H)$. Therefore Y is the objective function value achieved by the (random) solution returned by our algorithm. We proceed to show that $\text{Exp}[Y]$ is at least a fraction $2\mu_G$ of $\#E_{\text{opt}}$.

For one,

$$\begin{aligned} \text{Exp}[Y] &= \text{Exp} \left[\sum_{e \in \mathbf{E}(G)} Y(e) \right] \\ &= \sum_{e \in \mathbf{E}(G)} \text{Exp}[Y(e)] \\ &\geq \sum_{e \in E_{\text{cut}}^*} \text{Exp}[Y(e)] \\ &\geq \sum_{e \in E_{\text{cut}}^*} \text{Exp}[X(e)] \end{aligned}$$

by the way θ is extended over S . Indeed, for each $s \in S$, $\theta(s)$ is defined to be a vertex $s' \in \mathcal{L}(u)$ maximizing $\#\{\{s, v\} \in \mathbf{E}(G) : v \in \mathbf{V}(G) - S \wedge \{s', \theta(v)\} \in \mathbf{E}(H)\}$. Therefore,

$$\begin{aligned} \text{Exp}[Y] &\geq \sum_{e \in E_{\text{cut}}^*} \text{Exp}[X(e)] \\ &= \sum_{e \in \mathbf{E}(G)} \text{Exp}[X(e)] \\ &= \text{Exp}[X] \end{aligned}$$

since $X(e) = 0$ if $e \notin E_{\text{cut}}^*$. For another,

$$\begin{aligned}
\text{Exp}[X] &= \text{Exp} \left[\sum_{e \in \mathbf{E}(G)} X(e) \right] \\
&= \sum_{e \in \mathbf{E}(G)} \text{Exp}[X(e)] \\
&= \sum_{e \in E_{\text{opt}}} \text{Exp}[X(e)] \\
&= \sum_{e \in E_{\text{opt}}} \frac{1}{2 \mu_G} \\
&= \frac{\#E_{\text{opt}}}{2 \mu_G},
\end{aligned}$$

since the probability of any edge $e = \{u, u'\} \in E_{\text{opt}}$ being a cut-edge is $\frac{1}{2}$ and $\Pr[\theta(u') = \theta_{\text{opt}}(u')] = 1/\mu_G$, $u' \in \mathbf{V}(G) - S$. Combining this with $\text{Exp}[Y] \geq \text{Exp}[X]$, we obtain

$$\text{Exp}[Y] \geq \frac{\#E_{\text{opt}}}{2 \mu_G},$$

and the lemma is proved. \square

6 Fixed-parameter tractability

Parameterized complexity [8] is an approach to complexity theory which offers a means of analyzing algorithms in terms of their tractability. For many hard problems, the seemingly unavoidable combinatorial explosion can be restricted to a *small part* of the input, the *parameter*, so that the problems can be solved in polynomial-time when the parameter is fixed. The parameterized problems that have algorithms of $f(k) n^{O(1)}$ time complexity are called *fixed-parameter tractable*, where k is the parameter, f can be an arbitrary function depending only on k , and n denotes the overall input size. In the last decade, parameterized complexity has proved to be extremely useful in computational molecular biology, see for example [6,14,2].

We follow here this trend by showing in this section that the MAX- $(\mu_G, 1)$ -MATCHING problem for any bounded degree graph G is fixed-parameter tractable parameterized by the number of matched edges, *i.e.*, $\#\text{match}(G, H, \theta)$. For this, we adopt here a two-step procedure: we first define a new graph representation of the problem, and next use that graph to derive fixed-parameter tractability. At the heart of the algorithm is the *incompatibility graph* of any instance $\langle G, H, \mathcal{L} \rangle$ which is later shown to be a compact representation of the problem.

Definition 15 (Incompatibility graph) Let $\langle G, H, \mathcal{L} \rangle$ be a trim instance of the MAX- $(\mu_G, 1)$ -MATCHING problem and $<$ be an arbitrary total order on $\mathbf{V}(G)$. The incompatibility graph of (G, H, \mathcal{L}) , denoted by $I[G, H, \mathcal{L}]$, is defined by

$$\begin{aligned} \mathbf{V}(I[G, H, \mathcal{L}]) &= \{(u, v, u', v') : u < v \wedge \{u, v\} \in \mathbf{E}(G) \wedge \{u', v'\} \in \mathbf{E}(H) \\ &\quad \wedge u' \in \mathcal{L}(u) \wedge v' \in \mathcal{L}(v)\}, \text{ and} \\ \mathbf{E}(I[G, H, \mathcal{L}]) &= \{ \{(u_1, u_2, u'_1, u'_2), ((v_1, v_2, v'_1, v'_2))\} : \exists 1 \leq i, j \leq 2 \text{ such that} \\ &\quad u_i = v_j \text{ and } u'_i \neq v'_j \}. \end{aligned}$$

Less formally, each vertex of $I[G, H, \mathcal{L}]$ denotes a putative edge match in $\langle G, H, \mathcal{L} \rangle$ and two vertices of $I[G, H, \mathcal{L}]$ are connected by an edge if and only if the two corresponding edge matches are not compatible, *i.e.*,

$$\{ \{(u_1, u_2, u'_1, u'_2), ((v_1, v_2, v'_1, v'_2))\} \in \mathbf{E}(I[G, H, \mathcal{L}])$$

if no injective mapping $\theta : \mathbf{V}(G) \rightarrow \mathbf{V}(H)$ w.r.t. lists $\mathcal{L}(u)$ can match simultaneously edge $\{u_1, u_2\} \in \mathbf{E}(G)$ and edge $\{v_1, v_2\} \in \mathbf{E}(G)$. Most of the interest in the incompatibility graph $I[G, H, \mathcal{L}]$ stems from the following lemma, whose correctness follows immediately from the above definition.

Lemma 16 Let $\langle G, H, \mathcal{L} \rangle$ be a trim instance of the MAX- $(\mu_G, 1)$ -MATCHING problem. There exists an injective mapping $\theta : \mathbf{V}(G) \rightarrow \mathbf{V}(H)$ w.r.t. lists $\mathcal{L}(u)$, $u \in \mathbf{V}(G)$, such that $\#\text{match}(G, H, \theta) \geq k$ if and only if there exists an independent set of size at least k in the incompatibility graph $I[G, H, \mathcal{L}]$.

Thus, finding an injective mapping θ of G to H w.r.t. $\mathcal{L}(u)$, $u \in \mathbf{V}(G)$, that maximizes the number of matched edges (*i.e.*, $\#\text{match}(G, H, \theta)$) reduces to finding a maximum independent set in $I[G, H, \mathcal{L}]$. This equivalence gains in interest if we realize that, for any constant μ_G , if G is a bounded degree graph, then so is the incompatibility graph $I[G, H, \mathcal{L}]$.

Lemma 17 Let $\langle G, H, \mathcal{L} \rangle$ be an instance of the MAX- $(\mu_G, 1)$ -MATCHING problem. Then, $I[G, H, \mathcal{L}]$ has maximum degree at most $\mu_G^2 + 2\mu_G(\mu_G - 1)(\Delta(G) - 1) - 1$.

PROOF. For ease of exposition, let us first rewrite $\mathbf{E}(I[G, H, \mathcal{L}])$ as follows:

$\mathbf{E}(I[G, H, \mathcal{L}]) = \bigcup_{1 \leq i \leq 5} E_i$, where

$$\begin{aligned} E_1 &= \{ \{ (u, v, u', v'), (x, y, x', y') \} : u = x \wedge v = y \wedge (u' \neq x' \vee v' \neq y') \} \\ E_2 &= \{ \{ (u, v, u', v'), (x, y, x', y') \} : u = x \wedge v \neq y \wedge u' \neq x' \} \\ E_3 &= \{ \{ (u, v, u', v'), (x, y, x', y') \} : u \neq x \wedge v = y \wedge v' \neq y' \} \\ E_4 &= \{ \{ (u, v, u', v'), (x, y, x', y') \} : u = y \wedge u' \neq y' \} \\ E_5 &= \{ \{ (u, v, u', v'), (x, y, x', y') \} : v = x \wedge v' \neq x' \}. \end{aligned}$$

Let us look at a given vertex (u, v, u', v') of $\mathbf{V}(I[G, H, \mathcal{L}])$, and let us count how many edges at most can be incident to (u, v, u', v') :

- Edges from E_1 : intuitively, this corresponds to all the other possible cases of projection of the edge (u, v) of G onto an edge of H . Since u (resp. v) has at most μ_G images by \mathcal{L} , there are at most μ_G^2 different possible projections of (u, v) on an edge of H . Among them, only one (namely, edge (u', v')) does not imply an edge in E_1 . Thus, there are at most $\mu_G^2 - 1$ edges from E_1 .
- Edges from $E_2 \cup E_4$: intuitively, those edges correspond to edges of G of the form $e = \{x, y\}$, $x < y$, where
 - either $x = u$ but x and u do not have the same image in H
 - or $y = u$ but y and u do not have the same image in H
Considering both cases together, we see that there are at most $\mu_G - 1$ possibilities for x or y to be equal to u , while its image is different from u . Besides, for each of these $\mu_G - 1$ possible cases, there are at most μ_G possibilities for the other endpoint of e . Hence, for any fixed edge e having an endpoint equal to u , there are at most $\mu_G(\mu_G - 1)$ cases. Since G is of maximum degree $\Delta(G)$, there are at most $\Delta(G) - 1$ such possible edges e (because we do not count edge $\{u, v\}$), and thus altogether we have at most $\mu_G(\mu_G - 1)(\Delta(G) - 1)$ edges of $E_2 \cup E_4$ incident to vertex (u, v, u', v') in $I[G, H, \mathcal{L}]$.
- Edges from $E_3 \cup E_5$: this case is similar to the previous one, where we consider v instead of u . By symmetry, we conclude that we have a total of at most $\mu_G(\mu_G - 1)(\Delta(G) - 1)$ edges of $E_2 \cup E_4$ incident to vertex (u, v, u', v') in $I[G, H, \mathcal{L}]$.

Altogether, we get that the maximum degree of graph $I[G, H, \mathcal{L}]$ satisfies :

$$\Delta(I[G, H, \mathcal{L}]) \leq \mu_G^2 + 2\mu_G(\mu_G - 1)(\Delta(G) - 1) - 1.$$

□

It follows from the above lemma that $\Delta(I[G, H, \mathcal{L}]) = O(\Delta(G))$ when $\mu_G = O(1)$, and hence if G is a bounded degree graph, then so is $I[G, H, \mathcal{L}]$. Having disposed of these preliminaries steps, we now turn to proving fixed-parameter tractability of the MAX- $(\mu_G, 1)$ -MATCHING problem.

Proposition 18 *The MAX- $(\mu_G, 1)$ -MATCHING problem is solvable in $O(m(D+1)^k)$ time, where k is the number of matched edges, i.e., $\#\text{match}(G, H, \theta)$, $m = \mathbf{m}(G)$ and $D = \Delta(I[G, H, \mathcal{L}]) = (\mu_G - 1)(2\mu_G\Delta(G) - \mu_G + 1) = O(\Delta(G))$, and hence is fixed-parameter tractable for parameter k , provided that G is a bounded degree graph and μ_G is a constant.*

PROOF. By standard bounded search techniques [8], one can find an independent set of size k in a graph \mathcal{G} in $O(\mathbf{m}(\mathcal{G}) (\Delta(\mathcal{G}) + 1)^k)$ time, or return that no such subset exists. The proposition thus follows from applying this to the incompatibility graph $I[G, H, \mathcal{L}]$, where we use the fact that, by definition of $I[G, H, \mathcal{L}]$, $\mathbf{n}(I[G, H, \mathcal{L}]) = O(\mathbf{m}(G))$ for any constant μ_G . \square

7 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 proved the EXACT-(3, 2)-MATCHING problem for $\Delta(G) \leq 2$ to be polynomial-time solvable, and both the EXACT-(3, 2)-MATCHING problem for bipartite graphs G and H with $\Delta(G) \leq 1$ and $\Delta(H) \leq 2$ and the EXACT-(3, 1)-MATCHING problem for $\Delta(G) \leq 3$ and $\Delta(H) \leq 4$ to be **NP**-complete. Also, we showed that the MAX-(2, 1)-MATCHING problem for bounded degree bipartite graphs is **APX**-hard. This latter problem was shown to be fixed-parameter tractable parameterized by the number of matched edges.

We mention here some possible directions for future works. First, an interesting line of research is to further investigate the approximation of the MAX- (μ_G, μ_H) -MATCHING problem for bounded degree graphs G and H . For example, is the MAX-(2, 2)-MATCHING problem for bounded degree graphs G and H in **APX**? From a computational complexity point of view, the MAX- (μ_G, μ_H) -MATCHING problem for $\Delta(G) = \Delta(H) = 3$ remains open. Parameterized complexity of the MAX- (μ_G, μ_H) -MATCHING problem is completely unexplored in the case $\mu_H > 1$. In particular, is the MAX- (μ_G, μ_H) -MATCHING problem for bounded degree graphs G and H fixed-parameter tractable for any constant μ_G and μ_H ?

References

- [1] J. Akiyama, G. Exoo, and F. Harary, *Covering and packing in graphs IV: Linear arboricity*, Networks **11** (1981), 69–72.

- [2] J. Alber, J. Gramm, J. Guo, and R. Niedermeier, *Towards optimally solving the longest common subsequence problem for sequences with nested arc annotations in linear time*, Proc. 13th Annual Symposium on Combinatorial Pattern Matching (CPM), Fukuoka, Japan (A. Apostolico and M. Takeda, eds.), Lecture Notes in Computer Science, vol. 2373, Springer, 2002, pp. 99–114.
- [3] N. Alon, *The linear arboricity of graphs*, Israel J. of Mathematics **62** (1988), no. 3, 311–325.
- [4] N. Alon and J.H. Spencer, *The probabilistic method*, Wiley, 1992.
- [5] P. Berman and M. Karpinski, *On some tighter inapproximability results*, Proc. 26th International Colloquium on Automata, Languages and Programming (ICALP), Prague, Czech Republic (J. Wiedermann, P. van Emde Boas, and M. Nielsen, eds.), Lecture Notes in Computer Science, vol. 1644, Springer, 1999, pp. 200–209.
- [6] H.L. Bodlaender, R.G. Downey, M.R. Fellows, M.T. Hallett, and H.T. Wareham, *Parameterized complexity analysis in computational biology*, Computer Applications in the Biosciences **11** (1995), 49–57.
- [7] G. Brevier, R. Rizzi, and S. Vialette, *Pattern matching in protein-protein interaction graphs*, Proc. 16th Fundamentals of Computation Theory, 16th International Symposium (FCT), Budapest, Hungary (E. Csuhaj-Varjú and Z. Ésik, eds.), Lecture Notes in Computer Science, Springer, 2007, pp. 137–148.
- [8] R. Downey and M. Fellows, *Parameterized complexity*, Springer-Verlag, 1999.
- [9] P. Erdős and L Lovász, *Problems and results on 3-chromatic hypergraphs and some related questions*, Infinite and Finite Sets (Colloq., Keszthely, 1973; dedicated to P. Erdős on his 60th birthday) (A. Hajnal, R. Rado, and V Sós, eds.), Coll. Math. Soc. J. Bolyai, vol. 2, North-Holland, Amsterdam, 1975, pp. 609–627.
- [10] I. Fagnot, G. Lelandais, and S. Vialette, *Bounded list injective homomorphism for comparative analysis of protein-protein interaction graphs*, Journal of Discrete Algorithms **6** (2008), no. 2, 178–191.
- [11] 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), Gdansk, Poland (J. Jedrzejowicz and A. Szepietowski, eds.), Lecture Notes in Computer Science, vol. 3618, Springer, 2005, pp. 328–339.
- [12] M.R. Garey and D.S. Johnson, *Computers and intractability: a guide to the theory of NP-completeness*, W.H. Freeman, San Francisco, 1979.
- [13] 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.
- [14] J. Gramm, J. Guo, and R. Niedermeier, *Pattern matching for arc-annotated sequences*, ACM Transactions on Algorithms **2** (2006), no. 1, 44–65, To appear.

- [15] Y. Ho, A. Gruhler, et al., *Systematic identification of protein complexes in *Saccharomyces cerevisiae* by mass spectrometry*, Nature **415** (2002), no. 6868, 180–183.
- [16] M. Kalaei, V. Bafna, and R. Sharan, *Fast and accurate alignment of multiple protein networks*, Proc. 12th Annual International Conference on Computational Molecular Biology (RECOMB), National University of Singapore, Singapore, ACM, 2008, To appear.
- [17] 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.
- [18] M. Koyutürk, A. Grama, and W. Szpankowski, *Pairwise local alignment of protein interaction networks guided by models of evolution*, Proc. 9th Annual International Conference on Research in Computational Molecular Biology (RECOMB), Cambridge, MA, USA (S. Miyano, J. P. Mesirov, S. Kasif, S. Istrail, P. A. Pevzner, and M. S. Waterman, eds.), Lecture Notes in Bioinformatics, vol. 3500, Springer, 2005, pp. 48–65.
- [19] S. Micali and V.V. Vazirani, *An $O(\sqrt{|V||E|})$ algorithm for finding maximum matching in general graphs*, Proc. 21st Annual Symposium on Foundation of Computer Science (FOCS), IEEE, 1980, pp. 17–27.
- [20] C.H. Papadimitriou and M. Yannakakis, *Optimization, approximation and complexity classes*, J. of Computer and System Sciences **43** (1991), 425–440.
- [21] 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.
- [22] 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.
- [23] J. Scott, T. Ideker, R.M. Karp, and R. Sharan, *Efficient algorithms for detecting signaling pathways in protein interaction networks*, Journal of Computational Biology **13** (2006), 133–144.
- [24] 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), San Diego, California, USA (P.E. Bourne and D. Gusfield, eds.), ACM Press, 2004, pp. 282–289.
- [25] R. Sharan, S. Suthram, R.M. Kelley, T. Kuhn, S. McCuine, P. Uetz, T. Sittler, R.M. Karp, and T. Ideker, *Conserved patterns of protein interaction in multiple species*, Proc. Natl Acad. Sci. USA **102** (2005), no. 6, 1974–1979.
- [26] 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.

- [27] P. Uetz, L. Giot, et al., *A comprehensive analysis of protein-protein interactions in *Saccharomyces cerevisiae**, *Nature* **403** (2000), no. 6770, 623–627.
- [28] Q. Yang and S.-H. Sze, *Path matching and graph matching in biological networks*, *Journal of Computational Biology* **14** (2007), no. 1, 56–67.