

# Finding Exact and Maximum Occurrences of Protein Complexes in Protein-Protein Interaction Graphs

Guillaume Fertin<sup>1</sup>, Romeo Rizzi<sup>2</sup>, and Stéphane Vialette<sup>3</sup>

<sup>1</sup> Laboratoire d'Informatique de Nantes-Atlantique (LINA), FRE CNRS 2729  
Université de Nantes, 2 rue de la Houssinière, 44322 Nantes Cedex 3 - France  
`fertin@lina.univ-nantes.fr`

<sup>2</sup> Università degli Studi di Trento  
Facoltà di Scienze - Dipartimento di Informatica e Telecomunicazioni  
Via Sommarive, 14 I-38050 Povo - Trento (TN) - Italy  
`romeo.rizzi@unitn.it`

<sup>3</sup> Laboratoire de Recherche en Informatique (LRI), UMR CNRS 8623  
Faculté des Sciences d'Orsay  
Université Paris-Sud, 91405 Orsay - France  
`viallette@lri.fr`

**Abstract.** Comparing genomic properties of multiple species at varying evolutionary distances is a powerful approach for studying biological and evolutionary principles. 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, *i.e.*, proteins in different species that have diverged in the two lineages from a common ancestor. Two problems are considered: the **Exact- $(\mu_G, \mu_H)$ -Matching** problem and the **Max- $(\mu_G, \mu_H)$ -Matching** problem, where  $\mu_G$  (resp.  $\mu_H$ ) denotes in both problems the maximum number of orthologous proteins in  $H$  (resp.  $G$ ) of a protein in  $G$  (resp.  $H$ ). Following [FLV04], the **Exact- $(\mu_G, 1)$ -Matching** problem asks for an injective homomorphism of  $G$  to  $H$  w.r.t. orthologous proteins. The optimization version is called the **Max- $(\mu_G, 1)$ -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, the emphasis here is clearly on bounded degree graphs  $G$  and  $H$ , and extremal small values of parameters  $\mu_G$  and  $\mu_H$ .

We prove that the **Exact- $(\mu_G, 1)$ -Matching** problem for  $\Delta(G) \leq 2$  is polynomial-time solvable for any constant  $\mu_G$ , but that the **Exact- $(3, 2)$ -Matching** problem is **NP**-complete for bipartite graphs  $G$  and  $H$  with  $\Delta(G) \leq 2$  and  $\Delta(H) \leq 3$ . Concerning the optimization version, the **Max- $(2, 1)$ -Matching** problem is proved to be **APX**-hard for  $\Delta(G) \leq 3$  and  $\Delta(H) \leq 3$ . To counterbalance this discouraging result, it is shown here that the **Max- $(\mu_G, 1)$ -Matching** problem for bounded degree graph  $G$  is fixed-parameter-tractable parameterized by the number of matched edges and 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  $\mu_G$ , thereby proving that the **Max- $(2, 1)$ -Matching** problem for bounded degree graphs  $G$  and  $H$  is in **APX**, and hence **APX**-complete. Still in case  $\mu_H = 1$ , we complement these results by giving a fast randomized algorithm that achieves a ratio  $\mu_G^2$ , for any constant  $\mu_G$ .

## 1 Introduction

High-throughput analysis makes possible the study of protein-protein interactions at a genome-wide scale [Gav02,HG02,Uet00], 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<sup>+</sup>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 (even if some global features have been found). 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<sup>+</sup>03] developed the program PathBlast, which aligns two protein-protein interaction graphs combining topology and sequence similarity. Sharan *et al.* [SIK<sup>+</sup>04] studied the conservation of complexes<sup>1</sup> 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. 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 [SSK<sup>+</sup>05].

Following the line of research presented in [FLV04], 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 [FLV04], 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 the protein-protein interaction graph. 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  $\mu_G$  and  $\mu_H$ ): no protein has an unbounded number of orthologs in the other species, *i.e.*, each list has a constant size (upper bounded by parameter  $\mu_G$ ) and each protein has a constant number of occurrences among the lists (upper bounded by parameter  $\mu_H$ ). The present paper is devoted to analyzing the complexity of this problem (the **Exact**- $(\mu_G, \mu_H)$ -**Matching** problem) and of its natural optimization version (the **Max**- $(\mu_G, \mu_H)$ -**Matching** problem) in case of bounded degree graphs and extremal small values of parameters  $\mu_G$  and  $\mu_H$ .

The paper is organized as follows: Section 2 presents some preliminaries and introduces formally the problems. We prove in Section 3 new tight complexity results for the **Exact**- $(\mu_G, \mu_H)$ -**Matching** problem for bounded degree graphs. In Section 4, we show that the **Exact**- $(\mu_G, \mu_H)$ -**Matching** for bounded degree graphs is **APX**-hard. That result is complemented in Section 5 by showing that the **Exact**- $(\mu_G, \mu_H)$ -**Matching** problem for bounded degree graphs is in **APX**. Finally, we prove in Section 6 that the **Exact**- $(\mu_G, \mu_H)$ -**Matching** problem for bounded degree graphs parameterized by the number of matched edges is fixed-parameter tractable. Due to space constraints, several details and proofs are not presented in this paper.

## 2 Preliminaries

A *graph*  $G$  consists of a finite set  $\mathbf{V}(G) = \{u_1, u_2, \dots\}$  of elements called *vertices* together with a prescribed set  $\mathbf{E}(G)$  of undirected pairs of distinct vertices of  $\mathbf{V}(G)$ . We abbreviate  $\#\mathbf{V}(G)$  to  $\mathbf{n}(G)$ , and  $\#\mathbf{E}(G)$  to  $\mathbf{m}(G)$ . 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  $\theta$ , *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  $\theta$  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 a homomorphism  $\theta$  with the additional constraint that  $\theta(u) \subseteq$

<sup>1</sup> They focused on dense, clique-like interaction patterns.

$\mathcal{L}(u)$  for all  $u \in \mathbf{V}(G)$ . 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  $(\mu_G, \mu_H)$ -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- $(\mu_G, \mu_H)$ -Matching problem, is defined formally as follows.

**Exact- $(\mu_G, \mu_H)$ -Matching**

*Input* : Two graphs  $G$  and  $H$ , and  $(\mu_G, \mu_H)$ -bounded lists  $\mathcal{L}(u) \subseteq \mathbf{V}(H)$ ,  $u \in \mathbf{V}(G)$ .

*Question* : Is there an injective 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  $\mu_G$  and  $\mu_H$  are constant, *i.e.*,  $\mu_G = \mathcal{O}(1)$  and  $\mu_H = \mathcal{O}(1)$ .

**Proposition 1 ([FLV04]).** *The Exact- $(2, \mu_H)$ -Matching problem is solvable in linear-time for any constant  $\mu_H \geq 1$ .*

The basic idea of the proof of Proposition 1 is to transform each input  $(G, H, \mathcal{L})$  into a CNF formula  $\phi$  with at most two literals per clause in such a way that  $\phi$  is satisfiable if and only if there exists an injective homomorphism of  $G$  to  $H$  w.r.t.  $\mathcal{L}(u)$ ,  $u \in \mathbf{V}(G)$ . It is well-known that the 2-Sat problem is solvable in linear time [APT79]. Unfortunately, Proposition 1 is tight, and the situation is quite different for  $\mu_G \geq 3$ , as shown in the following proposition.

**Proposition 2 ([FLV04]).** *The Exact- $(3, 1)$ -Matching problem is NP-complete.*

It is worth noticing that Proposition 2 holds true even if both  $G$  and  $H$  are bipartite graphs or splits graphs [FLV04].

A first contribution in this paper is to complete the determination of the precise border between tractable and intractable cases for the Exact- $(\mu_G, \mu_H)$ -Matching problem. Moreover, in this paper, we begin the analysis of optimization versions of the problem. 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 application.

**Max- $(\mu_G, \mu_H)$ -Matching**

*Input* : Two graphs  $G$  and  $H$ , and  $(\mu_G, \mu_H)$ -bounded lists  $\mathcal{L}(u) \subseteq \mathbf{V}(H)$ ,  $u \in \mathbf{V}(G)$ .

*Solution* : An injective mapping  $\theta : \mathbf{V}(G) \mapsto \mathbf{V}(H)$  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 is the fact that  $\theta$  is no longer required to be an homomorphism in the Max- $(\mu_G, \mu_H)$ -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  $(G, H, \mathcal{L})$  be an instance of the Max- $(\mu_G, \mu_H)$ -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- $(\mu_G, \mu_H)$ -Matching problem,

since bad edges can never be matched. Notice that we can tell bad edges apart in  $\mathcal{O}(\mu_G^2 \mathbf{m}(G)) = \mathcal{O}(\mathbf{m}(G))$  time, provided  $\mu_G$  is a constant. Furthermore, by resorting on classical bipartite matching techniques, we can check in  $\mathcal{O}(\mathbf{n}(H) + \mathbf{m}(G) \sqrt{\mathbf{n}(G)})$  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  $(G, H, \mathcal{L})$  of the Max- $(\mu_G, \mu_H)$ -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- $(\mu_G, \mu_H)$ -Matching problem [FLV04]. We begin by giving an easy algorithm for the Exact- $(\mu_G, 1)$ -Matching problem in case  $\Delta(G) = 2$ .

**Proposition 3.** *The Exact- $(\mu_G, 1)$ -Matching problem for  $\Delta(G) \leq 2$  is solvable in  $\mathcal{O}(\mathbf{n}(G))$  time for any constant  $\mu_G$ .*

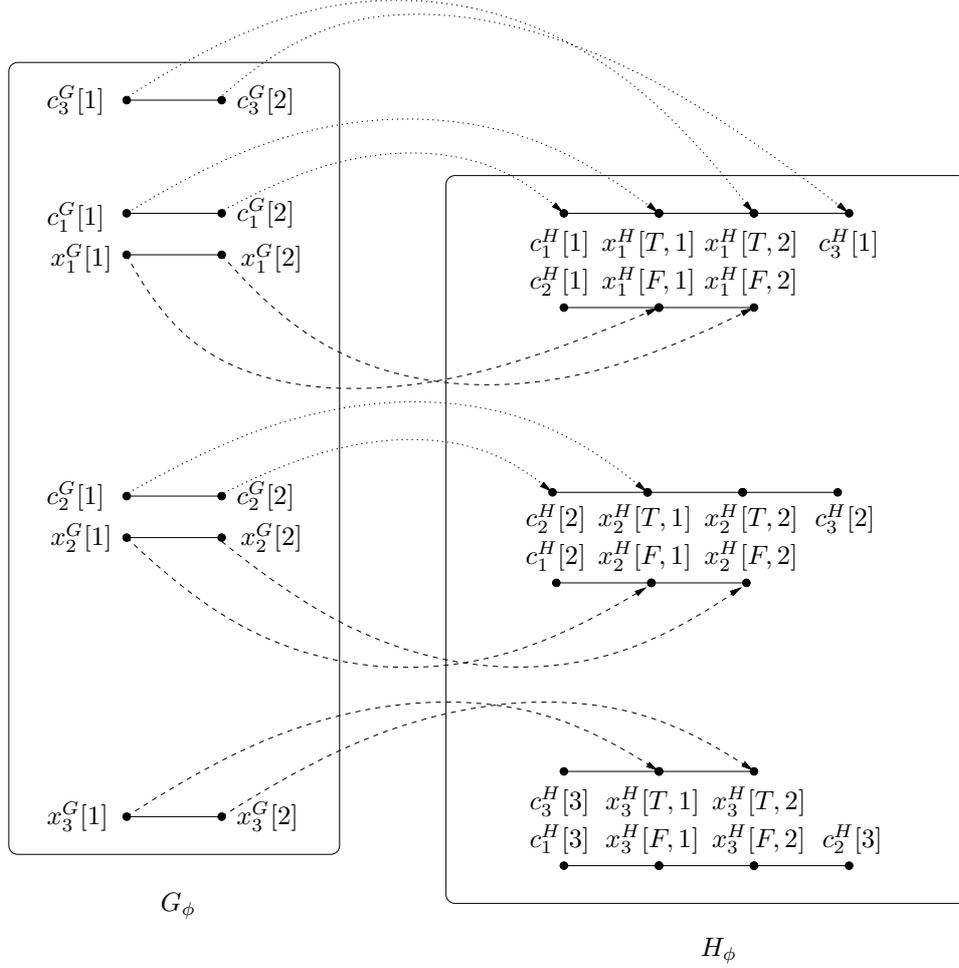
One may argue that the above proposition is too constrained to be of interest. Unfortunately, despite the simplicity of Proposition 3, the result is quite tight - taking into consideration  $\Delta(G)$  and  $\Delta(H)$  - as shown in the two following propositions (recall also that the Exact- $(2, \mu_H)$ -Matching problem is polynomial-time solvable for any constant  $\mu_H$  [FLV04]).

**Proposition 4.** *The Exact- $(3, 2)$ -Matching problem is NP-complete even if both  $G$  and  $H$  are bipartite graphs with  $\Delta(G) = 1$  and  $\Delta(H) = 3$ .*

*Proof (Of Proposition 4).* 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 [GJ79]. 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, hence a self-reduction would apply. Assume given an input  $\phi$  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  $\phi$  we associate a bipartite graph, denoted  $G_\phi$  - 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  $\phi$  we also associate a second bipartite graph, denoted  $H_\phi$ , 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 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  $\ell$ -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  $H_\phi$ . 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  $x_i^G[p] \in \mathbf{V}(G)$ , 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,  $\mu_G = 3$ ,  $\mu_H = 2$ ,  $\Delta(G_\phi) = 1$ , i.e.,  $G_\phi$  is a matching, and  $\Delta(H_\phi) = 2$ . Furthermore,  $H_\phi$  is 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  $\phi$  if and only if there exists an injective list homomorphism of  $G_\phi$  to  $H_\phi$  w.r.t. lists  $\mathcal{L}(u)$ ,  $u \in \mathbf{V}(G_\phi)$ .



**Fig. 1.** Illustration of the proof of Proposition 4 for the boolean formula  $\phi = (x_1 \vee \overline{x_2} \vee \overline{x_3}) \wedge (\overline{x_1} \vee x_2 \vee \overline{x_3}) \wedge (x_1 \vee x_2 \vee x_3)$ . Both  $G_\phi$  and  $H_\phi$  are bipartite graphs, and  $\Delta(G_\phi) = 1$  and  $\Delta(H_\phi) = 2$ . Shown here is the satisfying truth assignment  $f : X \mapsto \{\text{true}, \text{false}\}$  defined by  $f(x_1) = \text{true}$ ,  $f(x_2) = \text{true}$  and  $f(x_3) = \text{false}$ , together with the injective mapping  $\theta$  of  $G_\phi$  to  $H_\phi$  (denoted here by dashed and dotted lines).

Let  $f : X \mapsto \{\text{true}, \text{false}\}$  be a truth assignment for  $\phi$  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  $\ell$ -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  $\theta$  is an injective homomorphism of  $G_\phi$  to  $H_\phi$  w.r.t. lists  $\mathcal{L}(u)$ ,  $u \in \mathbf{V}(G_\phi)$ .

Conversely, suppose that there is an injective list homomorphism  $\theta$  of  $G_\phi$  to  $H_\phi$  w.r.t. lists  $\mathcal{L}(u)$ ,  $u \in \mathbf{V}(G_\phi)$ . 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  $\{x_i^G[1], x_i^G[2]\} \in \mathbf{E}(G_\phi)$ . Define a truth assignment  $f : X \mapsto \{\text{true}, \text{false}\}$  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  $\phi$ . 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  $\ell$ -th literal of  $\phi$  evaluates to **true** under the truth assignment  $f$ .  $\square$

**Proposition 5.** *The Exact-(3, 1)-Matching problem is NP-complete even when  $\Delta(G) = 3$  and  $\Delta(H) = 4$ .*

*Proof.* It is well known that deciding whether an input graph  $G$  has chromatic number 3 is **NP**-complete. It follows that, when given a graph  $G' = (V', E')$  and a subset  $F'$  of  $E'$ , deciding whether we can assign one of 3 possible colors to each node in  $V'$  in such a way that every two adjacent nodes  $u, v \in V'$  have the same color if and only if  $uv \in F'$  is **NP**-complete even when  $E' \setminus F'$  is a matching in  $G'$  and the edges in  $F'$  form node-disjoint paths in  $G'$ . Indeed, starting from  $G$ , explode each node  $v$  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  $F'$  to force all nodes in  $P'_v$  to search for a common color. The matching  $E' \setminus F'$  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  $uv$  is an edge in  $G$ .

Now, starting from the pair  $(G', F')$ , we show how to construct an “equivalent” instance  $(G, H, \mathcal{L})$  of problem **Exact-(3, 1)-Matching** with  $\Delta(G) = 3$  and  $\Delta(H) = 4$ . Indeed, take  $G := G'$  and let  $H$  be the graph with  $\mathbf{V}(H) := \cup_{v \in \mathbf{V}(G)} \{v_1, v_2, v_3\}$  and

$$E(H) := \bigcup_{uv \in F'} \{u_1v_1, u_2v_2, u_3v_3\} \cup \bigcup_{uv \in E(G) \setminus F'} \{u_1v_2, u_1v_3, u_2v_1, u_2v_3, u_3v_1, u_3v_2\}.$$

Clearly,  $\Delta(G) = \Delta(G') = 3$ , and  $\Delta(H) = 4$  since  $E(G) \setminus F'$  is a matching in  $G'$  and no node of  $G'$  is adjacent to more than two edges in  $F'$ . Moreover, by taking  $\mathcal{L}(v) = \{v_1, v_2, v_3\}$  for each node  $v$  of  $G$ , it is guaranteed that there exists a mapping  $c : \mathbf{V}(G') \mapsto \{1, 2, 3\}$  with  $c(u) = c(v)$  whenever  $uv \in F'$  and  $c(u) \neq c(v)$  whenever  $uv \in E \setminus F'$  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  $(G', F')$  and  $(G, H, \mathcal{L})$  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  $u \neq v$ , that is,  $\#\mathcal{L}^{-1}(u') \leq 1$  for each node  $u'$  of  $H$ .

The remainder of this section is devoted to the **Exact-( $\mu_G, 1$ )-Matching** problem. For each trim instance  $(G, H, \mathcal{L})$  of the **Exact-( $\mu_G, 1$ )-Matching** problem, define the *correspondence number* of the instance  $(G, H, \mathcal{L})$ , written  $C(G, H, \mathcal{L})$ , 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,  $0 \leq C(G, H, \mathcal{L}) \leq 1$ . Furthermore, if  $C(G, H, \mathcal{L}) = 1$ , then there exists an injective homomorphism  $\theta$  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. We now turn to proving a better lower bound.

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

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

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

## 4 Hardness of the **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 4 that the **Max-(3, 2)-Matching** problem is **NP**-complete even if both  $G$  and  $H$  are bipartite graphs with  $\Delta(G) = 1$  and  $\Delta(H) = 3$ . Moreover, by Proposition 5, we know that the **Max-(3, 1)-Matching** problem is **NP**-complete even when  $\Delta(G) = 3$  and  $\Delta(H) = 4$ . In this section, we strengthen these results by showing that the **Max-(2, 1)-Matching** problem for bounded degree graphs  $G$  and  $H$  is **APX**-complete (membership to **APX** is in fact deferred to the next section). This has to be compared with the **Exact-(2,  $\mu_H$ )-Matching** problem, which is linear-time solvable for any constant  $\mu_H$  [FLV04].

We propose a reduction from the **Max-2-Sat-3** problem. The input to the **Max-2-Sat-3** problem is a boolean formula  $\phi$  in conjunctive normal form in which each clause contains at most 2 literals and each variable appears in at most 3 of the clauses, counting together both positive and negative occurrences.

The optimization problem calls for a truth assignment that satisfies as many clauses as possible. It is known that the Max-2-Sat-3 problem is **APX**-hard [BK99,ACG<sup>+</sup>99]. 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, hence a self-reduction would apply.

Assume given an input  $\phi$  to the Max-2-Sat-3 problem. Let  $X = \{x_1, x_2, \dots, x_n\}$  denote the set of variables and  $C = \{c_1, c_2, \dots, c_m\}$  the set of clauses. We now detail the construction of the corresponding instance of the Max-(2, 1)-Matching problem. To  $\phi$ , we associate a bipartite graph  $G_\phi$  defined as follows. The set of vertices is  $\mathbf{V}(G_\phi) = V_X^G \cup V_C^G$  where  $V_X^G = \{x_i^G : 1 \leq i \leq n\}$  and  $V_C^G = \{c_j^G : 1 \leq j \leq m\}$ , and  $\{x_i^G, c_j^G\}$  is an edge in  $\mathbf{E}(G_\phi)$  if and only if the clause  $c_j$  contains a literal on  $x_i$ . To  $\phi$ , we also associate a second bipartite graph  $H_\phi$  defined as follows. The set of vertices is  $\mathbf{V}(H_\phi) = V_X^H[T] \cup V_X^H[F] \cup V_C^H[1] \cup V_C^H[2]$  where  $V_X^H[T] = \{x_i^H[T] : 1 \leq i \leq n\}$ ,  $V_X^H[F] = \{x_i^H[F] : 1 \leq i \leq n\}$ ,  $V_C^H[1] = \{c_j^H[1] : 1 \leq j \leq m\}$  and  $V_C^H[2] = \{c_j^H[2] : 1 \leq j \leq m\}$ . Now,  $\{x_i^H[T], c_j^H[\ell]\}$  is an edge in  $\mathbf{E}(H_\phi)$  if and only if the  $(3 - \ell)$ -th literal of  $c_j$  is a literal of  $x_i$  or the  $\ell$ -th literal of  $c_j$  is the positive literal  $x_i$  ( $\ell \in \{1, 2\}$ ). Similarly,  $\{x_i^H[F], c_j^H[\ell]\}$  is an edge in  $\mathbf{E}(H_\phi)$  if and only if the  $(3 - \ell)$ -th literal of  $c_j$  is a literal of  $x_i$  or the  $\ell$ -th literal of  $c_j$  is the negative literal  $\bar{x}_i$ . We now turn to defining the associated lists. To each  $x_i^G \in V_X^G$  we associate the list  $\mathcal{L}(x_i^G) = \{x_i^H[T], x_i^H[F]\}$ . To each  $c_j^G \in V_C^G$  we associate the list  $\mathcal{L}(c_j^G) = \{c_j^H[1], c_j^H[2]\}$ .

Clearly,  $\mu_G = 2$ ,  $\mu_H = 1$ ,  $\Delta(G) = 3$ ,  $\Delta(H) = 5$  (since each variable has both positive and negative occurrences), and both  $G_\phi$  and  $H_\phi$  are bipartite graphs. We now turn to proving correctness of the approximation-preserving reduction.

**Lemma 1.** *Every truth assignment for  $\phi$  that satisfies  $k$  clauses can be transformed, in polynomial-time, into an injective mapping  $\theta : \mathbf{V}(G_\phi) \mapsto \mathbf{V}(H_\phi)$  w.r.t. lists  $\mathcal{L}(u)$ ,  $u \in \mathbf{V}(G_\phi)$ , such that  $\#\text{Match}(G_\phi, H_\phi, \theta) = m + k$ .*

*Proof (Of Lemma 1).* Let  $f : X \mapsto \{\text{true}, \text{false}\}$  be a truth assignment for  $\phi$  that satisfies  $k$  clauses. Define a mapping  $\theta : \mathbf{V}(G_\phi) \mapsto \mathbf{V}(H_\phi)$  w.r.t. lists  $\mathcal{L}(u)$ ,  $u \in \mathbf{V}(G_\phi)$ , as follows. If  $\phi(x_i) = \text{true}$ , then define  $\theta(x_i^G) = x_i^H[T]$ , else define  $\theta(x_i^G) = x_i^H[F]$ . For every clause  $c_j$  whose second literal evaluates to true under the truth assignment  $f$ , define  $\theta(c_j^G) = c_j^H[2]$ . Define  $\theta(c_j^G) = c_j^H[1]$  for every other clause  $c_j \in C$ . Clearly,  $\theta$  is injective. Now, to prove  $\#\text{Match}(G_\phi, H_\phi, \theta) = m + k$ , we need the following two claims.

*Claim 1.* Let  $c_j \in C$  be a clause and  $x_i, x_p \in X$  be the two variables that occur in  $c_j$ . If  $c_j$  is not satisfied by the truth assignment  $f$ , then exactly one of  $\{\theta(x_i^G), \theta(c_j^G)\}$  and  $\{\theta(x_p^G), \theta(c_j^G)\}$  is an edge of  $H_\phi$ .

*Proof (Of Claim 1).* Wlog, let us assume that the first (resp. second) literal of  $c_j$  is a literal on  $x_i$  (resp.  $x_p$ ). Observe first that we must have  $\theta(c_j^G) = c_j^H[1]$  since  $c_j$  is not satisfied by the truth assignment  $f$ . Furthermore, by construction,  $\{\theta(x_i^G), \theta(c_j^G) = c_j^H[1]\}$  is an edge of  $H_\phi$ , whatever  $\theta(x_i^G)$  is. We are thus reduced to proving that  $\{\theta(x_p^G), \theta(c_j^G) = c_j^H[1]\}$  is not an edge of  $H_\phi$ . Suppose that the second literal of  $c_j$  is the negative literal  $\bar{x}_p$ . Since  $c_j$  is not satisfied by the truth assignment  $f$ , then it follows that  $f(x_p) = \text{true}$ . Therefore,  $\theta(x_p^G) = x_p^H[T]$ , and hence  $\{\theta(x_p^G) = x_p^H[T], \theta(c_j^G) = c_j^H[1]\}$  is not an edge of  $H_\phi$ . A similar argument applies if the second literal of  $c_j$  is the positive literal  $x_p$ .  $\square$

*Claim 2.* Let  $c_j \in C$  be a clause and  $x_i, x_p \in X$  be the two variables that occur in  $c_j$ . If  $c_j$  is satisfied by the truth assignment  $f$ , then both  $\{\theta(x_i^G), \theta(c_j^G)\}$  and  $\{\theta(x_p^G), \theta(c_j^G)\}$  are edges of  $H_\phi$ .

*Proof (Of Claim 2).* Wlog, let us assume that the first (resp. second) literal of  $c_j$  is a literal on  $x_i$  (resp.  $x_p$ ). We divide the proof of the claim in two parts: (i) the clause  $c_j$  is satisfied by its second literal and (ii) the clause  $c_j$  is satisfied by its first literal and not by its second literal.

Suppose first that the clause  $c_j$  is satisfied by its second literal. Then it follows that  $\theta(c_j^G) = c_j^H[2]$ . By construction,  $\{\theta(x_i^G), \theta(c_j^G) = c_j^H[2]\}$  is an edge of  $H_\phi$ , whatever  $\theta(x_i^G)$  is. What is left is thus to prove that  $\{\theta(x_p^G), \theta(c_j^G) = c_j^H[2]\}$  is an edge of  $H_\phi$ . Suppose that the second literal of  $c_j$  is the negative literal  $\bar{x}_p$ . Since  $c_j$  is satisfied by its second literal, then it follows that  $f(x_p) = \text{false}$ . Therefore,  $\theta(x_p^G) = x_p^H[T]$ , and hence  $\{\theta(x_p^G) = x_p^H[T], \theta(c_j^G) = c_j^H[2]\}$  is an edge of  $H_\phi$ . A similar argument applies if the second literal of  $c_j$  is  $x_p$ .

Suppose now that the clause  $c_j$  is satisfied by its first literal and not by its second literal. Then it follows that  $\theta(c_j^G) = c_j^H[1]$ . By construction,  $\{\theta(x_p^G), \theta(c_j^G) = c_j^H[1]\}$  is an edge of  $H_\phi$ , whatever the exact value of  $\theta(x_p^G)$ . We now prove that  $\{\theta(x_i^G), \theta(c_j^G) = c_j^H[2]\}$  is also an edge of  $H_\phi$ . Suppose that the first literal of  $c_j$  is the negative literal  $\overline{x_i}$ . Since  $c_j$  is satisfied by its second literal, then it follows that  $f(x_i) = \text{false}$ . Therefore,  $\theta(x_i^G) = x_i^H[1]$ , and hence  $\{\theta(x_i^G) = x_i^H[F], \theta(c_j^G) = c_j^H[2]\}$  is an edge of  $H_\phi$ . A similar argument applies if the first literal of  $c_j$  is the positive literal  $x_i$ .  $\square$

Combining Claim 1 and Claim 2, we thus obtain that if the truth assignment  $f$  satisfies  $k$  clauses, then  $\#\text{Match}(G_\phi, H_\phi, \theta) = 2k + (m - k) = m + k$ .  $\square$

It follows from Lemma 1 that, for any injective mapping  $\theta : \mathbf{V}(G_\phi) \mapsto \mathbf{V}(H_\phi)$  w.r.t. lists  $\mathcal{L}(u)$ ,  $u \in \mathbf{V}(G_\phi)$ , we have  $\#\text{Match}(G_\phi, H_\phi, \theta) \geq m$ .

**Lemma 2.** *Given an injective mapping  $\theta : \mathbf{V}(G_\phi) \mapsto \mathbf{V}(H_\phi)$  w.r.t. lists  $\mathcal{L}(u)$ ,  $u \in \mathbf{V}(G_\phi)$ , such that  $\#\text{Match}(G_\phi, H_\phi, \theta) = m + k$ , we can construct, in polynomial-time, a truth assignment for  $\phi$  that satisfies  $k$  clauses.*

*Proof (Of Lemma 2).* We first consider the structure of the injective mapping  $\theta$ .

*Claim 3.* Let  $c_j^G \in V_C^G$  and  $x_i^G, x_p^G \in V_X^G$  be such that  $\{x_i^G, c_j^G\} \in \mathbf{E}(G_\phi)$  and  $\{x_p^G, c_j^G\} \in \mathbf{E}(G_\phi)$ . Then, at least one of  $\{\theta(x_i^G), \theta(c_j^G)\}$  and  $\{\theta(x_p^G), \theta(c_j^G)\}$  is an edge of  $H_\phi$ .

*Proof (Of Claim 3).* By construction the corresponding clause  $c_j$  contains a literal on variable  $x_i$  and a literal on variable  $x_p$ . Wlog, let us assume that the first (resp. second) literal of the corresponding clause  $c_j$  is a literal on  $x_i$  (resp.  $x_p$ ). For one, if  $\theta(c_j^G) = c_j^H[2]$ , then  $\{\theta(x_i^G), \theta(c_j^G) = c_j^H[2]\}$  is an edge of  $H_\phi$ , whatever the exact value of  $\theta(x_i^G)$ . For another, if  $\theta(c_j^G) = c_j^H[1]$ , then  $\{\theta(x_p^G), \theta(c_j^G) = c_j^H[1]\}$  is an edge of  $H_\phi$ , whatever the exact value of  $\theta(x_p^G)$ .  $\square$

We now turn to constructing a truth assignment  $f : X \mapsto \{\text{true}, \text{false}\}$  for  $\phi$  as follows: for all  $x_i^G \in V_X^G$ , if  $\theta(x_i^G) = x_i^H[T]$  then define  $f(x_i) = \text{true}$ , else define  $f(x_i^G) = \text{false}$ .

*Claim 4.* Let  $c_j^G \in V_C^G$  and  $x_i^G, x_p^G \in V_X^G$  be such that  $\{x_i^G, c_j^G\} \in \mathbf{E}(G_\phi)$  and  $\{x_p^G, c_j^G\} \in \mathbf{E}(G_\phi)$ . If both  $\{\theta(x_i^G), \theta(c_j^G)\}$  and  $\{\theta(x_p^G), \theta(c_j^G)\}$  are edges of  $H_\phi$ , then the corresponding clause  $c_j$  is satisfied by the truth assignment  $f$ .

*Proof (Of Claim 4).* By construction, the corresponding clause  $c_j$  contains a literal on variable  $x_i$  and a literal on variable  $x_p$ . Wlog, let us assume that the first (resp. second) literal of the corresponding clause  $c_j$  is a literal on  $x_i$  (resp.  $x_p$ ).

Suppose first that  $\theta(c_j^G) = c_j^H[2]$ . For one, if  $\theta(x_p^G) = x_p^H[T]$ , then the second literal of the clause  $c_j$  is the positive literal  $x_p$ , and hence the truth assignment  $f(x_p) = \text{true}$  satisfies the clause  $c_j$ . For another, if  $\theta(x_p^G) = x_p^H[F]$ , then the second literal of the clause  $c_j$  is the negative literal  $\overline{x_p}$ , and hence the truth assignment  $f(x_p) = \text{false}$  satisfies the clause  $c_j$ . Suppose now that  $\theta(c_j^G) = c_j^H[1]$ . For one, if  $\theta(x_i^G) = x_i^H[T]$ , then the first literal of the clause  $c_j$  is the positive literal  $x_i$ , and hence the truth assignment  $f(x_i) = \text{true}$  satisfies the clause  $c_j$ . For another, if  $\theta(x_i^G) = x_i^H[F]$ , then the first literal of the clause  $c_j$  is the negative literal  $\overline{x_i}$ , and hence the truth assignment  $f(x_i) = \text{false}$  satisfies the clause  $c_j$ .  $\square$

According to Claim 3 and Claim 4, we thus obtain that if  $\#\text{Match}(G_\phi, H_\phi, \theta) = m + k$  then there exists a truth assignment  $f$  that satisfies  $k$  clauses of  $\phi$ .  $\square$

**Proposition 7.** *The Max-(2, 1)-Matching problem is APX-hard even if both  $G$  and  $H$  are bipartite graphs with  $\Delta(G) \leq 3$  and  $\Delta(H) \leq 5$ .*

*Proof (Of Proposition 7).* According to Lemma 1 and Lemma 2, there exists a truth assignment for  $\phi$  that satisfies  $k$  clauses if and only if there exists an injective mapping  $\theta : \mathbf{V}(G_\phi) \mapsto \mathbf{V}(H_\phi)$  w.r.t. lists  $\mathcal{L}(u)$ ,  $u \in \mathbf{V}(G_\phi)$ , such that  $\#\text{Match}(G_\phi, H_\phi, \theta) = m + k$ . It is now a simple matter check that this is an  $L$ -reduction [Pap94] with parameters  $\alpha = 2$  and  $\beta = 1$  from the Max-2-Sat-3 problem, which is known to be APX-hard [BK99, ACG<sup>+</sup>99], to the Max-(2, 1)-Matching problem with  $\Delta(G) \leq 3$  and  $\Delta(H) \leq 5$ .

By slightly complicating the proof, we can strengthen the above proposition.

**Proposition 8.** *The Max-(2,1)-Matching problem is **APX**-hard even if both  $G$  and  $H$  are bipartite graphs with  $\Delta(G) \leq 3$  and  $\Delta(H) \leq 3$ .*

## 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  $\Delta(H) \leq 3$ . We show in this section that the Max- $(\mu_G, 1)$ -Matching problem for bounded degree graphs  $G$  belongs to **APX** for any constant  $\mu_G$ , 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  $\mu_G^2$  for any constant  $\mu_G$ .

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  $G$ .

*Conjecture 1 (The linear arboricity conjecture [AEH81]).* The linear arboricity of every  $d$ -regular graph is  $\lceil (d+1)/2 \rceil$ .

This conjecture was shown to be asymptotically correct as  $d \rightarrow \infty$  [Alo88]. 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  $\Delta(G)$  [AS92]. The following lemma emphasizes on the connection between the linear arboricity and the matching number of a graph.

**Lemma 3.** *Let  $G$  be a graph. Then,  $\nu(G) \geq \mathbf{m}(G)(2\text{la}(G))^{-1}$ .*

*Proof (Of Lemma 3).* There exists a linear forest in  $G$  that contains at least  $\mathbf{m}(G)(\text{la}(G))^{-1}$  edges, and hence  $\nu(G) \geq \mathbf{m}(G)(2\text{la}(G))^{-1}$  since each connected component of a linear forest is a path.  $\square$

**Lemma 4.** *For any trim instance, the Max- $(\mu_G, 1)$ -Matching problem is approximable within ratio  $2\text{la}(G)$  in  $\mathcal{O}(\mathbf{n}(G) + \mathbf{m}(G)\sqrt{\mathbf{n}(G)})$  time for any constant  $\mu_G \geq 1$ .*

*Proof (Of Lemma 4).* Let  $(G, H, \mathcal{L})$  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  $\theta : \mathbf{V}(G) \mapsto \mathbf{V}(H)$  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,  $\theta$  is well-defined and is injective since  $\mu_H = 1$ .

So, if we let  $\theta$  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  $\#\mathcal{M} = \nu(G) \geq \mathbf{m}(G)(2\text{la}(G))^{-1}$  (Lemma 3), 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. For simplicity, let us assume that  $(G, H, \mathcal{L})$  is a trim instance. Finding a maximum matching in  $G$  is an  $\mathcal{O}(\mathbf{m}(G)\sqrt{\mathbf{n}(G)})$  time procedure [MV80]. Since constructing  $\theta$  is an  $\mathcal{O}(\mu_G^2 \nu(G) + \mathbf{n}(G) - 2\nu(G)) = \mathcal{O}(\mathbf{m}(G) + \mathbf{n}(G))$  time procedure, the algorithm, as a whole, runs in  $\mathcal{O}(\mathbf{n}(G) + \mathbf{m}(G)\sqrt{\mathbf{n}(G)})$  time.  $\square$

**Proposition 9.** *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  $\mu_G$ .*

*Proof (Of Proposition 9).* Combine Lemma 4 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 1.** *The Max- $(2, 1)$ -Matching problem is APX-complete even if both  $G$  and  $H$  are bipartite graphs with  $\Delta(G) \leq 3$  and  $\Delta(H) \leq 3$ .*

**Corollary 2.** *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  $\mu_G$ .*

We now turn to giving a fast randomized algorithm for the Max- $(\mu_G, 1)$ -Matching problem. The proof makes use of the *probabilistic method* [AS92], a powerful tool for demonstrating the existence of combinatorial objects.

**Lemma 5.** *Let  $(G, H, \mathcal{L})$  be a trim instance of the Max- $(\mu_G, 1)$ -Matching problem. For any  $\mu_G$ , there exists an injective mapping  $\theta : \mathbf{V}(G) \mapsto \mathbf{V}(H)$  w.r.t. lists  $\mathcal{L}(u)$ ,  $u \in \mathbf{V}(G)$ , such that  $\#\text{Match}(G, H, \theta) \geq \mu_G^{-2} \mathbf{m}(G)$ .*

*Proof (Of Lemma 5).* The proof is by the probabilistic method. For each  $u \in \mathbf{V}(G)$  with  $\mathcal{L}(u) = \{v_1, v_2, \dots, v_q\}$ ,  $q \leq \mu_G$ , suppose that  $\theta(u)$  is set to  $v_1, v_2, \dots$ , or  $v_q$  independently and equiprobably. Since  $\mu_H = 1$ , it follows that  $\theta$  is an injective mapping from  $\mathbf{V}(G)$  to  $\mathbf{E}(H)$  w.r.t. lists  $\mathcal{L}(u)$ ,  $u \in \mathbf{V}(G)$ . For each  $\{u, v\} \in \mathbf{E}(G)$ , let  $\mathcal{E}(\{u, v\}) = 1$  if  $\{\theta(u), \theta(v)\} \in \mathbf{E}(H)$ , and 0 otherwise. For any edge  $\{u, v\} \in \mathbf{V}(G)$ , the probability that it is matched by the injective mapping  $\theta$  is at least  $\mu_G^{-2}$  (since  $(G, H, \mathcal{L})$  is a trim instance), implying  $\mathbf{Exp}[\mathcal{E}(\{u, v\})] \geq \mu_G^{-2}$ . The expected number of edge matches by this random injective mapping  $\theta$  is  $\sum_{\{u, v\} \in \mathbf{E}(G)} \mathbf{Exp}[\mathcal{E}(\{u, v\})] \geq \mu_G^{-2} \mathbf{m}(G)$ . Then, there exists at least one injective mapping  $\theta : \mathbf{V}(G) \mapsto \mathbf{V}(H)$  w.r.t. lists  $\mathcal{L}(u)$ ,  $u \in \mathbf{V}(G)$ , such that  $\sum_{\{u, v\} \in \mathbf{E}(G)} \mathcal{E}(\{u, v\}) \geq \mu_G^{-2} \mathbf{m}(G)$ .  $\square$

**Corollary 3.** *There is linear-time randomized algorithm that achieves a performance ratio  $\mu_G^2$  for the Max- $(\mu_G, 1)$ -Matching problem restricted to trim instances with unbounded degree graphs  $G$  and  $H$ .*

*Proof (Of Corollary 3).* It is sufficient to observe that the proof of Lemma 5 yields a linear-time randomized  $\mu_G^2$ -approximation algorithm for unbounded degree graphs  $G$  and  $H$ .  $\square$

## 6 Fixed-parameter tractability

Parameterized complexity [DF99] 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  $f(k) n^{\mathcal{O}(1)}$  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. We designate the class of fixed-parameter tractable problems **FPT**. In the last decade, parameterized complexity has proved to be extremely useful in computational molecular biology, see for example [BDF<sup>+</sup>95, GGN02, AGGN02].

We follow here this trend by showing in this section that the Max- $(\mu_G, 1)$ -Matching problem for 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 these two algorithms is thus the *incompatibility graph* of any instance  $(G, H, \mathcal{L})$  which is latter shown to be a compact representation of the problem.

**Definition 2 (Incompatibility graph).** Let  $(G, H, \mathcal{L})$  be a trim instance of the  $\text{Max}-(\mu_G, 1)$ -Matching problem and  $<$  be an arbitrary total order on  $\mathbf{V}(G)$ . The incompatibility graph of  $(G, H, \mathcal{L})$ , written  $I[G, H, \mathcal{L}]$ , is defined by

$$\begin{aligned} \mathbf{V}(I[G, H, \mathcal{L}]) &= \{(u, v, u', v') : u < v \wedge \{u, v\} \in \mathbf{E}(G) \wedge \{u', v'\} \in \mathbf{E}(H) \wedge u' \in \mathcal{L}(u) \wedge v' \in \mathcal{L}(v)\} \\ \mathbf{E}(I[G, H, \mathcal{L}]) &= \bigcup_{1 \leq i \leq 5} E_i \end{aligned}$$

where

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

Observe that in  $E_4$  (resp.  $E_5$ ),  $u = y$  (resp.  $v = x$ ) implies  $v \neq x$  (resp.  $u \neq y$ ) since  $x < y = u < v$  (resp.  $u < v = x < y$ ) by definition of  $\mathbf{V}(I[G, H, \mathcal{L}])$ . Most of the interest in the incompatibility graph  $I[G, H, \mathcal{L}]$  stems from the following lemma.

**Lemma 6.** Let  $(G, H, \mathcal{L})$  be a trim instance of the  $\text{Max}-(\mu_G, 1)$ -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  $\theta$  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  $\mu_G$ , if  $G$  is a bounded degree graph, then so is the incompatibility graph  $I[G, H, \mathcal{L}]$ .

**Lemma 7.** Let  $(G, H, \mathcal{L})$  be an instance of the  $\text{Max}-(\mu_G, 1)$ -Matching problem. Then,  $I[G, H, \mathcal{L}]$  has maximum degree at most  $(\mu_G - 1)(2\mu_G\Delta(G) - \mu_G + 1)$ .

It follows from the above lemma that  $\Delta(I[G, H, \mathcal{L}]) = \mathcal{O}(\Delta(G))$  when  $\mu_G = \mathcal{O}(1)$ , 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  $\text{Max}-(\mu_G, 1)$ -Matching problem.

**Proposition 10.** The  $\text{Max}-(\mu_G, 1)$ -Matching problem is solvable in  $\mathcal{O}(\mathbf{m}(G)(D+1)^k)$  time, where  $k$  is the number of matched edges, i.e.,  $\#\text{Match}(G, H, \theta)$ , and  $D = \Delta(I[G, H, \mathcal{L}]) = (\mu_G - 1)(2\mu_G\Delta(G) - \mu_G + 1) = \mathcal{O}(\Delta(G))$ , and hence is fixed-parameter tractable for parameter  $k$ , provided that  $G$  is a bounded degree graph and  $\mu_G$  is a constant.

## 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  $\text{Exact}-(3, 2)$ -Matching problem and the  $\text{Max}-(2, 1)$ -Matching problem for bounded degree bipartite graphs to be **NP**-complete and **APX**-complete, respectively. The latter problem was shown to be fixed-parameter tractable parameterized by the number of matched edges.

We mention some possible directions for future works. An interesting line of research is to further investigate the approximation of the  $\text{Max}-(\mu_G, \mu_H)$ -Matching problem for bounded degree graphs  $G$  and  $H$ . For example, is the  $\text{Max}-(2, 2)$ -Matching problem for bounded degree graphs  $G$  and  $H$  in **APX**? Parameterized complexity of the  $\text{Max}-(\mu_G, \mu_H)$ -Matching problem is almost unexplored in the case  $\mu_H > 1$ . In particular, is the  $\text{Max}-(\mu_G, \mu_H)$ -Matching problem for bounded degree graphs  $G$  and  $H$  fixed-parameter tractable for any constant  $\mu_G$  and  $\mu_H$ ?

## References

- [ACG<sup>+</sup>99] G. Ausiello, P. Crescenzi, G. Gambosi, V. Kann, A. Marchetti-Spaccamela, and M. Protasi, *Complexity and Approximation: Combinatorial optimization problems and their approximability properties*, Springer-Verlag, 1999.
- [AEH81] J. Akiyama, G. Exoo, and F. Harary, *Covering and packing in graphs IV: Linear arboricity*, *Networks* **11** (1981), 69–72.
- [AGGN02] 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. of the 13th Annual Symposium on Combinatorial Pattern Matching (CPM 2002), LNCS, vol. 2373, Springer-Verlag, 2002, pp. 99–114.
- [Alo88] N. Alon, *The linear arboricity of graphs*, *Israel Journal of Mathematics* **62** (1988), no. 3, 311–325.
- [APT79] B. Aspvall, M.F. Plass, and R.E. Tarjan, *A linear-time algorithm for testing the truth of certain quantified boolean formulas*, *Information Processing Letters* **8(3)** (1979), 121–123.
- [AS92] N. Alon and J.H. Spencer, *The probabilistic method*, Wiley, 1992.
- [BDF<sup>+</sup>95] 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.
- [BK99] P. Berman and M. Karpinski, *On some tighter inapproximability results*, Proc. of the 26th International Colloquium on Automata, Languages and Programming (ICALP) (P. van Emde Boas J. Wiedermann and M. Nielsen, eds.), LNCS, vol. 1644, Springer, 1999, pp. 200–209.
- [DF99] R. Downey and M. Fellows, *Parameterized complexity*, Springer-Verlag, 1999.
- [FLV04] I. Fagnot, G. Lelandais, and S. Vialette, *Bounded list injective homomorphism for comparative analysis of protein-protein interaction graphs*, Proc. of the 1st Algorithms and Computational Methods for Biochemical and Evolutionary Networks (CompBioNets), KCL publications, 2004, pp. 45–70.
- [Gav02] 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.
- [GGN02] J. Gramm, J. Guo, and R. Niedermeier, *Pattern matching for arc-annotated sequences*, Proc. of the the 22nd Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS), LNCS, vol. 2556, 2002, pp. 182–193.
- [GJ79] M.R. Garey and D.S. Johnson, *Computers and intractability: a guide to the theory of NP-completeness*, W.H. Freeman, San Francisco, 1979.
- [HG02] Y. Ho and A. Gruhler *et al.*, *Systematic identification of protein complexes in *Saccharomyces cerevisiae* by mass spectrometry*, *Nature* **415** (2002), no. 6868, 180–183.
- [KSK<sup>+</sup>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.
- [MV80] S. Micali and V.V. Vazirani, *An  $O(\sqrt{|V||E|})$  algorithm for finding maximum matching in general graphs*, Proc. of the 21st Annual Symposium on Foundation of Computer Science (FOCS), IEEE, 1980, pp. 17–27.
- [Pap94] C.H. Papadimitriou, *Computational complexity*, Addison-Wesley, 1994.
- [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<sup>+</sup>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.
- [SIK<sup>+</sup>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. of the 8th annual international conference on Computational molecular biology (RECOMB 2004), ACM Press, 2004, pp. 282–289.
- [SSK<sup>+</sup>05] 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** (2005), no. 6, 1974–1979.
- [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.
- [Uet00] P. Uetz, L.Giot *et al.*, *A comprehensive analysis of protein-protein interactions in *Saccharomyces cerevisiae**, *Nature* **403** (2000), no. 6770, 623–627.

## Appendix (Program committee version only)

*Proof (Of Proposition 3).* 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  $\mathcal{O}(\mu_G^2 k) = \mathcal{O}(\mathbf{n}(G))$  time dynamic programming algorithm.

Suppose now that  $G$  is a cycle of length  $k+1$ . Again, 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 subproblem 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  $\mu_G^2$  subproblems, each of them can be solved in  $\mathcal{O}(\mathbf{n}(G))$  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  $\mathcal{O}(\mu_G^2 \mathbf{n}(G)) = \mathcal{O}(\mathbf{n}(G))$  time algorithm.  $\square$

*Proof (Of Proposition 6).* The proof is by direct application of the Lovasz local lemma (in the symmetric case). 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  $\theta$  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  $\theta$ . For one,

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

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,

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

According to the Lovasz local lemma in the symmetric case [AS92], we now thus obtain

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

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

---

*Proof (Of Proposition 8).* Consider the construction from Max-2-Sat-3 problem described above. First, notice that the only vertices of  $H_\phi$  with degree greater than three are the vertices in  $V_X^H[T] \cup V_X^H[F]$ . The basic idea is simply to reduce the degree of these vertices. To this aim, instead of the graph  $G_\phi$ , we propose the graph  $G'_\phi$  obtained from  $G_\phi$  by subdividing each edge of  $G_\phi$  with the insertion of a new vertex. Clearly,  $G'_\phi$  is bipartite and  $\Delta(G'_\phi) \leq \Delta(G_\phi) = 3$ . Furthermore, consider the new graph  $H'_\phi$  that differs from  $H_\phi$  in that, for every vertex  $x_i^H[T]$  (resp.,  $x_i^H[F]$ ) it contains three further vertices  $\{x_i^H[T, 1], x_i^H[T, 2], x_i^H[T, 3]\}$  (resp.,  $\{x_i^H[F, 1], x_i^H[F, 2], x_i^H[F, 3]\}$ ). These three vertices are the three neighbors of  $x_i^H[T]$  (resp.  $x_i^H[F]$ ), while the other edges of  $H'_\phi$  are introduced by the following rule: first,  $\{x_i^H[T; \ell], c_j^H[\ell]\}$  is an edge in  $\mathbf{E}(H'_\phi)$  if and only if the  $(3 - \ell)$ -th literal of  $c_j$  is a literal of  $x_i$  or the  $\ell$ -th literal of  $c_j$  is the positive literal  $x_i$ . Similarly,  $\{x_i^H[F, \ell], c_j^H[\ell]\}$  is an edge in  $\mathbf{E}(H'_\phi)$  if and only if the  $(3 - \ell)$ -th literal of  $c_j$  is a literal of  $x_i$  or the  $\ell$ -th literal of  $c_j$  is the negative literal  $\bar{x}_i$ .  $\square$

---

*Proof (Of Lemma 6).* Suppose that there exists an injective mapping  $\theta : \mathbf{V}(G) \mapsto \mathbf{V}(H)$  w.r.t. lists  $\mathcal{L}(u)$ ,  $u \in \mathbf{V}(G)$ , such that  $\#\{\{u, v\} \in \mathbf{E}(G) : \{\theta(u), \theta(v)\} \in \mathbf{E}(H)\} = k$ . For simplicity, suppose also there exists a total order  $<$  on  $\mathbf{V}(G)$ . Let  $S_k$  be the set of vertices of  $I[G, H, \mathcal{L}]$  defined as follows: for each edge  $\{u, v\} \in \mathbf{E}(G)$  such that  $u < v$  and  $\{\theta(u), \theta(v)\} \in \mathbf{E}(H)$ ,  $(u, v, \theta(u), \theta(v)) \in S_k$ . Clearly,  $S_k$  is of cardinality  $k$ . We claim that  $S_k$  is an independent set of  $I[G, H, \mathcal{L}]$ . Indeed, take any two distinct vertices  $U_1 = (u_1, v_1, \theta(u_1), \theta(v_1))$  and  $U_2 = (u_2, v_2, \theta(u_2), \theta(v_2))$  of  $S_k$ , and let us show that no edge can connect them in  $I[G, H, \mathcal{L}]$ . There are several cases to consider:

- $u_1 = u_2$  and  $v_1 = v_2$ . Since we start from an injective mapping  $\theta$ , this implies  $\theta(u_1) = \theta(u_2)$  and  $\theta(v_1) = \theta(v_2)$ . This means that  $U_1 = U_2$ , a contradiction.
- $u_1 = u_2$  and  $v_1 \neq v_2$ . According to the definition of  $I[G, H, \mathcal{L}]$ , if  $U_1$  and  $U_2$  are connected, then they must be connected by an edge of  $E_2$ . However, this is the case only if  $\theta(u_1) \neq \theta(u_2)$ . Since the mapping  $\theta$  we start from is injective, this case cannot occur, and therefore  $U_1$  and  $U_2$  are not connected by an edge in  $I[G, H, \mathcal{L}]$ .  
Thus, we can now suppose wlog that  $u_1 < u_2$ .
- $u_1 \neq u_2$  and  $v_1 = v_2$ . Similarly to the previous case, if  $U_1$  and  $U_2$  are connected, then they must be connected by an edge of  $E_3$ . However, this is the case only if  $\theta(v_1) \neq \theta(v_2)$ . Since  $\theta$  is injective, this case cannot occur, and therefore  $U_1$  and  $U_2$  are not connected by an edge in  $I[G, H, \mathcal{L}]$ .
- $u_1 = v_2$ . Note that since  $u_1 < v_1$  and  $u_2 < v_2$ , this implies  $v_1 \neq u_2$ . According to the definition of  $I[G, H, \mathcal{L}]$ , if  $U_1$  and  $U_2$  are connected, then they must be connected by an edge of  $E_4$ . However, this is the case only if  $\theta(u_1) \neq \theta(v_2)$ . Since  $\theta$  is injective, this case cannot occur, and therefore  $U_1$  and  $U_2$  are not connected by an edge in  $I[G, H, \mathcal{L}]$ .
- $v_1 = u_2$ . Note that since  $u_1 < v_1$  and  $u_2 < v_2$ , this implies  $u_1 \neq v_2$ . Similarly to the previous case, if  $U_1$  and  $U_2$  are connected, then they must be connected by an edge of  $E_5$ . However, this is the case only if  $\theta(v_1) \neq \theta(u_2)$ . Since  $\theta$  is injective, this case cannot occur, and therefore  $U_1$  and  $U_2$  are not connected by an edge in  $I[G, H, \mathcal{L}]$ .
- $u_1 \neq u_2$ ,  $u_1 \neq v_2$ ,  $v_1 \neq u_2$  and  $v_1 \neq v_2$ . In that case, by definition of  $I[G, H, \mathcal{L}]$ , no edge connects  $U_1$  to  $U_2$ .

Thus, for any two distinct vertices  $U_1$  and  $U_2$  in  $S_k$  no edge connects  $U_1$  to  $U_2$ , and therefore  $S_k$  is an independent set, of cardinality  $k$ , in  $I[G, H, \mathcal{L}]$ .

Conversely, suppose that there exists an independent set  $S_k$  of cardinality  $k$  in  $I[G, H, \mathcal{L}]$ . The mapping  $\theta$  we construct is the following:

- for any vertex  $(u, v, u', v') \in S_k$ ,  $u' = \theta(u)$  and  $v' = \theta(v)$ . Call this partial mapping  $\theta_1$ .
- for any vertex  $u$  in  $G$  that has no image by  $\theta_1$ , affect arbitrarily one of the images of  $\mathcal{L}(u)$ . Call this partial mapping  $\theta_2$ .

The construction of  $\theta_2$  is always feasible, since we suppose  $\mu_H = 1$ . This means that for every vertex  $u \in \mathbf{V}(G)$ , its images in  $\mathcal{L}(u)$  are “private”, and in particular cannot be used by another vertex. Hence, by the mapping constructed this way, every vertex of  $G$  has an image.

Let us now prove that the mapping is injective. For this, we first note that the case where  $u$  and  $v$  are two distinct vertices of  $G$  with  $\theta(u) = \theta(v)$  cannot appear, since  $\mathcal{L}(u) \cap \mathcal{L}(v) = \emptyset$  for any  $u, v \in \mathbf{V}(G)$ . Now let us suppose that  $\theta(u)$  and  $\theta(v)$  are two distinct images in  $H$  such that they have a common antecedent  $w$  in  $G$ . Clearly, by definition of  $\theta_2$ , this only could happen for two vertices  $u$  and  $v$  both having an image by  $\theta_1$ . Moreover, we note that  $\theta(u)$  and  $\theta(v)$  cannot be part of a single vertex  $(w, w, \theta(u), \theta(v))$  of  $S_k$  because in that case, by definition of  $I[G, H, \mathcal{L}]$ , we would have  $w < w$ , a contradiction. Thus  $\theta(u)$  and  $\theta(v)$  are part of distinct vertices  $P_1 = (p_1, q_1, p'_1, q'_1)$  and  $P_2 = (p_2, q_2, p'_2, q'_2)$  in  $S_k$ . There are four cases to consider:

- $\theta(u) = p'_1$  and  $\theta(v) = p'_2$ . Then  $P_1 = (w, q_1, \theta(u), q'_1)$  and  $P_2 = (w, q_2, \theta(v), q'_2)$ . In that case,  $P_1$  and  $P_2$  are either connected by an edge in  $E_1$  or by an edge in  $E_2$ , a contradiction.
- $\theta(u) = p'_1$  and  $\theta(v) = q'_2$ . Then  $P_1 = (w, q_1, \theta(u), q'_1)$  and  $P_2 = (p_2, w, p'_2, \theta(v))$ . In that case,  $P_1$  and  $P_2$  are connected by an edge in  $E_4$ , a contradiction.
- $\theta(u) = q'_1$  and  $\theta(v) = p'_2$ . Then  $P_1 = (p_1, w, p'_1, \theta(u))$  and  $P_2 = (w, q_2, \theta(v), q'_2)$ . In that case,  $P_1$  and  $P_2$  are connected by an edge in  $E_5$ , a contradiction.
- $\theta(u) = q'_1$  and  $\theta(v) = q'_2$ . Then  $P_1 = (p_1, w, p'_1, \theta(v))$  and  $P_2 = (p_2, w, p'_2, \theta(v))$ . In that case,  $P_1$  and  $P_2$  are either connected by an edge in  $E_1$  or by an edge in  $E_3$ , a contradiction.

By definition of  $I[G, H, \mathcal{L}]$ , we know that any vertex  $(u, v, u', v')$  of  $S_k$  is such that  $(u, v) \in \mathbf{E}(G)$  and  $(u', v') \in \mathbf{E}(H)$ ,  $u' \in \mathcal{L}(u)$  and  $v' \in \mathcal{L}(v)$ . Because  $\theta$  defined above is an injective mapping, and because  $S_k$  is of cardinality  $k$ , we conclude that  $\#\{u, v\} \in \mathbf{E}(G) : \{\theta(u), \theta(v)\} \in \mathbf{E}(H)\} = k$ , and the lemma is proved.  $\square$

---

*Proof (of Lemma 7).* 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  $\mu_G$  images by  $\mathcal{L}$ , there are at most  $\mu_G^2$  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  $\mu_G$  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 :

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

$\square$

---

*Proof (of Proposition 10).* By standard bounded search techniques, one can find an independent set of size  $k$  in a graph  $G$  in  $\mathcal{O}(\mathbf{n}(G) (\Delta(G) + 1)^k)$  time, or return that no such subset exists. The proposition thus follows from applying this to the incompatibility graph  $I[G, H, \mathcal{L}]$ , noting that  $\mathbf{n}(I[G, H, \mathcal{L}]) = \mathcal{O}(\mathbf{m}(G))$  since  $\mu_G$  is supposed constant.  $\square$

---