Paths, Trees and Matchings under Disjunctive Constraints [§]

Andreas Darmann [**] Ulrich Pferschy [*] Joachim Schauer [*]
Gerhard J. Woeginger [‡]

Abstract

We study the minimum spanning tree problem, the maximum matching problem and the shortest path problem subject to binary disjunctive constraints: A *negative disjunctive constraint* states that a certain pair of edges cannot be contained simultaneously in a feasible solution. It is convenient to represent these negative disjunctive constraints in terms of a so-called *conflict graph* whose vertices correspond to the edges of the underlying graph, and whose edges encode the constraints.

We prove that the minimum spanning tree problem is strongly $\mathcal{NP}$-hard, even if every connected component of the conflict graph is a path of length two. On the positive side, this problem is polynomially solvable if every connected component is a single edge (that is, a path of length one). The maximum matching problem is $\mathcal{NP}$-hard for conflict graphs where every connected component is a single edge.

Furthermore we will also investigate these graph problems under *positive disjunctive constraints*: In this setting for certain pairs of edges, a feasible solution must contain at least one edge from every pair. We establish a number of complexity results for these variants including APX-hardness for the shortest path problem.

Keywords: minimal spanning tree; matching; shortest path; conflict graph; binary constraints.

[§]This research has been supported by the Netherlands Organisation for Scientific Research (NWO), grant 639.033.403, and by BSIK grant 03018 (BRICKS: Basic Research in Informatics for Creating the Knowledge Society).

[**]University of Graz, Institute of Public Economics, Universitaetsstr. 15, A-8010 Graz, Austria, `andreas.darmann@uni-graz.at`

[*]University of Graz, Department of Statistics and Operations Research, Universitaetsstr. 15, A-8010 Graz, Austria, `{pferschy, joachim.schauer}@uni-graz.at`

[‡]TU Eindhoven, Department of Mathematics and Computer Science, P.O. Box 513, NL-5600 MB Eindhoven, The Netherlands, `gwoegi@win.tue.nl`

1 Introduction

We study variants of the minimum spanning tree problem (MST), of the maximum matching problem (MM) and of the shortest path problem (SP) in weighted, undirected, connected graphs. These variants are built around binary disjunctive constraints on certain pairs of edges.

- A negative disjunctive constraint expresses an incompatibility or a conflict between the two edges in a pair. From each conflicting pair, at most one edge can occur in a feasible solution.
- A positive disjunctive constraint enforces that at least one edge from the underlying pair is in a feasible solution.

Throughout we will represent these binary disjunctive constraints by means of an undirected constraint graph: Every vertex of the constraint graph corresponds to an edge in the original graph, and every edge corresponds to a binary constraint. In the case of negative disjunctive constraints this constraint graph will be called *conflict graph*, and in the case of positive disjunctive constraints this graph will be called *forcing graph*.

For a formal definition of the *minimum spanning tree problem with conflict graph* (MSTCG), the *maximum matching problem with conflict graph* (MMCG) and the *shortest path problem with conflict graph* (SPCG), let $G = (V, E)$ be an undirected connected graph with n vertices and m edges. Every edge e has a weight $w(e)$ (where w is a weight function $w : E \to \mathbb{R}$). Furthermore, the undirected graph $\bar{G} = (E, \bar{E})$ represents the conflict graph where each of the m vertices corresponds uniquely to an edge $e \in E$ of G. An edge $\bar{e} = (i, j) \in \bar{E}$ implies that the two vertices incident to $\bar{e}$ – that is, the two edges $i, j \in E$ – cannot occur together in a spanning tree or maximum matching of G. In contrast to graph G, the conflict graph $\bar{G}$ is not necessarily connected and may contain isolated vertices (i.e. edges of G which can be combined with all other edges in the minimum spanning tree solution). MSTCG asks for a minimum spanning tree T in G, given that adjacent vertices in $\bar{G}$ are not both together included in T, MMCG asks for a maximum matching in G, given that adjacent vertices in $\bar{G}$ do not both belong to the maximum matching and SPCG asks for a shortest **simple** path in G, given that adjacent vertices in $\bar{G}$ do not both belong to the shortest path.

Similarly, we define the problems *minimum spanning tree problem with forcing graph* (MSTFG), *maximum matching problem with forcing graph* (MMFG) and *shortest path problem with forcing graph* (SPFG) around positive disjunctive constraints. Every vertex of a forcing graph $\bar{H} = (E, \bar{E})$ corresponds to an edge $e \in E$ of G, and an edge $\bar{e} = (i, j) \in \bar{E}$ implies that at least one of the two vertices incident to $\bar{e}$ – that is, at least one of the two edges $i, j \in E$ – has to be included in a spanning tree or maximum matching of G. Again the graph $\bar{H}$ is not necessarily connected and may contain isolated vertices.

Note that for all considered problems MSTCG, MMCG, MSTFG, MMFG, SPCG and SPFG the existence of a feasible solution is not at all guaranteed.

In this paper we will characterize the complexity of MSTCG, MMCG, MSTFG, MMFG, and SPFG and we will identify graph classes for the conflict (forcing) graph $\bar{G}$ ($\bar{H}$) where the computationally complexity jumps from polynomially solvable to strongly $\mathcal{NP}$-hard. For illustrative reasons we introduce the following terminology.

Definition 1 *A* 2-*ladder is an undirected graph whose components are paths of length one, i.e. edges connecting pairs of vertices.*

Definition 2 *A* 3-*ladder is an undirected graph whose components are paths of length two.*

Results of this paper. For the minimum spanning tree problem we establish a sharp separation line between easy and hard instances. The results of Section 2 and Section 4 establish that problems MSTCG and MSTFG are strongly $\mathcal{NP}$-hard, even if the underlying conflict (forcing) graph is a 3-ladder. On the other hand, we show by a matroid intersection argument in Section 3 and in Section 5 that the minimum spanning tree problem is polynomially solvable for a 2-ladder as conflict (forcing) graph.

The considered variants of the maximum matching problem seem to be universally hard: Section 6 and Section 7 show that problems MMCG and MMFG are strongly $\mathcal{NP}$-hard even for 2-ladder conflict (forcing) graphs.

The shortest path problem with conflict graphs is known to be NPO PB-complete [Kan94], even if the conflict graph is a 2-ladder. For SPFG we will show in Section 8 as a complementary result that this problem is already APX-hard for a 2-ladder as forcing graph. Note that the results for SPCG and SPFG hold even for the unweighted case where the number of edges of the path is minimized.

Related results. Results of a similar flavor have been derived recently for the 0-1 knapsack problem with conflict graphs. While this problem is strongly $\mathcal{NP}$-hard for arbitrary conflict graphs, it was shown in [PS09] that pseudopolynomial algorithms (and hence also fully polynomial approximation schemes) exist if the given conflict graph is a tree, a graph of bounded treewidth or a chordal graph. Bin packing problems with special classes of conflict graphs were considered from an approximation point of view by [JO97] and [Jan99]. Complexity results for different classes of conflict graphs for a scheduling problem under makespan minimization are given in [BJ93]. Further references on combinatorial optimization problems with conflict graphs can be found in [PS09].

2 A strong $\mathcal{NP}$-hardness result for MSTCG

In this section we show that MSTCG is strongly $\mathcal{NP}$-hard even if the conflict graph $\bar{G}$ is a 3-ladder. As an example, consider a component of $\bar{G}$ that consists of the path $(e_1 e_2 e_3)$ on the three edges $e_1, e_2, e_3 \in E$: If a feasible spanning tree for the underlying graph G contains the edge e_2, then it must neither include edge e_1 nor edge e_3. And if a feasible tree contains edge e_1, then it must not contain e_2, but may contain e_3.

2.1 The graphs G_{MSTCG} and $\bar{G}_{MSTCG}$

We reduce the NP-complete problem *(3,B2)-SAT* [BKS03] to special instances of MSTCG which are described by a graph G_{MSTCG} in which a spanning tree has to be found subject to a conflict graph $\bar{G}_{MSTCG}$. *(3,B2)-SAT* is the special symmetric subproblem of $3-SAT$ in which each clause has size three and each literal occurs exactly twice. This means that each variable occurs exactly four times, twice negated and twice nonnegated.

Let I be an arbitrary instance of *(3,B2)-SAT* with k clauses C_j and n variables. We define the graph G_{MSTCG} in the following way (see Figure 1):

For each variable x_i we introduce a cycle of length four $CY_i = (y_i, x_i, \tilde{y}_i, \bar{x}_i)$. Vertex y_i is connected to a dedicated vertex r of the graph G_{MSTCG}. The opposite vertex $\tilde{y}_i$ is only connected to its neighbors on this cycle by edges $e(x_i) := (x_i, \tilde{y}_i)$ and $e(\bar{x}_i) := (\bar{x}_i, \tilde{y}_i)$. Clearly in each spanning tree of G_{MSTCG}, for reaching the vertex $\tilde{y}_i$, at least one of $e(x_i)$ and $e(\bar{x}_i)$ has to be included.

Furthermore for each clause C_j of I we define a vertex C_j in G_{MSTCG} that is connected to the cycles of length four in the following way: If the literal x_i resp. $\bar{x}_i$ occurs in clause C_j we connect vertex C_j by an edge to vertex x_i resp. $\bar{x}_i$, and call this edge e_{i1} resp. $\bar{e}_{i1}$, or e_{i2} resp. $\bar{e}_{i2}$ if the former name was already used. By the symmetric structure of *(3,B2)-SAT*, for each cycle CY_i all the edges e_{i1}, e_{i2}, $\bar{e}_{i1}$ and $\bar{e}_{i2}$ will be in G_{MSTCG}.

The conflict graph $\bar{G}_{MSTCG}$ on the edges of G_{MSTCG} consists of $2n$ 3-ladders, i.e. paths of length three, namely $(e_{i1}, e(\bar{x}_i), e_{i2})$ and $(\bar{e}_{i1}, e(x_i), \bar{e}_{i2})$ for $i = 1, \ldots, n$. This means that if the edge $e(x_i)$ is in a feasible spanning tree then neither edge $\bar{e}_{i1}$ nor $\bar{e}_{i2}$ can be in the tree. This is the central point of our reduction, because the correspondence between the instance I and the instance of MSTCG defined by these graphs is the following: If variable x_i is set to $TRUE$ in I then $e(x_i)$ is in the spanning tree and if x_i is set to $FALSE$, then $e(\bar{x}_i)$ and vice versa.

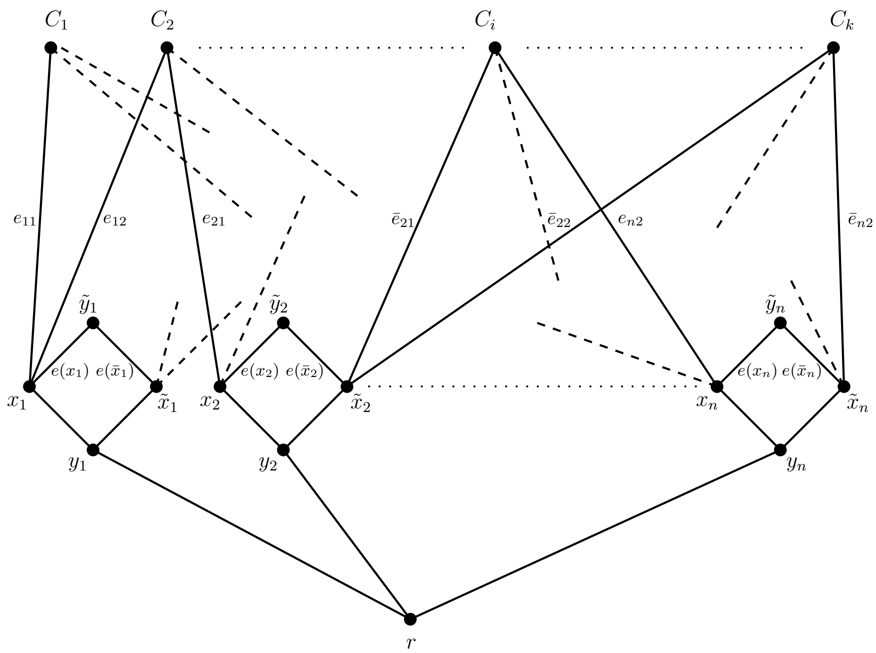

Figure 1: The graph G_{MSTCG}

2.2 MSTCG with a 3-ladder conflict graph is strongly $\mathcal{NP}$-hard

Theorem 1 *MSTCG is strongly $\mathcal{NP}$-hard, even if the conflict graph is a* 3*-ladder.*

Proof. Let I be an instance of *(3,B2)-SAT* and G_{MSTCG} and $\bar{G}_{MSTCG}$ the corresponding graphs constructed in Section 2.1. $MSTCG_I$ is then defined as instance of MSTCG described by the graph G_{MSTCG} and the conflict graph $\bar{G}_{MSTCG}$ with weight function $w = 0$. We prove the theorem by showing that the following holds:

$$I \text{ TRUE} \Longleftrightarrow \exists \text{ a spanning tree } T \text{ for } MSTCG_I$$

"$\Longleftarrow$": Let T be a spanning tree of $MSTCG_I$. By construction of the cycles CY_i in G_{MSTCG}, $e(x_i)$ or $e(\bar{x}_i)$ has to be in T to reach $\tilde{y}_i$. If $e(x_i)$ is in T, we set x_i in I to $TRUE$. In this case, by construction of the conflict graph $\bar{G}_{MSTCG}$, the edges $\bar{e}_{i1}$ and $\bar{e}_{i2}$ are blocked and any vertex (clause) C_j can be reached only via edges emanating from x_i and corresponding to a $TRUE$-assignment. Otherwise if $e(\bar{x}_i)$ is in T, we set x_i in I to $FALSE$ and get an analogous argument. But since T is a tree, every vertex C_j is reached and we get a satisfying truth assignment for I.

If both edges $e(x_i)$ and $e(\bar{x}_i)$ are in T, then there exists a tree T that reaches all clauses without using any of the edges e_{i1}, e_{i2}, $\bar{e}_{i1}$ and $\bar{e}_{i2}$. So there exists a satisfying truth assignment, where the setting of x_i is not relevant.

"$\Longrightarrow$": Given that there is an assignment A of the variables of I so that the instance is $TRUE$, we construct a spanning tree T of G_{MSTCG}. So let $T = \emptyset$ and $X = \{x_{l_1}, \ldots, x_{l_r}\}$ be the set of all variables in A set to $TRUE$ and $\bar{X} = \{\bar{x}_{k_1}, \ldots, \bar{x}_{k_s}\}$ the set of all variables set to $FALSE$.

$$T = T \cup (r, y_i) \ \forall i$$

$$T = T \cup e(x_{l_1}) \cup \ldots \cup e(x_{l_r}) \cup e(\bar{x}_{k_1}) \cup \ldots \cup e(\bar{x}_{k_s})$$

Mark all clauses C_j unmarked and let $C(x_i)$ and $C(\bar{x}_i)$ be the set of all clauses including x_i or $\bar{x}_i$, respectively.

for $l \in \{l_1 \ldots l_r\}$:

for $C_j \in C(x_l)$:

if C_j is unmarked and joined to CY_l by e_{lk}, $k \in \{1, 2\}$:

$T = T \cup e_{lk}$

Mark C_j as marked.

for $u \in \{k_1 \ldots k_s\}$:

for $C_j \in C(\bar{x}_u)$:

if C_j is unmarked and joined to CY_u by $\bar{e}_{uk}$, $k \in \{1, 2\}$:

$T = T \cup \bar{e}_{uk}$

Mark C_j as marked.

Since A is an assignment setting I to $TRUE$, clearly each clause C_j includes a literal set to $TRUE$. By the construction of T each vertex C_j is reached by exactly one edge corresponding to such a literal. This immediately yields the fact that T is a tree. $\square$

Corollary 2 *There cannot exist a polynomial time approximation algorithm for MSTCG, unless $\mathcal{P} = \mathcal{NP}$.*

Proof. In the proof of Theorem 1 even deciding if a spanning tree exists in the constructed instance was $\mathcal{NP}$-complete, leading to the desired result. $\square$

Since MSTCG is strongly $\mathcal{NP}$-hard given the conflict graph is a 3-ladder, MSTCG is also strongly $\mathcal{NP}$-hard in case the conflict graph is a path. Finally both results obviously imply that MSTCG is strongly $\mathcal{NP}$-hard for general conflict graphs.

Corollary 3 *Given the conflict graph is a path, MSTCG is strongly $\mathcal{NP}$-hard.*

Corollary 4 *MSTCG is strongly $\mathcal{NP}$-hard.*

3 MSTCG with disjoint conflicting pairs of edges is in $\mathcal{P}$

In this section we focus on MSTCG where the conflict graph is a 2-ladder, i.e. the conflict graph represents pairwise disjoint forbidden pairs of edges of E. More generally, we show that with the help of Edmonds' famous matroid-intersection theorem (Edmonds [Edm79], cf. [Sch03]) an optimal solution of MSTCG can be computed in polynomial time, whenever the conflict graph is a union of disjoint cliques. This result follows since (i) spanning trees correspond to bases of the graphic matroid [Sch03] and (ii) the conflict structure of the disjoint cliques is captured by a *partition matroid* (cf. [Jun04]).

Theorem 5 *(Edmonds [Edm79], cf. [Sch03])*
Let S be a set and let $c : S \to \mathbb{R}$. Given two matroids $M_1 = (S, I_1)$ and $M_2 = (S, I_2)$, a common base of M_1 and M_2 with minimum weight can be found in strongly polynomial time.

Since an optimal solution of MSTCG corresponds to a minimum-weight common base of the graphic matroid and the partition matroid, the above theorem yields the following result.

Theorem 6 *MSTCG with a conflict graph consisting of disjoint maximal cliques can be solved in strongly polynomial time.*

Corollary 7 *MSTCG with disjoint conflicting pairs of edges can be solved in strongly polynomial time.*

4 A strong $\mathcal{NP}$-hardness result for MSTFG

In this section we show that the separation between polynomially solvable and $\mathcal{NP}$-hard is the same for MST with conflict graph and with forcing graph. Complementing Theorem 1 we will show that MSTFG is strongly $\mathcal{NP}$-hard if the forcing graph $\bar{H}$ is a 3-ladder. Our reduction is based on the decision version of *MIN WEIGHTED 2-SAT-3* with unitary weight in which each variable occurs at most three times and every literal appears at most twice (i.e. every variable occurs at most twice negated and at most twice nonnegated). We will denote this problem as *2-SAT-3UB*. We refer to the number of variables set to $TRUE$ under satisfying truth assignment τ as the weight of τ.

The above special version of *2-SAT* can be shown to be APX-complete by checking the proof of the same result for *MIN WEIGHTED 2-SAT-3* with unitary weight derived by [AAGP97]. The $\mathcal{NP}$-completeness of the decision version is a straightforward consequence of the reduction.

4.1 The graphs G_{MSTFG} and $\bar{H}_{MSTFG}$

Let I be an instance of *2-SAT-3UB* with n variables $x_1, \dots, x_n$ and k clauses $C_1, \dots, C_k$, such that each literal appears at most twice and each positive literal is assigned weight one. We create an instance $MSTFG_I$ of MSTFG by building a graph G_{MSTFG} and a forcing graph $\bar{H}_{MSTFG}$ as follows.

To create $G_{MSTFG} = (V, E)$ (see Figure 2) we introduce for each variable x_i a triangle (i.e., a cycle of length 3) with a vertex α_i, two emanating edges x_i and $\bar{x}_i$ (corresponding to the literals x_i resp. $\bar{x}_i$) and an edge y_i opposite vertex α_i. All vertices α_i, $i = 1, \dots, n$, are connected to a dedicated vertex r.

For each clause C_j we introduce a vertex C_j and connect C_j to the vertex incident to y_i and x_i (resp. $\bar{x}_i$) by edge h_{ji} (resp. $\bar{h}_{ji}$) iff clause C_j contains literal x_i (resp. $\bar{x}_i$).

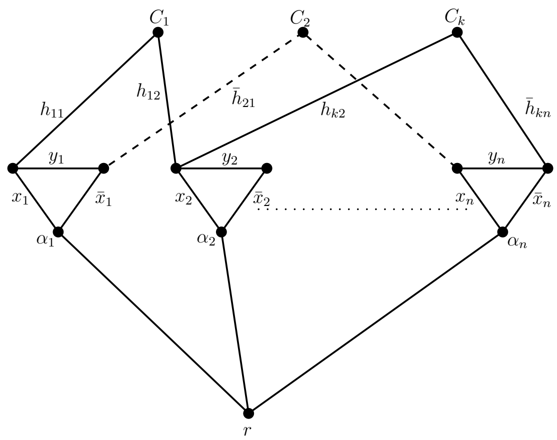

Figure 2: The graph G_{MSTFG}

We define the weight function $w : E \to \mathbb{N}_0$ by

$$w(e) := \begin{cases} 2n+1 & \text{if } e \in \{x_1, ..., x_n\} \\ 2n & \text{if } e \in \{\bar{x}_1, ..., \bar{x}_n\} \\ 4n^2 & \text{if } e \in \{h_{ji}, \bar{h}_{ji} \,|\, 1 \leq j \leq k,\ 1 \leq i \leq n\} \\ 0 & \text{otherwise.} \end{cases}$$

For every literal x_i contained in a clause C_j (causing an edge h_{ji}) we introduce in the forcing graph $\bar{H}_{MSTFG}$ an edge connecting x_i (as a vertex in $\bar{H}_{MSTFG}$) with the vertex representing the second edge $h_{j\ell}$ or $\bar{h}_{j\ell}$ emanating from C_j. Naturally, also x_ℓ (or $\bar{x}_\ell$) is joined by an edge with h_{ji}. The same construction is performed for every literal $\bar{x}_i$.

Note that the connected components of $\bar{H}_{MSTFT}$ are 3-ladders, 2-ladders and isolated vertices, since in I each literal occurs at most twice and each vertex C_j has degree two.

4.2 MSTFG with a 3-ladder forcing graph is strongly $\mathcal{NP}$-hard

Theorem 8 *Given the forcing graph is a 3-ladder, MSTFG is strongly $\mathcal{NP}$-hard.*

Proof. We show that the following holds for $L \leq n$:

$$\begin{gathered}\exists \text{ a satisfying truth assignment } \tau \text{ for } I \text{ with weight } \leq L \\ \Longleftrightarrow \\ \exists \text{ a spanning tree } T \text{ with } w(T) \leq 4kn^2 + 2n^2 + L\end{gathered}$$

"$\Rightarrow$": Let τ be a satisfying truth assignment for I with weight $\leq L$ and let X be the set of variables set $TRUE$ under τ. We construct a feasible spanning tree T of $MSTFG_I$ as follows: To the empty tree T add edge x_i if $x_i \in X$ and add $\bar{x}_i$ otherwise. If $x_i \in X$, then add edge h_{ji} for all vertices C_j adjacent to vertex x_i, unless an edge incident to C_j has been added already. Analogously, if $x_i \notin X$, then add edge $\bar{h}_{ji}$ for all vertices C_j adjacent to vertex $\bar{x}_i$, unless an edge incident to C_j has been added already. Finally, add the edges y_i and (α_i, r) to T for all $i = 1, \ldots, n$.

It is obvious that T is a spanning tree and it is clear by construction that the force restrictions imposed by $\bar{H}_{MSTFG}$ are satisfied by T. Since in the tree T, each vertex $C_1, ..., C_k$ is adjacent to exactly one edge we know that exactly k edges with weight $4n^2$ are contained in the tree. Now, since the weight of τ is not greater than L, at most L edges out of $\{x_1, ..., x_n\}$ and at least $n - L$ edges of $\{\bar{x}_1, ..., \bar{x}_n\}$ are contained in the tree. Since the remaining edges have zero weight, we get

$$\begin{aligned} w(T) &\leq k\,4n^2 + L(2n+1) + (n-L)2n \\ &= 4kn^2 + 2n^2 + L\,. \end{aligned}$$

"$\Leftarrow$": Let T be a feasible spanning tree of $MSTFG_I$ with $w(T) \leq 4kn^2+2n^2+L$ for some $L \leq n$. Thus, at most k edges of weight $4n^2$ are contained in T. Otherwise $w(T) \geq (k+1)4n^2 = 4kn^2+4n^2$ would hold which contradicts $L \leq n$. This implies that exactly k edges of weight $4n^2$ are contained in T to guarantee that all vertices C_j are connected. Hence, for each vertex (clause) C_j exactly one of the two emanating edges is contained in the tree T. Now the forcing graph $\bar{H}_{MSTFG}$ implies that for each vertex C_j at least one of the two edges that correspond to the literals constituting clause C_j in I must be contained in T, namely the one being in a forcing relation with the edge emanating from C_j but missing in T.

We next show that either x_i or $\bar{x}_i$ is contained in T, for all $i = 1, \ldots, n$. Since the vertices $C_1, ..., C_k$ are leaves of T, it immediately follows that at least one edge of $\{x_i, \bar{x}_i\}$ must be contained in T. Otherwise there would be no path from the endpoints of edges y_i to α_i. Assume that for some i both x_i and $\bar{x}_i$ are contained in T. Then we get $w(T) \geq 4kn^2 + 2n^2 + 2n + 1$ in contradiction to $L \leq n$.

To sum up, for each clause in I at least one of the edges that correspond to its literals is contained in T, and exactly one of $\{x_i, \bar{x}_i\}$ is in T for $i = 1, \ldots, n$. Therefore, the truth assignment τ defined by setting variable x_i $TRUE$ iff the edge x_i is contained in T is a satisfying truth assignment for I. Moreover, since T contains exactly k edges of weight $4n^2$ we can conclude that at most L edges of $\{x_1, ..., x_n\}$ are contained in T which proves that the weight of τ is not greater than L. □

5 MSTFG with disjoint conjunctive pairs of edges is in $\mathcal{P}$

Definition 3 *Let $G = (V, E)$ be an undirected and connected graph. The dual of the graphic matroid $M_1(G)$ is called the cographic matroid $M_1^*(G)$ of G.*

Lemma 9 *[Wel76] Let $X \subset E$. X is a base of the cographic matroid $M_1^*(G)$ $\Leftrightarrow$ $E \setminus X$ is a spanning tree of G.*

Let a 2-ladder be given as forcing graph with disjoint conjunctive pairs $x_{i_k} + x_{j_k} \geq 1$ with $k \in \{1, ..., \ell\}$ for some ℓ. Let T be a spanning tree of G. It is easy to see that T satisfies the conjunctive constraints $x_{i_k} + x_{j_k} \geq 1$ if and only if $E \setminus T$ satisfies the disjunctive constraints $x_{i_k} + x_{j_k} \leq 1$. Since $w(E \setminus T) = w(E) - w(T)$ it obviously follows that there is a one-to-one correspondence between minimum spanning trees satisfying the disjoint constraints $x_{i_k} + x_{j_k} \geq 1$ and maximum weight bases of the cographic matroid $M_1^*(G)$ satisfying the disjoint constraints $x_{i_k} + x_{j_k} \leq 1$. Thus solving MSTFG with disjoint conjunctive pairs corresponds to finding a maximum weight common base of the matroid $M_1^*(G)$ and the partition matroid restricted to a 2-ladder. Since a maximum weight common base of two matroids can be found in strongly polynomial time [Sch03], we get the following result.

Theorem 10 *MSTFG with a forcing graph consisting of disjoint conjunctive pairs, i.e. a 2-ladder, can be solved in strongly polynomial time.*

6 MMCG with disjoint conflicting pairs of edges is strongly $\mathcal{NP}$-hard

In this section we show that MMCG is strongly $\mathcal{NP}$-hard even if the conflict graph $\bar{G}$ is a 2-ladder, even for the unweighted case.

6.1 The graphs G_{MMCG} and $\bar{G}_{MMCG}$

We again use the problem *(3,B2)-SAT* for our reduction. The special instances of MMCG are described by a graph G_{MMCG} in which a maximum matching has to be found given a conflict graph $\bar{G}_{MMCG}$. Let I be an arbitrary instance of *(3,B2)-SAT* with k clauses C_j and n variables x_i. We define the graph G_{MMCG} in the following way (see Figure 3):

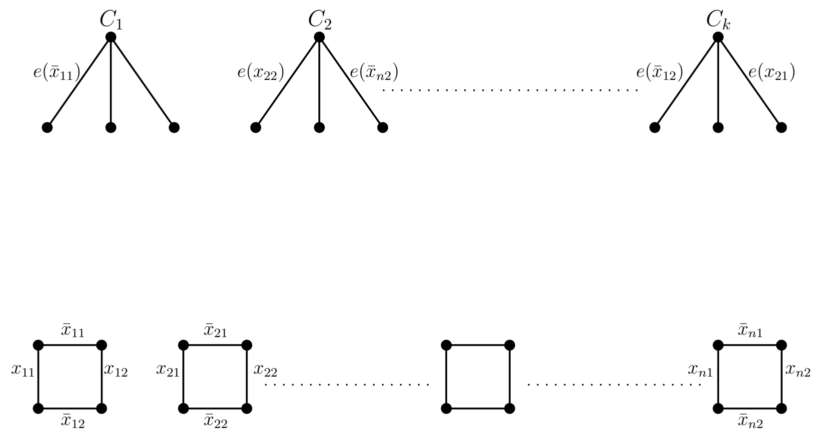

Figure 3: The graph G_{MMCG}

For each variable x_i we introduce a cycle of length four (CY_i) consisting of edges x_{i1}, x_{i2}, $\bar{x}_{i1}$ and $\bar{x}_{i2}$ such that x_{i1} and x_{i2} are not adjacent. These cycles are isolated components of G_{MMCG}. Moreover, we introduce for each clause C_j of I an isolated claw rooted at a vertex C_j with the following three edges: If the literal x_i occurs in clause C_j we denote one edge incident to vertex C_j by $e(x_{i1})$ or by $e(x_{i2})$ if the name $e(x_{i1})$ was already used. If the negated literal $\bar{x}_i$ occurs in clause C_j we denote one edge incident to vertex C_j by $e(\bar{x}_{i1})$ or by $e(\bar{x}_{i2})$ if the name $e(\bar{x}_{i1})$ was already used.

The conflict graph $\bar{G}_{MMCG}$ (a 2-ladder) on the edges of G_{MMCG} is defined by the isolated edges $(x_{i1}, e(\bar{x}_{i1}))$, $(x_{i2}, e(\bar{x}_{i2}))$, $(\bar{x}_{i1}, e(x_{i1}))$ and $(\bar{x}_{i2}, e(x_{i2}))$. The main idea of our reduction lies in the fact that a matching can take at most one edge for each claw induced by some C_j which then blocks an edge in the corresponding cycle CY_i by means of the 2-ladder. Clearly, a matching can contain at most two edges in a cycle of length four. By construction of CY_i the only possibility for choosing two matching edges is to take x_{11} and x_{12} or to take $\bar{x}_{11}$ and $\bar{x}_{12}$.

6.2 MMCG with a 2-ladder conflict graph is strongly $\mathcal{NP}$-hard

Theorem 11 *Given an instance I of (3,B2)-SAT the following holds for the corresponding instance $MMCG_I$ of MMCG constructed in the above way:*

$$I \ TRUE \Longleftrightarrow \exists \ \text{ a matching } M \text{ for } MSTCG_I \text{ of cardinality } k+2n$$

Proof.
"$\Longleftarrow$": Let M be a feasible matching of $MMCG_I$ of cardinality k+2n. Then by construction of G_{MMCG} M must contain exactly one edge of each claw corresponding to C_j and exactly two edges in each cycle CY_i. If these edges are x_{i1} and x_{i2}, set x_i in I to $TRUE$ else to $FALSE$. By the construction of $\bar{G}_{MMCG}$ and the fact that one edge is taken in each claw, it immediately follows that I is a feasible $TRUE$ instance.

"$\Longrightarrow$": Given a $TRUE$ assignment A of I we construct a matching M of G_{MSTCG}. Let $X = \{x_{l_1}, \ldots, x_{l_r}\}$ be the set of all variables in A set to $TRUE$ and $\bar{X} = \{\bar{x}_{k_1}, \ldots, \bar{x}_{k_s}\}$ the set of all variables set to $FALSE$. We start with a matching of size $2n$:

$$M = x_{l_1 1} \cup x_{l_1 2} \cup \ldots \cup x_{l_r 1} \cup x_{l_r 2} \ \cup \ \bar{x}_{k_1 1} \cup \bar{x}_{k_1 2} \cup \ldots \cup \bar{x}_{k_s 1} \cup \bar{x}_{k_s 2}$$

Mark all clauses C_j unmarked and let $C(x_i)$ and $C(\bar{x}_i)$ be the set of all clauses including x_i or $\bar{x}_i$ respectively.

for $l \in \{l_1 \ldots l_r\}$:
  for $C_j \in C(x_l)$:
    if C_j is unmarked and includes $e(x_{lk})$, $k \in \{1, 2\}$:
      $M = M \cup e(x_{lk})$
      Mark C_j as marked.
for $u \in \{k_1 \ldots k_s\}$:
  for $C_j \in C(\bar{x}_u)$:
    if C_j is unmarked and includes $e(\bar{x}_{uk})$, $k \in \{1, 2\}$:
      $M = M \cup e(\bar{x}_{uk})$
      Mark C_j as marked.

Since A is an assignment setting I to $TRUE$, clearly each clause C_j includes a literal set to $TRUE$. By the construction each claw corresponding to C_j adds exactly one edge to M. This immediately yields that M is in fact a matching of size $k + 2n$. □

Note that for $MMCG$ with a 2-ladder there exists a trivial $\frac{1}{2}$–approximation algorithm by computing a maximum weight matching MM in the graph G without considering the conflicts and then removing the edge with lighter weight of each conflicting pair in the solution.

7 MMFG with disjoint conjunctive pairs of edges is strongly $\mathcal{NP}$-hard

For showing that MMFG is strongly $\mathcal{NP}$-hard with a 2-ladder as forcing graph, we use again the problem *2-SAT-3UB*. (cf. Section 4).

We define the graph G_{MMFG} in the following way (see Figure 4) which resembles the graph used in Section 6: For each variable x_i we introduce a cycle of length four (CY_i) that is build exactly like in Section 6. But now for each clause C_j of I we introduce an isolated path of length two with the vertex of degree two called C_j. We name the two edges of these paths in the following way: If the literal x_i occurs in clause C_j we denote one edge incident to vertex C_j by h_{i1}^j or by h_{i2}^j if the name h_{i1}^ℓ was already used for some index ℓ. If the negated literal $\bar{x}_i$ occurs in clause C_j we denote one edge incident to vertex C_j by $\bar{h}_{i1}^j$ or by $\bar{h}_{i2}^j$ if the name $\bar{h}_{i1}^\ell$ was already used for some index ℓ. Clearly in this graph, by the property of I, only one of the two edges h_{i2}^j and $\bar{h}_{i2}^l$ can exist.

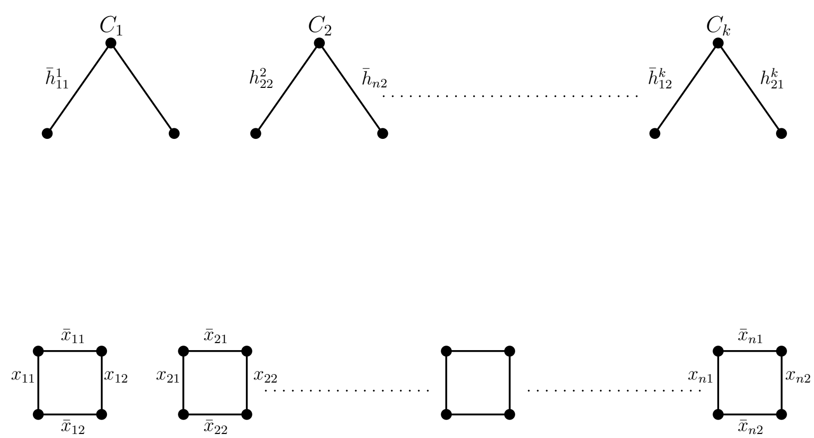

Figure 4: The graph G_{MMFG}

Extending the construction of Section 6, we will now use weights, chosen in the following way:

$$w(e) := \begin{cases} n^4 & \text{if } e \in \{x_{i1}, x_{i2} \mid i = 1, \dots, n\} \\ n^4 + \frac{1}{2} & \text{if } e \in \{\bar{x}_{i1}, \bar{x}_{i2} \mid i = 1, \dots, n\} \\ n^2 & \text{if } e \in \{h_{i1}^j, h_{i2}^j, \bar{h}_{i1}^j, \bar{h}_{i2}^j \mid \forall\, i, j\} \end{cases}$$

If a literal x_i appears in a clause C_j inducing some edge $h_{i\ell}^j$ in G_{MMFG}, in the forcing graph $\bar{H}_{MMFG}$ we join $x_{i\ell}$ with the other edge emanating from C_j. The same is done for a literal $\bar{x}_i$ and its corresponding edges $\bar{x}_{i1}$ and $\bar{x}_{i2}$ (if they exist). By the structure of I and the construction of G_{MMFG}, $\bar{H}_{MMFG}$ is a 2-ladder.

Theorem 12 *Given the forcing graph is a 2-ladder, MMFG is strongly $\mathcal{NP}$-hard.*

Proof. We show that the following holds for $L \leq n$:

$$\exists \text{ a satisfying truth assignment } \tau \text{ for } I \text{ with weight } \leq L$$
$$\Longleftrightarrow$$
$$\exists \text{ a matching } M \text{ with } w(M) \geq kn^2 + 2n^5 + (n - L)$$

"$\Rightarrow$": Let τ be a satisfying truth assignment for I with weight $\leq L$ and let X be the set of variables set $TRUE$ under τ. Create a feasible solution M of $MMFG_I$ resulting from the above described construction as follows: If $x_i \in X$ then add edges x_{i1} and x_{i2} to the matching M, otherwise add $\bar{x}_{i1}$ and $\bar{x}_{i2}$. For each clause C_j add the edge $h^j_{i\ell}$ (resp. $\bar{h}^j_{i\ell}$) if $x_{i\ell}$ (resp. $\bar{x}_{i\ell}$) was added to M and no edge incident to C_j was added before. With this we we get the following weight for M:

$$w(M) = kn^2 + 2n\,n^4 + (n - L)$$

"$\Leftarrow$": Let M be a feasible matching of $MMFG_I$ with $w(M) \geq kn^2+2n^5+(n-L)$. By the definition of the weights, this implies that in each cycle CY_i exactly two edges were taken and in each path of length two corresponding to C_j exactly one edge was taken. Moreover, in the cycles CY_i at least $(n - L)$ times the two edges $\bar{x}_{i1}$ and $\bar{x}_{i2}$ were taken. So let τ be a truth assignment that results from setting x_i to $TRUE$ if x_{i1} and x_{i2} are in M and to $FALSE$ otherwise. Clearly the weight of τ is at most L. To show that τ is feasible for instance I, assume that there is a clause C_j such that neither of the two literals in C_j are set to $TRUE$ in τ. By the construction of $\bar{H}_{MMFG}$ this implies that both edges adjacent to C_j are in M, contradicting the matching property. $\square$

8 Shortest Path under Disjunctive Constraints

Because of the following result for negative disjunctions will only consider the shortest path problem with disjoint conjunctive pairs of edges in this section:

Theorem 13 *[Kan94] The shortest path problem with forbidden pairs is NPO PB-complete even in the unweighted case.*

For the shortest path problem with disjoint conjunctive pairs of edges we will derive an APX-hardness result by a reduction from *2-SAT-3UB*. We define the graph G_{SPFG} corresponding to an instance I of *2-SAT-3UB* (see Figure 5): For each variable x_i we introduce a cycle CY_i of length five with edges starting at a vertex x_i in clockwise direction $(x_{i1}, d_i, x_{i2}, \bar{x}_{i2}, \bar{x}_{i1})$. Each vertex x_{i+1} (resp. C_1 for $i = n$) is incident to the edges x_{i2} and $\bar{x}_{i2}$ and thus also contained in CY_i. For each clause C_j of I we introduce a cycle CY^j of length four with one vertex C_j. This vertex is connected to its opposite C_{j+1} (resp. t if $j = k$) by two paths consisting of $h^j_{i\ell}$ or $\bar{h}^j_{i\ell}$, $\ell \in \{1, 2\}$, and another dummy edge where the edges $h^j_{i\ell}$ and $\bar{h}^j_{i\ell}$ are defined in exactly the same way as in Section 7.

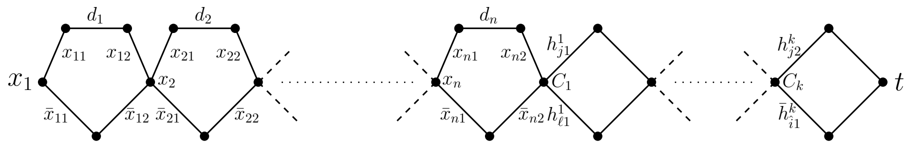

Figure 5: The graph G_{SPFG}

If a literal x_i appears in a clause C_j inducing some edge $h^j_{i\ell}$ in G_{SPFG}, in the forcing graph $\bar{H}_{SPFG}$ we join $x_{i\ell}$ with the other edge emanating from C_j in CY^j. The same is done for a literal $\bar{x}_i$ and its corresponding edges $\bar{x}_{i1}$ and $\bar{x}_{i2}$ (if they exist). By the structure of I and the construction of G_{SPFG}, $\bar{H}_{SPFG}$ is a 2-ladder.

Theorem 14 *Given the forcing graph is a 2-ladder, SPFG is strongly $\mathcal{NP}$-hard even in the unweighted case.*

Proof. We show that the following holds for $W \leq n$:

$$\exists \text{ a satisfying truth assignment } \tau \text{ for } I \text{ with weight } W$$
$$\Longleftrightarrow$$
$$\exists \text{ a path } P \text{ between } x_1 \text{ and } t \text{ with length } l(P) = W + 2n + 2k$$

"$\Rightarrow$": Let τ be a satisfying truth assignment for I with weight $W \leq n$ and let X be the set of variables set $TRUE$ under τ. Create a feasible solution P of the instance $SPFG_I$ resulting from the above described construction as follows: If $x_i \in X$ then add edges x_{i1}, d_i and x_{i2} to P, otherwise add $\bar{x}_{i1}$ and $\bar{x}_{i2}$. For each clause C_j add the edge $h^j_{i\ell}$ (resp. $\bar{h}^j_{i\ell}$) if $x_{i\ell}$ (resp. $\bar{x}_{i\ell}$) was added to P and no edge incident to C_j in the cycle CY^j was added before. Complete the path by adding the missing dummy edges. With this we get that the length of P equals $W + 2n + 2k$. Obviously, P fulfills the positive disjunctive constraints imposed by $\bar{H}_{SPFG}$.

"$\Leftarrow$": Let P be a path of $SPFG_I$ with $l(P) = W + 2n + 2k$ ($W \leq n$). This immediately implies that the edges d_i occur exactly W times in P since every path from x_1 to t will require at least two edges for each of the $n+k$ cycles. Now let τ be a truth assignment that results from setting x_i to $TRUE$ if d_i is in P and to $FALSE$ otherwise. Clearly the weight of τ equals W. To show that τ is feasible for instance I, assume that there is a clause C_j such that neither of the two literals in C_j are set to $TRUE$ in τ. B y the construction of $\bar{H}_{SPFG}$ this implies that both edges incident to the vertex C_j in CY^j are in P, contradicting the assumption that P is a path. $\square$

This reduction can easily be extended to an AP-reduction (cf. [APMS$^+$99]) by introducing instead of the edges d_i in G_{SPFG} a path of length $2n+2k$ between x_{i1} and x_{i2}. With this we get that a solution of I with weight W corresponds to a solution of $SPFG_I$ with length $(W+1)(2n+2k)$.

Theorem 15 *Given the forcing graph is a 2-ladder, SPFG is APX-hard even in the unweighted case.*

References

[AAGP97] P. Alimonti, G. Ausiello, L. Giovaniello, and M. Protasi. On the complexity of approximating weighted satisfiability problems. Rapporto Tecnico RAP 38.97, Dipartimento di Informatica e Sistemistica, Università degli Studi di Roma La Sapienza, 1997.

[APMS+99] G. Ausiello, M. Protasi, A. Marchetti-Spaccamela, G. Gambosi, P. Crescenzi, and V. Kann. *Complexity and Approximation: Combinatorial Optimization Problems and Their Approximability Properties*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 1999.

[BJ93] H.L. Bodlaender and K. Jansen. On the complexity of scheduling incompatible jobs with unit-times. In *MFCS '93: Proceedings of the 18th International Symposium on Mathematical Foundations of Computer Science*, pages 291–300, London, UK, 1993. Springer.

[BKS03] P. Berman, M. Karpinski, and A.D. Scott. Approximation hardness of short symmetric instances of max-3sat. *Electronic Colloquium on Computational Complexity (ECCC)*, (049), 2003.

[Edm79] J. Edmonds. Matroid intersection. *Annals of Discrete Mathematics*, 4:39–49, 1979.

[Jan99] K. Jansen. An approximation scheme for bin packing with conflicts. *Journal of Combinatorial Optimization*, 3:363–377, 1999.

[JO97] K. Jansen and S. Öhring. Approximation algorithms for time constrained scheduling. *Information and Computation*, 132(2):85–108, 1997.

[Jun04] D. Jungnickel. *Graphs, Networks and Algorithms (Algorithms and Computation in Mathematics)*. Springer, 2004.

[Kan94] V. Kann. Polynomially bounded minimization problems that are hard to approximate. *Nordic Journal of Computing*, 1(3):317–331, 1994.

[PS09] U. Pferschy and J. Schauer. The knapsack problem with conflict graphs. *Journal of Graph Algorithms and Applications*, to appear 2009.

[Sch03] A. Schrijver. *Combinatorial Optimization, Polyhedra and efficiency*, volume B. Springer, 2003.

[Wel76] D.J. Welsh. *Matroid theory*. Academic Press, 1976.