Approximating Infeasible 2VPI-Systems

Neele Leithäuser[1], Sven O. Krumke[2], and Maximilian Merkert[2]

[1] ITWM Fraunhofer Institut für Techno- und Wirtschaftsmathematik, Fraunhofer-Platz 1, 67663 Kaiserslautern, Germany
neele.leithaeuser@itwm.fraunhofer.de

[2] Dept. of Mathematics, University of Kaiserslautern, Paul-Ehrlich-Str. 14, 67663 Kaiserslautern, Germany
{krumke,merkert}@mathematik.uni-kl.de

Abstract. It is a folklore result that testing whether a given system of equations with two variables per inequality (a 2VPI system) of the form $x_i - x_j = c_{ij}$ is solvable, can be done efficiently not only by Gaussian elimination but also by shortest-path computation on an associated constraint graph. However, when the system is infeasible and one wishes to delete a minimum weight set of inequalities to obtain feasibility (MinFs2$^=$), this task becomes NP-complete.

Our main result is a 2-approximation for the problem MinFs2$^=$ for the case when the constraint graph is planar using a primal-dual approach. We also give an α-approximation for the related maximization problem MaxFs2$^=$ where the goal is to maximize the weight of feasible inequalities. Here, α denotes the arboricity of the constraint graph. Our results extend to obtain constant factor approximations for the case when the domains of the variables are further restricted.

1 Introduction

The problem of checking whether a system of linear (in-)equalities admits a solution is efficiently solvable by means of linear optimization methods. However, in many applications it is known that a linear system is unsolvable. An instance $I = (X, C)$ of the maximum feasible subsystem problem (MaxFs) consists of a finite set X of variables and a set C of constraints on the variables, and the goal is to select a subset $C' \subseteq C$ of the constraints of maximum size such that the corresponding reduced system (X, C') is feasible. This problem has a wide range of applications (see e.g. [2] and the references therein) and is well-known to be NP-complete. The corresponding minimization problem MinFs asks to delete a minimum number of constraints in order to obtain a feasible system.

In this paper we consider the versions of MinFs and MaxFs where the constraints are restricted to the special form $x_i - x_j = c_{ij}$, in particular there are only two variables per inequality. Such a system is commonly referred to as a 2VPI-system. We are also given a nonnegative weight for each constraint, specifying the cost of removing it from the system. The problems MinFs2$^=$ and MaxFs2$^=$ of deleting a minimum cost set of constraints or retaining a maximum cost set of constraints, respectively, are still NP-hard to solve, even in case of unit costs for the constraints.

M.C. Golumbic et al. (Eds.): WG 2012, LNCS 7551, pp. 225–236, 2012.

With each instance $I = (X, C)$ of a 2VPI-system, one can associate a corresponding *constraint graph* G_I as follows. For each variable $x_i \in X$ there is a vertex i in G_I and for each constraint of the form $x_i - x_j = c_{ij}$ there is a directed arc a from i to j of length $\ell(a) := c_{ij}$. This arc may be traversed in both directions, where the length for traversing in direction from j to i is given by $-c_{ij}$. It is then easy to see that the system given in $I = (X, C)$ is feasible if and only if G_I does not contain a negative length (undirected) cycle. Thus, the consistency of such a system can be tested by an all-pairs shortest path computation in $O(mn)$ time, where $n = |X|$ and $m = |C|$ denote the number of variables and constraints, respectively.

This graph-theoretic view on a 2VPI-system which we are going to take in this paper means that the problem MinFs2$^=$ is equivalent to the following problem: Delete a minimum cost set of the arcs of a given graph such that the resulting graph does not contain any negative length cycle.

MaxFs and MinFs have been studied extensively in terms of complexity and approximability, see, e.g., [2,10,4,13] and the references therein, and also many heuristic algorithms have been proposed (e.g., [3,1]), but MaxFs2$^=$ and MinFs2$^=$ have not been studied specifically so far in literature.

We present the first constant factor approximation algorithm for MaxFs2$^=$ when restricted to planar graphs. We also give an exact pseudo-polynomial time algorithm for series-parallel graphs and a polynomial time algorithm for extension-parallel graphs. Moreover we derive new hardness results for generalized versions of the problems, where the variables have values in some domain other than $\mathbb{R}$ or the operations are replaced by the group operations in some group.

Our paper is organized as follows: Section 3 contains our new hardness results which among other things show that MinFs2$^=$ and MaxFs2$^=$ are still NP-hard to solve, even if the corresponding constraint graph is planar. In Section 4 we give a factor 2-approximation for MinFs2$^=$ on planar graphs. Our algorithm is based on the primal-dual framework by Goemans and Williamson [8] and their work on feedback vertex problems in [9]. In fact, although we apply a very similar technique as Goemans and Williamson do in [9] for minimum feedback vertex problems in planar graphs, we obtain an improved performance guarantee of 2 compared to 3 in [9] (their best result is a factor of $\frac{9}{4}$).[1] In Section 5 we obtain approximation algorithms for the maximization version MaxFs2$^=$. In Section 6 we extend our results to the case of additional restrictions on the domains of the variables.

2 Preliminaries and Problem Definition

An instance I of MaxFs2$^=$ (and MinFs2$^=$) is given by a finite set X of variables and a finite set $\mathcal{E}$ of equations of the form $x_i - x_j = c_{ij}$ over the variables in X. Also, for each equation $e \in \mathcal{E}$ we are given a nonnegative weight w_e.

[1] We stress that, unfortunately, our results do not imply improved approximation results for the feedback vertex problem.

We call a subset $\mathcal{E}' \subseteq \mathcal{E}$ of the equations *consistent* if the included equations can be satisfied simultaneously, otherwise $\mathcal{E}'$ is termed *inconsistent*. The goal in MaxFs2$^=$ is to find a consistent subset $\mathcal{E}'$ of the equations of maximum weight $w(\mathcal{E}') = \sum_{e \in \mathcal{E}'} w_e$. Similarly, MinFs2$^=$ asks to remove a minimum weight subset of the equations to obtain a consistent system.

With each instance I of MaxFs2$^=$ (and MinFs2$^=$), the associated *constraint graph* G_I contains a vertex i for each variable $x_i \in X$ and for each constraint $x_i - x_j = c_{ij}$ a directed arc a from i to j of length $\ell(a) := c_{ij}$. As mentioned before, this arc may be traversed in both directions, where the length for traversing in direction from j to i is given by $-c_{ij}$. In all what follows we identify a constraint e with the corresponding arc in G_I. As a consequence, in addition to its length, any arc e has an associated weight $w_e \geq 0$.

Since we allow traversing arcs against their directions at the corresponding negative length, G_I contains a negative length (undirected) cycle if and only if G_I contains a cycle of positive length. Thus, we call any (undirected) cycle in G_I of nonzero length an *inconsistent cycle*. By the observations made in the introduction, a subset of the equations is consistent if and only if the corresponding subgraph of G_I does not contain an inconsistent cycle.

Thus, one can state MaxFs2$^=$ (MinFs2$^=$) equivalently in graph theoretic terms as follows: Given a directed graph $G = (V, E)$ with lengths and weights on the arcs, find a maximum weight subset of the arcs (delete a minimum weight subset of the arcs) such that the resulting subgraph does not contain any inconsistent cycle. We call such a subgraph a *consistent subgraph*. In the sequel we assume that all graphs are (weakly) connected, since otherwise we can consider the problem on each connected component separately.

3 Complexity of MinFs2$^=$ and MaxFs2$^=$

Although MaxFs2$^=$ and MinFs2$^=$ were defined in such a way that each equality contains exactly two variables, in the following we will nevertheless consider equations with only one variable, since this apparently more general case can easily be reduced to MaxFs2$^=$ and MinFs2$^=$, respectively: Introduce a new variable x_0 and add it with appropriate sign to every equation that has only one variable, e.g., an equation $x_i = c$ is replaced by $x_i - x_0 = c$. This problem has an optimal solution x^* with $x_0^* = 0$: Add $-\hat{x}_0$ to every component of an arbitrary given solution $\hat{x}$ to obtain a new solution with $x_0 = 0$ and equally many (exactly the same) satisfied equations.

Theorem 1. MaxFs2$^=$ *is APX-hard.*

Proof. We provide an L-reduction from the APX-complete optimization problem Max-2-Sat: Given an instance I_1 of Max-2-Sat with variables x_i, $i = 1, \ldots, n$ and clauses $y_{c_{j1}} \vee y_{c_{j2}}, j = 1, \ldots, m$ with exactly two literals, $y_{c_{jl}} \in \{x_{c_{jl}}, \overline{x}_{c_{jl}}\}$, construct an instance I_2 of MaxFs2$^=$ as follows: For every clause $y_{c_{j1}} \vee y_{c_{j2}}$, $j \in \{1, \ldots, m\}$ create the following 14 equations:

$$y_{c_{j1}} - \overline{y}_{c_{j2}} = 0$$
$$y_{c_{j1}} - \overline{y}_{c_{j2}} = 1$$
$$(\text{where } \overline{\overline{x}}_{c_{jl}} := x_{c_{jl}})$$
$$x_{c_{j1}} = 1, \quad x_{c_{j2}} = 1, \quad \overline{x}_{c_{j1}} = 1, \quad \overline{x}_{c_{j2}} = 1$$
$$x_{c_{j1}} = 0, \quad x_{c_{j2}} = 0, \quad \overline{x}_{c_{j1}} = 0, \quad \overline{x}_{c_{j2}} = 0$$
$$x_{c_{j1}} - \overline{x}_{c_{j1}} = 1, \quad x_{c_{j2}} - \overline{x}_{c_{j2}} = 1$$
$$x_{c_{j1}} - \overline{x}_{c_{j1}} = -1, \quad x_{c_{j2}} - \overline{x}_{c_{j2}} = -1$$

This transformation is not cost preserving, but it is not too hard to verify that it is in fact an L-reduction: There exists a solution x for the instance I_2 of MaxFs2$^=$ which satisfies k_x equations if and only if there exists a solution x^{SAT} for I_1 that satisfies at least $k_{x^{SAT}} = k_x - 6m$ clauses. Moreover, $\mathrm{OPT}(I_2) \leq 7m = 14 \cdot \frac{m}{2} \leq 14 \cdot \mathrm{OPT}(I_1)$.

Note that for the definition of MaxFs2$^=$ we do not need to have a multiplication on the domains and also no algorithm presented in this paper relies on the fact that there is a multiplication on $\mathbb{R}$. In fact a group is perfectly sufficient. For these algebraic versions we have a similar complexity result, which essentially uses the same construction as in Theorem 1 if the group has an element of order greater than 2 and otherwise uses a reduction from MaxCut:

Theorem 2. *For every nontrivial group (G,+),* MaxFs2$^=$*, where all variables must attain values in G, is APX-hard.* □

A simple greedy method shows that furthermore MaxFs2$^=$ can be approximated within the constant factor $|G|$ for any finite group. The algorithm iteratively chooses a variable x_i. Let U denote the equations which are (currently) unary with respect to x_i, i.e., which currently contain no variable other than x_i. It then assigns a value to x_i which maximizes the number of satisfied equations in U (if U is empty, an arbitrary value is assigned to x_i) and updates the remaining equations by substituting the chosen value for x_i. It is easy to see that this provides a $|G|$-approximation.

By a reduction from Partition we can prove the following result:

Theorem 3. MaxFs2$^=$ *is NP-hard, even if the constraint graph is series-parallel.* □

Theorem 4. MaxFs2$^=$ *is strongly NP-hard, even if all weights are unit weights and the constraint graph is planar.*

Proof. Due to lack of space, we only sketch the proof. We provide a reduction from the strongly NP-complete problem Rectilinear Steiner Tree Problem mrst ([6, ND13]). We are given a set $P \subset \mathbb{Z} \times \mathbb{Z}$ of points in the plane and an integer k. The task is to decide whether there is a finite set $Q \subset \mathbb{Z} \times \mathbb{Z}$ such that there is a spanning tree of total weight k or less for the vertex set $V \cup Q$, where the weight of an edge $\{(x_1, y_1), (x_2, y_2)\}$ is measured with respect to the rectilinear metric $|x_1 - x_2| + |y_1 - y_2|$.

As common, we call the points in P terminals. Given an arbitrary instance I_1 from MRST with a terminal set $P = \{v_1, \ldots, v_n\} \subset \mathbb{Z} \times \mathbb{Z}$ where $v_i = (x_i, y_i)$, we compute

$$x_{\min} := \min_{i=1,\ldots,n} x_i \qquad y_{\min} := \min_{i=1,\ldots,n} y_i$$
$$x_{\max} := \max_{i=1,\ldots,n} x_i \qquad y_{\max} := \max_{i=1,\ldots,n} y_i$$

and set up a grid-graph $G^* = (V, E^*)$ with nodes $V = \{v_{x,y} : x \in \mathbb{Z} \cap [x_{\min}, x_{\max}], y \in \mathbb{Z} \cap [y_{\min}, y_{\max}]\}$. The edges connect vertices that are horizontally or vertically adjacent, that is, $E^* = \{e = [(x_1, y_1), (x_2, y_2)] : |x_1 - x_2| = 1, |y_1 - y_2| = 1\}$. Since the original reduction in [6] constructs only terminal points within a grid of maximal extension in $O(n^3)$, we can assume that instance I_1 also induces a grid which is polynomially bounded in n and hence only polynomially many nodes have to be introduced in G^*.

We now construct an instance I_2 of MinFs2$^=$ on the plane dual graph $G = (G^*)^*$ of G^* as follows: In G the vertices in G^* correspond to faces and the edges incident to a vertex in G^* form the boundary of the corresponding face in G. Assign a value to every node of G^* as follows:

$$c(v_i) := \begin{cases} i, & \text{if } i \in \{1, \ldots, n-1\} \\ -\frac{1}{2}n(n-1), & \text{if } i = n \\ 0, & \text{otherwise.} \end{cases}$$

It can then be shown that we can always assign lengths to the edges of G such that the clockwise length of the face corresponding to vertex v_i equals $c(v_i)$.

The graph G corresponds in fact to an instance of MinFs2$^=$. Observe that for no strict subset of the terminals their (face) values add up to zero. It can now be seen that the feasible solutions for the instance I_2 of MinFs2$^=$ correspond exactly to Steiner trees for I_1 and that I_2 has a consistent subset of at least $m - k$ edges if and only if for I_1, there is a Steiner Tree with at most k edges.

4 Approximation Algorithm for MinFs2$^=$ on Planar Graphs

The problem MinFs2$^=$ is a special case of the well-known Hitting Set Problem. In the general Hitting Set Problem, we are given a collection $\mathcal{C}$ of subsets of a finite set E and weights $w_e \geq 0$ for all $e \in E$. The goal is to find a minimum weight subset $A \subset E$ such that A contains at least one element from each subset in $\mathcal{C}$. In the case of MinFs2$^=$, the collection $\mathcal{C}$ is the set of all inconsistent cycles and the finite set of ground elements is the set of all arcs in the constraint graph with their respective weights.

Goemans and Williamson [8] gave a general primal-dual framework for designing approximation algorithms for Hitting Set. The problem can be formulated as an Integer Linear Program whose linear relaxation and dual read as follows:

$$\begin{aligned} \min \sum_{e\in E} w_e x_e & & & & \max \sum_{C\in\mathcal{C}} y_C & & & \\ s.t. \sum_{e\in C} x_e & \geq 1 & \forall C \in \mathcal{C} & & s.t. \sum_{C\in\mathcal{C}: e\in C} y_C & \leq w_e & \forall e \in E \\ x_e & \geq 0 & \forall e \in E & & y_C & \geq 0 & \forall C \in \mathcal{C} \end{aligned}$$

The primal-dual-method constructs simultaneously a feasible solution x for the Integer Linear Program and a feasible solution y for the dual of the LP-relaxation. The algorithm starts with the trivial dual feasible solution $y = 0$ and uses a violation oracle VIOLATION, which outputs for a given subset S of the ground set E a subset of the sets not hit by S. It then increases the dual variables of all the sets in VIOLATION(S) until one of the dual packing constraints becomes tight for some element, which then gets added to the solution. At the end, a cleanup step is performed. Algorithm 1 shows the translation of the general algorithm for MINFS2$^=$. We assume without loss of generality that $w_e > 0$ for all $e \in E$, since edges of zero weight can be removed from the graph at the beginning without adding to the solution cost.

A key theorem from Goemans and Williamson [8] is the following:

Theorem 5 ([8]). *The primal-dual algorithm 1 delivers a solution for* HITTING SET *of weight at most* $\gamma \sum_{C\in\mathcal{C}} y_C \leq \gamma OPT$ *if for any infeasible* $\overline{E}$ *and any minimal augmentation* F *of* $\overline{E}$*, the collection* $\mathcal{V}(\overline{E})$ *returned by the violation oracle on input* $\overline{E}$ *satisfies:*

$$\sum_{C\in\mathcal{V}(\overline{E})} |C \cap F| \leq \gamma |\mathcal{V}(\overline{E})|.$$

A *minimal augmentation* is a feasible solution F containing $\overline{E}$ which is inclusionwise minimal, that is, for any $e \in F$ it holds that $F \setminus \{e\}$ is infeasible.

In order to apply the result of Theorem 5 we need to design an appropriate violation oracle. The techniques will be similar to that in [9], where Goemans and Williamson construct a primal-dual 3-approximation for feedback problems on planar graphs. We will use the following intuitive definitions:

Definition 1 (cf. [9]). *Let G be a directed plane graph (i.e., G is planar and has been embedded in the plane, so it makes sense to talk about faces of G). A face of G whose boundary forms an inconsistent cycle is called a* inconsistent face. *Every cycle C of G divides the plane into two regions, the* interior *(the one with finite diameter) and the* exterior. *We define $f(C)$ to be the set of all faces in the interior of C. We say that a cycle C_1* contains *a cycle C_2 if $f(C_1) \supseteq f(C_2)$ and write $C_2 \subseteq_f C_1$.*

The relation "$\subseteq_f$" defines a partial order on the set $\mathcal{C}$ of inconsistent cycles of G. The inclusionwise minimal inconsistent cycles with respect to this partial order are of particular interest. We will abuse notation slightly and call them the minimal inconsistent cycles.

Two cycles C_1 and C_2 cross *if none of the sets $f(C_1) \cap f(C_2)$, $f(C_1) \setminus f(C_2)$, $f(C_2) \setminus f(C_1)$ is empty. A family of cycles is called* laminar *if no two of its cycles*

Algorithm 1 Primal-Dual Approximation Algorithm for MinFs2$^{=}$

Input: Graph $G = (V, E)$ with arc lengths and weights $w_e \geq 0$.
Output: An edge set $A \subseteq E$ such that the subgraph $(V, E \setminus A)$ does not contain any inconsistent cycles, i.e., an edge set A such that $C \cap A \neq \emptyset$ for each inconsistent cycle C.

$y = 0$ {dual solution}
$A = \emptyset$ {primal solution, initially empty}
Set $x_e = 0$ for all edges $e \in E$.
$l = 0$ {iteration count}
while there is an inconsistent cycle in $(V, E \setminus A)$ **do**
  $\mathcal{V} = \text{VIOLATION}(A)$
  Increase y_C uniformly for all $C \in \mathcal{V}$ until $\exists e_l \notin A : \sum_{C:e_l \in C} y_C = w_{e_l}$
  $A = A \cup \{e_l\}$
end while
for all $j = l, \ldots, 1$ **do**
  if $A \setminus e_l$ is feasible **then**
    $A = A \setminus \{e_l\}$
  end if
end for

are crossing, i.e., any two cycles either do not share an interior face or one of them contains the other.

The definition of our violation oracle is the first and also the most important point where we make use of the planarity of the constraint graph $G := G_I$: Define VIOLATION(A) to be the set of all minimal inconsistent cycles in the graph obtained by deleting all edges in A from G. As we will show, the minimal inconsistent cycles are all inconsistent faces and therefore it is clear that VIOLATION can be computed in polynomial time.

Choose an orientation for the plane and orient the boundaries of the faces of G accordingly. Summing up all (the lengths of) those boundaries we see that every edge contributes to exactly two summands and with different sign, so the result must be 0. Also, if we only sum up all the boundaries of the faces inside some cycle C, all inner boundaries cancel, leaving only the edges of C.

To put this into a formula, define $\ell(C)$ to be the length of the cycle C (=sum of the signed lengths of its edges) with clockwise orientation. If C is the boundary of a face F, we also define $\ell(F) := \ell(C)$, except for the exterior face, where we set $\ell(F) := -\ell(C)$. This gives us

$$\sum_{F\text{: is a face of } G} \ell(F) = 0 \tag{1}$$

and

$$\ell(C) = \sum_{F \in f(C)} \ell(F) \text{ for all cycles } C \text{ of } G. \tag{2}$$

Lemma 1. *If there is an inconsistent cycle C in G, there are (at least) two faces corresponding to inconsistent cycles, one of which lies in the interior of C and the other one in the exterior of C.*

Proof. We first show that the existence of one inconsistent face implies that there are actually two inconsistent faces. To see this use Equation (1). If there were only one inconsistent face, its boundary would be the only nonzero summand, which means that the sum is nonzero contradicting (1).

Let $C \in \mathcal{C}$ be an inconsistent cycle in G. Consider the subgraph G_{int} of G, which only contains C and all edges and vertices in the interior of C. By construction, C is the boundary of an inconsistent face of G_{int} (namely the exterior face) and by the above claim G_{int} must have another inconsistent face, which lies in the interior of C. But this is, of course, also an inconsistent face in G.

The existence of an inconsistent face in the exterior of C follows analogously if we consider the subgraph G_{ext} of G, which only contains C and all edges and vertices in the exterior of C, instead of G_{int}.

Proposition 1. *Every minimal cycle in $\mathcal{C}$ with respect to $\subseteq_f$ is the boundary of a face of G and, therefore, face-minimal inconsistent cycles do not cross.*

Proof. Follows immediately from Lemma 1.

Lemma 2. *A graph is consistent if and only if all face-minimal cycles are consistent.*

Proof. Immediately from Proposition 1.

Inspired by Lemma 2, given a partial solution A, our violation oracle returns the collection of face-minimal inconsistent cycles in $G = (V, E \setminus A)$. Observe that this violation oracle can be implemented to run in polynomial time, since there is only a linear number of faces, which can be checked exhaustively. Our main ingredient for the analysis of the primal-dual algorithm for MinFs2$^=$ is the following:

Theorem 6. *Let G be a planar graph and let $\mathcal{M}$ be a collection of face-minimal inconsistent cycles. Then, for any minimal solution A we have*

$$\sum_{C \in \mathcal{M}} |A \cap C| \leq 2|A| \leq 2|\mathcal{M}|.$$

In order to prove the theorem, we need a number of auxiliary results. Let A be an inclusionwise minimal solution for MinFs2$^=$. By the minimality of A, for every $e \in A$ there must be an inconsistent cycle C_e such that $C_e \cap A = \{e\}$. In fact, if the intersection were empty or contained more than $\{e\}$ for every inconsistent cycle, then we could remove e from A because it either hits only cycles which are already being hit or it does not hit any cycle. We call such a cycle C_e with $C_e \cap A = \{e\}$ a *witness cycle* of e.

Due to lack of space, the proof of the following lemma is deferred to the full version of the paper:

Lemma 3. *Let $A \subset E$ be an inclusionwise minimal solution. Then, there exists a laminar family of witness cycles $C_e \in \mathcal{C}$, $e \in A$.* □

We are now ready to complete the proof of Theorem 6. We first see from Lemma 2 that the violation oracle $\mathcal{V}$, which returns all face-minimal inconsistent cycles is valid in the sense that if G contains an inconsistent cycle, $\mathcal{V}$ will be non-empty.

Let $\mathcal{F} = \{C_e | e \in A\}$ be a laminar family of witness cycles for A, which exists by Lemma 3. Since $\mathcal{F}$ is laminar, we can construct a forest with node set $\mathcal{F}$ representing the partial order induced by "$\subseteq_f$" restricted to the elements of $\mathcal{F}$. We add a root node r, connect it to all maximal elements of $\mathcal{F}$, and denote the resulting tree by T. For the analysis we now assign an edge $e \in A$ to the node of T corresponding to its witness cycle; furthermore, every element C of $\mathcal{M}$ can be assigned to (the node of T corresponding to) the smallest witness cycle that contains C; if there is no such witness cycle, C is assigned to the root node r. Let $\mathcal{M}_e$ denote the set of elements of $\mathcal{M}$ which are assigned to node C_e. Note that $\mathcal{M}_e$ might be empty for some nodes of T, but never for leaves. This is because by Lemma 1, there must be some inconsistent face inside every witness cycle.

We wish to bound $\sum_{C \in \mathcal{M}_e} |A \cap C|$ from above: For a cycle $C \in \mathcal{M}_e$, we know that $|A \cap C|$ can only contain the edge e and edges assigned to the children of C_e, since no element of $\mathcal{M}$ crosses any element of $\mathcal{F}$ and every witness cycle contains exactly one element of A (and therefore separates its inside faces from all elements of A but the aforementioned ones). By definition of T, the number of those candidate edges is $\deg_T(C_e)$, i.e., the degree of node C_e in the tree T. Since every edge touches at most two inconsistent faces, each edge can only appear once in $\sum_{C \in \mathcal{M}_e} |A \cap C|$, so

$$\sum_{C \in \mathcal{M}_e} |A \cap C| \leq \deg_T(C_e).$$

If $\mathcal{M}_e = \emptyset$, we can use 0 as a trivial better bound. Summing up over all $e \in A$, we obtain:

$$\sum_{C \in \mathcal{M}} |A \cap C| = \sum_{e \in A} \sum_{C \in \mathcal{M}_e} |A \cap C| \leq \sum_{e \in A : \mathcal{M}_e \neq \emptyset} \deg_T(C_e)$$

The average vertex-degree of a tree with n nodes is $\frac{2n-2}{n}$; we would like to use this, but the sum on the right-hand side does not contain all vertex-degrees of T, but only some of them. Here the key observation is that all the leaves of T appear in the sum; and the absence of any node with degree ≥ 2 can only decrease the average vertex-degree, which is below 2. A special case is the root node: It may have degree 1 but $\mathcal{M}_r$ can be empty. Taking this into account, we get

$$\sum_{e \in A : \mathcal{M}_e \neq \emptyset} \deg_T(C_e) \leq 2|\{e \in A : \mathcal{M}_e \neq \emptyset\}| - 1 \leq 2|\mathcal{M}| - 1.$$

This completes the proof of Theorem 6. Together with Theorem 5, we thus obtain:

Corollary 1. *There exists a polynomial time* 2*-approximation algorithm for* MinFs2$^=$ *when the associated constraint graph is planar.* □

5 Approximation Algorithms for MaxFs2$^=$

It is easy to obtain a 2-approximation algorithm for MaxFS2$^\leq$, that is, for the case when all constraints are inequalities of the form $x_i - x_j \leq c_{ij}$. Numbering the vertices in the constraint graph G_I from 1 to n, one splits the arcs into two groups, one that contains those arcs going from smaller to higher numbers and one that contains the other arcs going from higher to smaller numbers. Thus, the arc set of G_I is partitioned into two parts each of which forms an acyclic subgraph (and hence is consistent in the case of inequalities). One of the subgraphs contains at least one half of the total weight of the arcs and is, thus, a 2-approximation to MaxFS2$^\leq$.

Unfortunately, this approach fails for MaxFs2$^=$. On the other hand, if the constraint graph associated with an instance of MaxFs2$^=$ is a forest, it is trivially consistent. This observation can be used to obtain an approximation for MaxFs2$^=$. Recall that the *arboricity* of a graph is the minimum number of forests in which the edge set of a graph can be partitioned. This value can be computed in polynomial time as shown in [5,12]. Moreover, a simple planar graph with n vertices can be edge-partitioned into three forests in $O(n)$ time [14].

Suppose first that the constraint graph does not contain parallel edges. Then, computing a decomposition of the constraint graph G into the minimum number of forests and then selecting the one with the largest weight by averaging yields an approximation with a factor α, where α is the arboricity of G. Now, suppose that G does contain parallel edges. Let e_1 and e_2 be such two parallel edges. If they have the same length ℓ, we can collapse them into one edge with length ℓ and weight $w(e_1) + w(e_2)$; otherwise no solution at all can contain both e_1 and e_2, since they form an inconsistent cycle. In this case, we remove the lighter edge. Thus, we can eliminate all parallel edges without affecting solution quality.

Theorem 7. MaxFs2$^=$ *can be approximated within a factor of* α*, where* α *is the arboricity of the reduced graph, where parallel edges have been processed as above.* □

An immediate corollary is the following:

Corollary 2. *Choosing a maximum weight spanning tree of the constraint graph yields an approximation within a factor of* α*.* □

6 Extensions to Bounded Domains

In this section, we replace the domain $\mathbb{R}$ for each variable x_i by an interval $[a_i, b_i]$. We then have additional box constraints $a_i \leq x_i \leq b_i$, which are not optional, but need all to be satisfied. We call the extension of MinFs2$^=$ and MaxFs2$^=$ to interval domains MinFs2$^=$-D and MaxFs2$^=$-D respectively. Note that we can

test whether all box constraints can be satisfied simultaneously in polynomial time by a a shortest-path computation.

Analogously to inconsistent cycles, we define (domain) inconsistent paths: A *(domain) inconsistent path* $p = v_1, \ldots, v_k$ is a simple path with the property that either

$$a_{v_1} + \sum_{i=1}^{k-1} \ell(v_i, v_{i+1}) > b_{v_k} \quad \text{or} \quad b_{v_1} + \sum_{i=1}^{k-1} \ell(v_i, v_{i+1}) < a_{v_k}.$$

Obviously, an inconsistent path or an inconsistent cycle render the whole system invalid. In fact, we can show that it is also sufficient to destroy all inconsistent paths and cycles in order to have a domain feasible system.

Due to lack of space, most of the proofs of the following results will be deferred to the full version of the paper.

Theorem 8. *The problem of hitting all inconsistent paths by a subset of edges of minimum weight is NP-complete in general graphs. It remains strongly NP-complete on planar graphs, even if no vertex is incident to more than three arcs.*

Theorem 9. *The problems* MinFs2$^=$-D *and* MaxFs2$^=$-D *are polynomial solvable on trees. The result still holds if we have domains of the form* $[a_v, b_v] \cap \mathbb{Z}$.

Theorem 10. MinFs2$^=$-D *on graphs without inconsistent cycles can be approximated within a factor of* $O(\log n)$*, where* n *is the number of vertices. In the special case of planar graphs, there is a constant approximation factor.*

Proof. Given a graph without inconsistent cycles, each path between two nodes i and j has the same length d_{ij}. We can therefore determine which node pairs are connected by inconsistent paths. Consequently, we can regard the problem as a multicut problem with the conflicting nodes as terminal pairs and use the approximation algorithm from [7] which provide the stated approximation factors. □

Using the result of Theorem 9 with the techniques of Section 5 we obtain:

Theorem 11. MaxFs2$^=$-D *has an* α*-approximation guarantee on general graphs, where* α *is again the arboricity of the reduced graph.* MinFs2$^=$-D *has a constant approximation guarantee on planar graphs.*

References

1. Amaldi, E., Bruglieri, M., Casale, G.: A two-phase relaxation-based heuristic for the maximum feasible subsystem problem. Computers and Operations Research 35, 1465–1482 (2008)
2. Amaldi, E., Kann, V.: The complexity and approximability of finding maximum feasible subsystems of linear relations. Theoretical Computer Science 147(1-2), 181–210 (1995)

3. Chinneck, J.: Fast heuristics for the maximum feasible subsystem problem. INFORMS Journal on Computing 13(3), 211–223 (2001)
4. Elbassioni, K., Raman, R., Ray, S., Sitters, R.A.: On the approximability of the maximum feasible subsystem problem with 0/1-coefficients. In: Proceedings of the 20th Annual ACM-SIAM Symposium on Discrete Algorithms, pp. 1210–1219 (2009)
5. Gabow, H.N., Westermann, H.H.: Forests, frames, and games: algorithms for matroid sums and applications. Algorithmica 7(1), 465–497 (1992)
6. Garey, M.R., Johnson, D.S.: The rectilinear steiner tree problem is NP-complete. SIAM Journal on Applied Mathematics, 826–834 (1977)
7. Garg, N., Vazirani, V.V., Yannakakis, M.: Approximate max-flow min-(multi) cut theorems and their applications. SIAM Journal on Computing 25, 235 (1996)
8. Goemans, M.X., Williamson, D.P.: The primal-dual method for approximation algorithms and its application to network design problems. In: [11], pp. 144–191. PWS Publishing Company (1997)
9. Goemans, M.X., Williamson, D.P.: Primal-dual approximation algorithms for feedback problems in planar graphs. Combinatorica 18(1), 37–59 (1998)
10. Greer, R.: Trees and hills: Methodology for maximizing functions of systems of linear relations. Annals of Discrete Mathematics 22 (1984)
11. Hochbaum, D.S. (ed.): Approximation algorithms for NP-hard problems. PWS Publishing Company, 20 Park Plaza, Boston, MA 02116–4324 (1997)
12. Nash-Williams, C.: Decomposition of finite graphs into forests. Journal of the London Mathematical Society 1(1), 12 (1964)
13. Pfetsch, M.: Branch-and-cut for the maximum feasible subsystem problem. SIAM Journal on Optimization 19(1), 21–38 (2008)
14. Schnyder, W.: Embedding planar graphs on the grid. In: Proceedings of the First Annual ACM-SIAM Symposium on Discrete Algorithms, pp. 138–148 (1990)