Combinatorial properties of a general domination problem with parity constraints

Johannes Hatzl [1]

Department of Management Science
University of Waterloo
200 University Avenue West
Waterloo, Ontario, N2L 3G1
Canada

Stephan Wagner

Department of Mathematical Sciences
Mathematics Division
Stellenbosch University
Private Bag X1, Matieland 7602
South Africa

Abstract

We consider various properties of a general parity domination problem: given a graph G, one is looking for a subset S of the vertex set such that the open/closed neighborhood of each vertex contains an even/odd number of vertices in S (it is prescribed individually for each vertex which of these applies). We define the parameter $s(G)$ to be the number of solvable instances out of 4^n possibilities and study the properties of this parameter. Upper and lower bounds for general graphs and trees are given as well as a remarkable recurrence formula for rooted trees and—more generally—an algorithm for graphs of bounded tree-width. Furthermore, we give explicit formulas in several special cases and investigate random graphs.

Key words: domination problem, parity constraint, matrix algebra

Email addresses: jhatzl@uwaterloo.ca (Johannes Hatzl), swagner@sun.ac.za (Stephan Wagner).

[1] This research is partially supported by the MITACS Network of Centres of Excellence through the project *High Performance Optimization: Theory, Algorithm Design and Engineering Applications.*

1 Introduction

The classical problem of domination asks for a subset S of the vertex set of a graph $G = (V(G), E(G))$ of minimum cardinality such that $N[v] \cap S \neq \emptyset$ for all $v \in V(G)$, where $N[v] := \{u \in V(G) \mid \exists e = (u, v) \in E(G)\} \cup \{v\}$ denotes the closed neighborhood of v. Quite a lot of different modifications and generalizations of this problem are known. For instance, the *k-tuple domination problem* [12] asks for a minimum set S such that $|N[v] \cap S| \geq k$ for all vertices v. Similarly, in the *k-domination problem* [9,10] the task is to find a set S of minimum cardinality such that $|N(v) \cap S| \geq k$ for all vertices v, where $N(v) := \{u \in V(G) \mid \exists e = (u, v) \in E(G)\}$ denotes the open neighborhood of v. Even more generally, one can prescribe a set R_v for every vertex v and pose the question whether there exists a set S such that $|N[v] \cap S| \in R_v$ (or $|N(v) \cap S| \in R_v$) for all vertices v.

The special cases $R_v = \{1, 2, 3, \ldots\}$ and $R_v = \{k, k+1, \ldots\}$ have already been mentioned. These and other variants, such as $R_v = \{1\}$, are discussed in the book of Haynes, Hedetniemi and Slater [13]. Another interesting kind of domination problem involves parity constraints. It has been treated in a series of papers by Amin, Slater and others [1–4], motivated by the following remarkable result of Sutner [17]:

Theorem 1 (Sutner [17]). *For every graph $G = (V(G), E(G))$, there exists a set $S \subseteq V(G)$ such that $|N[v] \cap S|$ is odd for every $v \in V(G)$.*

This means that the domination problem for $R_v = \{1, 3, 5, \ldots\}$ is always solvable if we consider closed neighborhoods. Thus, it is pretty natural to consider a general parity assignment problem, where each R_v is either $\{1, 3, 5, \ldots\}$ or $\{0, 2, 4, \ldots\}$. It has been treated quite extensively in [1–4], where the notions of "parity dimension" and "all parity realizable graphs" have been introduced. Wagner [18] gives a recursive procedure for determining the parity dimension of a tree, which is then applied to enumeration problems involving the parity dimension, in particular to counting all parity realizable trees.

In another recent paper, Gassner and Hatzl [11] discuss an even more general parity domination problem from an algorithmic point of view: for every vertex v, we have exactly one of the following four constraints:

- $|N(v) \cap S| \equiv 0 \mod 2$,
- $|N(v) \cap S| \equiv 1 \mod 2$,
- $|N[v] \cap S| \equiv 0 \mod 2$,
- $|N[v] \cap S| \equiv 1 \mod 2$.

I.e., the open/closed neighborhood has to contain an even/odd number of vertices in S. Another reason to consider domination problems with parity con-

straints lies in the fact that the problem can be stated easily in terms of matrix algebra: in the following, we denote by A and $I + A$ the open neighborhood matrix (adjacency matrix) and the closed neighborhood matrix respectively (I is the identity matrix). Furthermore, we use a vector $a \in \{0,1\}^{V(G)}$ as a representation for the neighborhood information (i.e., whether the open or closed neighborhood is considered for a certain vertex): if the entry a_v that corresponds to a vertex v is 0, the open neighborhood is of interest for this vertex, and the closed neighborhood otherwise. Another vector $b \in \{0,1\}^{V(G)}$ represents the prescribed parities. Using these vectors, our requirements can be written as

$$(A + \operatorname{diag}(a))x = b \tag{1}$$

over the field $\mathbb{F}_2$. Obviously, $x_v = 1$ if and only if $v \in S$.

In this paper, we are interested in the number of solvable instances—a parameter that plays an analogous role to the parity dimension (for the domination problem with parity constraints considering closed subsets only): let the *solvability number* $s(G)$ denote the number of solvable instances for a graph G, i.e., the number of pairs $(a, b) \in \{0,1\}^{V(G)} \times \{0,1\}^{V(G)}$ such that there exists a vector x satisfying the system of linear equations in (1).

Basic linear algebra gives us the the following simple lemma:

Lemma 2. *Let $G = (V(G), E(G))$ be a graph, then*

$$s(G) = \sum_{a \in \{0,1\}^{V(G)}} 2^{\operatorname{rk}(A + \operatorname{diag}(a))}, \tag{2}$$

where $\operatorname{rk}(B)$ *denotes the rank of a matrix B over $\mathbb{F}_2$.*

Remark 3. Replacing 2 by a variable x in the above formula, we obtain a polynomial

$$S_G(x) = \sum_{a \in \{0,1\}^{V(G)}} x^{\operatorname{rk}(A + \operatorname{diag}(a))}$$

with interesting properties: $S_G(0) = 1$ if G is the empty graph, and $S_G(0) = 0$ otherwise; $S_G(1) = 2^{|V(G)|}$, and $\frac{S'_G(1)}{S_G(1)}$ gives the average rank of $A + \operatorname{diag}(a)$, as a varies over all possible vectors.

Corollary 4.

$$2^{|V(G)|} \leq s(G) \leq 4^{|V(G)|}.$$

Thus, $\frac{\log_2 s(G)}{|V(G)|}$ can be seen as a "normalisation" of $s(G)$ which always lies between 1 and 2. In Section 2, we will improve these bounds for arbitrary graphs. Moreover, explicit formulas for some graphs are given. Section 3 is dedicated to trees and a recurrence formula for rooted trees. This recursion enables us to improve the lower bound for trees. Afterwards, an algorithm which computes $s(G)$ in linear time for graphs with bounded tree width is

discussed. The last section deals with the expected value of $s(G)$ for random graphs.

2 Special cases and inequalities

In this section, explicit formulas for special graphs are deduced. All these formulas are obtained using equation (2) and arguments from linear algebra.

Proposition 5. *The solvability numbers of the empty graph and the complete graph on n vertices are 3^n and $2 \cdot 3^n - 5 \cdot 2^{n-2} + (-2)^{n-2}$ respectively.*

Proof. The formula for the empty graph is trivial: just note that the rank of $A + \mathrm{diag}(a)$ equals the number of 1's in a (since $A = 0$), from which the formula follows from the binomial theorem.
The formula for the complete graph is slightly trickier: if $a \neq 0$, one of the rows of $A + \mathrm{diag}(a)$ consists entirely of 1's. We subtract this row from all others and obtain a matrix whose rows (except one) contain at most one 1. It is easy to see that the rank is thus the number of 0's in a, increased by 1. Applying the binomial theorem again yields the main term $2 \cdot 3^n$. Since the rank of the adjacency matrix A is n if n is even and $n-1$ if n is odd (by a similar argument—note that the vector $(1, 1, \ldots, 1)$ is spanned by the rows of A if and only if n is even), we obtain $2 \cdot 3^n - 2^n$ if n is even and $2 \cdot 3^n - 3 \cdot 2^{n-1}$ if n is odd, which reduces to $2 \cdot 3^n - 5 \cdot 2^{n-2} + (-2)^{n-2}$ for all n. □

Proposition 6. *The solvability numbers of the path P_n and the cycle C_n on n vertices are $s(P_n) = \frac{5}{6} \cdot 4^n + \frac{1}{6} \cdot (-2)^n$ and $s(C_n) = \frac{5}{8} \cdot 4^n - \frac{1}{4} \cdot (-2)^n$, respectively.*

Proof. Again, we want to determine the number of vectors a for which $A + \mathrm{diag}(a)$ has a specific rank. For this purpose, we only have to find nontrivial solutions x to the equation $(A+\mathrm{diag}(a))x = 0$. Let us start with the path P_n, and let its vertices be $v_1, v_2, \ldots, v_n$, where v_1 and v_n are the leaves. The corresponding entries of a and x are denoted by $a_1, a_2, \ldots, a_n$ and $x_1, x_2, \ldots, x_n$. Suppose that x_1 is given. Then x_2 is uniquely defined by the equation $a_1 x_1 + x_2 = 0$, x_3 is uniquely defined by $x_1 + a_2 x_2 + x_3 = 0$, and so on. Hence, a solution to $(A + \mathrm{diag}(a))x = 0$ is uniquely determined by x_1 (if it exists). It is plain that $x_1 = 0$ always leads to the trivial solution, so we may assume $x_1 = 1$. In order to determine whether a nontrivial solution exists for a given vector a, we can use a finite automaton that reads the entries $a_1, a_2, \ldots$ of a (in this order). A state is given by two consecutive entries x_i, x_{i+1} of x. The initial state ($i = 0$) is $(0, 1)$ (0 for the non-existing vertex v_0 and 1 by the assumption that $x_1 = 1$). The general equation

$$x_{i-1} + a_i x_i + x_{i+1} = 0$$

implies that the state transitions are

$$a_i = 0 : (0,1) \to (1,0),\ (1,0) \to (0,1),\ (1,1) \to (1,1),$$
$$a_i = 1 : (0,1) \to (1,1),\ (1,0) \to (0,1),\ (1,1) \to (1,0).$$

Figure 1 shows the resulting automaton. A nontrivial solution exists if and only if the final state is $(x_n, x_{n+1}) = (1,0)$. If a nontrivial solution exists, it is unique.

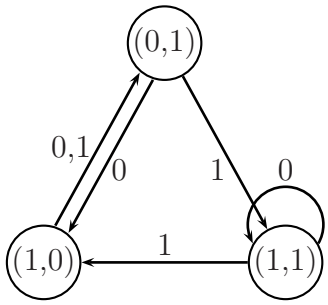


Fig. 1. An automaton used for determining the rank of $A + \mathrm{diag}(a)$ in the case of the path.

Now it is an easy exercise to determine the number of vectors a for which a nontrivial solution exists (and for which the rank of $A + \mathrm{diag}(a)$ is thus $n-1$ rather than n) from the adjacency matrix of the automaton, which in turn yields the formula for $s(P_n)$: the number of vectors for which $A + \mathrm{diag}(a)$ has full rank n is given by $\frac{2}{3} \cdot 2^n + \frac{1}{3} \cdot (-1)^n$, whereas the number of vectors for which $A + \mathrm{diag}(a)$ has rank $n-1$ is given by $\frac{2}{3} \cdot 2^{n-1} + \frac{1}{3} \cdot (-1)^{n-1}$.

The procedure is basically the same for the cycle. However, we have to prescribe two initial values x_1 and x_2, for which there are four possibilities. $(x_1, x_2) = (0,0)$ leads to the trivial solution again. The automaton is the same as for the path—a nontrivial solution exists if and only if $(x_1, x_2) = (x_{n+1}, x_{n+2})$ for some initial values (x_1, x_2) in the set $\{(0,1), (1,0), (1,1)\}$. If a nontrivial solution exists for exactly one of them, the rank of $A + \mathrm{diag}(a)$ is $n-1$. However, if a nontrivial solution exists for all three, the rank is $n-2$. To take account of this, we also count the number of vectors a for which this occurs. This is done by means of another automaton, whose states are triples of the states of the first automaton (Figure 2). The automaton returns to its initial state after n steps if and only if there is a nontrivial solution for all three choices of (x_1, x_2).

Now it turns out that the rank of $A + \mathrm{diag}(a)$ is n for $\frac{2}{3} \cdot 2^{n-1} + \frac{1}{3} \cdot (-1)^{n-1}$ vectors, $n-1$ for 2^{n-1} vectors and $n-2$ for $\frac{2}{3} \cdot 2^{n-2} + \frac{1}{3} \cdot (-1)^{n-2}$ vectors, which yields the formula for $s(C_n)$. □

A formula for the star will be given in Section 3. Similarly, more tedious calculations yield the following formulas:

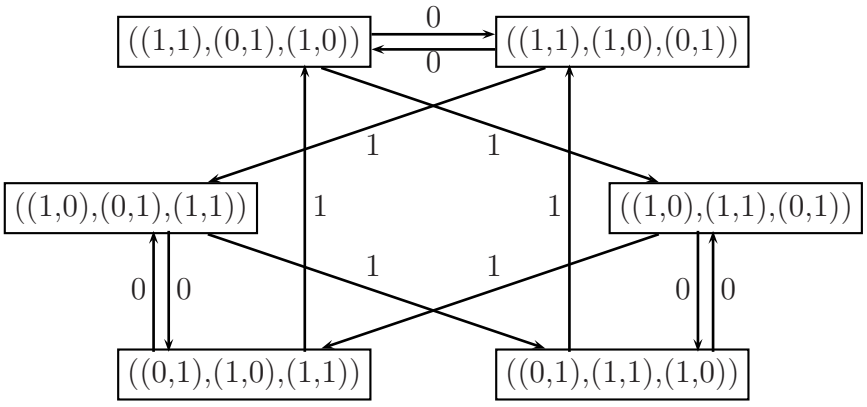


Fig. 2. An automaton used for determining the rank of $A + \mathrm{diag}(a)$ in the case of the cycle.

Proposition 7. *The solvability numbers of the fan F_n and the wheel W_n on n vertices are*

$$s(F_n) = \tfrac{35}{48} \cdot 4^n - \tfrac{1}{24} \cdot (-2)^n + \tfrac{5}{4\sqrt{-7}} \left((-1+\sqrt{-7})^{n-1} - (-1-\sqrt{-7})^{n-1} \right)$$

and

$$s(W_n) = \tfrac{85}{128} \cdot 4^n + \tfrac{15}{64} \cdot 2^n + \tfrac{7}{32} \cdot (-2)^n - \tfrac{25}{32} \left((-1+\sqrt{-7})^{n-1} + (-1-\sqrt{-7})^{n-1} \right),$$

respectively.

Theorem 8. *Let E_n be the empty graph on n vertices. Then,*

$$s(G) \geq s(E_n)$$

holds for all graphs G with $|V(G)| = n$. Moreover, the inequality is strict if $G \neq E_n$.

Proof. First we prove that there is always a vector a_0 such that $A + \mathrm{diag}(a_0)$ has full rank. This is done by means of induction. The statement is trivial for $n = 1$. For the induction step, let v be the vertex that corresponds to the last row of A. By the induction hypothesis, we can choose a vector a_0' such that $A' + \mathrm{diag}(a_0')$, where A' is the adjacency matrix of $G \setminus \{v\}$, is invertible. Hence, there is a unique set R of rows of $A' + \mathrm{diag}(a_0')$ whose sum is equal to the last row of A (without the last element). If there is an even number of neighbors of v among the vertices corresponding to R, we set $b = 1$, otherwise we set $b = 0$. Now, we obtain a vector a_0 by appending b to a_0'. Then, $A + \mathrm{diag}(a_0)$ has full rank again: by our choice of b, the last row is linearly independent of the others, which in turn are linearly independent by the assumption that $A' + \mathrm{diag}(a_0')$ is invertible. This finishes the induction.

Therefore, there is always at least one vector a_0 for which $A + \mathrm{diag}(a_0)$ has full rank. Since changing an entry of a changes the rank at most by 1, there are at most $\sum_{k=0}^{l} \binom{n}{k}$ vectors a for which $A + \mathrm{diag}(a)$ has rank $\leq l$. Equality only holds for the empty graph (the only graph for which the rank can be 0). By Lemma 2, this proves the claim. □

Similarly, an upper bound for $s(G)$ is given in the next theorem.

Theorem 9. *Let P_n be the path with n vertices. Then,*

$$s(G) \leq s(P_n)$$

holds for all graphs G with $|V(G)| = n$. Moreover, the inequality is strict if $G \neq P_n$.

Proof. In the proof of Proposition 6 it was shown that $|\{a \in \{0,1\}^n : \mathrm{rk}(A + \mathrm{diag}(a)) < n\}| = \frac{2}{3}2^{n-1} + \frac{1}{3}(-1)^{n-1}$ where A is the adjacency matrix of P_n. Moreover, it was proved that $\mathrm{rk}(A + \mathrm{diag}(a))$ is n or $n-1$ for all $a \in \{0,1\}^n$. Thus, it suffices to show that

$$|\{a \in \{0,1\}^n : \mathrm{rk}(A + \mathrm{diag}(a)) < n\}| \geq c(n), \tag{3}$$

where

$$c(n) := \frac{2}{3}\,2^{n-1} + \frac{1}{3}\,(-1)^{n-1},$$

holds for all symmetric $n \times n$ matrices A. Without loss of generality, we can restrict ourselves to adjacency matrices, i.e., to matrices where all diagonal entries are 0. In order to prove this we use induction on n. It is easy to see that the inequality given in (3) holds for $n = 1$ and $n = 2$. Moreover, if $n = 2$ we have equality if and only if A is the adjacency matrix of P_2.
Let A be an $(n+1) \times (n+1)$ adjacency matrix and assume that (3) holds for n and $n-1$. For the inductive step, we give upper bounds on the cardinality of the following two sets:

$$S_1 := \{a \in \{0,1\}^{n+1} : \mathrm{rk}(A + \mathrm{diag}(a)) < n+1 \text{ and } a_1 = 1\}, \tag{4}$$
$$S_2 := \{a \in \{0,1\}^{n+1} : \mathrm{rk}(A + \mathrm{diag}(a)) < n+1 \text{ and } a_1 = 0\}. \tag{5}$$

Let us assume we are given a symmetric matrix $A + \mathrm{diag}(a)$ with $a_1 = 1$. In order to calculate its rank, we add the first row to all rows i with $a_{i,1} = 1$. Then we obtain a matrix $\tilde{A}$, where $\tilde{a}_{i,1} = 0$ holds for all $i = 2, \ldots, n+1$ and $\tilde{a}_{1,1} = 1$. Consequently, the rank of $\tilde{A}$ equals the rank of its $(1,1)$-minor, i.e., the matrix that results from deleting the first row and first column. Note that this minor is still a symmetric matrix (since $a_{i,1}a_{1,j}$ is added to each entry $a_{i,j}$, which is clearly symmetric), so (by the induction hypothesis) there are at least $c(n)$ different diagonal vectors for which it does not have full rank. Moreover, each of the possible diagonals appears exactly once as a varies over

all possible vectors. Thus, using the induction hypothesis, $|S_1| \geq c(n)$ follows. The upper bound on the cardinality of S_2 can be obtained similarly. We assume without loss of generality that $a_{2,1} = a_{1,2} = 1$. Otherwise, all entries in the first row, respectively column, are 0 and we can conclude that $|S_2| = 2^n$. Consequently, equation (3) follows immediately for $n+1$, because $2^n + c(n) \geq c(n+1)$. Now we use some elementary row and column operations again to calculate the rank of $A + \mathrm{diag}(a)$. First we add the second row to all rows i with $a_{i,1} = 1$. Afterwards, we add the second column to all columns j with $b_{1,j} = 1$. After interchanging the first and the second row, we end up with a matrix $\tilde{A}$ for which $\tilde{a}_{1,1} = \tilde{a}_{2,2} = 1$ and $\tilde{a}_{i,1} = 0$ for all $i = 2, \ldots, n+1$ and $\tilde{a}_{2,j} = 0$ for all $j = 3, \ldots, n+1$. Due to this fact, the matrix $\tilde{A}$ has full rank if and only if the matrix obtained from $\tilde{A}$ by deleting the first two rows and columns has full rank. Let us denote this $(n-2) \times (n-2)$ matrix by $\tilde{A}'$. Note that $\tilde{A}'$ is also symmetric, since $a_{i,1}a_{2,j} + a_{1,j}a_{i,2} + a_{i,1}a_{1,j}(a_{2,2} + a_2)$ is added to each entry $a_{i,j}$, which is symmetric in i and j again. Furthermore, each of the possible diagonals of $\tilde{A}'$ appears exactly twice as a varies over all possible vectors. Using the induction hypothesis, $|S_1| \geq 2c(n-1)$ can be concluded. Summarizing, we get

$$\begin{aligned} &|\{a \in \{0,1\}^{n+1} : \mathrm{rk}(A + \mathrm{diag}(a)) < n+1\}| \\ &\qquad = |S_1| + |S_2| \geq c(n) + 2c(n-1) = c(n+1), \quad (6) \end{aligned}$$

which completes the induction step. To see why $s(G) < s(P_n)$ if $G \neq P_n$, note that $|S_1| + |S_2| = c(n) + 2c(n-1)$ holds if and only if A is the adjacency matrix of P_{n+1}. Using the induction hypothesis again, this completes the proof. □

Theorem 10. *Let G be a graph and G^c its complement. Then we have*

$$\tfrac{1}{2}s(G) \leq s(G^c) \leq 2s(G).$$

The constant 2 is best-possible in this inequality.

Proof. Let A, A' be the adjacency matrices of G and G^c. Furthermore, for a vector a, let $a' = e - a$, where e is the vector whose entries are all 1 (i.e., all entries of a are switched). Finally, let E be the $|V(G)| \times |V(G)|$ matrix whose entries are all 1. Then, it is clear that

$$A' + \mathrm{diag}(a') = E - (A + \mathrm{diag}(a)).$$

Furthermore, if we add the vector e as a row to some matrix M and $E - M$, the space that is spanned by the rows of the two matrices becomes the same, and the rank increases at most by one. Hence, the ranks of M and $E - M$ differ at most by one. Applying this to $A + \mathrm{diag}(a)$ and summing over all a, we have, by Lemma 2,

$$\tfrac{1}{2}s(G) \leq s(G^c) \leq 2s(G),$$

as claimed. The formulas for the empty graph and the complete graph show that the constant 2 is indeed best-possible. □

3 A property of distinguished vertices and a recursion for rooted trees

Let v be a distinguished vertex (the "root") of a graph G. Given the neighborhood vector a and the parity vector b, we say that an instance is "solvable" if there exists a solution x in (1) and "almost solvable" if there exists a solution x for the instance (a, b'), where b' results from replacing the entry b_v belonging to v by $1 - b_v$. Furthermore, if a solution vector x is given, we say that we "use the root" if the entry x_v that corresponds to v is 1, and that we "don't use the root" otherwise. Now, the following lemma holds, which proves extremely useful in the treatment of trees:

Lemma 11. *Given a root $v \in V(G)$, a neighborhood vector a and a parity vector b, the following situations (and no others) are possible:*

(1) the instance is neither solvable nor almost solvable,
(2) the instance is solvable using the root or without using the root, but not almost solvable,
(3) the instance is almost solvable using the root or without using the root, but not solvable,
(4) the instance is solvable using the root and almost solvable without using the root,
(5) the instance is almost solvable using the root and solvable without using the root,
(6) the instance is solvable and almost solvable if we use the root, but neither solvable nor almost solvable if we don't use it,
(7) the instance is solvable and almost solvable if we don't use the root, but neither solvable nor almost solvable if we use it.

Furthermore, the number of instances belonging to the second case is the same as the number of instances belonging to case (3), the number of instances belonging to case (4), and the number of instances belonging to case (5); we denote this quantity by $x(G, v)$. Similarly, the number of instances belonging to case (6) is the same as the number of instances belonging to case (7); we denote this quantity by $y(G, v)$. Then, $3x(G, v) + 2y(G, v) = s(G)$ holds for all vertices v.

Proof. Set $M = A + \mathrm{diag}(a)$. Since the kernel and the image of a symmetric matrix are orthogonal, we know that precisely one of the following possibilities hold:

- there is a vector x with $x_v = 1$ such that $Mx = 0$,
- there is a vector x such that $Mx = e_v$, where e_v is the unit vector whose entry is 1 at the position that corresponds to v and 0 otherwise.

It is easy to see that the first statement is equivalent to the fact that all solvable or almost solvable instances (a, b) can be solved (almost solved) with or without using the root. On the other hand, $Mx = e_v$ is equivalent to the fact that every instance that can be solved can also be almost solved and vice versa. Hence, if the first statement holds, all (almost) solvable instances (a, b) belong to case (2) or (3), and there are no instances that can be solved as well as almost solved. Whether an instance belongs to case (2) or (3) only depends on b_v, and thus the number of instances is the same for each of these two cases.

If the second statement holds, all (almost) solvable instances (a, b) belong to case (4) or (5) (depending only on b_v) if $x_v = 1$, and to case (6) or (7) if $x_v = 0$. In the former case, note that x becomes a solution to $Mx = 0$ if a_v is changed from 0 to 1 or vice versa. Hence, the instances belonging to cases (2) and (3) and the instances belonging to cases (4) and (5) are equinumerous.

Finally, we have to prove the fact that there are equally many instances belonging to cases (6) and (7). But this is also obvious, since they form affine spaces of the same dimension which differ only by the first column of M. □

The following proposition provides a recursion for the auxiliary parameters $x(G, v)$ and $y(G, v)$ in the case of rooted trees, which enables us to deal with all sorts of questions concerning the solvability number of trees:

Proposition 12. *Let T be a rooted tree and v its root. Furthermore, let the branches of T be $T_1, \ldots, T_k$ and their roots (the neighbors of v) $v_1, \ldots, v_k$. Then the identities*

$$x(T, v) = \prod_{i=1}^{k} (2x(T_i, v_i) + 2y(T_i, v_i))$$

and

$$y(T, v) = 4 \left(\prod_{i=1}^{k} (3x(T_i, v_i) + 2y(T_i, v_i)) - \prod_{i=1}^{k} (2x(T_i, v_i) + 2y(T_i, v_i)) \right)$$

hold, which can be simplified as follows:

$$s(T) = 8 \prod_{i=1}^{k} s(T_i) - 5 \prod_{i=1}^{k} t(T_i, v_i),$$

$$t(T, v) = 8 \prod_{i=1}^{k} s(T_i) - 6 \prod_{i=1}^{k} t(T_i, v_i),$$

where $t(T, v) = 2x(T, v) + 2y(T, v)$ is another auxiliary parameter.

Proof. Let an instance (a, b) be given and let $(a^{(i)}, b^{(i)})$ be the restriction to the i-th branch. Obviously, (a, b) can only be solvable or almost solvable if all restrictions $(a^{(i)}, b^{(i)})$ are. We distinguish three cases:

(1) There are two instances $(a^{(i)}, b^{(i)})$ and $(a^{(j)}, b^{(j)})$ which belong to cases (2) and (3) of Lemma 11 respectively. Then, we must not use the root v in order to solve the subproblem in the i-th subtree, but we have to use it in order to solve the subproblem in the j-th subtree; in view of this contradiction, (a, b) is neither solvable nor almost solvable in this case.

(2) There is at least one instance $(a^{(i)}, b^{(i)})$ which belongs to case (2) (or (3)) of Lemma 11. We consider the first possibility, since the second can be treated analogously. Then, we must not use the root v, but we can solve each of the subproblems, and since we may decide in the i-th branch whether we want to use v_i or not, the instance (a, b) is solvable as well as almost solvable in this case. Hence, it belongs to case (6) or (7) of Lemma 11. There are

$$4 \cdot 2 \left(\prod_{i=1}^{k} (3x(T_i, v_i) + 2y(T_i, v_i)) - \prod_{i=1}^{k} (2x(T_i, v_i) + 2y(T_i, v_i)) \right)$$

possible instances for this case (note that there are still 4 ways to choose a_v and b_v!), and since cases (6) and (7) are equinumerous, we obtain the formula for $y(T, v)$.

(3) If there are no instances $(a^{(i)}, b^{(i)})$ which belong to case (2) or (3) of Lemma 11, then it follows analogously that (a, b) belongs to one of the cases (2)–(5) (note that we can either use or not use the root v in order to solve the subproblems in this case), the particular case depending on the choice of a_v and b_v. The formula for $x(T, v)$ follows as a simple consequence.

Now, substitution of $s(T_i) = 3x(T_i, v_i) + 2y(T_i, v_i)$ and $t(T_i, v_i) = 2x(T_i, v_i) + 2y(T_i, v_i)$ yields

$$x(T, v) = \prod_{i=1}^{k} t(T_i, v_i)$$

and

$$y(T, v) = 4 \left(\prod_{i=1}^{k} s(T_i) - \prod_{i=1}^{k} t(T_i, v_i) \right),$$

and the formulas for $s(T)$ and $t(T, v)$ follow immediately. □

The recursive formula provides us with another method to prove the formula for the path as well as an explicit formula for the star:

Corollary 13. *Let P_n and S_n be the path and the star on n vertices respectively. Then, the formulas $s(P_n) = \frac{5}{6}4^n + \frac{1}{6}(-2)^n$ and $s(S_n) = 8 \cdot 3^{n-1} - 5 \cdot 2^{n-1}$ hold.*

Proof. Note that $s(K_1) = 3$ and $t(K_1, v) = 2$ (where v is the only vertex of K_1). Using this fact the formula for both graphs is immediately obtained from the above proposition (in the case of the path, one has to solve a simple linear recursion). □

It can even be shown that the star has the smallest number of solvable instances among all trees with a given number of vertices, whereas the path has the largest number of solvable instances (which is clear in view of Theorem 9).

Proposition 14. *Let S_n be the star graph with n vertices. Then,*

$$s(S_n) \leq s(T)$$

holds for all trees $T = (V(T), E(T))$ with $|V(T)| = n$.

Proof. We show using induction on n that $s(S_n) \leq s(T)$ (where $s(S_n) = 8 \cdot 3^{n-1} - 5 \cdot 2^{n-1}$), that $t(T, v)$ attains its minimum value among all trees with n vertices if $T = S_n$ and v is one of its leaves (with a minimum value of $16 \cdot 3^{n-2} - 2^n$), and that $x(T, v) = s(T) - t(T, v)$ is minimum if T is the star and v its center (with a minimum value of 2^{n-1}). This is obvious for $n = 1$ or $n = 2$. For the induction step, note that

$$\begin{aligned} s(T) &= 8 \prod_{i=1}^{k} s(T_i) - 5 \prod_{i=1}^{k} t(T_i, v_i) \\ &= 8 s(T_j) \prod_{\substack{i=1 \\ i \neq j}}^{k} s(T_i) - 5(s(T_j) - x(T_j, v_j)) \prod_{\substack{i=1 \\ i \neq j}}^{k} t(T_i, v_i) \\ &= \left(8 \prod_{\substack{i=1 \\ i \neq j}}^{k} s(T_i) - 5 \prod_{\substack{i=1 \\ i \neq j}}^{k} t(T_i, v_i) \right) s(T_j) + 5 \prod_{\substack{i=1 \\ i \neq j}}^{k} t(T_i, v_i) \cdot x(T_j, v_j) \end{aligned}$$

for all j, and that $8 \prod_{\substack{i=1 \\ i \neq j}}^{k} s(T_i) - 5 \prod_{\substack{i=1 \\ i \neq j}}^{k} t(T_i, v_i) > 0$. Hence, by the induction hypothesis, every branch of a tree for which $s(T)$ is minimum has to be a star, rooted at its center (or a single vertex). The same way of reasoning works for the tree for which $t(T, v)$ is minimum. Finally, the branches of a tree for which $x(T, v)$ is minimum have to be stars (or possibly single vertices), each rooted at one of its leaves.

For $s(T)$, the argument is easy now: without loss of generality, we may assume that v is a leaf, and the claim readily follows.

For $t(T, v)$, we have to maximize the expression

$$8 \prod_{i=1}^{k} (8 \cdot 3^{n_i - 1} - 5 \cdot 2^{n_i - 1}) - 6 \prod_{i=1}^{k} (8 \cdot 3^{n_i - 1} - 6 \cdot 2^{n-1})$$

subject to the condition $\sum_{i=1}^{k} n_i = n-1$ (here, n_i is the number of vertices in the i-th branch). First, suppose that $k \geq 2$, consider two branches l, m, and set $n_l + n_m = p$. We write A and B for

$$\prod_{\substack{i=1 \\ i \neq l,m}}^{k} (8 \cdot 3^{n_i-1} - 5 \cdot 2^{n_i-1}) \quad \text{and} \quad \prod_{\substack{i=1 \\ i \neq l,m}}^{k} (8 \cdot 3^{n_i-1} - 6 \cdot 2^{n_i-1})$$

respectively and note that $A \geq B$. Now, we have

$$\begin{aligned} 8\prod_{i=1}^{k}(8 \cdot 3^{n_i-1} - 5 \cdot 2^{n_i-1}) - 6\prod_{i=1}^{k}(8 \cdot 3^{n_i-1} - 6 \cdot 2^{n-1}) & \\ = 8A(8 \cdot 3^{n_l-1} - 5 \cdot 2^{n_l-1})(8 \cdot 3^{n_m-1} - 5 \cdot 2^{n_m-1}) & \\ - 6B(8 \cdot 3^{n_l-1} - 6 \cdot 2^{n_l-1})(8 \cdot 3^{n_m-1} - 6 \cdot 2^{n_m-1}) & \\ = (25A - 27B) \cdot 2^{p+1} + 128(4A - 3B) \cdot 3^{p-2} & \\ - \frac{16(10A - 9B)}{3}\left(3^p \left(\frac{2}{3}\right)^{n_l} + 2^p \left(\frac{3}{2}\right)^{n_l}\right), & \end{aligned}$$

and this expression is a concave function in n_l. Hence, the minimum is attained at the borders, namely if $n_l = 1$ or $n_l = p - 1$. But this means that the overall minimum can only be attained if all but one n_l equal 1. Thus let $n_1 = n_2 = \ldots = n_{k-1} = 1$ and $n_k = n - k$. Then we have to minimize

$$\begin{aligned} 8 \cdot 3^{k-1}(8 \cdot 3^{n-k-1} - 5 \cdot 2^{n-k-1}) - 6 \cdot 2^{k-1}(8 \cdot 3^{n-k-1} - 6 \cdot 2^{n-k-1}) & \\ = 64 \cdot 3^{n-2} + 9 \cdot 2^n - 8 \cdot 3^n \left(\frac{2}{3}\right)^k - \frac{20}{3} \cdot 2^n \left(\frac{3}{2}\right)^k, & \end{aligned}$$

which is again a concave function (in k). Comparing the values at the borders, we see that the minimum is attained for $k = 1$, which corresponds to a star, v being one of its leaves.

Finally, in order to minimize $x(T, v)$, we have to minimize

$$\prod_{i=1}^{k} f(n_i),$$

where $\sum_{i=1}^{k} n_i = n - 1$ and $f(x) = 16 \cdot 3^{x-2} - 2^x$ for $x > 1$ and $f(1) = 2$. However, the simple inequality $f(x) \geq 2^x$ holds, with equality if and only if $x = 1$. Therefore, the minimum is attained if and only if $n_1 = n_2 = \ldots = 1$, and the minimal value is 2^{n-1}. This finishes the induction and hence the whole argument. □

To conclude this section, we remark that the recursion for $s(T)$ and $t(T, v)$ can easily be translated to the world of generating functions (compare [18])—for instance, if one wants to determine the average solvability number of a rooted

ordered tree, the following functional equations for the generating functions $S(x) = \sum_T s(T)x^{|T|}$ and $T(x) = \sum_T t(T,v)x^{|T|}$ hold:

$$S(x) = \frac{8x}{1-S(x)} - \frac{5x}{1-T(x)},$$
$$T(x) = \frac{8x}{1-S(x)} - \frac{6x}{1-T(x)}.$$

By an asymptotic analysis, it can be followed from these equations that the average solvability number of a rooted ordered tree on n vertices is asymptotically

$$\sqrt{\frac{3567+523\sqrt{41}}{7544}} \cdot \left(\frac{293-41\sqrt{41}}{8}\right)^n.$$

4 Tree-decomposition and tree-width

In this section, we define the terms tree-width and the tree-decomposition and state some well known properties that are used in this paper. These concepts were originally introduced by Robertson and Seymour [16] in connection with graph minor theory. However, it has been shown that that many $\mathcal{NP}$-complete problems like independent set, Hamiltonian circuit, vertex cover and Steiner trees can be solved in polynomial time on graphs with bounded tree-width (see [5]). Thus, this graph class is also interesting from an algorithmical point of view. Recently, Gassner and Hatzl [11] developed a linear time algorithm for a domination problem with parity constraints on graphs with bounded tree-width. In this paper, similar ideas are used in order to find the number of solvable instances for graphs of bounded tree-width in linear time. However, as for many other algorithms that solve problems on graphs with bounded tree-width the constant factor hidden in the $\mathcal{O}$-notation turns out to be very large. Thus, the algorithm discussed here does not lead to a reasonable algorithm from a practical point of view and a lot of additional ideas are required in order to speed up the running time.

Extensive surveys on tree-width can be found in the papers of Bodlaender [5] and Kloks [14]. Bodlaender [6] also developed a linear time algorithm which determines, for a given graph $G = (V, E)$, whether the tree-width of G is at most k (for fixed k), and if so, outputs a tree-decomposition of G with tree-width at most k. However, the computation of the tree-width is $\mathcal{NP}$-hard for general graphs (see [7]). Examples for graphs that have bounded tree-width are trees (tree-width 1), series-parallel graphs (tree-width 2), outerplanar graphs (tree-width 2), cactus graphs (tree-width 2) and Halin graphs (tree-width 3).

Definition 15. A tree-decomposition of a graph $G = (V(G), E(G))$ is a pair $(T, \mathcal{X})$ consisting of a tree $T = (V(T), E(T))$ and a family of vertex sets, i.e., $\mathcal{X} = \{X_t : X_t \subseteq V(G), t \in V(T)\}$ such that

(T1) $V(G) = \bigcup_{t \in V(T)} X_t$;
(T2) for every edge $(u, v) \in E(G)$ there exists a vertex $t \in V(T)$ such that $u \in X_t$ and $v \in X_t$;
(T3) if $t_2 \in V(T)$ is on the path between t_1 and t_3 in T then $X_{t_1} \cap X_{t_3} \subseteq X_{t_2}$.

Properties (T1) and (T2) imply that the given graph G is the union of the subgraphs that are induced by the vertex sets X_t. Moreover, property (T3) is equivalent to the fact that for each vertex $v \in V$ the sets X_t containing v induce a subtree of T. An example of a tree-decomposition is given in Figure 3(a) and 3(b).

Note that there are many different tree-decompositions for a given graph, e.g., it is always possible that the tree $T = (V(T), E(T))$ consists of only a single vertex, $E(T) = \emptyset$ and $\mathcal{X} = \{V(G)\}$. However, we are interested in tree-decompositions for which the cardinalities of the sets X_t are small. It is easy to see that if G contains a clique of size k there is a set X_t in the tree-decomposition of G with $|X_t| \geq k$, because there has to exist a set containing all vertices of the clique. On the other hand, if G is a tree, there exists a tree-decomposition in such a way that all sets in $\mathcal{X}$ have cardinality at most two.
The following definition of tree-width makes these ideas more formal.

Definition 16. The width of a tree-decomposition $(T, \mathcal{X})$ of a graph $G = (V(G), E(G))$ is the number

$$\max\{|X_t| - 1 : t \in V(T)\},$$

and the tree-width $\mathrm{tw}(G)$ of G is the least width of any tree-decomposition of G.

Using the comments above, it follows that the tree-width of any tree is 1 and that $\mathrm{tw}(K_n) = n - 1$, where K_n is the complete graph with n vertices. Thus, in some sense the tree-width measures how tree-like a given graph is.
Another important property of a tree-decomposition which is essential for many algorithms is the so-called separator property. This follows almost directly from (T3) and a formal proof can for example be found in Diestel's book [8].

Lemma 17. *Let $e = (t_1, t_2)$ be an edge in T and T_1, T_2 be the components of $T \setminus e$ with $t_1 \in T_1$ and $t_2 \in T_2$. Then $X_{t_1} \cap X_{t_2}$ separates the vertex set $\bigcup_{t \in T_1} X_t$ from $\bigcup_{t \in T_2} X_t$.*

It is useful to introduce so called *nice tree-decompositions*, which have a particular structure, that helps to keep notation simple. A tree-decomposition $(T, \mathcal{X})$ is called nice if it satisfies the following properties:

- T is a rooted binary tree with root vertex r.
- The vertices of T belong to one of the following four types:

(1) *Leaf vertex* i is a leaf of T and $|X_i| = 1$;
(2) *Introduce vertex* i has one child j and $X_i = X_j \cup \{v\}$ for some vertex $v \in V(G)$;
(3) *Forget vertex* i has one child j and $X_i = X_j \backslash \{v\}$ for some vertex $v \in V(G)$;
(4) *Join vertex* i has two children j and l with $X_i = X_j = X_l$.

The nice tree-decomposition is not necessary for the proposed algorithm, but enables us to simplify the description of the algorithm. In [14] it is shown that if a graph G has tree-width k then there exists a tree-decomposition $(T, \mathcal{X})$ with $|V(T)| = \mathcal{O}(n)$. Moreover, given such a tree-decomposition $(T, \mathcal{X})$ a nice tree-decomposition of G that has also $\mathcal{O}(n)$ vertices and the same width k can be found in $\mathcal{O}(n)$ time. This result ensures that it suffices to develop a linear time algorithm for a given nice tree-decomposition.
A nice tree-decomposition is given in Figure 3(c).

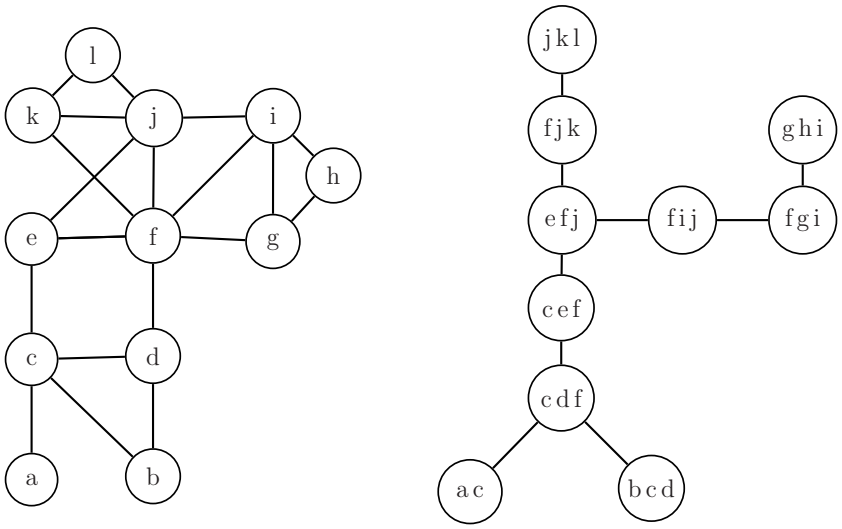
5 Calculating the solvability number of graphs with bounded tree-width

Many problems on graphs with bounded tree-width can be solved in polynomial time using a dynamic programming approach. Indeed, $s(G)$ can also be obtained using this technique. In order to describe our approach in more detail some notation is introduced. Let G be a graph and $(T, \mathcal{X})$ its nice tree-decomposition with tree-width k and root vertex r. A subtree of T rooted at a vertex t is denoted by $T(t)$. Furthermore, the subgraph

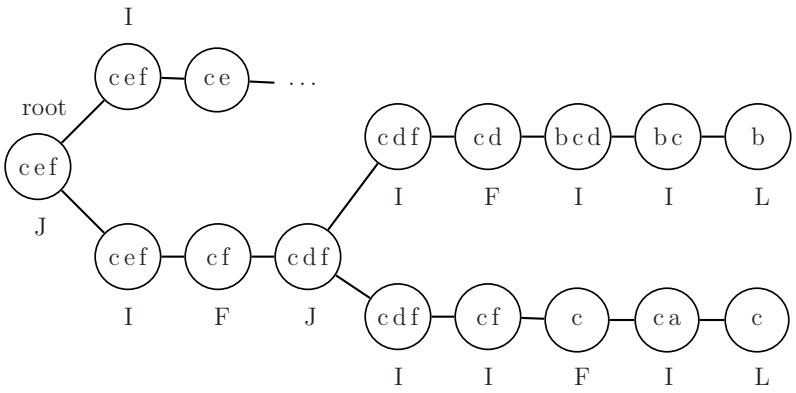
$$G_t = (V_t, E_t) = G\left[\bigcup_{j \in T(t)} X_j\right]$$

is assigned to $t \in V(T)$, where $G[U]$ denotes the induced subgraph of U. Obviously, $G_r = G$ holds. In the following, we will count the number of solvable instances for each subgraph G_t. This is done in a bottom-up manner from the leaves of the tree to the root r. At the end the number of solvable instances of G_r corresponds to $s(G)$.

Suppose we consider an instance given by the vectors $a, b \in \{0,1\}^{|V(G)|}$. Moreover, a subset $S \subseteq V_t$ for some $t \in V(T)$ is given which should be extended to a feasible solution by adding some vertices from the set $V(G) \setminus V_t$. Then using



(a) A graph G with tree-width 2. (b) A tree-decomposition of G with width 2.



(c) A part of a nice tree-decomposition of G with the different vertex types.

Fig. 3. An example for a graph, its tree-decomposition and a nice tree-decomposition.

the separator property all vertices $v \in V_t \setminus X_t$ have to satisfy the parity constraints. Otherwise, we cannot extend S to a feasible solution by just adding vertices from the set $V(G) \setminus V_t$. However, a vertex $v \in X_t$ may violate its parity constraint with respect to S because the extension of S may include vertices of $V \setminus V_t$ that are adjacent to v. Thus, the following definition seems to be appropriate for a dynamic programming approach. In the following definition and thereafter $\mathcal{P}(U)$ denotes the power set of U.

Definition 18. Let $G = (V, E)$ be a graph, $U \subseteq V$ be a subset of the vertices and $\bar{a} \in \{0,1\}^{|U|}$ and $\bar{b} \in \{0,1\}^{|U|}$ be vectors. Furthermore, let f be a function that maps a set $A \subseteq U$ to a subset of $\mathcal{P}(U)$, i.e., $f : \mathcal{P}(U) \to \mathcal{P}(\mathcal{P}(U))$ and if $A \subseteq U$ then $f(A) = \{A_1, \ldots, A_k\}$ for some pairwise different $A_i \subseteq U$. An

instance given by $a \in \{0,1\}^{|V(G)|}$ and $b \in \{0,1\}^{|V|}$ is called $(\bar{a}, \bar{b}, f)$-feasible if and only if $a_u = \bar{a}_u$ and $b_u = \bar{b}_u$ hold for all $u \in U$ and the following two conditions are satisfied for all $A \subseteq U$:

(1) For all $A_i \in f(A)$ there exists a set $S \subseteq V$ with $S \cap U = A_i$ satisfying the following properties:

$$|N(v) \cap S| \equiv b_v \mod 2 \quad \forall v \in A : a_v = 0, \tag{7}$$
$$|N[v] \cap S| \equiv b_v \mod 2 \quad \forall v \in A : a_v = 1, \tag{8}$$
$$|N(v) \cap S| \not\equiv b_v \mod 2 \quad \forall v \in U \setminus A : a_v = 0, \tag{9}$$
$$|N[v] \cap S| \not\equiv b_v \mod 2 \quad \forall v \in U \setminus A : a_v = 1, \tag{10}$$
$$|N(v) \cap S| \equiv b_v \mod 2 \quad \forall v \in V \setminus U : a_v = 0, \tag{11}$$
$$|N[v] \cap S| \equiv b_v \mod 2 \quad \forall v \in V \setminus U : a_v = 1. \tag{12}$$

(2) For all $B \subseteq U$ that are not in $f(A)$ there does not exist such a set $S \subseteq V$.

Let us denote the set of functions $\{f \mid f : \mathcal{P}(U) \to \mathcal{P}(\mathcal{P}(U))\}$ by $\mathcal{F}$. It follows from the definition that each instance is $(\bar{a}, \bar{b}, f)$-feasible for one triple $(\bar{a}, \bar{b}, f)$, where $\bar{a}, \bar{b} \in \{0,1\}^{|U|}$ and $f \in \mathcal{F}$. The idea of the algorithm is to compute the number of instances that are $(\bar{a}, \bar{b}, f)$-feasible for all triples $(\bar{a}, \bar{b}, f)$. We will use $s_U(\bar{a}, \bar{b}, f)$ for the number of instances that are $(\bar{a}, \bar{b}, f)$-feasible. Once these numbers are available, the number of solvable instances can easily be computed. In fact, it suffices to check all triples $(\bar{a}, \bar{b}, f)$ where f has the property that $f(U) \neq \emptyset$. Obviously, there is a huge number of triples that have to be stored, but in fact a lot of functions f will imply that $s_U(\bar{a}, \bar{b}, f) = 0$. Some situations where this is the case are covered in Lemma 11. The following lemma states even more triples $(\bar{a}, \bar{b}, f)$ for which $s_U(\bar{a}, \bar{b}, f) = 0$ follows immediately.

Lemma 19. *Let $G = (V, E)$ be a graph with $v \in V$ and $N[v] \subseteq U$ for some $U \subseteq V$. Furthermore, vectors $\bar{a}, \bar{b} \in \{0,1\}^{|U|}$ are given. Then, $s_U(\bar{a}, \bar{b}, f) = 0$ for any function $f : \mathcal{P}(U) \to \mathcal{P}(\mathcal{P}(U))$ satisfying one of the following conditions:*

(1) $\bar{a}_v = 1$ and there exists a set $A \subseteq U$ with $v \in A$, $A_1 \in f(A)$ and $|A_1 \cap N[v]| \not\equiv \bar{b}_v \mod 2$.
(2) $\bar{a}_v = 1$ and there exists a set $A \subseteq U$ with $v \notin A$, $A_1 \in f(A)$ and $|A_1 \cap N[v]| \equiv \bar{b}_v \mod 2$.
(3) $\bar{a}_v = 0$ and there exists a set $A \subseteq U$ with $v \in A$, $A_1 \in f(A)$ and $|A_1 \cap N(v)| \not\equiv \bar{b}_v \mod 2$.
(4) $\bar{a}_v = 0$ and there exists a set $A \subseteq U$ with $v \notin A$, $A_1 \in f(A)$ and $|A_1 \cap N(v)| \equiv \bar{b}_v \mod 2$.

Proof. Let f be a function satisfying the first condition. Due to the fact that $N[v] \subseteq U$ it can be concluded that $|N[v] \cap S| = |N[v] \cap S \cap U|$ holds for any

subset $S \subseteq V$. However, $S \cap U = A_1$ and $|N[v] \cap A_1| \equiv b_v \mod 2$ have to be satisfied in order to have an $(\bar{a}, \bar{b}, f)$-feasible solution. This contradicts (8) in Definition 18 and $s_U(\bar{a}, \bar{b}, f) = 0$ follows.
The cases where f is a function satisfying one of the other conditions in the lemma can be done in an analogous way. □

The next lemma states in more detail what happens if a single vertex v is added to a graph G. We assume that $a_v = 1$ and $b_v = 0$, but it is straightforward to give similar results for the remaining three cases. Note that this is exactly what we need in order to deal with introduce vertices in a nice tree-decomposition. Here, the set U' in the next lemma corresponds to a vertex X_i of the tree-decomposition and the set U corresponds to the child X_j of X_i, i.e., $X_i = X_j \cup \{v\}$. Note that by the definition of a tree-decomposition $N(v) \subseteq X_i$ holds in G_i.

Lemma 20. *Let $G = (V, E)$ and $G' = (V', E')$ be two graphs such that $V' = V \cup v$, $v \notin V$, and G is an induced subgraph of G'. Assume that $U \subseteq V$ with $N(v) \subseteq U$ and let us denote $U' = U \cup v$. For a given function $f : \mathcal{P}(U) \to \mathcal{P}(\mathcal{P}(U))$ we define a function $f' : \mathcal{P}(U') \to \mathcal{P}(\mathcal{P}(U'))$ as follows:*

(1) If $A_i \in f(A)$ for some $A \subseteq U$ and $|A_i \cap N(v)| \equiv 0 \mod 2$, then $A_i \in f'(A \cup v)$ and $A_i \cup \{v\} \in f'((A \setminus N(v)) \cup ((U \setminus A) \cap N(v)))$.
(2) If $A_i \in f(A)$ for some $A \subseteq U$ and $|A_i \cap N(v)| \equiv 1 \mod 2$, then $A_i \in f'(A)$ and $A_i \cup \{v\} \in f'(v \cup (A \setminus N(v)) \cup ((U \setminus A) \cap N(v)))$.
(3) All sets $A'_i \subseteq U'$ with $A'_i \in f'(A')$ for some $A' \subseteq U'$ are covered by the cases 1 and 2 above.

Let $\bar{a}, \bar{b} \in \{0,1\}^{|U|}$ and $\bar{a}', \bar{b}' \in \{0,1\}^{|U'|}$ be given, such that $\bar{a}_u = \bar{a}'_u$ and $\bar{b}_u = \bar{b}'_u$ hold for all $u \in U$ and $\bar{a}'_v = 1$ and $\bar{b}'_v = 0$.
If an instance in G given by $a, b \in \{0,1\}^{|V|}$ is $(\bar{a}, \bar{b}, f)$-feasible, then the instance in G' given by $a', b' \in \{0,1\}^{|V|+1}$ with $a_w = a'_w$ and $b_w = b'_w$ for all $w \in V$ and $a'_v = 1$ and $b'_v = 0$ is $(\bar{a}', \bar{b}', f')$-feasible.

Proof. In order to show that the instance $a', b' \in \{0,1\}^{|V|+1}$ is $(\bar{a}', \bar{b}', f')$-feasible assume that A'_i and A' are subsets of U' such that $A'_i \in f'(A')$. In the following, we give sets $S' \subseteq V'$ such that (7) — (12) are satisfied for $U' \subseteq V'$. Let us consider four different cases:

(1) $v \notin A'_i$ and $v \notin A'$
In this case we know that $A'_i \in f(A')$ which implies that there exists a set $S \subseteq V$ such that (7) — (12) are satisfied in G. Moreover, $|A'_i \cap N(v)| \equiv 1 \mod 2$ holds. Thus, setting $S' = S$ it can be concluded that (7) — (12) hold in G' because $v \notin S$ and $|A_i \cap N(v)| \equiv 1 \not\equiv b_v \mod 2$. Hence, for all $A'_i \in f'(A')$ with $v \notin A'_i$ and $v \notin A'$ there is a set S' such that the first condition in Definition 18 is satisfied.

(2) $v \notin A'_i$ and $v \in A'$
In this case $A'_i \in f(A' \setminus v)$ and $|A'_i \cap N(v)| \equiv 0 \mod 2$ hold. Again, there exists a set $S \subseteq V$ satisfying (7) — (12) in G. Due to the fact that $v \notin A'_i$ it is easy to see that $S' = S$ satisfies (7) — (12) in G'.

(3) $v \in A'_i$ and $v \notin A'$
Here, it can be concluded that A' equals $(A \setminus N(v)) \cup ((U \setminus A) \cap N(v))$ for some $A \subseteq U$ with $A'_i \setminus v \in f(A)$ and $|A_i \cap N(v)| \equiv 0 \mod 2$. Let us define $S' = S \cup v$, where S is again the set satisfying (7) — (12) in G. Due to the fact that $N(v) \subseteq U$ we only have to check equations (7) — (10) for all $u \in N[v]$. Let us start with the vertex v itself. It is easy to see that $|N[v] \cap S'| = |\{v\}| + |N(v) \cap S| = |\{v\}| + |N(v) \cap A_i| \equiv 1 \neq b_v \mod 2$ holds. Finally, we consider vertices $u \in N(v)$. For those vertices $|N[u] \cap S'| \equiv 1 + |N[u] \cap S| \mod 2$ ($|N(u) \cap S'| \equiv 1 + |N(u) \cap S| \mod 2$) hold. Thus, $u \in A'$ if and only if $u \in U \setminus A$ and S' fulfills (7) — (12).

(4) $v \in A'_i$ and $v \in A'$
This case is similar to the previous one. The set A' can be written as $v \cup (A \setminus N(v)) \cup ((U \setminus A) \cap N(v))$ for some $A \subseteq U$ with $A'_i \setminus v \in f(A)$ and $|A_i \cap N(v)| \equiv 1 \mod 2$ and the same arguments as above can be used.

In order to conclude the proof it remains to show that the second condition of Definition 18 is satisfied. However, this can be done in almost the same way. It is straightforward to prove that if there does not exist a set S in G that satisfies all the equations (7) — (12), then there is no set S' in G' for which all the equations are met. A formal proof is omitted. □

Theorem 21. *The number $s(G)$ can be computed in linear time for graphs G with bounded tree-width.*

Proof. Using a dynamic programming approach from the leaves to the root leads to a linear time algorithm. To see this note that for a leaf vertex $|X_m| = 1$ holds and it can be decided in constant time how many instances are (a, b, f)-solvable for all triples. Lemma 20 shows how a join vertex has to be handled. In fact, we have to add one new vertex v and consider all four possibilities, i.e., $a_v \in \{0, 1\}$ and $b_v \in \{0, 1\}$. Again, all the necessary computations can be done in constant time assuming that the size of each set X_m in the tree-decomposition is bounded by a constant.
Let us now discuss join vertices now. Assume that X_m is a join vertex with children X_j and X_l. Furthermore, $s_{X_j}(\bar{a}, \bar{b}, f)$ and $s_{X_l}(\bar{a}, \bar{b}, f)$ are known in the graphs G_j and G_l respectively for all triples $(\bar{a}, \bar{b}, f)$. The task is to compute $s_{X_m}(\bar{a}, \bar{b}, f)$ in the graph G_m. Note that the vertex set X_m separates V_j from V_l in G_m. Using the inclusion-exclusion principle one can conclude that in G_m

$$|N[v] \cap S| = |N[v] \cap S \cap V_j| + |N[v] \cap S \cap V_l| - |N[v] \cap S \cap V_j \cap V_l| \quad (13)$$

holds for all $S \subseteq V_m$ and $v \in V_m$. Due to the definition of a join vertex it

follows immediately that $V_j \cap V_l = X_m$. Moreover, in equation (13), $N[v]$ can also be replaced by $N(v)$ and the equation still remains valid.
Let us assume we are given vectors $a, b \in \{0,1\}^{|V_m|}$; then we have to determine the function $f : \mathcal{P}(X_m) \to \mathcal{P}(\mathcal{P}(X_m))$ for which the instance (a, b) in G_m is $(\bar{a}, \bar{b}, f)$-solvable, where $\bar{a}_v = a_v$ and $\bar{b}_v = b_v$ for all $v \in V_m$. We define the vectors $a', b' \in \{0,1\}^{|V_j|}$ and $a'', b'' \in \{0,1\}^{|V_l|}$, where $a'_v = a_v$ and $b'_v = b_v$ for all $v \in V_j$ and $a''_v = a_v$ and $b''_v = b_v$ for all $v \in V_l$ and assume that the instance in G_j given by a', b' is $(\bar{a}, \bar{b}, f')$-solvable for some $f' : \mathcal{P}(X_j) \to \mathcal{P}(\mathcal{P}(X_j))$. Similarly, assume that the instance a'', b'' in G_l is $(\bar{a}, \bar{b}, f'')$-solvable for some $f'' : \mathcal{P}(X_l) \to \mathcal{P}(\mathcal{P}(X_l))$. If $A_i \in f'(A')$ and $A_i \in f''(A'')$ for some $A_i, A', A'' \subseteq X_i$, then there exist sets S' and S'' satisfying (11) and (12) in G_j and G_l respectively. Moreover, $S' \cap S'' = A_i$ and $(S' \cup S'') \cap X_m = A_i$. Let us define $S := S' \cup S''$; then it follows from the separator property that $|N[v] \cap S| \equiv b_v$ for all $v \in V_m \setminus X_m$ with $a_v = 1$ and $|N(v) \cap S| \equiv b_v$ for all $v \in V_i \setminus X_i$ with $a_v = 0$. Moreover, $|N[v] \cap S \cap X_i|$ and $|N(v) \cap S \cap X_i|$ can be calculated for all $v \in X_i$. Due to the fact that $A_i \in f'(A')$, the parities of $|N[v] \cap S' \cap V_j| = |N[v] \cap S \cap V_j|$ and $|N(v) \cap S' \cap V_j| = |N(v) \cap S \cap V_j|$ can be determined for all $v \in X_m$. The same can be done for $|N[v] \cap S \cap V_l|$ and $|N(v) \cap S \cap V_l|$ using f''. Using all this information, the parity of $|N[v] \cap S|$ and $|N(v) \cap S|$ can be obtained for all $v \in X_m$ making use of equation (13). If the set $A \subseteq X_m$ contains exactly those vertices v for which the parity $|N[v] \cap S|$, respectively $|N(v) \cap S|$, equals b_v, then it can be concluded that $A_i \in f(A)$.
This procedure has to be repeated for all $A', A'' \subseteq X_m$ with $A_i \in f'(A')$ and $A_i \in f''(A'')$, and it leads to all sets $A \subseteq X_m$ with $A_i \in f(A)$. Using the same idea for all sets $A_i \subseteq X_m$ with $A_i \in f'(A')$ and $A_i \in f''(A'')$ for some A' and A'', the function f can be completely determined. Obviously, everything can be done in constant time if the cardinality of X_m is bounded by a constant.
Finally, we have to deal with forget vertices. Assume that X_m is a forget vertex in the nice tree-decomposition, i.e., it has one child X_j and $X_m = X_j \setminus v$, and that an instance in G_j given by $a, b \in \{0,1\}^{|V_j|}$ is $(\bar{a}, \bar{b}, f)$-solvable for some $\bar{a}, \bar{b} \in \{0,1\}^{|X_j|}$ and for some $f : \mathcal{P}(X_j) \to \mathcal{P}(\mathcal{P}(X_j))$. Note that due to the separator property v is not adjacent to any vertex $u \in V(G) \setminus V_m$ in G. We have to make sure that the parity constraint is fulfilled for v. Consequently, we are only interested in $A_i \in f(A)$ where $v \in A$. In fact, we define a function f' in the following way: Suppose $v \in A$ for some $A \subseteq X_j$ and $A_i \in f(A)$, then $A_i \setminus v \in f'(A \setminus v)$ if $v \in A_i$ and $A_i \in f'(A \setminus v)$ otherwise. It is easy to check that if an instance $a, b \in \{0,1\}^{|V_j|}$ is $(\bar{a}, \bar{b}, f)$-solvable, then $a, b \in \{0,1\}^{|V_m|}$ is $(\bar{a}', \bar{b}', f')$-solvable if $\bar{a}'_u = \bar{a}_u$ and $\bar{b}'_u = \bar{b}_u$ for all $u \in X_m$. This manipulation can again be done in constant time if $|X_m|$ and $|X_j|$ are bounded by a constant.
This shows that each step in the dynamic programming approach can be done in constant time. Due to the fact that there exists a nice tree-decomposition with $\mathcal{O}(n)$ vertices the running time is linear. This completes the proof. □

Basically, the algorithm generalizes the approach that has been given in Section 2 for determining the solvability number of special graphs. In view of the high degree of symmetry, a lot of simplifications were possible in these cases. Finally, we remark that the discussed algorithm immediately implies the following corollary:

Corollary 22. *Let $G = (V, E)$ be a graph with adjacency matrix A and $a \in \{0,1\}^{|V|}$ a vector. If G has bounded treewidth, then the binary rank of $A + \mathrm{diag}(a)$ can be computed in linear time.*

6 Random graphs

In this section, we will show that the expected value of the solvability number $s(G)$ is of order 4^n for a random graph in $\mathcal{G}(n, \frac{1}{2})$ (i.e., each edge is inserted with probability $\frac{1}{2}$), so that the "typical" value of $s(G)$ is pretty close to its maximum. In particular, the following theorem holds:

Theorem 23. *Let $G \in \mathcal{G}(n, \frac{1}{2})$ be a random graph with vertex set $[n] = \{1, 2, \ldots, n\}$. Then the inequality*

$$\mathbb{E}(s(G)) > \tfrac{1}{2} \cdot 4^n$$

holds.

Proof. The proof of this theorem essentially follows the approach of Amin, Clark and Slater [1]. First of all, fix a vector $a \in \{0,1\}^n$, and let A denote the (random) adjacency matrix of G. Note that the number of solutions of the matrix equation $(A + \mathrm{diag}(a))x = 0$ over $\mathbb{F}_2$ is exactly $X = 2^{n - \mathrm{rk}(A + \mathrm{diag}(a))}$. We will calculate the expected value of this random variable rather than that of $2^n X^{-1}$ which we are actually interested in. To this end, we determine the probability that a set S is a solution for the instance $(a, 0)$. We write p_S for this probability and obtain

$$\mathbb{E}(X) = \sum_{S \subseteq [n]} p_S.$$

Let s_1, s_2 be the number of vertices $v \in S$ such that $a_v = 1$ and $a_v = 0$ respectively, and let S_1, S_2 be the corresponding sets. Furthermore, set $s = |S|$. S can only be a solution if

- every vertex in S_1 has an odd number of neighbors in S,
- every vertex in S_2 has an even number of neighbors in S,
- every vertex in $[n] \setminus S$ has an even number of neighbors in S.

If G_S is the restriction of G on S, the first two statements are equivalent to the property that S_1 is the set of vertices of odd degree in G_S. By the following lemma, the probability for this is $2^{\binom{s-1}{2}}/2^{\binom{s}{2}} = 2^{1-s}$ if $S \neq \emptyset$ and if s_1 is even:

Lemma 24 (Read and Robinson [15]). *Let $U \subseteq V$ with $|V| = n$. If $|U|$ is even, the number of simple graphs on V where U is the set of vertices having odd degree is $2^{\binom{n-1}{2}}$.*

Furthermore, the probability that the number of neighbors in S is even equals $\frac{1}{2}$ for every vertex in $[n] \setminus S$, as long as $S \neq \emptyset$. By independence, we thus have $p_S = 2^{1-s}2^{-(n-s)} = 2^{1-n}$.

$S = \emptyset$ is always a solution, so $p_S = 1$ in this case. So if k is the number of 1's in a, we obtain

$$\mathbb{E}(X) = \sum_{\substack{s_1=0 \\ s_1 \text{ even}}}^{k} \binom{k}{s_1} \sum_{s_2=0}^{n-k} \binom{n-k}{s_2} 2^{1-n} + (1 - 2^{1-n}) = \begin{cases} 2 - 2^{1-n} & k \neq 0, \\ 3 - 2^{1-n} & k = 0. \end{cases}$$

By Jensen's inequality,

$$2^n \, \mathbb{E}(X^{-1}) \geq \begin{cases} \frac{2^n}{2-2^{1-n}} & k \neq 0, \\ \frac{2^n}{3-2^{1-n}} & k = 0. \end{cases}$$

Summing over all vectors a finally shows that

$$\mathbb{E}(s(G)) \geq \frac{2^n(2^n-1)}{2-2^{1-n}} + \frac{2^n}{3-2^{1-n}} > \frac{1}{2} \cdot 4^n.$$

□

References

[1] A. T. Amin, L. H. Clark, and P. J. Slater. Parity dimension for graphs. *Discrete Math.*, 187(1-3):1–17, 1998.

[2] A. T. Amin and P. J. Slater. Neighborhood domination with parity restrictions in graphs. In *Proceedings of the Twenty-third Southeastern International Conference on Combinatorics, Graph Theory, and Computing (Boca Raton, FL, 1992)*, volume 91, pages 19–30, 1992.

[3] A. T. Amin and P. J. Slater. All parity realizable trees. *J. Combin. Math. Combin. Comput.*, 20:53–63, 1996.

[4] A. T. Amin, P. J. Slater, and G.-H. Zhang. Parity dimension for graphs—a linear algebraic approach. *Linear Multilinear Algebra*, 50(4):327–342, 2002.

[5] H. L. Bodlaender. A tourist guide through treewidth. *Acta Cybernet.*, 11(1-2):1–21, 1993.

[6] H. L. Bodlaender. A linear-time algorithm for finding tree-decompositions of small treewidth. *SIAM J. Comput.*, 25(6):1305–1317, 1996.

[7] A. Brandstädt, V. B. Le, and J. P. Spinrad. *Graph classes: a survey*. SIAM Monographs on Discrete Mathematics and Applications. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1999.

[8] R. Diestel. *Graph theory*, volume 173 of *Graduate Texts in Mathematics*. Springer-Verlag, Berlin, third edition, 2005.

[9] J. F. Fink and M. S. Jacobson. n-domination in graphs. In *Graph theory with applications to algorithms and computer science (Kalamazoo, Mich., 1984)*, Wiley-Intersci. Publ., pages 283–300. Wiley, New York, 1985.

[10] J. F. Fink and M. S. Jacobson. On n-domination, n-dependence and forbidden subgraphs. In *Graph theory with applications to algorithms and computer science (Kalamazoo, Mich., 1984)*, Wiley-Intersci. Publ., pages 301–311. Wiley, New York, 1985.

[11] E. Gassner and J. Hatzl. A parity domination problem in graphs with bounded tree width and distance-hereditary graphs. 2006. Submitted.

[12] F. Harary and T. W. Haynes. The k-tuple domatic number of a graph. *Math. Slovaca*, 48(2):161–166, 1998.

[13] T. W. Haynes, S. T. Hedetniemi, and P. J. Slater. *Fundamentals of domination in graphs*, volume 208 of *Monographs and Textbooks in Pure and Applied Mathematics*. Marcel Dekker Inc., New York, 1998.

[14] T. Kloks. *Treewidth, Computation and Approximation*. Berlin : Springer-Verlag, 1994.

[15] R. C. Read and R. W. Robinson. Enumeration of labelled multigraphs by degree parities. *Discrete Math.*, 42(1):99–105, 1982.

[16] N. Robertson and P. D. Seymour. Graph minors. II. Algorithmic aspects of tree-width. *J. Algorithms*, 7(3):309–322, 1986.

[17] K. Sutner. Linear cellular automata and the Garden-of-Eden. *Math. Intelligencer*, 11(2):49–53, 1989.

[18] S. Wagner. Counting all parity realizable trees. 2007. Submitted.