Sapienza Università di Roma

Dipartimento di Scienze Statistiche

Dottorato di Ricerca in Ricerca Operativa
XXIV ciclo

The Stable Set Problem: Some Structural Properties and Relaxations

Ph.D. Candidate:
Carla Michini

Coordinator:
Prof. Stefano Lucidi

Advisor:
Prof. Antonio Sassano

Rome, March 2012

Contents

Acknowledgements

I would like to thank my advisor, Prof. Antonio Sassano, for sharing with me his knowledge and for supporting me with his fatherly guidance. He has constantly passed on his enthusiasm to me, providing me with challenging opportunities for my scientific growth. From him, I have learnt that the passion for research should be fed, above all, with the curiosity of answering new questions, rather than with the pretension of establishing definitive conclusions.

I am also particularly obliged to Prof. Gérard Cornuéjols, who welcomed me in his research group at Tepper School of Business during my stay at CMU as a visiting scholar. He has offered me the chance to work on very exciting research topics, helping and following me continuously. I feel very honoured of having worked under his supervision.

Keywords

Stable set problem

Edge formulation

Simplex pivot

Vertex adjacency

Combinatorial diameter

Corner relaxation

Fixing

List of Figures

Preface

A stable set in a graph $G(V, E)$ is a subset S of the node set V such that no two nodes of S are adjacent. The *stable set polytope* $STAB(G)$ is the convex hull of the incidence vectors of stable sets in G. Suppose that G has node weights $c \in \mathbb{R}_+^{|V|}$. The *Maximum Weight Stable Set Problem* asks for the stable set of maximum weight and is a NP-hard combinatorial optimization problem.

A natural formulation of the stable set problem is the so-called *edge formulation* defined by $|E|$ two-variable constraints, expressing the simple condition that two adjacent nodes cannot belong to a stable set. The polytope defined by the linear relaxation of the edge formulation is the *fractional stable set polytope*, denoted in the following by $FSTAB(G)$. The edge formulation yields a very weak approximation of the stable set polytope, as $FSTAB(G)$ coincides with $STAB(G)$ only in the case of bipartite graphs. Neverthless, the simpler geometrical structure of $FSTAB(G)$ provides deep theoretical insights as well as interesting algorithmic opportunities. For instance, it is a well-known result that vertices of $FSTAB(G)$ must be $(0, \frac{1}{2}, 1)$-valued (Balinski, 1969). Furthermore, Nemhauser and Trotter proved that variables assuming binary values in an optimal solution to the LP-relaxation of the edge formulation retain the same values in some optimal solution of the original (integer) problem. The main purpose of this thesis consists in the definition of some additional structural properties of the fractional stable set polytope, to be exploited for solving efficiently instances of the Maximum Weight Stable Set Problem.

Instrumental to this goal is a graphic characterization of basic solutions of $FSTAB(G)$, based on a result of Campelo and Cornuéjols. To each basic solution x_B we associate a *basic subgraph* G_B, that is a subgraph of G whose connected components are rooted trees and 1-trees with an odd cycle. This graphic representation has the property that a component of x_B is fractional if and only if the corresponding node belongs to a 1-tree.

A first topic of the thesis concerns vertex adjacency on the fractional stable set polytope. With regard to adjacency of 0-1 vertices, no information is lost by looking at this relaxed polytope. Indeed, by the *Trubin property*, two 0-1 vertices are adjacent in $STAB(G)$ if and only if they are adjacent in $FSTAB(G)$. Exploiting the graphic characterization of bases, adjacency of bases is redefined in terms of simple graphic operations (corresponding to simplex pivots), that turn a given basis into an adjacent one. Between all possible pivots, we characterize degenerate and non-degenerate ones, and we differentiate those leading to an integer or to a fractional vertex. The graphic characterization of bases is also crucial to prove another structural property of the fractional stable set polytope, concerning the adjacency of its vertices. In particular, we extend a necessary and sufficient condition due to Chvátal for adjacency of (integer) vertices of the stable set polytope to arbitrary (and possibly fractional) vertices of the fractional stable set polytope. These results lead us to prove that the Hirsch Conjecture is true for the fractional stable set polytope, i.e. the combinatorial diameter of this fractional polytope is at most equal to the number of edges of the given graph. We actually refine this bound in the non-bipartite case, by proving a tighter bound, namely $|V|$. We finally design a *simplex-like* algorithm for the Maximum Weight Stable Set Problem, that relies on the adjacency properties outlined above. Primal algorithms applicable to stable set problem had already been developed by Balas and Padberg (for the set partitioning problem), by Nemhauser and Ikura (in the bipartite case) and by Firla et al. (for general 0-1 linear programs). Our algorithm, which is also primal, exploits the adjacency properties of the fractional stable set polytope to generate only integer solutions without using cutting plane methods. Preliminary computational results are encouraging but show that the main drawback of the algorithm consists in the occurence of cycling, due to the high degree of degeneracy of the polytope. Despite this, our approach seems promising, as it opens to the perspective of an exact combinatorial method of solution for the stable set problem, provided that an anti-cycling rule is embedded in the current design of the algorithm.

The second topic of the dissertation is the analysis of the strength of different corner relaxations for the edge formulation of the *Maximum Cardinality Stable Set Problem*. The corner relaxation is a central concept in cutting plane theory, as most general purpose cutting planes are valid for the corner relaxation of a mixed-integer linear program (MILP). Given a basis B of $FSTAB(G)$, the corner relaxation is the convex hull of the integer points of the problem obtained from the MILP by dropping non-negativity on the basic variables. For the edge formulation of the stable set problem, Campelo and Cornuéjols provided a full description of the corner polyhedron associated to a given basis, proving in addition that the split and the Chvátal closures coincide and can be obtained by intersecting the corner polyhedra over all the (feasible) bases.

In a paper concerning the facial structure of the set packing polyhedron, Padberg already observed that "the fractional vertex from which we generate the group problem and hence the Gomory cuts is generally degenerate, in the sense that part of the tight constraints have zero basic slacks. These constraints are not used in the group-theoretic approach though the shape of the cone defined by the fractional vertex and the (feasible) edges emanating from it may depend critically (and obviously, does so in the case of the node-covering problem) on this set of constraints. This indicates a possible direction in which to extend the group theoretic approach". Recently, Fischetti and Monaci provided an empirical assessment of the strength of the corner and other related relaxations on benchmark problems. We followed the line of research indicated by Padberg and we validated with theoretical arguments the empirical results obtained by Fischetti and Monaci. Our main contribution to this issue consists in a tight analysis of the bounds given by the corner relaxation and three of its extensions in the special case of the edge formulation of the stable set problem, for which a full description of the corner polyhedron is available. Our theoretical analysis confirms the intuition of Padberg, showing that degeneracy plays a major role, as the difference in the bounds given by corner relaxations from two different optimal bases can be significantly large. Therefore, exploiting multiple degenerate bases for cut generation could give better bounds than working with just a single basis.

Finally, a concave reformulation for Set Covering problems is presented, where integrality constraints are dropped and the original linear objective function is replaced by a concave one, penalizing fractional values. For such reformulation, any integer local optimum corresponds to a heuristic solution of the original problem. To determine local optima of our concave reformulation, we apply the Frank-Wolfe algorithm with a multistart approach. The choice of a suitable parametric concave function allows us to regulate the smoothness of the objective function and to achieve sparseness of the local optimum. When applied to the edge formulation of the stable set problem, additional properties of local optima can be established. Namely, if the parameter of the objective function belongs to a certain range, binary valued variables of the local optimum can be fixed, allowing a reduction of the dimension of the problem. Computational experiments show that the concave heuristic is effective on some difficult benchmark problems.

In Chapter 1 we introduce the stable set problem and some basic notions of graph theory, polyhedral theory, linear and integer programming. In Chapter 2 we describe a graphic characterization of bases of the fractional stable set polytope, that will lay the foundation of the subsequent results. In Chapter 3 we characterize graphically simplex pivots and we then describe some structural properties concerning the adjacency of vertices on the fractional stable set polytope. We extend Chvátal's condition to arbitrary vertices of our fractional polytope and we prove that the Hirsch conjecture is true in this case. Finally we describe our simplex-like algorithm and present preliminary computational results. In Chapter 4 we study the strength of the corner and other related relaxations in the case of the edge formulation of the stable set problem. In Chapter 5 we propose a concave heuristic for the stable set problem and we present an extension of the well-known fixing theorem of Nemhauser and Trotter. Finally, in Chapter 6, we present the research directions and the perspectives of our future work.

Chapter 1

Introduction to the Stable Set Problem

In this chapter we introduce the *stable set problem* and we recall some basic notions that will be useful to derive the main results presented in the next chapters.

In Section 1.1 we introduce some basic notation and concepts of graph theory; in Sections 1.2 we present basic notions of polyhedral theory; in Sections 1.3 and 1.4 we recall concepts of linear and integer programming, respectively. Finally, in Sections 1.5 and 1.6, we introduce the stable set problem and some combinatorial problems related to it, and in Section 1.7 we describe some properties of the edge formulation of the stable set problem, due mainly to Nemhauser and Trotter.

1.1 Graph Theory

In this section we introduce some basic notation and notions of graphs theory. For further readings we refer to [16] and [29].

A *graph* G is a pair $G = (V, E)$, where V is the set of *nodes* of the graph, and E consists of 2-element subsets of V, called *edges* of the graph. Therefore $V \cap E = \emptyset$. Graphs are usually represented graphically by drawing a dot for each node and by joining two of such dots by a line, if the corresponding two nodes define an edge. A graph with node set V is said to be a graph *on* V. The node set and the edge set of a graph G are denoted by $V(G)$ and $E(G)$, respectively. The number of nodes of a graph G is the *order* of the graph. The graph of order 0 is called the *empty graph* and is denoted by $\emptyset$, while we refer as *singleton* or *isolated node* to a graph of order 1.

If $G(V, E)$ is a graph and $e = (v_1, v_2) \in E$, then we say that v_1 and v_2 are the *ends* of e. The ends of an edge are said to be *incident* with the edge, and viceversa. Two nodes which are incident with a common edge are *adjacent*. An edge with identical ends is called a *loop*. A graph is *simple* if it has no loops and if no edges join the same pair of nodes. We will only deal with *finite graphs*, i.e. graphs whose order is finite, that are also simple.

Let $G(V, E)$ be a (non-empty) graph. The set of *neighbours* of a node u in G, denoted by $N(u)$ is defined as $N(u) := \{v \in V : (u, v) \in E\}$. More generally for $U \subseteq V$, we define the neighbours of U as $N(U) := \{v \in V : (u, v) \in E, u \in U, v \in V \setminus U\}$. The *degree* $d(v)$ of a node v is the number of edges incident with v in G. In a simple graph, $d(v) = |N(v)|$. Given $U \subseteq V$, we define the set of edges with both endpoints in U, as $\Gamma(U) := \{(u, v) \in E : u, v \in U\}$.

A *cut* of graph $G(V, E)$ is a partition of its nodes into two disjoint subsets $(U, V \setminus U)$, with $U \subseteq V$. The *cutset* $\delta(U)$ of the cut $(U, V \setminus U)$ is the set of edges *crossing* the cut, i.e. $\delta(U) = \{(u, v) \in E : u \in U, v \in V \setminus U\}$, the set of edges whose endpoints are in different subsets of the partition.

The *union* of two graphs $G(V, E)$ and $G'(V', E')$ is defined as $G \cup G'(V \cup V', E \cup E')$, while their *intersection* is $G \cap G'(V \cap V', E \cap E')$. If $G \cap G' = \emptyset$, then G and G' are *disjoint*. If $V' \subseteq V$ and $E' \subseteq E$, then G' is a *subgraph* of G or, equivalently, G is a *supergraph* of G'. If $V(G') = V(G)$, then we say that G' is a *spanning* subgraph of G. Given $V' \subseteq V$, we define the subgraph of G *induced* by V', denoted by $G[V']$, as the subgraph of G with node set V' and edge set $E' = \Gamma(V')$.

A *walk* in G is a sequence of nodes $\{v_1, v_2, \ldots, v_k\}$, $k \geq 1$, such that $(v_j, v_{j+1}) \in E$ for $j = 1, \ldots, k-1$. A walk is *closed* if $k > 1$ and $v_k = v_1$. A walk without any repeated nodes is a *path*. A closed walk with no repeated nodes other than the first and the last ones, is called a *cycle*. The *length* of the path $\{v_1, v_2, \ldots, v_k\}$ is $k-1$; the length of the cycle $\{v_1, v_2, \ldots, v_k = v_1\}$ is $k-1$. A cycle of length k is *odd* or *even* depending on whether k is odd or even. The distance $d(u, v)$ in G of two nodes u and v is the length of the shortest path (i.e. the path of minimum length) connecting u and v in G. An edge which joins two nodes of a cycle but does not belong to the cycle is a *chord* of the cycle. We say that a cycle is *induced* in G, if it defines an induced subgraph of G which does not contain chords.

A non-empty graph G is called *connected* if any two of its nodes are linked by a path in G. If a graph is not connected, we say that it is *disconnected*. Connection is an equivalence relation on the node set V. Thus, there is a partition of V into non-empty subsets $V_1, \ldots, V_k$ such that two nodes u and v are connected if and only if they both belong to the same set V_i. The subgraphs $G[V_1], \ldots, G[V_k]$ are called the *connected components* of G.

We now introduce some special classes of graphs. A simple graph where each pair of distinct nodes is joined by an edge is called a *complete graph*. A *clique* is a subset of the nodes of the graph, such that every two nodes in the subset are connected by an edge. In other words, the nodes of a clique induce a complete subgraph on G. The *complement graph* $\bar{G}$ of G is a graph on $V(G)$ such that two nodes on $\bar{G}$ are adjacent if and only if they are not adjacent in G. A *bipartite graph* G is a graph such that its node set $V(G)$ can be partitioned into two subsets X and Y, so that each edge has one end in X and one end in Y; such a partition (X, Y) is called a *bipartition* of the graph.

Theorem 1.1.1. *[16] A graph is bipartite if and only if it contains no odd cycle*

An acyclic graph is called *forest*. A connected forest is a *tree*. Therefore, a forest is a graph whose connected components are trees. The nodes of degree 1 of a tree are called *leaves*. Every tree with at least two nodes has a leaf.

Theorem 1.1.2. *Let T be a graph. The following statements are equivalent:*

(i) T is a tree;

(ii) There is a unique path connecting any two nodes of T;

(iii) Removing any edge of T yields a disconnected graph;

(iv) Adding an edge that connects any two non-adjacent nodes of T yields a graph with a unique cycle.

Theorem 1.1.3. *A connected graph with n nodes is a tree if and only if it has $n-1$ edges.*

The *root* r of a tree T is a special node of $V(T)$. When a tree T is *rooted*, a partial order on $V(T)$ is defined, according to the distance of any node of the tree from its root. Let G be a connected graph. A *spanning tree* of G is a spanning subgraph of G that is tree. Every connected graph admits a spanning tree. A spanning tree of a connected graph can be determined by applying the *depth-first search* algorithm, or the *breadth-first search* algorithm.

We define a *pseudoforest* as a graph in which each connected component has at most one cycle. A *1-tree* is a connected psudoforest, that is a connected graph containing exactly one cycle. Thus, the connected components of a pseudoforest are either trees or 1-trees. Removing an edge from the cycle of a 1-tree yields a tree. Conversely, adding an edge connecting two non-adjacent nodes of a tree yields a 1-tree.

We associate to any simple graph $G(V,E)$ a $|E| \times |V|$ *incidence matrix* $A(G) = [a_{ij}]$, where a_{ij} is 1 if edge e_i is incident with node v_j, and 0 otherwise.

A *node coloration* of a graph G is an assignment of positive integers to the nodes of G, so that no two nodes labelled with the same integer are adjacent. If G admits a node coloration of its nodes with n colors, G is said to be *n-colorable*. The smallest integer k such that G admits a k-coloration is called the *chromatic number* of G and is denoted by $\chi(G)$.

A set of edges of G such that no two edges of the set are incident with the same node, is called a *matching*. The size of the largest matching is the *matching number* and is denoted by $\nu(G)$. A matching is said to be *perfect* if it covers all the node set $V(G)$.

A set of edges such that every node of the graph is incident to at least one edge of the set is called *edge cover*. The *edge covering number* is the cardinality of the smallest edge cover in G, denoted by $\rho(G)$.

A set of nodes of G that are pairwise non-adjacent is called *stable set*. The *stability* *number* $\alpha(G)$ is the cardinality of the largest stable set in G.

A set of nodes of G is a *node cover* if each edge of G has at least one endpoint in the set. The cardinality of any smallest node cover of G is the *node covering number* and is denoted by $\tau(G)$.

1.2 Polyhedral theory

Given a matrix $A \in \mathbb{R}^{m \times n}$ and a vector $b \in \mathbb{R}^n$, the solution set of the system of linear inequalities $\{x \in \mathbb{R}^n \colon Ax \leq b\}$ is called a *polyhedron.* A bounded polyhedron is called a *polytope.* Thus, a polyhedron is defined by the intersection of a finite number of halfspaces.

The dimension $\dim(P)$ of a polyhedron P is defined as the dimension of the affine space spanned by P. Precisely, the maximum number of affinely independent points of P is equal to $1 + \dim(P)$.

An inequality $a^T x \leq b_0$ is *valid* with respect to a polyhedron P if $P \subseteq \{x \colon a^T x \leq b_0\}$. If $a^T x \leq b_0$ is a valid inequality of P, the set $F \subseteq P$ given by $F = \{x \in P \colon a^T x = b_0\}$ is a *face* of P. A *facet* is a non-empty face of P of dimension $\dim(P) - 1$.

A polyhedron is *full dimensional* if it has an *interior point*, i.e. a point satisfying all defining inequalities with strict inequality. In this case, the inequalities defining the polyhedron are all essential (up to the multiplication by a positive number) and they are in one-to-one correspondence with facets. If, instead, the polyhedron is not full dimensional, then there exist linear equations that are satisfied by all points of the polyhedron.

If $v \in P$ is a point of the polyhedron that is a face of P, then v is called a *vertex* of P. A polyhedron is *pointed* if it contains a vertex or, equivalently, if it does not contain a line. A non-empty polyhedron $P = \{x \in \mathbb{R}^n \colon Ax \leq b\}$ is pointed if and only if A has full column rank. A vertex of a polyhedron can be characterized geometrically as a point which is not contained in the segment connecting any other two points of the polyhedron.

A polytope is the convex hull of finetely many points, i.e. its vertices. To describe a polyhedron, instead, it does not suffice to consider its vertices. A classical result of Minkowski and Weyl establishes that we can represent any polyhedron P as the sum of a polytope and a convex polyedral cone: $P = \text{conv}(V) + \text{cone}(R)$, where V and R are finite subsets of $\mathbb{R}^n$. Precisely, V is the set of vertices of the polyhedron, and R is the set of its *extreme rays.* Thus, each point of a polyhedron can be expressed as a convex combination of vertices plus a conic combination of rays.

Given a constraint $a^T x \leq b_0$, we say that the constraint is *binding* or *active* at $\bar{x}$ if $a^T \bar{x} = b_0$. We now introduce an algebraic definition of a vertex as a feasible solution at which there are n linearly independent active constraints.

Definition 1.2.1. *[13] Let P be a polyhedron defined by linear equality and inequality constraints and consider a point $x \in \mathbb{R}^n$.*

- *x is a* basic solution *if*
  - *(i) All equality constraints are active;*
  - *(ii) Out of the constraints that are active at x, there are n of them that are linearly independent.*
- *If x is a basic solution that satisfies all the constraints, we say that it is a* basic feasible solution*.*

Theorem 1.2.1. *[13] Let P be a non-empty polyhedron. $x \in P$ is a vertex of P if and only if it is a basic feasible solution.*

A polyhedron P is in *standard form* if it is expressed as $P = \{x \in \mathbb{R}^n \colon Ax = b, x \geq 0\}$, where A is a $m \times n$ full row rank matrix. A non-singular $m \times m$ submatrix B of A is called *basis.* A basic solution for a polyhedron in standard form is therefore a vector $x \in \mathbb{R}^n$ of the form $x = \begin{pmatrix} x_B \\ x_N \end{pmatrix}$ such that $x_B = B^{-1}b$ and $x_N = \mathbf{0}_{n-m}$. Two different basic solutions are *adjacent* if there are $n-1$ linearly independent constraints that are active at both of them. For standard form polyhedra, two bases are said to be adjacent if they share all but one basic column. Each basis of a polyhedron P corresponds to a basic solution of P. However, different bases may lead to the same basic solution. This phenomenon is closely related to *degeneracy.* Degeneracy occurs in a basic solution x, if there are more than n binding constraints. In other words, the number of active constraints in x is greater than the minimum necessary. In standard form polyhedra, degeneracy occurs if more than $n-m$ of the components of x are zero.

1.3 Linear Programming

A *linear programming* (LP) problem consists in minimizing (or maximizing) a linear function subject to linear equality and inequality constraints, i.e. over a polyhedron. There are many standard forms in which linear programming problems can be written. First we give a classical result of linear programming concerning optimality of vertices.

Theorem 1.3.1. *Consider the linear programming problem of minimizing a linear cost function $c^T x$ over a non-empty polyhedron P. Then, either the problem is unbounded, i.e. the optimal cost is $-\infty$, or there exists an optimal solution. If the problem has a finite solution and P does not contain a line, then there exists an optimal solution which is a vertex of P.*

Consider the following linear programming problem

$$\begin{aligned} \min \quad & c^T x \\ \text{s.t.} \quad & Ax = b \\ & x \geq 0, \end{aligned} \tag{LP}$$

where $A \in \mathbb{R}^{m \times n}$ is a matrix whose rows are linearly independent and $b \in \mathbb{R}^n$.

The best known algorithm for linear programming, the *simplex method* was defined by Dantzig in 1951 and is still one of the most efficient methods to solve LP problems. We will first describe the principles of the simplex method assuming that the linear program is non-degenerate. In this case, from a geometric point of view, the simplex method starts from a vertex of $P = \{x \in \mathbb{R}^n_+ \colon Ax = b\}$ and looks for an edge of the polyhedron P connecting v to another vertex v', in which the value of the objective function is strictly lower. Repeating this procedure, in a finite number of steps, an optimal vertex is reached, i.e. a vertex such that all its neighboring vertices are not improving for the objective function. Algebraically, at each step of the simplex the current basis B is transformed into an adjacent one by applying a *simplex pivot*, i.e. by exchanging a column of the basis matrix B with a column of the nonbasic matrix N, provided that the constraint matrix A has been expressed according to such decomposition as $A = [\ B \quad N\ ]$. For each nonbasic variable x_j the corresponding *reduced cost* $\bar{c}_j$ is computed, which represents the rate of change of the objective function per unit increase of x_j. At each step, the variable entering the basis is chosen as the nonbasic variables with minimum negative reduced cost. The basic variable which exits the basis is instead chosen according to a *ratio test*, in order to guarantee that the new basic solution is also feasible. We refer to [13] for further details on the classical version of the simplex method, and we recommend [14] for alternative implementations used in current solvers.

In case of degeneracy, a pivot operation may result in no change of basic feasible solution. In particular, stalling or cycling phenomena can occur. Several anti-cycling rules have been proposed to guarantee the finite termination of the simplex method. An approach that seems more efficient in practice to deal with degeneracy is based on *bound perturbation* and related methods [14]. When the current basic feasible solution is such that all reduced costs are nonnegative, the current solution is optimal. Note that, in presence of degeneracy, this condition is only sufficient.

1.4 Integer Programming

An *integer linear program* (ILP) is an optimization problem of the form

$$\begin{aligned} \min \quad & c^T x \\ \text{s.t.} \quad & Ax \geq b \\ & x \in \mathbb{Z}^n. \end{aligned} \tag{ILP}$$

In a *mixed integer linear program* (MILP) only a subset of the variables are integer constrained, i.e. $x_i \in \mathbb{Z}$, $i \in I \subset \{1, \ldots, n\}$. Analogously, in a *mixed 0-1 linear program* a subset of the variables are constrained to assume binary values, i.e. $x_i \in \{0, 1\}$, $i \in I \subset \{1, \ldots, n\}$. In this thesis, we will mainly deal with *pure 0-1 linear programs*, where all variables are constrained to assume binary values, i.e. $x \in \{0, 1\}^n$. Unlike linear programming problems, mixed integer linear programs are *NP-hard* [22].

In particular, pure 0-1 linear programs are also classified as NP-hard, and in fact this problem is one of Karp's 21 NP-complete problems [49]. Consider the pure 0-1 linear program

$$\begin{aligned} \min \quad & c^T x \\ \text{s.t.} \quad & x \in S \subseteq \{0,1\}^n. \end{aligned} \tag{BLP}$$

A feasible solution $\bar{x} \in S$ such that $c^T\bar{x} \leq c^T x$ for all $x \in S$ is an *optimal solution* of the problem; if $S = \emptyset$, the problem is *infeasible*; if $S \neq \emptyset$ and for each $x \in S$ there exists $\bar{x} \in S$ such that $c^T\bar{x} < c^T x$, the problem is *unbounded*. Consider the convex hull of points in S. Minimizing the objective function $c^T x$ over $\text{conv}(S)$ yields a linear program which is equivalent to (BLP). In fact, by Theorem 1.3.1, if this problem has an optimal solution, then an optimum lies on a vertex of $\text{conv}(S)$. Thus, in principle, it is possible to solve the pure 0-1 linear program (BLP) by solving the linear program $\min\{c^T x \colon x \in \text{conv}(S)\}$. Unfortunately, in general, the description of $\text{conv}(S)$ in terms of linear inequalities is not known.

A *formulation* is a representation of S by a system of linear inequalities $Ax \geq b$ such that $\{x \in \mathbb{R}^n \colon Ax \geq b\} \cap \{0,1\}^n = S$. The *linear relaxation* of a formulation $Ax \geq b$ is the linear programming problem

$$\begin{aligned} \min \quad & c^T x \\ \text{s.t.} \quad & Ax \geq b \\ & 0 \leq x \leq 1. \end{aligned} \tag{LR}$$

Let P be the feasible set of (LR). Clearly $P \supseteq S$, implying that $z^{LR} = \min\{c^T x \colon x \in P\} \leq \min\{c^T x \colon x \in S\} = z^{BP}$. Therefore, the optimal value of (LR), yields a *lower bound* on the optimal value of (BLP). The difference $z^{BP} - z^{LR}$ is the *integrality gap* of the 0-1 program. Moreover, given any feasible solution $\bar{x}$ of (BLP), the difference $c^T\bar{x} - z^{LR}$ is an indication of the quality of $\bar{x}$. An interesting question concerns the issue of establishing a measure of the quality of formulations. The classical criterion used to establish whether a formulation is better than another one, is to verify which one of the two better approximates $\text{conv}(S)$. In fact, it is assumed that the better such approximation is, the easier the integer program can be solved.

Two exact methods to solve an integer linear program are the *branch-and-bound* method and the *cutting planes* algorithm.

Given a valid inequality $\alpha x \geq \beta$, and a point $\bar{x} \notin \text{conv}(S)$, $\alpha x \geq \beta$ is a *cut* for $\bar{x}$ if $\alpha\bar{x} < \beta$, i.e. the inequality $\alpha x \geq \beta$ *cuts off* $\bar{x}$. Cutting planes are crucial, as they can be used to tighten relaxations of integer programs. Cutting planes algorithms generate a sequence of successively tighter relaxations, by adding at each step one or more cuts to the current linear programming relaxation of the integer program. The first finitely terminating cutting plane algorithm for integer programming was a cutting plane algorithm proposed by Gomory in 1958 [38], which used some detailed information from the optimal simplex tableau. *Gomory cuts* were initially considered impractical and ineffective, due mainly to some numerical issues. Surprisingly, in the mid 90's Gomory cuts were shown to be very effective in combination with branch-and-bound and techniques to overcome numerical instabilities [25]. Nowadays, all commercial MILP solvers use Gomory cuts, also because they can be very efficiently generated from a simplex tableau, whereas many other types of cuts are expensive to separate. For further readings on cutting planes method see [24].

The branch-and-bound method was introduced in the early 60's by Land and Doig [52] and is an implicit enumeration technique based of the subdivision of the feasible set into subsets. Roughly speaking, the algorithm computes a lower bound in every unexplored subset and uses these bounds to discard certain subsets from further consideration. The enumeration generated in the procedure is represented by a tree, whose nodes correspond to the list of subproblems to be solved. For a 0-1 program, the usual subdivision of the feasible region into subsets consists in progressively fixing variables of the problem to 0 or 1.

Up to the early 80's, pure branch-and-bound was the common method used by practitioners to solve mixed integer programs. An important improvement consisted in the introduction of cutting planes at the root node, i.e. the node of the branch-and-bound tree where no variable has been fixed yet, to possibly tighten the original linear programming relaxation. Another important improvement to branch-and-bound came when cutting planes started to be added not only at the root node, but also at other nodes of the branch-and-bound tree. When a cut is generated at a node where some of the variables have been fixed, it is only guaranteed to be valid for all the descendants of that node. On the other hand, to derive an inequality that is valid at the root node, and therefore for the whole branch-and-bound tree, the cut has to be *lifted* by computing the coefficients of the fixed variables. The method that we have just outlined combines both the cutting planes and the implicit enumeration approach, and is known as *branch-and-cut*.

1.5 The stable set problem

Let $G(V, E)$ be a graph and $c\colon V \to \mathbb{Q}_+$ be any weighting of the nodes of G. A *stable set* (*independent set*, *node packing*) is a subset S of the node set V such that no two nodes of S are adjacent. We denote by $\alpha(G, c)$ the maximum weight stable set in G. Determining the *maximum weight stable set* is well known to be NP-hard, even in the case where $c = \mathbf{1}$, which corresponds to the problem of finding the *maximum cardinality stable set*. The *stable set polytope*, denoted by $STAB(G)$ is the convex hull of the incidence vectors of stable sets in G. Formally,

$$STAB(G) := \text{conv}\{\chi^S \in \{0,1\}^{|V|}\colon S \subseteq V \text{ is a stable set of } G\}.$$

Consequently,

$$\begin{aligned} \alpha(G, c) = \quad & \max \quad c^T x \\ & \text{s.t.} \quad x \in STAB(G). \end{aligned} \tag{STAB}$$

For general graphs, a complete description of the facets of $STAB(G)$ is not known. In the following, we describe different classes of facets of the stable set polytope and, for each of them, we present graphs for which such description is sufficient to characterize $STAB(G)$. For further readings of the facial structure of the stable set polytope we refer to [45] and [63]. The first sets of linear inequalities valid for $STAB(G)$ are the following:

$$x_i \geq 0 \qquad \forall \ i \in V \tag{1.1}$$

$$x_i + x_j \leq 1 \qquad \forall \ (i,j) \in E. \tag{1.2}$$

Constraints (1.2) express the simple condition that the endpoints of an edge cannot both belong to a stable set. Constraints (1.1) and (1.2) define a formulation of (STAB) which is commonly referred to as the *edge formulation*. We denote the polytope defined by inequalities (1.1) and (1.2) by $FSTAB(G) = \{x \in \mathbb{R}^n_+ \colon x_i + x_j \leq 1 \ \forall \ (i,j) \in E\}$ and we address it as the *fractional stable set polytope*.

Proposition 1.5.1. *$FSTAB(G)$ coincides with $STAB(G)$ if and only if G is bipar-*

tite and has no isolated nodes.

If G contains some isolated nodes, we would need to add to the description of $FSTAB(G)$ also constraints of type $x_i \leq 1$, one for each isolated node $i \in V$. Throughout the thesis we will make the following assumption.

Assumption 1.5.1. *$G(V, E)$ is an undirected, simple graph without isolated nodes.*

Therefore, we can consider the following linear relaxation of the edge formulation, where all constraints of type $x_i \leq 1$ are redundant, and consequently discarded.

$$\begin{aligned} \max \quad & c^T x \\ \text{s.t.} \quad & x_i + x_j \leq 1 \qquad \forall\, (i,j) \in E \\ & x_i \geq 0 \qquad \forall\, i \in V. \end{aligned} \tag{FSTAB}$$

The minimal graphs (under taking induced subgraphs) for which $FSTAB(G) \supset STAB(G)$ are the odd cycles. In fact, $x_i^* = \frac{1}{2}\ \forall\, i \in V$ belongs to $FSTAB(G)$ but is not feasible with respect to $STAB(G)$. A class of inequalities valid for $STAB(G)$ which takes into account odd cycles consists in the so-called *odd cycle constraints*:

$$\sum_{i \in C} x_i \leq \frac{|C| - 1}{2} \qquad \text{for each odd cycle } C. \tag{1.3}$$

Graphs for which (1.1), (1.2) and (1.3) suffice to describe $STAB(G)$ are called *t-perfect*. Bipartite graphs are trivially t-perfect. Other classes of graphs that have been proven to be t-perfect are *almost bipartite graphs*, *series-parallel graphs*, *nearly bipartite planar graphs* and *strongly t-perfect graphs*. For details and further readings we refer to [45]. An important result on t-perfect graphs establishes that it is possible to find a maximum weight stable set on t-perfect graphs in polynomial time, see [45].

Another system of linear inequalities valid for the stable set polytope consists of the *clique constraints*

$$\sum_{i \in Q} x_i \leq 1 \qquad \text{for each clique } Q. \tag{1.4}$$

A clique inequality (1.4) asserts that a stable set cannot pick more than one node in a given clique. Remark that clique constraints (1.4) specialize in edge constraints, if

$|Q| = 2$ and to odd cycle constraints, in the case where $|Q| = 3$, i.e. for triangles. The graphs for which (1.1) and (1.4) suffice to describe $STAB(G)$ are called *perfect*. Perfect graphs were introduced by Berge in 1961. In general, for a graph G, the *clique number* $\omega(G)$, i.e. the size of a maximum clique in G, is not greater than the *chromatic number* $\chi(G)$, i.e. the minimum number of colors that is needed to label the nodes of the graph in such a way that no two adjacent nodes have the same color. For a perfect graph, $\omega(G') = \chi(G')$ for each induced subgraph G' of G. It was conjectured by Berge (1961, 1962) and proven by Lovász (1972) that the complement of a perfect graph is itself perfect. For a list of classes of perfect graphs, see e.g. [45]. A crucial contribution for the characterization of perfect graphs was given in 2002 by Chudnovsky, Robertson, Seymour and Thomas, who proved the following result, conjectured by Berge in 1962.

Theorem 1.5.1 (The Strong Perfect Graph Theorem). *A graph is perfect if and only if it does not contain an odd cycle of length at least five, or its complement, as an induced subgraph.*

The Strong Perfect Graph Theorem implies that the minimal *imperfect* graphs are the odd cycles of length at least five, and their complements.

In this thesis we mainly investigate some structural properties of the fractional stable set polytope $FSTAB(G)$: even if $FSTAB(G)$ turns out to be a very weak approximation of $STAB(G)$, its simple geometrical structure allows us to state useful characterizations of its bases and vertices, enhancing the understanding of several interesting geometrical properties.

1.6 Some combinatorial problems related to the stable set problem

Let $M = \{1, \ldots, m\}$ be a finite set and $M_1, M_2, \ldots, M_n$ be a given collection of subsets of M. Define $N = \{1, \ldots, n\}$ and suppose that for each $j \in N$, c_j is the weight associated to the subset M_j of the collection. A subset F of N is called a *cover*, *packing* or *partitioning* if it intersects each element M_j of the collection at least once, at most once or exactly once, respectively. In the *set covering* problem we would like to find a cover F of *minimum weight*; in the *set packing* problem we would like to find a packing F of *maximum weight*, while in the *set partitioning* both minimization and maximization versions are possible. In order to define an integer programming formulation for such problems,it is possible to introduce a 0-1 $m \times n$ matrix A, encoding incidence relations of the family $\{M_j\colon j \in N\}$. The entries a_{ij} of A are such that $a_{ij} = 1$ if $i \in M_j$ and $a_{ij} = 0$ if $i \notin M_j$. The families of set covering, packing and partitioning problems can be then formulated as

$$\min \left\{ \textstyle\sum_{j=1}^{n} c_j x_j \colon Ax \geq \mathbf{1}, x \in \{0,1\}^n \right\}, \tag{SC}$$

$$\max \left\{ \textstyle\sum_{j=1}^{n} c_j x_j \colon Ax \leq \mathbf{1}, x \in \{0,1\}^n \right\}, \tag{SP}$$

$$\min \left\{ \textstyle\sum_{j=1}^{n} c_j x_j \colon Ax = \mathbf{1}, x \in \{0,1\}^n \right\}. \tag{SPP}$$

In the following, we will denote by SC, SP and SPP the convex hulls of the feasible points of (SC),(SP) and (SPP), respectively. Moreover, we will indicate with (LSC), (LSP) and (LSPP) the linear relaxations of (SC), (SP) and (SPP), respectively.

First of all, note that (SP) is a special case of (SPP).

The edge formulation of the stable set problem is clearly a set packing problem on a graph. Conversely, (SP) can be formulated as a stable set problem on the *intersection graph* associated to matrix A. Precisely, denote by A_j the j-th column of matrix A of (SP). The intersection graph $G_A(V, E)$ has a node for every column of A, and one edge for every pair of non-orthogonal columns of A, i.e. $(i, j) \in E$ if and only if $A_i^T A_j \geq 1$. Denote by A_G the edge-node incidence matrix of G_A and assign to each node $i \in V$ the weight c_i of the i-th variable of (SP). Consider the stable set problem

$$\begin{aligned} \max \quad & c^T x \\ \text{s.t.} \quad & A_G x \leq \mathbf{1} \\ & x \in \{0,1\}^n. \end{aligned} \tag{1.5}$$

Remark that x is a feasible (resp. optimal) solution of (1.5) if and only if it is a feasible (resp. optimal) solution of (SP). A direct consequence of this observation is that one way of solving set packing problems consists in solving a stable set problem on the corresponding intersection graph. However, while these two integer problems are equivalent, the two associated linear relaxations are not. Precisely

$$\max\left\{\sum_{j=1}^{n} c_j x_j \colon Ax \leq \mathbf{1}, x \geq \mathbf{0}\right\} \leq \max\left\{\sum_{j=1}^{n} c_j x_j \colon A_G x \leq \mathbf{1}, x \geq \mathbf{0}\right\}.$$

Other combinatorial problems on graphs which are strictly related to the stable set problem are: the *node coverig* problem, the *maximum clique* problem, the *edge covering* problem and the *matching* problem.

Let G be any graph, and let $\nu(G)$, $\tau(G)$, $\alpha(G)$, $\rho(G)$ denote the matching, node covering, stability and edge covering numbers, respectively. Two basic results linking these numbers are the so-called *Gallai Identities* (1959).

Lemma 1.6.1. *For any graph G, $\alpha(G) + \tau(G) = |V(G)|$.*

Lemma 1.6.2. *For any graph G with no isolated nodes, $\nu(G) + \rho(G) = |V(G)|$.*

It is easy to verify that, if S is a minimum node cover of G, then $V(G) \setminus S$ is a maximum stable set. Consider the edge formulation of the stable set problem: $STAB(G) = \{x \in \{0,1\}^n \colon x_i + x_j \leq 1\}$. The affine transformation $y = \mathbf{1} - x$ maps $STAB(G)$ into $NC(G) = \{y \in \{0,1\}^n \colon y_i + y_j \geq 1\}$, that is the *node covering polytope*, i.e. the convex hull of the incidence vectors of node covers in G.

Concerning the relationship between matchings and edge covers, a minimal edge cover is minimum if and only if it contains a maximum matching. Conversely, a maximal matching is maximum if and only if it is contained in a minum edge cover. Moreover, for any graph G, $\nu(G) \leq \tau(G)$ and, for bipartite graphs the following identity holds.

Theorem 1.6.1 (König's Minmax Theorem)**.** *If G is bipartite, then $\nu(G) = \tau(G)$.*

Note that the dual of (FSTAB) corresponds to the problem of covering weighted nodes by a minimum number of edges:

$$\begin{aligned} \min \quad & \sum_{(i,j)\in E} y_{ij} & \\ \text{s.t.} \quad & \sum_{(i,j)\in E} y_{ij} \geq c_j & \forall\, j \in V \\ & y_{ij} \geq 0 & \forall\, (i,j) \in E. \end{aligned} \tag{1.6}$$

If $c_j = 1 \ \ \forall\ j \in V$, (1.6) corresponds to the linear relaxation of the edge covering problem. In this case, by strong duality, the optimum value of (1.6) is equal to the optimal value of (FSTAB). If G is bipartite, $STAB(G) = FSTAB(G)$ implies $\alpha(G) = \rho(G)$ and, by Lemma 1.6.2, $\alpha(G) + \nu(G) = |V(G)|$.

Finally, the maximum clique problem on a graph G is equivalent to the maximum stable set problem on the complement graph of G, i.e. $\bar{G}(V, \bar{E})$. Therefore, we can use the edge formulation to formulate the problem of finding the maximum weight clique on $G(V, E)$:

$$\begin{array}{rll} \max & c^T x & \\ \text{s.t.} & x_i + x_j \leq 1 & \forall\ (i,j) \notin E \\ & x_i \geq 0 & \forall\ i \in V. \end{array}$$

1.7 Properties of the edge formulation

In this section we introduce some useful results about the fractional stable set polytope $FSTAB(G)$, due mainly to Nemhauser and Trotter [61]. The first result that we present states that vertices of $FSTAB(G)$ are half-integral and was originally established by Balinski [11]; detailed proofs of this result are given in [60, 69].

Theorem 1.7.1. *(Balinski [11]) Let x be a vertex of $FSTAB(G)$. Then, $x_i = 0$, $\frac{1}{2}$ or 1 for $i = 1, \ldots, n$.*

An interesting sufficient condition for local optimality is given by the following theorem.

Theorem 1.7.2. *(Nemhauser and Trotter [61]) If $S \subseteq V$ is an optimal stable set in $G[S \cup N(S)]$, the subgraph of G induced by $S \cup N(S)$, then $S \subseteq S^*$, where S^* is an optimal stable set in G.*

The next theorem illustrates how to perform a fixing of the variables that are integer valued in an optimal solution of (FSTAB).

Theorem 1.7.3. *(Nemhauser and Trotter [61]) Suppose x^* is an optimal $(0, \frac{1}{2}, 1)$-valued solution of (FSTAB). Define sets $S = \{i \in V : x^*_i = 1\}$ and $\bar{S} = \{i \in V : x^*_i =$*

$0\}$. *There exists a maximum stable set in G that contains S and does not contain $\bar{S}$.*

Theorem 1.7.3 implies that those variables which assume binary values in an optimal solution of (FSTAB) retain the same values in some optimal solution of (STAB). This means that, to solve (STAB), one can solve (FSTAB) and then find a stable set on the subgraph of G induced by the nodes $i \in V : x_i^* = \frac{1}{2}$.

The next theorem establishes a necessary and sufficient condition for $x_i^* = \frac{1}{2} \;\; \forall i = 1, \ldots, n$ to be a (unique) optimal solution of (FSTAB). Recall that, for any $S \subseteq V$, the neighbors of S are defined as $N(S) = \{j \in V \setminus S \colon (i,j) \in E \text{ for some } i \in S\}$.

Theorem 1.7.4. *(Nemhauser and Trotter [61]) The solution $x_i^* = \frac{1}{2} \;\; \forall i = 1, \ldots, n$ is an optimal (resp. the unique optimal) solution of (FSTAB) if and only if $|S| \leq |N(S)|$ (resp. $|S| < |N(S)|$) for every non-empty stable set S.*

Chapter 2

A graphic characterization of bases of the Fractional Stable Set Polytope

In this chapter, we present a graphic characterization of bases of $FSTAB(G)$ in terms of special subgraphs of G. This result will be at the base of all our subsequent results. In fact, the understanding of the bases of the fractional stable set polytope will provide us with a deep insight of the problem, allowing for the definition of further structural properties, presented in the next chapters. The characterization of the bases of $FSTAB(G)$ is based on a result of Campelo and Cornuéjols [17] and is closely related to a previous characterization of the vertices of $FSTAB(G)$, established by Nemhauser and Trotter [60].

Remark that, by introducing a slack variable for each edge constraint, (FSTAB) can be rewritten as

$$\begin{array}{lll} \max & c^T x & \\ \text{s.t.} & x_i + x_j + y_{ij} = 1 & \forall\ (i,j) \in E \\ & x_i \geq 0 & \forall\ i \in V \\ & y_{ij} \geq 0 & \forall\ (i,j) \in E. \end{array} \tag{FSTAB}$$

Analogously, we can express the fractional stable set polytope as $FSTAB(G) = \{(x, y) \in \mathbb{R}^n_+ \times \mathbb{R}^m_+ \colon x_i + x_j + y_{ij} = 1\ \ \forall\ (i,j) \in E\}$. According to this redefinition of $FSTAB(G)$, every node of G indexes a x variable of $FSTAB(G)$, while each edge of G indexes a y slack variable of $FSTAB(G)$. Therefore, in the following, we will call the x and y variables *node variables* and *edge variables* (or *edge-slack variables*), respectively. We will also say that a node is 0, $\frac{1}{2}$ or 1 valued in (x, y), if the corresponding node variable is 0, $\frac{1}{2}$ or 1 valued, respectively. Analogously, we say that an edge is 0, $\frac{1}{2}$ or 1 valued in (x, y), if the corresponding edge variable is 0, $\frac{1}{2}$ or 1 valued, respectively.

Let A denote the edge-node incidence matrix of G. Let $\mathcal{B}$ stand for the set of all bases of the constraint matrix $[A \quad I]$. Note that the rows of (FSTAB) are linearly independent, therefore a basis consists of m columns. We denote by B an element of $\mathcal{B}$ and by N the resulting nonbasic submatrix. To avoid heavy notation, we may also use B and N to denote the corresponding sets of indices.

In Section 2.1, we present a characterization of vertices of $FSTAB(G)$ given by Nemhauser and Trotter [60]. In Section 2.2, we describe a necessary condition, given by Campelo and Cornuéjols [17], stating that any basic solution of $FSTAB(G)$ can be associated to a special pseudoforest of G. We also establish the converse, i.e. we prove that this condition is sufficient as well. Finally, we briefly relate the first characterization and the second.

2.1 Vertices of the Fractional Stable Set Polytope

A first characterization of the vertices of the fractional stable set polytope $FSTAB(G)$ was given by Nemhauser and Trotter in [60]. In their paper, Nemhauser and Trotter present a decomposition theorem which yields a characterization of vertices of $FSTAB(G)$ in terms of certain *elementary vertices*. In this section, we briefly summarize their fundamental results, which will be useful for a graphic characterization of basic solutions of $FSTAB(G)$.

Given a simple graph $G(V, E)$, consider a subset of the nodes $F \subseteq V$. Let $x^F \in FSTAB(G)$ be defined as

$$x_j^F = \begin{cases} \frac{1}{2} & \text{if } j \in F \\ 0 & \textit{otherwise.} \end{cases} \tag{2.1}$$

Denote by $G[F]$ the subgraph of G induced by nodes in F. If x^F is a vertex of $FSTAB(G)$ and $G[F]$ is connected, then x^F is called an *elementary fractional vertex*. The following proposition establishes a characterization of elementary fractional vertices.

Proposition 2.1.1. *(Nemhauser and Trotter [60]) Let $F \subseteq V$ and $x^F \in FSTAB(G)$ be defined as in (2.1). Then x^F is an elementary fractional vertex of $FSTAB(G)$ if and only if $G[F]$ contains an odd cycle.*

Feasible integer solutions of (FSTAB) correspond to integer vertices of $FSTAB(G)$. Two vertices of $FSTAB(G)$ are said to be *disjoint* if their sum is feasible for $FSTAB(G)$. Nemhauser and Trotter proved a decomposition theorem for vertices of $FSTAB(G)$, describing them in terms of disjoint integer vertices and elementary fractional vertices.

Theorem 2.1.1. *(Nemhauser and Trotter [60]) A vector $x \in \mathbb{R}^n$ is a vertex of $FSTAB(G)$ if and only if it can be expressed as $x = x^0 + x^1 + \ldots + x^k$, where*

(i) x^0 is an integer vertex of $FSTAB(G)$;

(ii) $x^1, \ldots, x^k$ are elementary fractional vertices of $FSTAB(G)$;

(iii) $x^0, x^1, \ldots, x^k$ are mutually disjoint.

This characterization implies that an arbitrary vertex of $FSTAB(G)$ can be decomposed uniquely in the sum of an integer vertex and elementary fractional vertices. The converse is also true, that is any sum of vertices which yields a feasible solution of $FSTAB(G)$ produces a vertex of $FSTAB(G)$. Nemhauser and Trotter also remark that, unlike what happens in the matching problem, where elementary fractional vertices are in one-to-one correspondence with the odd cycles of G, (FSTAB) has an elementary fractional vertex for every induced subgraph of G that is connected and contains an odd cycle.

2.2 A graphic characterization of bases

In this section, we present a graphic characterization of bases of $FSTAB(G)$. Such graphic characterization is deeply related to the results presented in Section 2.1.

A graphic characterization of the bases of $FSTAB(G)$ was given by Campelo and Cornuéjols [17, 18]. This result, reported in the next theorem, is a necessary condition for a half integral vector to be a basic solution of $FSTAB(G)$. The condition states that, given a basic solution of $FSTAB(G)$ it is possible to associate it to a subgraph of G, whose connected components are either rooted trees or 1-trees with an odd cycle. We call such pseudoforest a *1-pseudoforest*. We show that this condition is also sufficient, that is given a 1-pseudoforest, a basic solution of $FSTAB(G)$ is automatically defined.

Let $B \in \mathcal{B}$ be a basis, feasible or infeasible, of $[A \quad I]$. Let V_B and V_N represent the set of basic and nonbasic nodes, indexing variables x_B and x_N, respectively. Similarly, partition edges into E_B and E_N and slack variables correspondingly, into y_B and y_N. In order to characterize the structure of the basis, consider $G_B(V, E_N)$, which is obtained from G by removing the basic edges. Let $C_i(V_i, E_i)$, $i = 1, \ldots, k$ be the connected components of G_B.

Recall that a graph is a *1*-tree if it is connected and the number of its nodes equals the number of its edges. A 1-tree contains a unique cycle. Define I_0 and I_1 as the subsets of $\{1, \ldots, k\}$ indexing tree and 1-tree components of G_B, respectively. Remark that every singleton of G_B can be seen as a trivial tree, containing only one node and no edges.

Theorem 2.2.1. *(Campelo and Cornuéjols [17]) For every $B \in \mathcal{B}$, G_B is a 1-pseudoforest, i.e. each connected component of G_B is either a rooted tree or a 1-tree with an odd cycle. Each tree has exactly one nonbasic node, which corresponds to its root. The nodes of every 1-tree are all basic.*

Given $B \in \mathcal{B}$, for all $i = 1, \ldots, k$ denote by B_i the submatrix of B defined by the rows and columns indexed by E_i and $V_i \cap B$, respectively. Remark that, for every isolated node of G_B, $V_i \cap B = \emptyset$. We denote by I_0' the tree components of G_B that are not singletons and we assume w.l.o.g. that singletons are the last $k - k'$ connected components of G_B.

Lemma 2.2.1. *(Campelo and Cornuéjols [18]) Given $B \in \mathcal{B}$, for $i \in I_0'$, $B_i^{-1}\mathbf{1} \in \{0,1\}^{|V_i|}$. For $i \in I_1$, $B_i^{-1}\mathbf{1} = \left(\frac{1}{2}\right)\mathbf{1}$.*

Given a basis $B \in \mathcal{B}$, consider the associated 1-pseudoforest G_B. For every rooted tree of G_B, i.e. for each C_i with $i \in I_0$, denote by $\tau(C_i)$ the root of the tree. Similarly, for every 1-tree component C_j, $j \in I_1$ of G_B, denote by $\kappa(C_j)$ its unique (odd) cycle.

In the next chapters, in order to prove several results, the converse of Theorem 2.2.1 will be needed. To this purpose, it is very important to establish that there is a one-to-one correspondence between 1-pseudoforests and bases of $FSTAB(G)$. In the next theorem, we prove that the converse of Theorem 2.2.1 holds.

Theorem 2.2.2. *Let $G_B(V, E_N)$ be a 1-pseudoforest of G, i.e. a subgraph of G whose connected components are rooted trees and 1-trees with an odd cycle. Denote by $C_i(V_i, E_i), i = 1, \dots, k$ the connected components of G_B. Let $I_0 \subseteq \{1, \dots, k\}$ index the tree components of G_B and, for $i \in I_0$, let $\tau(C_i)$ be the root of the tree component C_i. Define $V_N = \bigcup_{i \in I_0} \tau(C_i)$, $V_B = V \setminus V_N$, $E_B = E \setminus E_N$. Then $B = V_B \cup E_B$ is a basis of $FSTAB(G)$.*

Proof. First, let us group the equations of (FSTAB) according to the edges of E_N and E_B respectively, to get

$$\bar{A}x + y_N = 1, \tag{2.2}$$

$$\hat{A}x + y_B = 1, \tag{2.3}$$

where $A = \begin{bmatrix} \bar{A} \\ \hat{A} \end{bmatrix}$. Notice that $\bar{A}$ is the edge-node incidence matrix of G_B. Precisely, if A_i is the $|E_i| \times |V_i|$ incidence matrix of C_i, then $\bar{A}$ can be organized as

$$\bar{A} = \left[\begin{array}{cccc|c} A_1 & & & & \\ & A_2 & & & \\ & & \ddots & & 0 \\ & & & A_{k'} & \end{array}\right],$$

where $k' \leq k$ is the number of connected components of G_B containing at least two nodes. Remark that the last zero columns correspond to those components of G_B which consist of isolated nodes.

For every non-trivial tree component C_i, $i \in I_0$, $i \leq k'$, let us partition $A_i = [B_i \ N_i]$, where N_i consists of a single column, which is the one indexed by node $\tau(C_i)$. It is easy to check that each matrix B_i is square and invertible, because it can be expressed as a triangular matrix, by reordering the nodes of the tree from the leaves towards the root $\tau(C_i)$, according to the partial order defined by the root of the tree.

For every 1-tree component C_i, $i \in I_1$, define $B_i = A_i$. Also in this case each matrix B_i is square and invertible, as it can be expressed as a block matrix of the form

$$B_i = \begin{bmatrix} T_i & D_i \\ 0 & K_i \end{bmatrix}, \tag{2.4}$$

where K_i is the edge-node incidence matrix of the odd cycle and T_i, togheter with an extra column of D_i, is the edge-node incidence matrix of the acyclic part of the 1-tree. Note that, by conveniently reordering the nodes of the 1-tree, it is possible to express K_i and T_i as a circulant matrix and a triangular matrix, respectively, implying that B_i is invertible.

Therefore, a basis of (2.2), (2.3) is given by

$$B = \begin{bmatrix} \bar{B} & 0 \\ \hat{B} & I \end{bmatrix}, \tag{2.5}$$

where $\bar{B}$ and $\hat{B}$ are submatrices of $\bar{A}$ and $\hat{A}$, respectively and

$$\bar{B} = \begin{bmatrix} B_1 & & & \\ & B_2 & & \\ & & \ddots & \\ & & & B_{k'} \end{bmatrix}. \tag{2.6}$$

Because $\bar{B}$ is a block diagonal matrix whose blocks are non-singular, $\bar{B}$ is non-singular as well, implying that B is a basis.

□

Theorems 2.2.1 and 2.2.2 establish a precise correspondence between bases of (FSTAB) and 1-pseudoforests of G. In the remainder of the thesis, given a basis $B \in \mathcal{B}$, we will say that $G_B(V, E_N)$ is the *basic subgraph* of G associated to B. In the next theorem, this correspondence is extended to basic solutions of (FSTAB). In particular, the next theorem highlights the connection between the variables that are $\frac{1}{2}$-valued in a

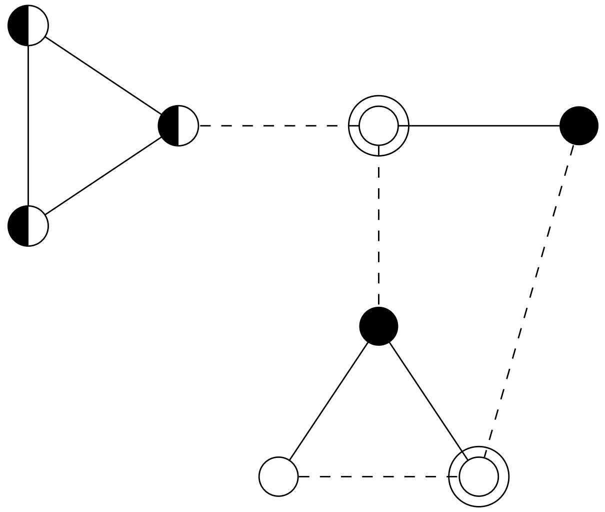

Figure 2.1: A basic feasible solution and the associated 1-pseudoforest. Black nodes are 1-valued, white nodes are 0-valued, and half coloured nodes are $\frac{1}{2}$-valued. The root of each tree component is indicated by a circle. Dashed and plain edges index basic and nonbasic slack variables, respectively.

basic solution of $FSTAB(G)$ and the nodes belonging to the 1-tree components of the associated basic subgraph. Figure 2.1 illustrates the one-to-one correspondence between 1-pseudoforests of G and bases of $FSTAB(G)$. Throughout the thesis we will always indicate graphically 1-valued nodes by black circles, 0-valued nodes by white circles, and $\frac{1}{2}$-valued nodes by half coloured circles. The nonbasic edges of a basic subgraph will be represented by plain lines, while basic edges will be represented by dashed lines.

Theorem 2.2.3. *Let $B \in \mathcal{B}$ be a basis of (FSTAB). Denote by $\bar{x}$ the basic solution associated to B. Then:*

(i) all nodes in 1-tree components of G_B index $\frac{1}{2}$-valued components of $\bar{x}$;

(ii) all nodes of tree components of G_B index $(0, 1)$-valued components of $\bar{x}$. On each tree component, nodes that are at even distance from the root are 0-valued, while those that are at odd distance from the root are 1-valued.

Proof. The result immediately follows from (2.5), (2.6) and Lemma 2.2.1. Note that, given a tree component $C_i(V_i, E_i)$, $i \in I_0$, the assignment of binary values to its nodes is uniquely determined by $\tau(C_i) = V_i \setminus B$, the only nonbasic node of C_i, which takes value 0. $\square$

Example 2.2.1. *Consider the graph of Figure 2.1 and the basis associated to the 1-pseudoforest depicted in the figure. By labeling the nodes according to the order described in the proof of Theorem 2.2.2, it is possible to express the basis matrix B as*

	1	2	4	6	7	8	(1,3)	(2,5)	(3,4)	(5,6)
(1,2)	1	1								
(2,3)		1								
(4,5)			1							
(6,7)				1	1					
(7,8)					1	1				
(6,8)				1		1				
(1,3)	1						1			
(2,5)		1						1		
(3,4)			1						1	
(5,6)				1						1

We say that a tree (or a forest) is *alternating* with respect to a basic solution x of $FSTAB(G)$, if all its edges connect a 1-valued node to a 0-valued node.

Remark 2.2.1. *Given $B \in \mathcal{B}$ and the associated basic solution x, each tree component of the basic subgraph G_B is alternating. This easily follows by recalling that the edges of G_B are nonbasic, and therefore 0-valued in x, and from the fact that nodes of tree components of G_B are 0-1 valued.*

Remark 2.2.2. *How are the characterizations of sections 2.1 and 2.2 related one to the other? Given a basic feasible solution $x \in FSTAB(G)$, Theorem 2.1.1 states that x can be expressed as the sum of an integer vertex x^0, and k elementary fractional vertices $x^1, \ldots, x^k$. By definition, elementary fractional vertices are such that $\frac{1}{2}$-valued nodes induce a connected subgraph of G that contains an odd cycle. Such subgraph admits one or more underlying 1-trees, representing degenerate bases associated to the same elementary fractional vertex. On the other hand, the subgraph of G induced by the nodes that are 0-1 valued in x, admits one or more spanning alternating forests, which represent degenerate bases associated to x^0. Remark that condition (iii) of Theorem 2.1.1 is related to the hypothesis that x is a vertex, i.e. a basic feasible solution.*

Chapter 3

Vertex Adjacency and the Hirsch Conjecture for the Fractional Stable Set Polytope

Given a graph, the *edge formulation* of the *stable set problem* is defined by two-variable constraints, one for each edge, expressing the simple condition that two adjacent nodes cannot belong to a stable set. We study vertex adjacency in the *fractional stable set polytope* $FSTAB(G)$, i.e. the polytope yielded by the linear relaxation of the edge formulation, that is defined by the following inequalities:

$$\begin{aligned} x_i + x_j + y_{ij} = 1 & \quad \forall\, (i,j) \in E \\ x_i \geq 0 & \quad \forall\, i \in V \\ y_{ij} \geq 0 & \quad \forall\, (i,j) \in E. \end{aligned}$$

Even if this polytope is a weak approximation of the *stable set polytope*, its simple geometrical structure provides deep theoretical insight as well as interesting algorithmic opportunities. Exploiting a graphic characterization of the bases, we first redefine simplex pivots in terms of simple graphic operations, that turn a given basis into an adjacent one. Between all possible pivots, we characterize degenerate and non-degenerate ones, and we differentiate those leading to an integer or to a fractional vertex. The graphic characterization of bases is crucial to prove another structural property of the fractional stable set polytope, concerning the adjacency of its vertices. In particular, we extend a necessary and sufficient condition due to Chvátal for adjacency of (integer) vertices of the stable set polytope to arbitrary (and possibly fractional) vertices of the fractional stable set polytope. These results lead us to prove that the Hirsch Conjecture is true for the fractional stable set polytope, i.e. the combinatorial diameter of this fractional polytope is at most equal to the number of edges of the given graph.

In Section 3.1 we present some results from the literature regarding adjacency of integer vertices on $FSTAB(G)$. In Section 3.2 we give a graphic characterization of simplex pivots, and we precisely distinguish degenerate and non-degenerate feasible pivots. In Section 3.3 we characterize adjacency between integer and fractional vertices. In Section 3.4 we establish a graphic characterization for two arbitrary vertices to be adjacent on $FSTAB(G)$. This result generalizes Chvátal's condition about adjacency of integer vertices. In Section 3.5 we prove Hirsch conjecture for $FSTAB(G)$. Finally, in section 3.6, we exploit the graphic properties of bases and pivots to define a simple-like algorithm for the Maximum Weight Stable Set Problem.

3.1 Adjacency of integer vertices

The structural properties of the stable set problem and the special characterization of bases of the edge relaxation, outlined in Chapters 1 and 2, translate into special adjacency properties for vertices of $FSTAB(G)$. Exploiting such properties, it could in principle be possible to solve the stable set problem through a modified version of the simplex method, which generates only integer solutions without using cutting plane techniques. This issue has been addressed by Balas and Padberg [7, 8, 9], with regard to the set partitioning and the set packing polytopes which are defined, respectively, as:

$$\text{SPP} = conv\{x \in \{0,1\}^n \colon Ax = \mathbf{1}\},$$

$$\text{SP} = conv\{x \in \{0,1\}^n \colon Ax \leq \mathbf{1}\},$$

where A is an $m \times n$ matrix of zeros and ones (see Section 1.6).

Define also

$$\text{LSP} = \{x \in \mathbb{R}^n_+ \colon Ax \leq \mathbf{1}\},$$

the constraint set of the linear relaxation (LSP) of (SP).

Recall from Chapter 1 that the stable set problem (STAB) is in fact a special case of the set packing problem (SP), since the constraint matrix has exactly two ones per row. Moreover, remind that for a given set packing problem, one can equivalently solve the stable set problem associated to the corresponding intersection graph.

Two bases of a linear program are called *adjacent* if they differ in exactly one column. Two *basic feasible solutions* are called *adjacent* if they correspond to adjacent vertices in the polytope defined by the constraint set of the problem. This distinction is necessary since, because of degenaracy, two adjacent bases may be associated with the same vertex, and two adjacent basic feasible solutions may be associated with two non-adjacent bases. A basis is defined *integer* if the associated basic solution has all its components integer.

Balas and Padberg [7] showed that, for every feasible integer basis to LSP, there are at least as many adjacent feasible integer bases as there are nonbasic columns. Moreover, they proved that, given two basic feasible integer solutions x^1 and x^2 to LSP, x^2 can be obtained from x^1 by a sequence of p pivots, where p is the number of indices $j \in \{1,\dots,n\}$ for which x^1_j is nonbasic and $x^2_j = 1$, in such a way that each basic solution of the sequence is feasible and integer. In the next theorem this result is stated in the case where x^2 is an optimal solution to (SP).

Theorem 3.1.1. *(Balas and Padberg [7]) Let x^1 be a feasible integer (but not optimal) solution to (LSP) associated with the basis B_1. If x^2 is an optimal solution to (SP), then there exists a sequence of adjacent bases $B_{10}, B_{11}, \dots, B_{1p}$ such that $B_{10} = B_1$, $B_{1p} = B_2$ is a basis associated with x^2, and*

(i) the basic solutions $x^{1i} = B_{1i}^{-1}\mathbf{1}$, $i = 0, 1, \dots, p$, are all feasible and integer;

(ii) $cx^{10} \le cx^{11} \le \dots \le cx^{1p}$;

(iii) $p = |J_1 \cap Q_2|$, where J_1 is the index set of nonbasic variables associated with B_1, while $Q_2 = \big\{j \in \{1,\dots,n\} : x^2_j = 1\big\}$.

Remark that, since any vertex of LSP is optimal for some vector c, Theorem 3.1.1 implies that, given any two basic feasible integer solutions x^1 and x^2 to LSP, x^2 can be obtained from x^1 by a sequence of p pivots. Moreover, since $p \le m$, part (iii) of the theorem proves Hirsch conjecture for this special class of linear programs, with respect to integer solutions. A similar statement can also be deduced from a result of Trubin [70]. Trubin, in a totally different fashion, showed that all edges of SP are also edges of LSP. Because two vertices of a polytope are adjacent if they lie on an edge, the *Trubin property* implies the existence of a path containing only integer vertices between any two integer vertices of the feasible set.

A direct consequence of Theorem 3.1.1 is that, given a basic feasible integer solution x^1 to LSP, there is a better integer solution x^2 which is a vertex adjacent to x^1 on LSP. However, identifying such adjacent vertices can be a very difficult task, because set packing problems tend to be highly degenerate. This implies that there are many bases associated to the same solution and that there is a very large number of vertices of LSP adjacent to a given vertex.

In a subsequent paper [8], Balas and Padberg proposed a constructive characterization of adjacency between integer vertices of LSP. In order to state their result, we introduce some preliminary notation. Consider a basic feasible integer solution x^1 to LSP and a basis B_1 associated to x^1. Denote by $N = \{1, \ldots, n\}$ the index set of all variables, and by $I_1 = \{1, \ldots, m\}$ and $J_1 = N \setminus I_1$ the index sets of basic and non-basic variables, respectively. Let a_j be the j-th column of A and define $\bar{a}_j = B_1^{-1} a_j$, $\bar{a}^j = \begin{pmatrix} \bar{a}_j \\ -e_j \end{pmatrix}$, where e_j is the $(n-m)$ unit vector. Define also $Q_1 = \{j \in N | x_j^1 = 1\}$ and $\bar{Q}_1 = N \setminus Q_1$.

Theorem 3.1.2. *(Balas and Padberg [8]) Let x^1 be a basic feasible integer solution to (LSP). Then x^2 is a basic feasible integer solution to (LSP), if and only if there exists $Q \subseteq J_1$ such that*

$$\sum_{j \in Q} \bar{a}_{kj} = \begin{cases} 0 \; or \; 1, & k \in Q_1, \\ 0 \; or \; -1, & k \in I_1 \cap \bar{Q}_1, \end{cases}$$

and

$$x_j^2 = \begin{cases} 1 & j \in Q_2 = Q \cup S, \\ 0 & otherwise, \end{cases}$$

where

$$S = \Big\{k \in Q_1 | \sum_{j \in Q} \bar{a}_{kj} = 0\Big\} \bigcup \Big\{k \in I_1 \cap \bar{Q}_1 | \sum_{j \in Q} \bar{a}_{kj} = -1\Big\}.$$

When this condition holds, then

$$x^2 = x^1 - \sum_{j \in Q} \bar{a}_{kj}.$$

Given a basic feasible integer solution x^1, a set $Q \subseteq J_1$ which satisfies the condition of Theorem 3.1.2 is called *decomposable*, if it can be partitioned into two subsets, such that each one of them satisfies the condition. This notion is useful for the next theorem, which characterizes adjacency of integer vertices of LSP. The equivalence between statements (i) and (iii) was established by Balas and Padberg [8] and implies the one between (i) and (ii), proven earlier by Trubin [70].

Theorem 3.1.3. *(Balas and Padberg [8], Trubin [70]) Let x and y be any two vertices of SP. Let $J(x)$ be the index set of nonbasic variables associated to an arbitrary basis of x. Then the following statements are equivalent:*

(i) x and y are adjacent in LSP;

(ii) x and y are adjacent in SP;

(iii) $Q(y) \cap J(x)$ is not decomposable, where $Q(y) = \{j \in N : y_j = 0\}$.

Another characterization of adjacency was given by Chvátal [20] with regard to the stable set polytope $STAB(G)$. Given two sets X and Y, denote by $X \Delta Y$ their symmetric difference, i.e. the set of elements belonging to one but not both of two given sets.

Theorem 3.1.4. *(Chvátal [20]) Given a graph $G(V, E)$, let x and y be two vertices of $STAB(G)$. Denote by X and Y the stable sets associated to x and y, respectively. Then x and y are adjacent in $STAB(G)$ if and only if the subgraph of G induced by $X \Delta Y$ is connected.*

A direct consequence of this result is a characterizartion of adjacency in the matching polytope. Recall from Section 1.1 that a matching $M \subseteq E$ in a given graph $G(V, E)$ is a set of edges without common nodes. The *matching polyhedron* is the convex hull of incidence vectors of all matchings of G. The *line graph* $L(G)$ of G is defined as a graph where each node of $L(G)$ is associated to an edge of G and two nodes of $L(G)$

are adjacent if and only if the corresponding edges in G share a common endpoint. Therefore, there is a one-to-one correspondence between matchings of G and stable sets in $L(G)$. Moreover, the subgraph of $L(G)$ induced by $X \subseteq E$ is connected if and only if the subgraph obtained from G by removing edges in $E \setminus X$ and deleting all isolated nodes is connected.

Corollary 3.1.1. *(Chvátal [20]) Given a graph $G(V,E)$, let x and y be the incidence vectors of matchings X and Y in G. Then x and y are adjacent in the matching polytope if and only $X \Delta Y$ defines a connected graph.*

In the following, we say that two stable sets of $G(V,E)$ are *adjacent*, if the corresponding incidence vectors are adjacent in $STAB(G)$.

Given a stable sets X, define an *alternating subgraph* of $G(V,E)$ as a bipartite subgraph $H(V', E')$, where nodes of X are connected only to nodes of $V \setminus X$ and such that for any $(i,j) \in E \setminus E'$, if $i \in V'$ and $j \notin V'$, then $j \notin X$. Clearly, for any pair of adjacent stable sets (X,Y), the subgraph of G induced by $X \Delta Y$, which is connected by Theorem 3.1.4, is also an alternating subgraph. This observation connects Chvátal result to an earlier result due to Balinski [10], who related the optimality of a stable set X to the existence of an *augmenting subgraph*, that is an alternating subgraph $H(V', E')$, such that the weight-sum of $V' \setminus X$ exceeds that of $V' \cap X$ (see also Edmonds [31]).

Theorem 3.1.5. *(Balinski [10]) Let X be a stable set in $G(V,E)$ and denote by x its incidence vector. X has maximum weight if and only if it admits no augmenting subgraph.*

3.2 A graphic representation of simplex pivots on the Fractional Stable Set Polytope

Recall from Chapter 2 that, given a basis $B \in \mathcal{B}$, by Theorem 2.2.1, it is possible to associate to B a subgraph G_B of $G(V,E)$, consisting of single nodes, trees and 1-trees with an odd cycle.

The *basic subgraph* G_B is obtained from G by removing the basic edges, that correspond to basic slack variables of $FSTAB(G)$. We have denoted by $C_i(V_i, E_i)$, $i = 1, \dots, k$ the connected components of G_B and we have defined I_0 and I_1 as the index sets of tree and 1-tree components, respectively. For each $i \in I_1$ we have denoted by $\kappa(C_i)$ the unique cycle of 1-tree C_i; for each $j \in I_0$ we have denoted by $\tau(C_j)$ the only nonbasic node of tree C_j.

Such characterization of bases of $FSTAB(G)$ allows us to describe graphically simplex pivots as well. A similar task was tackled by Ikura and Nemhauser for bipartite graphs [47]. Given a basis $B \in \mathcal{B}$, simplex pivots on $FSTAB(G)$ can be characterized in terms of elementary transformations of G_B into $G_{B'}$, where $B' \in \mathcal{B}$ is adjacent to B in $FSTAB(G)$. In this section, we present an overview of all possible transformations of G_B into $G_{B'}$. We denote by (x, y) and (x', y') the basic solutions associated to B and B', respectively. In the following, unless explicitly specified, we don't assume neither B nor B' to be feasible.

3.2.1 Pivoting in a nonbasic edge of a 1-tree

Consider a connected component $C_i(V_i, E_i)$, $i \in I_1$. Suppose we want edge $(i_1, i_2) \in E_i$ to enter the basis. If $(i_1, i_2) \in \kappa(C_i)$, $C'_i(V_i, E_i \setminus (i_1, i_2))$ is a tree; otherwise, it is a subgraph consisting of two connected components, $C_{i1}(V_{i1}, E_{i1})$ and $C_{i2}(V_{i2}, E_{i2})$, such that $V_i = V_{i1} \cup V_{i2}$ and $E_i = E_{i1} \cup E_{i2} \cup \{(i_1, i_2)\}$. Clearly, one of the two components is a 1-tree, while the other one is a tree. Assume w.l.o.g. that C_{i1} is a 1-tree and C_{i2} is a tree. To construct a basis B' adjacent to B, we then need to remove a variable from the basis, in order to obtain a subgraph of $G(V, E)$, satisfying the conditions of Theorem 2.2.2. In the following, we describe separately two different situations that can occurr.

Pivoting out a basic edge. A first possibility consists in pivoting out an edge $(s, t) \notin E_i$, such that $s, t \in V_i$. If $(i_1, i_2) \in \kappa(C_i)$, the distance $d(s, t)$ (i.e. the number of edges of the path connecting s and t) on C'_i must be even. Then, $C''_i(V_i, E_i \setminus (i_1, i_2) \cup (s, t))$ is a 1-tree with an odd cycle that can replace C_i in G_B, yielding a new subgraph $G_{B'}$ (Fig. 3.1). If $(i_1, i_2) \notin \kappa(C_i)$ and $s, t \in V_{i2}$, $d(s, t)$ on C_{i2} must be even, and a new subgraph $G_{B'}$ can be obtained from G_B by replacing C_i with the 1-trees C_{i1} and $C'_{i2}(V_{i2}, E_{i2} \cup (s, t))$, both containing an odd cycle (Fig. 3.2). If $(i_1, i_2) \notin \kappa(C_i)$, $s \in V_{i1}$

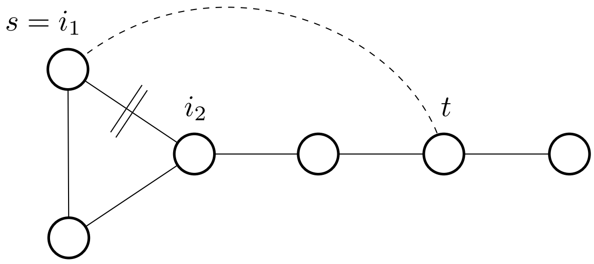

Figure 3.1: $C_i(V_i, E_i)$, $i \in I_1$. $(i_1, i_2) \in \kappa(C_i)$ enters the basis. $(s,t) \notin E_i$: $s, t \in V_i$ exits the basis.

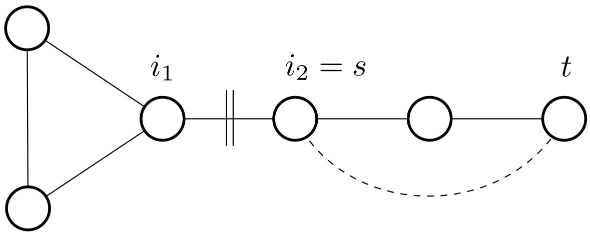

Figure 3.2: $C_i(V_i, E_i)$, $i \in I_1$. $(i_1, i_2) \notin \kappa(C_i)$ enters the basis. $(s,t) \notin E_i$: $s, t \in V_{i2}$ exits the basis.

and $t \in V_{i2}$, $C_i''(V_i, E_i \setminus (i_1, i_2) \cup (s,t))$ is a 1-tree with an odd cycle, that can replace C_i in G_B, yielding the basic sungraph G'_B (Fig. 3.3). In any case, $B' = B \setminus (s,t) \cup (i_1, i_2)$ is a basis of $FSTAB(G)$.

Alternatively, it is possible to pivot in an edge $(s,t) \in \delta(V_i)$. Suppose $(i_1, i_2) \in \kappa(C_i)$ (resp. $(i_1, i_2) \notin \kappa(C_i)$). Then, (s,t) should be such that $s \in V_i$ (resp. $s \in V_{i2}$) and $t \in V_j$, $j \neq i$. This amounts to connecting C_i' (resp. C_{i2}) to C_j, yielding component $C_{ij}(V_i \cup V_j, E_i \setminus (i_1, i_2) \cup E_j \cup (s,t))$ (resp. $C_j'(V_{i2} \cup V_j, E_{i2} \cup E_j \cup (s,t))$). Replacing C_i and C_j with C_{ij} (resp. with C_{i1} and C_j') in G_B yields the basic subgraph $G_{B'}$, associated to basis $B' = B \setminus (i_1, i_2) \cup (s,t)$ (Fig. 3.4 and Fig. 3.5).

Pivoting out a basic node. Suppose $(i_1, i_2) \in \kappa(C_i)$ (resp. $(i_1, i_2) \notin \kappa(C_i)$). It is possible to pivot out a node $z \in V_i$ (resp. $z \in V_{i2}$). This amounts to replacing in G_B the 1-tree C_i with the tree C_i' (resp. with the 1-tree C_{i1} and the tree C_{i2}), where z is the nonbasic node of the new tree component, that is $\tau(C_i') = z$ (resp. $\tau(C_{i2}) = z$). The new basis is $B' = B \setminus (i_1, i_2) \cup \{z\}$ (Fig. 3.6 and 3.7).

Proposition 3.2.1. *Given a basis $B \in \mathcal{B}$, consider a 1-tree component $C_i(V_i, E_i)$,*

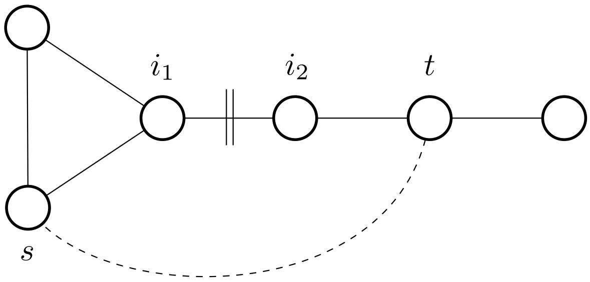

Figure 3.3: $C_i(V_i, E_i)$, $i \in I_1$. $(i_1, i_2) \notin \kappa(C_i)$ enters the basis. $(s, t) \notin E_i$: $s \in V_{i1}$, $t \in V_{i2}$ exits the basis.

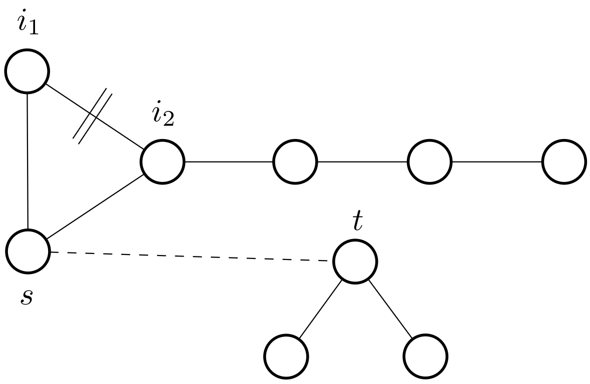

Figure 3.4: $C_i(V_i, E_i)$, $i \in I_1$. $(i_1, i_2) \in \kappa(C_i)$ enters the basis. $(s, t) \notin E_i$: $s \in V_i$, $t \in V_j$, $j \neq i$ exits the basis.

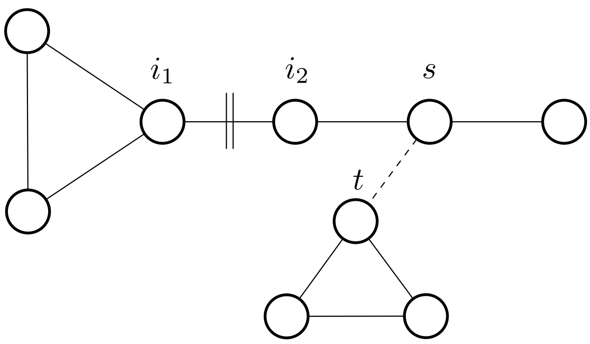

Figure 3.5: $C_i(V_i, E_i)$, $i \in I_1$. $(i_1, i_2) \notin \kappa(C_i)$ enters the basis. $(s, t) \notin E_i$: $s \in V_{i2}$, $t \in V_j$, $j \neq i$ exits the basis.

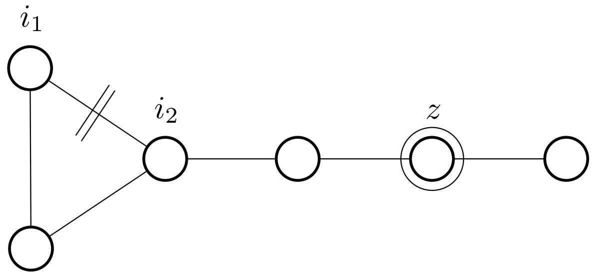

Figure 3.6: $C_i(V_i, E_i)$, $i \in I_1$. $(i_1, i_2) \in \kappa(C_i)$ enters the basis. $z \in V_i$ exits the basis.

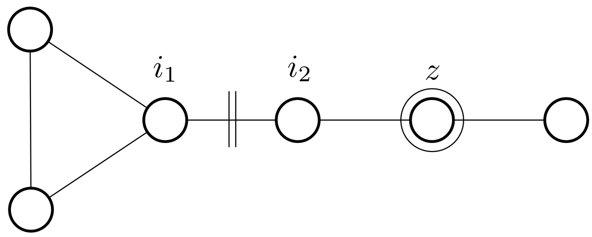

Figure 3.7: $C_i(V_i, E_i)$, $i \in I_1$. $(i_1, i_2) \notin \kappa(C_i)$ enters the basis. $z \in V_{i2}$ exits the basis.

$i \in I_1$ and an edge $(i_1, i_2) \in E_i$ to be pivoted in. The following pivots are those (the only ones) that are degenerate:

(i) pivoting in $(i_1, i_2) \in \kappa(C_i)$, and pivoting out an edge $(s, t) \notin E_i$ s.t. $s, t \in V_i$ and $d(s, t)$ on C_i' is even;

(ii) pivoting in $(i_1, i_2) \notin \kappa(C_i)$, and pivoting out an edge $(s, t) \notin E_i$ s.t. $s, t \in V_{i2}$ and $d(s, t)$ on C_{i2} is even;

(iii) pivoting in $(i_1, i_2) \notin \kappa(C_i)$, and pivoting out an edge $(s, t) \notin E_i$ s.t. $s \in V_{i1}$ and $t \in V_{i2}$;

(iv) pivoting in $(i_1, i_2) \in \kappa(C_i)$ (resp. $(i_1, i_2) \notin \kappa(C_i)$), and pivoting out an edge (s, t) s.t. $s \in V_i$ (resp. $s \in V_{i2}$), $t \in V_j$, $j \neq i$, $j \in I_1$.

Proof. (i), (ii) and (iii) correspond to the pivots represented in Fig. 3.1, Fig. 3.2 and Fig. 3.3, respectively. Observe that only nodes in V_i are involved in the trasformation of G_B. Moreover, C_i is replaced with one or more 1-tree components. Therefore, by Theorem 2.2.3, the new basis B' is such that $x'_{B'} = x_B$.

Analogously, to prove (iv), observe that $j \in I_1$ implies that only nodes belonging to 1-trees (namely, those in $V_i \cup V_j$) are involved in the transformation of G_B (see, for example, Fig. 3.5). Moreover, in $G_{B'}$ they still belong to 1-trees, while the rest of the subgraph is unchanged. Therefore, by Theorem 2.2.3, the new basis B' is such that $x'_{B'} = x_B$.

To prove that there are no other degenerate pivots, recall that pivoting in a nonbasic edge and pivoting out a basic node amounts to trasforming a 1-tree (or part of it) into a tree (see, for example, Figures 3.6 and 3.7). Therefore, those components of $x'_{B'}$ that are indexed by nodes of the new tree component are, by Theorem 2.2.3, 0-1 valued. This implies $x'_{B'} \neq x_B$. The last pivot to be considered is the one where a nonbasic edge enters the basis, and a basic edge $(s,t) \in \delta(V_i)$ exits the basis, s.t. $t \in V_j$, $j \in I_0$. In this tranformation of G_B, a non-empty subset of the nodes in V_i become part of a tree component. These nodes will be then, by Theorem 2.2.3, 0-1 valued in $x'_{B'}$, showing that $x'_{B'} \neq x_B$. □

Consider a pivot such that a nonbasic edge (i_1, i_2) of 1-tree C_i enters the basis. We call the pivot *trivially infeasible*, if the new basis $B' \in \mathcal{B}$ is such that the corresponding basic solution (x', y') is infeasible and $y'_{i_1 i_2} < 0$.

Proposition 3.2.2. *Given a basis $B \in \mathcal{B}$, consider a 1-tree component $C_i(V_i, E_i)$, $i \in I_1$ and an edge $(i_1, i_2) \in E_i$ s.t. $(i_1, i_2) \in \kappa(C_i)$ (resp. $(i_1, i_2) \notin \kappa(C_i)$). The following pivots yield a new basis $B' \in \mathcal{B}$, which is trivially infeasible:*

(i) pivoting in (i_1, i_2), and pivoting out a node $z \in V_i$ (resp. $z \in V_{i2}$) s.t. $d(z, i_2)$ on $C'_i(V_i, E_i \setminus (i_1, i_2))$ (resp. on C_{i2}) is odd;

(ii) pivoting in (i_1, i_2), and pivoting out an edge (s,t) s.t. $s \in V_i$ (resp. $s \in V_{i2}$), $t \in V_j$, $j \in I_0$, and $d(\tau(C_j), i_2)$ on $C_{ij}(V_i \cup V_j, E_i \setminus (i_1, i_2) \cup E_j \cup (s,t))$ (resp. on $C'_j(V_{i2} \cup V_j, E_{i2} \cup E_j \cup (s,t))$) is odd.

Proof. *(i)* The new component C'_i (resp. C_{i2}) is a tree s.t. $\tau(C'_i) = z$ (resp. $\tau(C_{i2}) = z$). Therefore, in the new tree component, all nodes at odd distance from z will index 1-valued components of $x'_{B'}$ (this follows from the fact that edges of a basic subgraph are nonbasic and therefore 0-valued). Because $d(z, i_2)$ is odd, $x'_{i_2} = 1$. This means that $x'_{i_1} + x'_{i_2} = \frac{1}{2} + 1$ and that $y'_{i_1 i_2} = -\frac{1}{2}$, implying that (x', y') is trivially infeasible.

(ii) The new component C_{ij} (resp. C'_j) is a tree s.t. $d(\tau(C_{ij}), i_2)$ (resp. $d(\tau(C'_j), i_2)$) is odd, implying that $x'_{i_2} = 1$. Therefore, $x'_{i_1} + x'_{i_2} = \frac{1}{2} + 1$ and $y'_{i_1 i_2} = -\frac{1}{2}$, showing that (x', y') is trivially infeasible. □

In the following, given a tree T and a node $i \in V(T)$, we denote by $H(i,T)$ the set of nodes of T that are at odd distance from i, i.e. $H(i,T) = \{j \in V(T) \colon d(i,j) \text{ on } T_i \text{ is odd}\}$.

Proposition 3.2.3. *Given a feasible basis $B \in \mathcal{B}$, consider a 1-tree component $C_i(V_i, E_i)$, $i \in I_1$ and an edge $(i_1, i_2) \in E_i$ to be pivoted in. The following pivots are those that are non-degenerate and yield a feasible basis $B' \in \mathcal{B}$:*

(i) *pivoting in $(i_1, i_2) \in \kappa(C_i)$, and pivoting out a node $z \in V_i \setminus H(i_2, C'_i)$ s.t., $H(i_2, C'_i)$ is a stable set of $G(V,E)$ and there is no edge $(v,u) \in \delta(H)$ with $v \in H$ and $u \in V_r$, $r \in I_1$;*

(ii) *pivoting in $(i_1, i_2) \notin \kappa(C_i)$, and pivoting out a node $z \in V_{i2} \setminus H(i_2, C_{i2})$ s.t., $H(i_2, C_{i2})$ is a stable set of $G(V,E)$, and there is no edge $(v,u) \in \delta(H)$ s.t. $v \in H$ and $u \in V_r$, $r \in I_1$;*

(iii) *pivoting in $(i_1, i_2) \in \kappa(C_i)$, and pivoting out an edge (s,t) with $s \in V_i$, $t \in V_j$, $j \in I_0$, such that $s \in H(i_2, C'_i)$, $H(i_2, C'_i)$ is a stable set of $G(V,E)$, and there is no edge $(v,u) \in \delta(H)$ s.t. $v \in H$ and $u \in V_r$, $r \in I_1$;*

(iv) *pivoting in $(i_1, i_2) \notin \kappa(C_i)$, and pivoting out an edge (s,t) with $s \in V_{i2}$, $t \in V_j$, $j \in I_0$, such that $s \in H(i_2, C_{i2})$, $H(i_2, C_{i2})$ is a stable set of $G(V,E)$, and there is no edge $(v,u) \in \delta(H)$ s.t. $v \in H$ and $u \in V_r$, $r \in I_1$.*

Proof. Pivots (i)-(iv) are not degenerate and not trivially infeasible, by Propositions 3.2.1 and 3.2.2, respectively. Consider case (i). In $G_{B'}$, C_i has been replaced by the tree component C'_i, s.t. $\tau(C'_i) = z$. Because the edges of $G_{B'}$ are 0-valued, $H(i_2, C'_i)$ includes all nodes that were 1/2-valued in x_B and that are 1-valued in $x'_{B'}$. To prove feasibility of (x', y'), we have to show that $y'_{uv} \geq 0 \;\; \forall (u,v) \in E$. By feasibility of B, all edges in $E(V \setminus V_i)$ index nonnegative slack variables. As $H(i_2, C'_i)$ is a stable set, $E(H) = \emptyset$. We show that $\forall (v,u) \in \delta(H)$, (v,u) is s.t. $v \in H$, $u \in V_r$, $r \in I_0$ and $d(u, \tau(C_r))$ on C_r is even. Indeed, by hypothesis, C_r cannot be a 1-tree. If $r \in I_0$ and $d(u, \tau(C_r))$ on C_r is odd, it would follow $x_u = x'_u = 1$ and $y_{uv} = -\frac{1}{2}$, contradicting feasibility of (x,y). This proves that nodes in $H(i_2, C'_i)$ can only be connected to nodes that are 0-valued in x'. Feasibility of B' for cases (ii)-(iv) can be proven analogously to case (i).

We now prove that there are no more non-degenerate pivots where (i_1, i_2) enters the basis, which yield a new feasible basis B'. First, observe that all pivots where (i_1, i_2) enters the basis and $(s,t) \notin E_i$ with $s, t \in V_i$ exits the basis are degenerate (see Figures 3.1-3.3). Similarly, those pivots where (i_1, i_2) enters the basis and $(s,t) \notin E_i$ with $s \in V_i$ and $t \in V_j$, $j \in I_1$ is pivoted out are degenerate (see, for example, Fig. 3.5). Suppose we pivot in $(i_1, i_2) \in \kappa(C_i)$ and we pivot out a node $z \in V_i$. We show that conditions stated in (i) are necessary for feasibility (this can be analogously shown for (ii)). Indeed, $z \in H(i_2, C'_i)$ would imply that B' is trivially infeasible, as i_1 and i_2 would be both 1-valued in the new basic solution x'. If $H(i_2, C'_i)$ is not a stable set, (x', y') violates all the edge constraints indexed by $E(H)$. Finally, if there exists an edge $(v, u) \in \delta(H)$ s.t. $v \in H$ and $u \in V_r$, $r \in I_1$, it follows that $x'_v + x'_u = 1 + \frac{1}{2}$ and $y'_{vu} = -\frac{1}{2}$, implying infeasibility of (x', y'). Analogously, it can be proven the necessity of the conditions stated in (iii) and (iv) for the case where we pivot in (i_1, i_2) and we pivot out an edge $(s,t) \notin E_i$ s.t. $s \in V_i$ and $t \in V_j$, $j \in I_0$. $\square$

3.2.2 Pivoting in a nonbasic edge of a tree

Consider a connected component $C_i(V_i, E_i)$, $i \in I_0$. Suppose we want edge $(i_1, i_2) \in E_i$ to enter the basis. Then $C'_i(V_i, E_i \setminus (i_1, i_2))$ consists of two tree components, $C_{i1}(V_{i1}, E_{i1})$ and $C_{i2}(V_{i2}, E_{i2})$, such that $V_i = V_{i1} \cup V_{i2}$ and $E_i = E_{i1} \cup E_{i2} \cup (i_1, i_2)$. Assume w.l.o.g. that $i_1 \in V_{i1}$, $i_2 \in V_{i2}$, $\tau(C_i) \in V_{i1}$. To construct a basis B' adjacent to B, we need to remove a variable from the basis, in order to obtain a subgraph of $G(V, E)$, satisfying the conditions of Theorem 2.2.2. In the following, we describe separately two different situations that can occurr.

Pivoting out a basic edge. If the edge to be pivoted out $(s,t) \notin E_i$ connects two nodes of the tree, such that $s, t \in V_{i2}$, and $d(s,t)$ on C_{i2} is even, then the new component $C'_{i2}(V_{i2}, E_{i2} \cup (s,t))$ is a 1-tree with an odd cycle, and the subgraph $G_{B'}$, obtained from G_B by replacing C_i with C_{i1} and C'_{i2}, satisfies the conditions of Theorem 2.2.2 (Fig. 3.8). If (s,t) is such that $s \in V_{i1}$ and $t \in V_{i2}$ the new component $C''_i(V_i, E_i \setminus (i_1, i_2) \cup (s,t))$ is a tree, and the subgraph $G_{B'}$ obtained from G_B by replacing C_i with C''_i satisfies the conditions of Theorem 2.2.2 (Fig. 3.9). Finally, if

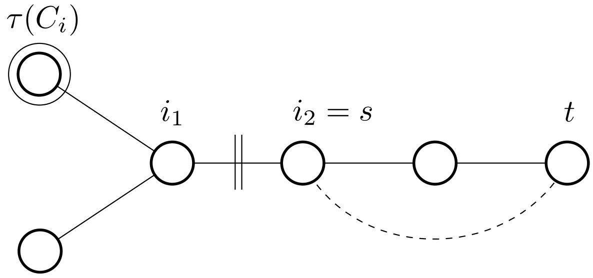

Figure 3.8: $C_i(V_i, E_i)$, $i \in I_0$. $(i_1, i_2) \in E_i$ enters the basis. $(s,t) \notin E_i$: $s, t \in V_{i2}$ exits the basis.

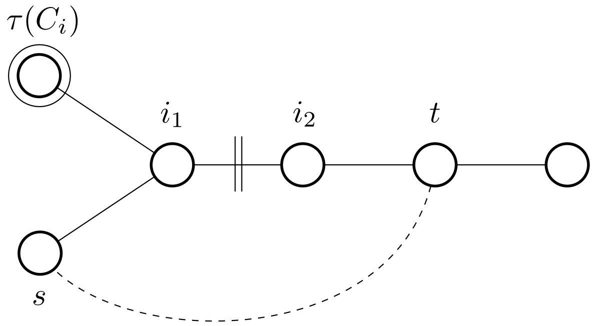

Figure 3.9: $C_i(V_i, E_i)$, $i \in I_0$. $(i_1, i_2) \in E_i$ enters the basis. $(s,t) \notin E_i$: $s \in V_{i1}$, $t \in V_{i2}$ exits the basis.

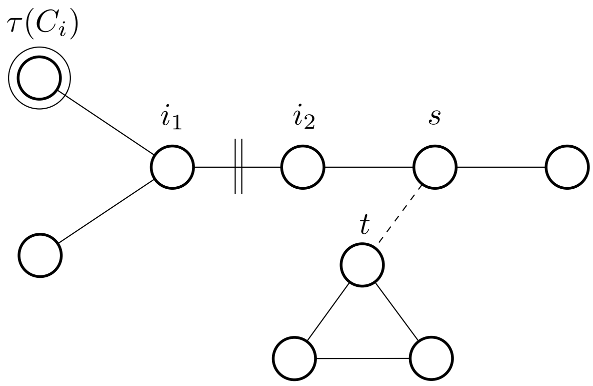

Figure 3.10: $C_i(V_i, E_i)$, $i \in I_0$. $(i_1, i_2) \in E_i$ enters the basis. $(s,t) \notin E_i$: $s \in V_{i2}$, $t \in V_j$, $j \neq i$ exits the basis.

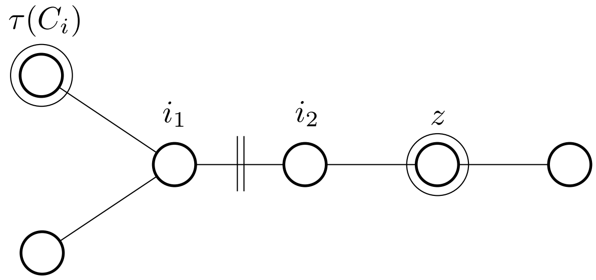

Figure 3.11: $C_i(V_i, E_i)$, $i \in I_0$. $(i_1, i_2) \in E_i$ enters the basis. $z \in V_{i2}$ exits the basis.

(s, t) is such that $s \in V_{i2}$ and $t \in V_j$, $j \neq i$, pivoting out (s, t) amounts to connecting C_{i2} to C_j, which yields component $C'_j(V_{i2} \cup V_j, E_{i2} \cup E_j \cup (s, t))$. Replacing C_i and C_j with C_{i1} and C'_j in G_B defines the basic subgraph $G_{B'}$ (Fig. 3.10). In any case, $B' = B \setminus (s, t) \cup (i_1, i_2)$ is a basis of $FSTAB(G)$ associated to $G_{B'}$.

Pivoting out a basic node. In this case, it is possible to pivot out a node $z \in V_{i2}$. This amounts to replacing in G_B the tree C_i with the trees C_{i1} and C_{i2}, where $\tau(C_{i2}) = z$ (Fig. 3.11).

Proposition 3.2.4. *Given a basis $B \in \mathcal{B}$, consider a tree component $C_i(V_i, E_i)$, $i \in I_0$ and an edge $(i_1, i_2) \in E_i$ to be pivoted in. The following pivots are those (the only ones) that are degenerate:*

(i) pivoting in (i_1, i_2), and pivoting out an edge (s, t) s.t. $s \in V_{i2}$, $t \in V_j$, $j \neq i$, $j \in I_0$ and $d(\tau(C_i), \tau(C_j))$ on $C_{ij}(V_i \cup V_j, E_i \cup E_j \cup (s, t))$ is even;

(ii) pivoting in (i_1, i_2), and pivoting out an edge $(s, t) \neq (i_1, i_2)$ s.t. $s \in V_{i1}$, $t \in V_{i2}$, and $d(s, t)$ on C_i is odd;

(iii) pivoting in (i_1, i_2), and pivoting out a node $z \in V_{i2}$, s.t. $d(\tau(C_i), z)$ on C_i is even.

Proof. (i). In this case, components C_i and C_j are replaced in G_B with components C_{i1} and $C'_j(V_j \cup V_{i2}, E_j \cup E_{i2} \cup (s, t))$, which are both trees. Clearly, variables indexed by nodes of $V_{i1} \cup V_j$ retain their values in x'. We have to prove that, $\forall\, v \in V_{i2}$, $d(v, \tau(C_i))$ on C_i and $d(v, \tau(C_j))$ on C'_j have the same parity. Indeed, given a tree component of a basic subgraph, all nodes at even (resp. odd) distance from the nonbasic node of the tree are 0-valued (resp. 1-valued) in the corresponding basic solution. Recall also that, given two nodes of a tree, there exists only one path connecting them on the tree. Therefore, given the tree C_i of G_B, if $d(s,\tau(C_i))$ on C_i is even (resp. odd), all the nodes in $\{v \in V_{i2}$ s.t. $d(s,v)$ on C_i is even$\}$ are 0-valued (resp. 1-valued), while all the nodes in $\{v \in V_{i2}$ s.t. $d(s,v)$ on C_i is odd$\}$ are 1-valued (resp. 0-valued). Analogously, given the tree C'_j of $G_{B'}$, if $d(s,\tau(C_j))$ on C'_j is even (resp. odd), all the nodes in $\{v \in V_{i2}$ s.t. $d(s,v)$ on C'_j is even$\}$ are 0-valued (resp. 1-valued), while all the nodes in $\{v \in V_{i2}$ s.t. $d(s,v)$ on C'_j is odd$\}$ are 1-valued (resp. 0-valued). Observe also that $\forall\, v \in V_{i2}$, $d(v,s)$ on both C_i and C'_j is equal to $d(v,s)$ on C_{i2}. Therefore, we have only to prove that $d(s,\tau(C_i))$ on C_i and $d(s,\tau(C_j))$ on C'_j have the same parity. This easily follows from the fact that $d(\tau(C_i),\tau(C_j))$ on $C_{ij}(V_i \cup V_j, E_i \cup E_j \cup (s,t))$, which is even by hypothesis, can be expressed as the sum of $d(s,\tau(C_i))$ on C_i and $d(s,\tau(C_j))$ on C'_j. Consequently, the two terms have the same parity.

(ii). Pivoting out (s,t) amounts to connecting back C_{i1} and C_{i2}, that yields the new tree component $C''_i(V_i, E_i \cup (i_1,i_2) \setminus (s,t))$. Because $d(s,t)$ on C_i is odd, the slack variable y_{st}, associated to edge (s,t) leaving the basis, is 0-valued in (x,y) (Fig. 3.9). This shows that we have performed a degenerate pivot.

(iii). Being $d(\tau(C_i),z)$ on C_i even, $x_z = 0$. Pivoting out a 0-valued basic node clearly yields a degenerate pivot.

To prove that there are no more degenerate pivots where (i_1,i_2) enters the basis, consider first the case where a node $z \in V_{i2}$ is pivoted out. If $d(\tau(C_i),z)$ on C_i is odd, all 0-valued nodes of V_{i2} become 1-valued, and viceversa. If, instead, we pivot out an edge $(s,t) \notin E_i$, connecting two nodes of V_{i2} at even distance in C_{i2}, the resulting component is a 1-tree. This implies that nodes of V_{i2} become $\frac{1}{2}$-valued. When $(s,t) \neq (i_1,i_2)$ is s.t. $s \in V_{i1}$, $t \in V_{i2}$ and $d(s,t)$ on C_i is even, it follows that $y_{st} \in \{-1,1\}$ and the pivot swaps the assignments of zeros and ones on nodes of V_{i2}. If (s,t) is s.t. $s \in V_{i2}$, $t \in V_j$, $j \neq i$, we introduce in the basic subgraph component $C'_j(V_j \cup V_{i2}, E_j \cup E_{i2} \cup (s,t))$. If $j \in I_1$, nodes of V_{i2} become $\frac{1}{2}$-valued. If $j \in I_0$ and $d(\tau(C_i),\tau(C_j))$ on $C_{ij}(V_i \cup V_j, E_i \cup V_j \cup (s,t))$ is odd, then $d(s,\tau(C_i))$ on C_i and $d(s,\tau(C_j))$ on C'_j have different parity and, as a consequence, all nodes in V_{i2} change their values from 0 to 1, and viceversa. □

Consider a pivot such that a nonbasic edge (i_1,i_2) of tree C_i enters the basis. Again, we define a pivot to be *trivially infeasible* if the new basis $B' \in \mathcal{B}$ is such that the corresponding basic solution (x', y') is infeasible and $y'_{i_1 i_2} < 0$.

Proposition 3.2.5. *Given a basis $B \in \mathcal{B}$, consider a tree component $C_i(V_i, E_i)$, $i \in I_0$ and suppose that an edge $(i_1, i_2) \in E_i$, s.t. $d(\tau(C_i), i_1)$ on C_i is odd, enters the basis. The following pivots yield a basis $B' \in \mathcal{B}$ that is trivially infeasible:*

(i) pivoting in (i_1, i_2), and pivoting out an edge (s, t) s.t. $s \in V_{i2}$, $t \in V_j$, $j \in I_1$;

(ii) pivoting in (i_1, i_2), and pivoting out an edge (s, t) s.t. $s \in V_{i2}$, $t \in V_j$, $j \neq i$, $j \in I_0$ and $d(i_2, \tau(C_j))$ on $C'_j(V_j \cup V_{i_2}, E_j \cup E_{i2} \cup (s, t))$ is odd;

(iii) pivoting in (i_1, i_2), and pivoting out an edge $(s, t) \notin E_i$ such that $s, t \in V_{i2}$, and $d(s, t)$ on C_{i2} is even;

(iv) pivoting in (i_1, i_2), and pivoting out an edge $(s, t) \notin E_i$ such that $s \in V_{i1}$, $t \in V_{i2}$ and $d(s, t)$ on C_i is even;

(v) pivoting in (i_1, i_2), and pivoting out a node $z \in V_{i2}$, s.t. $d(i_2, z)$ on C_{i2} is odd.

Proof. First, observe that $d(\tau(C_i), i_1)$ on C_i being odd, implies $x_{i_1} = 1$. (i). In G_B, we are replacing C_i and C_j with the tree C_{i1} and the 1-tree $C'_j(V_j \cup V_{i2}, E_j \cup E_{i2} \cup (s, t))$ (Fig. 3.10). As a consequence, all nodes of V_{i2} become $\frac{1}{2}$-valued, while the other ones retain their values. Therefore $x'_{i1} + x'_{i2} = 1 + \frac{1}{2}$, and $y'_{i_1 i_2} = -\frac{1}{2}$, implying infeasibility of B'.

(ii). In G_B, we are replacing C_i and C_j with the trees C_{i1} and $C'_j(V_j \cup V_{i2}, E_j \cup E_{i2} \cup (s, t))$. Because $\tau(C'_j) = \tau(C_j)$ and, by hypothesis, $d(i_2, \tau(C_j))$ on C'_j is odd, it follows that $x'_{i2} = 1$. Therefore $x'_{i1} + x'_{i2} = 1 + 1$, and $y'_{i_1 i_2} = -1$, implying infeasibility of B'.

(iii). In G_B, we are replacing C_i with the tree C_{i1} and the 1-tree $C'_{i2}(V_{i2}, E_{i2} \cup (s, t))$ (Fig. 3.8). As a consequence, all nodes of V_{i2} become $\frac{1}{2}$-valued, while the other ones retain their values. Therefore $x'_{i1} + x'_{i2} = 1 + \frac{1}{2}$, and $y'_{i_1 i_2} = -\frac{1}{2}$, implying infeasibility of B'.

(iv). In G_B, C_i is replaced with $C''_i(V_i, E_i \setminus (i_1, i_2) \cup (s, t))$. Notice that $d(s, t)$ on C_i being even, all nodes of V_{i2} have their values switched from 0 to 1, and viceversa. Therefore $x'_{i1} + x'_{i2} = 1 + 1$, and $y'_{i_1 i_2} = -1$, implying infeasibility of B'.

(v). In G_B, C_i is replaced with C_{i1} and C_{i2} and it is set $\tau(C_{i2}) = z$ (Fig. 3.11). Notice that $d(i_2, z)$ on C_{i2} being odd, all nodes of V_{i2} have their values switched from 0 to

1, and viceversa. Therefore $x'_{i1} + x'_{i2} = 1 + 1$, and $y'_{i_1 i_2} = -1$, implying infeasibility of B'. □

Proposition 3.2.6. *Given a feasible basis $B \in \mathcal{B}$, consider a tree component $C_i(V_i, E_i)$, $i \in I_0$ and suppose that an edge $(i_1, i_2) \in E_i$, s.t. $d(\tau(C_i), i_2)$ on C_i is odd, enters the basis. Define $H = H(i_2, C_{i2})$. The following pivots are those that are non-degenerate and yield a feasible basis $B' \in \mathcal{B}$:*

(i) pivoting in (i_1, i_2), and pivoting out a node $z \in V_{i2}$ s.t. $z \notin H$, H is a stable set of $G(V, E)$, and there is no edge $(v, u) \in \delta(H)$ s.t. $v \in H$ and $u \in V_r$, $r \in I_1$ or $r \in I_0$ and $d(u, \tau(C_r))$ on C_r is odd;

(ii) pivoting in (i_1, i_2), and pivoting out an edge $(s, t) \notin E_i$ such that $s \in V_{i1}$, $t \in V_{i2}$, $d(s, t)$ on C_i is even, H is a stable set of $G(V, E)$, and there is no edge $(v, u) \in \delta(H)$ s.t. $v \in H$ and $u \in V_r$, $r \in I_1$ or $r \in I_0$ and $d(u, \tau(C_r))$ on C_r is odd;

(iii) pivoting in (i_1, i_2), and pivoting out an edge (s, t) s.t. $s \in V_{i2}$, $t \in V_j$, $j \in I_0$, $d(t, \tau(C_j))$ on C_j is even, $s \in H$, H is a stable set of $G(V, E)$, and there is no edge $(v, u) \in \delta(H)$ s.t. $v \in H$ and $u \in V_r$, $r \in I_1$ or $r \in I_0$ and $d(u, \tau(C_r))$ on C_r is odd;

(iv) pivoting in (i_1, i_2), and pivoting out an edge (s, t) s.t. $s, t \in V_{i2}$, $d(s, t)$ on C_i is even, and there is no edge $(v, u) \in \delta(H)$ s.t. $v \in H$ and $u \in V_r$, $r \in I_0$ and $d(u, \tau(C_r))$ is odd;

(v) pivoting in (i_1, i_2), and pivoting out an edge (s, t) s.t. $s \in V_{i2}$, $t \in V_j$, $j \in I_1$, and there is no edge $(v, u) \in \delta(H)$ s.t. $v \in H$ and $u \in V_r$, $r \in I_0$ and $d(u, \tau(C_r))$ is odd.

Proof. Pivots (i)-(v) are not trivially infeasible, by Proposition 3.2.5, because $d(\tau(C_i), i_2)$ on C_i is odd. By Proposition 3.2.4, it immediately follows that pivots (i), (ii), (iv) and (v) are non-degenerate. (iii) is also a non-degenerate pivot, because $d(\tau(C_i), \tau(C_j))$ on $C_{ij}(V_i \cup V_j, E_i \cup E_j \cup (s, t))$ is equal to $d(\tau(C_i), s)$ on C_i, which is even, plus $d(s, t) = 1$ on C_i, plus $d(t, \tau(C_j))$ on C_j, which is even.

First of all, observe that all nodes of H index 0-valued components of x, while all nodes of $V_{i2} \setminus H$ index 1-valued components of x. This is implied by the fact that all edges of a basic subgraph are nonbasic, therefore 0-valued.

Consider case (i). In $G_{B'}$, C_i has been replaced by the tree components C_{i1} and C_{i2}, s.t. $\tau(C_{i2}) = z$. As $z \notin H$, all nodes of H index now 1-valued components of x', whereas nodes in $V_{i2} \setminus H$ index 0-valued components of x'. Moreover, all slack variables y'_{ij} are nonnegative, because, by hypothesis, H is a stable set and nodes of H are only connected to 0-valued nodes of tree components. This proves that pivot (i) is non-degenerate and feasible, and that all other pivots where (i_1, i_2) enters the basis and a node $z \in V_{i2}$ exits the basis are either degenerate or they yield an infeasible basis .

Similarly, pivot (ii) (resp. (iii)) can be shown to be the only non-degenerate pivot leading to a new feasible basis, s.t. (i_1, i_2) enters the basis an edge $(s, t) \notin E_i$ with $s \in V_{i1}$ and $t \in V_{i2}$ (resp. $(s, t) \notin E_i$ with $s \in V_{i2}$, $t \in V_j$, $j \in I_0$) exits the basis.

Consider case (iv). In $G_{B'}$, C_i has been replaced by the tree component C_{i1} and the 1-tree component $C'_{i2}(V_{i2}, E_{i2} \cup (s, t))$. All nodes of V_{i2} index now 1/2-valued components of x'. Moreover, all slack variables y'_{ij} are nonnegative because, by hypothesis, there is no edge connecting a node of H to any 1-valued node of a tree component. This proves that pivot (iv) is feasible, and that all other pivots where (i_1, i_2) enters the basis and an edge (s, t) s.t. $s, t \in V_{i2}$ exits the basis yield an infeasible basis.

Feasibility of pivot (v) can be proven following the same line of reasoning of case (iv). Analogously, all other pivots where (i_1, i_2) enters the basis and an edge $(s, t) \notin E_i$ s.t. $s \in V_{i2}$ and $t \in V_j$, $j \in I_1$ exits the basis lead to an infeasible basis. □

3.2.3 Pivoting in the nonbasic node of a tree

Consider a connected component $C_i(V_i, E_i)$, $i \in I_0$. Suppose we want node $z = \tau(C_i)$ to enter the basis . To construct a basis B' adjacent to B, we need to remove a variable from the basis, in order to obtain a subgraph of $G(V, E)$ satisfying the conditions of Theorem 2.2.2. Again, we consider two different cases.

Pivoting out a basic edge. If the edge to be pivoted out $(s, t) \notin E_i$ is such that $s, t \in V_i$ and $d(s, t)$ on C_i is even, the new component $C''_i(V_i, E_i \cup (s, t))$ is a 1-tree with an odd cycle and the subgraph $G_{B'}$, obtained from G_B by replacing C_i with C''_i, satisfies the conditions of Theorem 2.2.2. Therefore, $B' = B \setminus (s, t) \cup \{z\}$ is a basis of $FSTAB(G)$

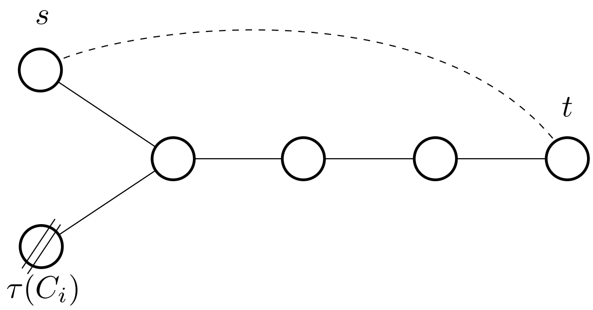

Figure 3.12: $C_i(V_i, E_i)$, $i \in I_0$. $\tau(C_i)$ enters the basis. $z' \in V_i$, $z' \neq \tau(C_i)$ exits the basis.

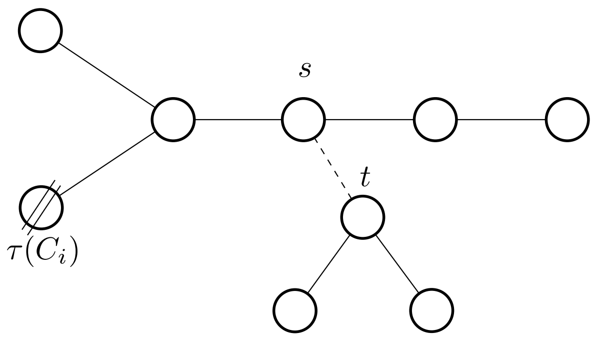

Figure 3.13: $C_i(V_i, E_i)$, $i \in I_0$. $\tau(C_i)$ enters the basis. $(s, t) \notin E_i$, s.t. $s, t \in V_i$ exits the basis.

associated to $G_{B'}$ (Fig. 3.12).

If (s, t) is such that $s \in V_i$ and $t \in V_j$, $j \neq i$, pivoting out (s, t) amounts to connecting C_i to C_j, yielding component $C_{ij}(V_i \cup V_j, E_i \cup E_j \cup (s, t))$. Replacing C_i and C_j with C_{ij} in G_B yields the basic subgraph $G_{B'}$, associated to basis $B' = B \setminus \{z\} \cup (s, t)$ (Fig. 3.13).

Pivoting out a basic node. In this case, it is possible to pivot out a node $z' \in V_i$, $z' \neq z$. This implies that, in $G_{B'}$, the only nonbasic node of the tree C_i is z', that is $\tau(C_i) = z'$ (Fig. 3.14).

Proposition 3.2.7. *Given a basis $B \in \mathcal{B}$, consider a tree component $C_i(V_i, E_i)$, $i \in I_0$ and its nonbasic node $\tau(C_i)$. The following pivots are degenerate:*

(i) pivoting in $\tau(C_i)$ and pivoting out a node $z' \neq \tau(C_i)$ s.t. $d(\tau(C_i), z')$ on C_i is even;

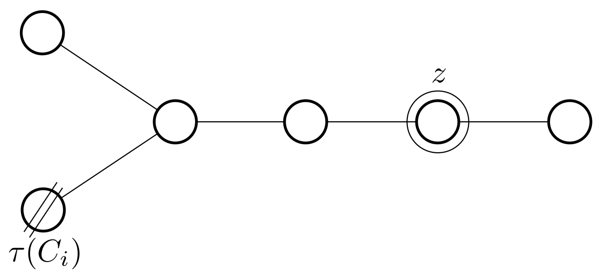

Figure 3.14: $C_i(V_i, E_i)$, $i \in I_0$. $\tau(C_i)$ enters the basis. $(s,t) \notin E_i$, s.t. $s \in V_i$, $t \in V_j$, $j \neq i$ exits the basis.

(ii) *pivoting in $\tau(C_i)$ and pivoting out an edge (s,t) s.t. $s \in V_i$, $t \in V_j$, $j \neq i$, $j \in I_0$ and $d(\tau(C_i), \tau(C_j))$ on $C_{ij}(V_i \cup V_j, E_i \cup E_j \cup (s,t))$ is even.*

Proof. (i). In $G_{B'}$, C_i is such that its only nonbasic node is now z'. Because $d(\tau(C_i), z')$ on C_i is even, we are swapping two 0-valued variables, yielding a degenerate pivot.

(ii). In this case, components C_i and C_j are replaced in G_B with the tree C_{ij}, s.t. $\tau(C_{ij})= \tau(C_j)$. Clearly, variables indexed by nodes of $V \setminus V_i$ retain their values in x'. We have to prove that, $\forall\ v \in V_i$, $d(v, \tau(C_i))$ on C_i and $d(v, \tau(C_j))$ on C_{ij} have the same parity. We follow the same reasoning used in proof of Proposition 3.2.4(i). $d(s, \tau(C_i))$ on C_i and $d(s, \tau(C_j))$ on C_{ij} have the same parity because, by hypothesis, $d(\tau(C_i), \tau(C_j))$ on $C_{ij}(V_i \cup V_j, E_i \cup E_j \cup (s,t))$ is even. Therefore, $x'_s = x_s$ and consequently $x'_v = x_v\ \forall\ v \in V_i$. □

Given a tree T and a node $i \in V(T)$, define $J(i,T)$ as the set of nodes of T that are at even distance from i, i.e. $J(i,T) = \{j \in V(T) \colon d(i,j) \text{ on } T_i \text{ is even}\}$.

Proposition 3.2.8. *Given a feasible basis $B \in \mathcal{B}$, consider a tree component $C_i(V_i, E_i)$, $i \in I_0$ and suppose that node $\tau(C_i)$ enters the basis. Define $J = J(\tau(C_i)), C_i)$. The following pivots are those that are non-degenerate and yield a feasible basis $B' \in \mathcal{B}$:*

(i) *pivoting in $\tau(C_i)$, and pivoting out a node $z' \in V_i$, s.t. $z' \notin J$, J is a stable set of $G(V,E)$, and there is no edge $(v,u) \in \delta(J)$ s.t. $v \in J$ and $u \in V_r$, $r \in I_1$ or $r \in I_0$ and $d(u, \tau(C_r))$ on C_r is odd;*

(ii) *pivoting in* $\tau(C_i)$*, and pivoting out an edge* (s,t) *s.t.* $s \in V_i$*,* $t \in V_j$*,* $j \in I_0$*,* $d(t, \tau(C_j))$ *on* C_j *is even,* $s \in J$*,* J *is a stable set of* $G(V,E)$*, and there is no edge* $(v,u) \in \delta(J)$ *s.t.* $v \in J$ *and* $u \in V_r$*,* $r \in I_1$ *or* $r \in I_0$ *and* $d(u, \tau(C_r))$ *on* C_r *is odd;*

(iii) *pivoting in* $\tau(C_i)$*, and pivoting out an edge* (s,t) *s.t.* $s,t \in V_i$*,* $d(s,t)$ *on* C_i *is even, and there is no edge* $(v,u) \in \delta(J)$ *s.t.* $v \in J$ *and* $u \in V_r$*,* $r \in I_0$ *and* $d(u, \tau(C_r))$ *is odd;*

(iv) *pivoting in* $\tau(C_i)$*, and pivoting out an edge* (s,t) *s.t.* $s \in V_i$*,* $t \in V_j$*,* $j \in I_1$*, and there is no edge* $(v,u) \in \delta(J)$ *s.t.* $v \in J$ *and* $u \in V_r$*,* $r \in I_0$ *and* $d(u, \tau(C_r))$ *is odd.*

Proof. First of all, observe that all nodes of J index 0-valued components of x, while all nodes of $V_i \setminus J$ index 1-valued components of x. This is implied by the fact that all edges of a basic subgraph are nonbasic, therefore 0-valued. By Proposition 3.2.7, it immediately follows that pivots (*i*), (*iii*), and (*iv*) are non-degenerate. (*ii*) is also a non-degenerate pivot, because $d(\tau(C_i), \tau(C_j))$ on $C_{ij}(V_i \cup V_j, E_i \cup E_j \cup (s,t))$ is equal to $d(\tau(C_i), s)$ on C_i, which is even ($s \in J$), plus $d(s,t) = 1$ on C_i, plus $d(t, \tau(C_j))$ on C_j, which is even (by hypothesis).

Consider case (*i*). In $G_{B'}$, C_i is such that its only nonbasic node is now z, that is $\tau(C_i) = z$. As $z' \notin J$, all nodes of J index now 1-valued components of x', whereas nodes in $V_i \setminus J$ index 0-valued components of x'. Moreover, all slack variables y'_{ij} are nonnegative, because J is a stable set and nodes of J are only connected to 0-valued nodes of tree components. This proves that pivot (*i*) is non-degenerate and feasible and that all other pivots, where $\tau(C_i)$ enters the basis and another node $z' \in V_i$ exits the basis, are either degenerate or they yield an infeasible basis.

Similarly, pivot (*ii*) can be shown to be the only non-degenerate pivot leading to a new feasible basis, s.t. $\tau(C_i)$ enters the basis an edge $(s,t) \notin E_i$ with $s \in V_i$ and $t \in V_j$, $j \in I_0$ exits the basis.

Consider case (*iii*). In $G_{B'}$, C_i has been replaced by the 1-tree component $C''_i(V_i, E_i \cup (s,t))$. All nodes of V_i index now 1/2-valued components of x'. Moreover, all slack variables y'_{ij} are nonnegative because, by hypothesis, there is no edge connecting a node of J to any 1-valued node of a tree component. This proves that pivot (*iii*) is feasible, and that all other pivots, where $\tau(C_i)$ enters the basis and an edge (s,t) s.t. $s,t \in V_i$ exits the basis, yield an infeasible basis.

Feasibility of pivot (iv) can be proven following the same line of reasoning of case (iii). Analogously, all other pivots, where $\tau(C_i)$ enters the basis and an edge $(s,t) \notin E_i$ s.t. $s \in V_i$ and $t \in V_j$, $j \in I_1$ exits the basis, lead to an infeasible basis. □

Remark 3.2.1. *Each of the non-degenerate feasible pivots described in Propositions 3.2.3, 3.2.6 and 3.2.8 transforms a basis B_1 associated to vertex x^1, into an adjacent basis B_2 associated to vertex x^2 and is such that both $G_{B_1}[W]$ and $G_{B_2}[W]$ are connected, where $W = \{u \in V : x^1_u \neq x^2_u\}$.*

3.3 Some properties concerning adjacency

Given a connected graph $G(V,E)$, assume it is not bipartite, i.e. it contains an odd cycle. Then $FSTAB(G)$ admits fractional vertices. In [63], Padberg gave the following characterization of the fractional vertices of $FSTAB(G)$ that are adjacent to a given integral vertex. We propose an alternative proof of this result, based on the graphic characterization of bases and pivots of $FSTAB(G)$, given in Chapter 2 and Section 3.2.

Proposition 3.3.1. *(Padberg [63]) Given an integer vertex x^I of $FSTAB(G)$, and a fractional vertex x^F, adjacent to x^I on $FSTAB(G)$, consider any two adjacent bases B_I and B_F, associated to x^I and x^F, respectively. Then, B_F can be written as follows (after, possibly, permuting some rows and columns):*

$$B_F = \begin{bmatrix} D & 0 & 0 \\ 0 & G & 0 \\ F_1 & F_2 & I \end{bmatrix},$$

where D has exactly two $+1$ entries in each row and contains exactly one cyclic submatrix of odd order.

Proof. By Theorem 2.2.3, any basis B_I associated to x^I is such that G_{B_I} is a spanning forest of $G(V,E)$. By Propositions 3.2.6 and 3.2.8, the only feasible pivots leading to a fractional vertex adjacent to x^I are (iv) of Proposition 3.2.6, and (iii) of Proposition 3.2.8. Both these pivots yield a new basis B_F associated to a fractional vertex x^F, such that G_{B_F} contains exactly one 1-tree component, say C_1. Therefore, by Theorem 2.2.2, B_F can be expressed as in (2.5) and (2.6), where B_1 is the edge-node incidence matrix of C_1 and $B_2, \ldots, B_k$ are associated to tree components of B_F. Recall that B_1 can be then expressed as in (2.4), implying that B_F contains exactly one cyclic submatrix of odd order. □

Proposition 3.3.1 can then be restated as follows.

Lemma 3.3.1. *Given an integer vertex x^I of $FSTAB(G)$, and a fractional vertex x^F, adjacent to x^I on $FSTAB(G)$, consider any two adjacent bases B_I and B_F, associated to x^I and x^F, respectively. Then, G_{B_F} contains exactly one 1-tree component.*

A direct consequence of Lemma 3.3.1 is that we are able to characterize integer vertices adjacent to the solution $x^*_i = \frac{1}{2}$, $i = 1, \ldots, n$.

Proposition 3.3.2. *Let $S \subseteq V$ be a stable set of $G(V, E)$ and x^S be the incidence vector associated to S. Then x^S is adjacent to x^* on $FSTAB(G)$ if and only if there exists a spanning tree T of G, such that for each $(i, j) \in T$, $i \in S$, $j \notin S$.*

Proof. Suppose there exists a spanning tree T of G, such that for each $(i, j) \in T$, $i \in S$, $j \notin S$. Define $C_1(V, T)$ and $\tau(C_1) = z$, where $z \in V$ is an arbitrary node such that $z \notin S$. By Theorems 2.2.2 and 2.2.3, $B_S = (V \setminus z) \cup (E \setminus T)$ is a basis associated to $G_{B_S} = C_1$, whose corresponding basic feasible solution is x^S. Because G is not bipartite, there should exist an edge connecting two nodes $i, j \notin S$. Pivoting in z and pivoting out (i, j) yields basis $B^* = B_S \setminus (i, j) \cup \{z\}$, whose corresponding basic graph G_{B^*} is composed by a single 1-tree component, namely $C_1'(V, T \cup (i, j))$. By Theorem 2.2.3, the basic feasible solution associated to B^* is x^*.

Suppose now that x^S is adjacent to x^* on $FSTAB(G)$. By Lemma 3.3.1, there exist two adjacent bases B_S and B^*, associated to x^S and x^*, respectively, such that G_{B^*} contains exactly one 1-tree component. Because all components of x^* are fractional, by Theorem 2.2.3, it follows that G_{B^*} consists of a single 1-tree component $C_1(V, W)$. By Proposition 3.2.3, this implies that there exist an edge $(i, j) \in \kappa(C_1)$ to be pivoted in and a node $z \notin S$ to be pivoted out, such that $B_S = B^* \cup (i, j) \setminus \{z\}$ is associated

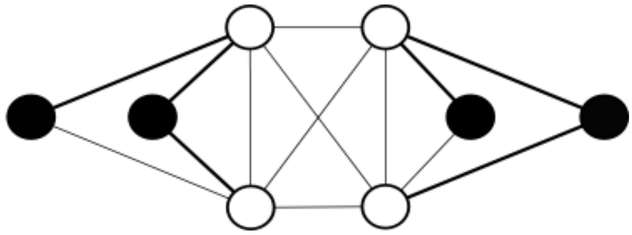

Figure 3.15: For the given graph G, the stable set formed by black nodes is not adjacent to x^* in $FSTAB(G)$

to x^S. Clearly, G_{B_S} consists of a single tree component. Because all the edges of G_{B_S} are nonbasic, and therefore 0-valued, it follows that each edge in $W \setminus (i,j)$ connects a node in S and a node in $V \setminus S$. □

Recall that, to solve (STAB), we can assume w.l.o.g. x^* to be the optimal solution of (FSTAB). Indeed, if this is not the case, by Theorem 1.7.3, we can fix (0,1)-valued variables of the optimal solution and reduce the problem to the subgraph of G induced by $\frac{1}{2}$-valued nodes. Under this assumption, Proposition 3.3.2 establishes whether the optimal solution of (STAB) is adjacent in $FSTAB(G)$ to x^*, which is optimal to (FSTAB). Figure 3.15 shows a graph where this is not the case.

We now generalize Proposition 3.3.2 to an arbitrary fractional vertex of $FSTAB(G)$. We introduce first some preliminary definitions. Given a vector $x \in \{0, \frac{1}{2}, 1\}^{|V|}$, define $F(x) = \left\{i \in V : x_i = \frac{1}{2}\right\}$, $I(x) = \{i \in V : x_i \in \{0,1\}\}$, $S(x) = \{i \in V : x_i = 1\}$. Given a fractional vertex $\bar{x}$, denote by $G[F(\bar{x})]$ the subgraph of G induced by nodes of $F(\bar{x})$.

Proposition 3.3.3. *Let $S \subseteq V$ be a stable set of $G(V,E)$ and x^S be the incidence vector associated to S. Then x^S is adjacent to $\bar{x}$ on $FSTAB(G)$ if and only if there exists a spanning tree T of $G[F(\bar{x})]$, such that for each $(i,j) \in T$, $i \in S$, $j \notin S$ and $\forall\, i \in I(\bar{x})\ x_i^S = \bar{x}_i$.*

Proof. Suppose there exists a spanning tree T of $G[F(\bar{x})]$, such that for each $(i,j) \in T$, $i \in S$, $j \notin S$ and $\forall\, i \in I(\bar{x})\ x_i^S = \bar{x}_i$. By Theorems 2.2.1 and 2.2.3, any basis of $\bar{x}$

is associated to a basic subgraph whose tree and 1-tree components span $F(\bar{x})$ and $I(\bar{x})$, respectively. Consider any basis $\bar{B}$ associated to $\bar{x}$ and assume w.l.o.g. that the first r components $\bar{C}_1(V_1, E_1), \ldots, \bar{C}_r(V_r, E_r)$ of $G_{\bar{B}}$ are trees. Define $C_i = \bar{C}_i$, $i = 1, \ldots, r$ and $C_{r+1}(F(\bar{x}), T)$,with $\tau(C_{r+1}) = z$, where $z \in F(\bar{x}) \setminus S$. By Theorems 2.2.2 and 2.2.3,

$$B_S = \left(V \setminus \bigcup_{i=1}^{r+1} \tau(C_i) \right) \cup \left(E \setminus \left(\bigcup_{i=1}^{r} E_i \cup T \right) \right)$$

is a basis associated to x^S, whose corresponding basic subgraph G_{B_S} consists of components $C_1, \ldots, C_{r+1}$. Because $\bar{x}$ is a vertex of $FSTAB(G)$, the subgraph of G induced by $F(\bar{x})$ is not bipartite. Hence, there should exist an edge connecting two nodes $i, j \in (F(\bar{x}) \setminus S)$. Pivoting in z and pivoting out (i, j) yields basis $\hat{B} = B_S \setminus (i, j) \cup \{z\}$, whose basic graph $G_{\hat{B}}$ is composed by the trees $\hat{C}_i = C_i$, $i = 1, \ldots, r$, plus a single 1-tree component, namely $\hat{C}_{r+1}(F(\bar{x}), T \cup (i, j))$. By Theorem 2.2.3, the basic feasible solution associated to $\hat{B}$ is $\bar{x}$, i.e. $\hat{B}$ is a degenerate basis associated to $\bar{x}$ adjacent to B_S.

Suppose now that x^S is adjacent to $\bar{x}$ on $FSTAB(G)$. By Lemma 3.3.1, there exist two adjacent bases B_S and $\bar{B}$, associated to x^S and $\bar{x}$, respectively, such that $G_{\bar{B}}$ contains r tree components $\bar{C}_1, \ldots, \bar{C}_r$, and one 1-tree component $\bar{C}_{r+1}(F(\bar{x}), W)$. By Proposition 3.2.3, this implies that there exist an edge $(i, j) \in \kappa(\bar{C}_{r+1})$ to be pivoted in and a node $z \in F(\bar{x}) \setminus S$ to be pivoted out, such that $B_S = \bar{B} \cup (i, j) \setminus \{z\}$ is associated to x^S. Clearly, G_{B_S} consists of tree components $C_i = \bar{C}_i$, $i = 1, \ldots, r$ and $C_{r+1}(F(\bar{x}), W \setminus (i, j))$. Because all the edges of G_{B_S} are nonbasic, and therefore 0-valued, it follows that each edge in $W \setminus (i, j)$ connects a node in S and a node in $V \setminus S$. □

It is natural to ask whether all integer vertices of $FSTAB(G)$ are adjacent to some fractional vertex of $FSTAB(G)$. We next present a necessary and sufficient condition for an integer vertex to be adjacent to a fractional one.

Proposition 3.3.4. *Let $S \subseteq V$ be a stable set of $G(V, E)$ and x^S be the incidence vector associated to S. Then x^S is adjacent to a fractional vertex of $FSTAB(G)$ if and only if there exists an odd cycle of G with at most two consecutive nodes belonging to $V \setminus S$.*

Proof. Suppose that there exists an odd cycle κ of G with at most two consecutive nodes u and v belonging to $V \setminus S$. Then, there exists a basis B of x^S such that all nodes of κ and all edges of κ but (u, v) belong to a tree component C_i of G_B. Assume w.l.o.g. that nodes in $V(C_i) \setminus S$ are connected in G only to nodes of other trees which belong to $V \setminus S$. (If this is not the case, i.e. if i is a node of C_i belonging to $V \setminus S$, connected in G to another tree C_j through a node $j \in S$, pivoting out (i, j) and pivoting in $\tau(C_i)$ results in a degenerate pivot.) Then, pivoting out (u, v) and pivoting in $\tau(C_i)$ yields a fractional vertex such that all nodes of C_i are $\frac{1}{2}$-valued.

If, conversely, x^S is adjacent to the fractional vertex $\bar{x}$, then, by Proposition 3.3.3 there exists a spanning tree T of $G[F(\bar{x})]$, such that for each $(i, j) \in T$, $i \in S$ and $j \notin S$. Recalling that $G[F(\bar{x})]$ is not bipartite, it follows that there exist two nodes $u, v \in F(\bar{x})$ such that $(u, v) \in E$. Therefore $T \cup (u, v)$ contains an odd cycle with only two consecutive nodes in $V \setminus S$. □

By Proposition 3.3.4, an integer vertex may not be adjacent to any fractional vertex. An example is shown in Fig. 3.16. The converse is also not necessarily true, as we show in the next Proposition. We first introduce an algorithm to determine a suitable spanning tree of a connected graph $G(V, E)$.

Algorithm 3.3.1. Initialization*: Given a node $v \in V$, set $S^0 = \{v\}$, $\bar{S}^0 = \emptyset$ and $T^0 = \emptyset$.*

1. *If $S^k \cup \bar{S}^k = V$, stop. $H(V, T^k)$ is a spanning tree of G. Otherwise, go to 2.*
2. *Choose a node $v \in V \setminus (S^k \cup \bar{S}^k)$ adjacent to a node $u \in S^k \cup \bar{S}^k$.*
3. *If $v \in N(S^k)$, set $S^{k+1} = S^k$ and $\bar{S}^{k+1} = \bar{S}^k \cup \{v\}$, and $T^{k+1} = T^k \cup (u, v)$. Go to 1.*
4. *If $v \in N(\bar{S}^k)$, set $S^{k+1} = S^k \cup \{v\}$, $\bar{S}^{k+1} = \bar{S}^k$, and $T^{k+1} = T^k \cup (u, v)$. Go to 1.*

Lemma 3.3.2. *Let $G(V, E)$ be a connected graph. Given $v \in V$, there exists a stable set $S \subseteq V$ with $v \in S$, and a spanning tree $T \subseteq E$, such that $\forall\ (i, j) \in T$, $i \in S$, $j \in V \setminus S$.*

Proof. Apply Algorithm 3.3.1 setting $S^0 = \{v\}$. At each iteration k we add to the current tree T^k an edge $(u, w) \in \delta(S^k \cup \bar{S}^k)$. Therefore, T^{k+1} is connected and

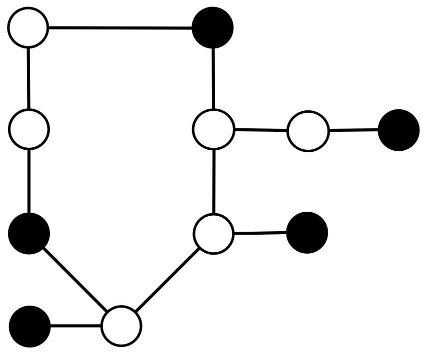

Figure 3.16: The stable set formed by black nodes is not adjacent to any fractional vertex of $FSTAB(G)$

acyclic. At termination, a spanning tree is returned (G is connected). Note that, by construction, each node that is added to S^k is not adjacent to any other node in S^k, proving that S^k is a stable set. Finally, each edge (u, w) that is added to T^k connects a node of S^k to a node in $\bar{S}^k$. □

Proposition 3.3.5. *Let $\bar{x}$ be a vertex of $FSTAB(G)$. Then there exists an integer vertex x^S of $FSTAB(G)$ adjacent to $\bar{x}$ if and only if $G[F(\bar{x})]$ is connected.*

Proof. Suppose that $G[F(\bar{x})]$ is connected and observe that nodes of $F(\bar{x})$ are only adjacent to 0-valued nodes of $I(\bar{x})$. We can apply Lemma 3.3.2 to construct, in $G[F(\bar{x})]$, a tree T associated to a stable set P. The subgraph $C(F(\bar{x}), T)$, with $\tau(C) = z$ such that $z \notin P$, defines a tree component that, together with $G_{\bar{B}}[I(\bar{x})]$ forms a basic subgraph of

$$x_j^S = \begin{cases} 1 & j \in F(\bar{x}) \cap P \\ 0 & j \in F(\bar{x}) \setminus P \\ \bar{x}_j & j \in I(\bar{x}), \end{cases}$$

where $S = P \cup I(\bar{x})$ is a stable set of G, and x^S is the associated incidence vector. The converse is immediately implied by Proposition 3.3.3. □

We now prove that, from any vertex, we can reach x^* along a path on $FSTAB(G)$ of length at most $|V|$.

Theorem 3.3.1. *Let x be a basic feasible solution to (FSTAB) associated with the basis B. There exists a sequence of adjacent bases $B_{I0}, B_{I1}, \ldots, B_{Ip}$ such that $B_{I0} = B$, $B_{Ip} = B^*$ is a basis associated with x^*, and*

(i) the basic solutions $x^{Ii} = B_{Ii}^{-1}\mathbf{1}$, $i = 0, 1, \ldots, p$, are all feasible and such that $I(x^{I0}) \supseteq I(x^{I1}) \supseteq \cdots \supseteq I(x^{Ip}) = \emptyset$;

(ii) p is the number of nonbasic nodes in B, that is equal to the number of tree components of G_B.

Proof. The proof is constructive. Assume w.l.o.g. that G is connected (if not, apply the following reasoning to each connected component of G).

If some nodes of G are $\frac{1}{2}$-valued in x, $G_{B_{I0}}$ contains p tree components $C_i^{I0}(V_i^{I0}, E_i^{I0})$, $i = 1, \ldots, p$ and q 1-tree components $C_i^{I0}(V_i^{I0}, E_i^{I0})$, $i = p+1, \ldots, p+q$. First, perform d degenerate pivots, in order to merge progressively any two tree components connected by an edge with 0-valued slack (pivot (ii) of Proposition 3.2.7). Basis B_{Id} is such that the subgraph $G_{B_{Id}}$ consists of $p' = p - d$ tree components and there are no 0-valued edges connecting them. Recall that G is connected. Therefore, it is possible to progressively merge each tree component to some 1-tree through pivots (iv) of Proposition 3.2.8. Observe that feasibility is preserved at each step, because nodes increasing their values from 0 to $\frac{1}{2}$ can only be connected to 0-valued nodes of other tree components, or to nodes of the same tree, that simultaneously switch their values to $\frac{1}{2}$. After p' pivots, we get a basis B_{Ip}, such that $G_{B_{Ip}}$ is composed only by 1-tree components. Therefore $B^* = B_{Ip}$ is a basis associated to x^*.

If, conversely, all nodes of G are integer valued in x, consider the p tree components $C_i^{I0}(V_i^{I0}, E_i^{I0})$, $i = 1, \ldots, p$ of forest $G_{B_{I0}}$. First, perform d degenerate pivots, in order to merge progressively any two tree components connected by an edge with 0-valued slack (pivot (ii) of Proposition 3.2.7). Basis B_{Id} is such that the forest $G_{B_{Id}}$ consists of $p' = p - d$ connected components and there are no 0-valued edges connecting them.

If for some $k \in \{1, \ldots, p'\}$ there exist two nodes $u, v \in V_k^{Id}$ such that $x_u^{Id} = x_v^{Id} = 0$, C_k^{Id} contains an odd cycle. Then we can perform a pivot that leads to basis $B_{I(d+1)} = B_{Id} \setminus (u, v) \cup \tau(C_k^{Id})$ (pivot (iii) of Proposition 3.2.8). The associated basic feasible solution is:

$$x_j^{I(d+1)} = \begin{cases} \frac{1}{2} & j \in V_k^{Id} \\ x_j^{Id} & \textit{otherwise} \end{cases}$$

Note that $x^{I(d+1)}$ is feasible, because none of the 0-valued nodes of V_k^{Id} was connected to a 1-valued node of another tree component. Because G is connected, we can progressively enlarge the 1-tree component by merging it, at each iteration, with a tree component of the current basic graph. At each step, the nonbasic node of a tree component enters the basis, and a 1-valued edge connecting the tree to the current 1-tree exits the basis (pivot (iv) of Proposition 3.2.8). The corresponding pivot operations are $p' - 1$. Therefore, after $d + 1 + (p' - 1) = p$ pivots, we obtain a basis consisting of a single 1-tree, associated to x^*. Moreover, at each step we get a basic feasible solution satisfying (i).

If for all $k \in \{1, \ldots, p'\}$ there do not exist two nodes $u, v \in V_k^{Id}$ such that $x_u^{Id} = x_v^{Id} = 0$, V_k^{Id} induces a bipartite subgraph of G. Therefore, any odd cycle of G contains at least three consecutive nodes which do not belong to S. Then, by Proposition 3.3.4, x^I is not adjacent to any fractional vertex. We can then perform $q - d$ pivots, in order to obtain a basis B^{Iq} associated to an integer solution x^{Iq}, that is adjacent to a fractional vertex. Precisely, we can merge two tree components, say C_1^{Id} and C_2^{Id}, through a 1-valued edge (u_1, u_2), with $u_1 \in V_1^{Id}$ and $u_2 \in V_2^{Id}$, which exits the basis. Suppose we pivot in $\tau(C_1^{Id})$. The new basis $B_{I(d+1)} = B_{Id} \setminus (u_1, u_2) \cup \tau(C_1^{Id})$ is associated to the basic feasible solution:

$$x_j^{I(d+1)} = \begin{cases} 1 - x_j^{Id} & j \in V_1^{Id} \\ x_j^{Id} & otherwise \end{cases}$$

Note that $x^{I(d+1)}$ is feasible, because nodes increasing their values from 0 to 1 can only be connected to 0-valued nodes of other tree components, or to nodes of the same tree, that simultaneously switch their values to 0.

Let $d^0 = d$. If $G_{B^{I(d^0+1)}}$ has some 0-valued edges connecting its tree components, we can merge them by performing d^1 degenerate pivots, in order to obtain a basis $B^{I(d^0+1+d^1)}$ associated to $x^{I(d^0+1+d^1)} = x^{I(d^0+1)}$, with the smallest number of connected components in its corresponding basic subgraph. We can repeat this procedure t times, until at step $q = d^0 + t + \sum_{i=1}^{t} d^i$, one of the $p - q$ trees of $G_{B^{Iq}}$, say C_k^{Iq}, has two nodes $u, v \in V_k^{Iq}$ such that $x_u^{Iq} = x_v^{Iq} = 0$. Note that this must happen if $q = p - 1$, because G is not bipartite. Pivoting in $\tau(C_k^{Iq})$ and pivoting out (u, v) leads to basis $B^{I(q+1)}$, which is associated to the fractional basic feasible solution:

$$x_j^{I(q+1)} = \begin{cases} \frac{1}{2} & j \in V_k^{Iq} \\ x_j^{Iq} & otherwise \end{cases}$$

Finally, merging the remaining $p - (q+1)$ tree components to the current 1-tree, yields basis B^{Ip}, which consists of a single 1-tree component and is associated to x^*. This proves both (i) and (ii). An example of the constructive procedure used in the proof is shown in Figure 3.17. □

Corollary 3.3.1. *Let x be a vertex of $FSTAB(G)$. There exists a path from x to x^* along edges of $FSTAB(G)$ of length at most p, where p is the number of nonbasic nodes of a basis B of x.*

3.4 Vertex adjacency

In this section, we generalize Chvátal's graphic characterization of integer vertices which are adjacent in $FSTAB(G)$ (see Theorem 3.1.4), to arbitrary vertices of $FSTAB(G)$.

Let us first introduce some preliminary notation. Given two vertices x^1 and x^2 of $FSTAB(G)$, we define the *generalized symmetric difference* $x^1 \otimes x^2 = \{u \in V : x_i^1 \neq x_i^2\}$. Define also $V^{ij} = \{u \in V : x_u^1 = i,\ x_u^2 = j\}$, where $i, j \in \{0, \frac{1}{2}, 1\}$.

Theorem 3.4.1. *Given a graph $G(V,E)$, let (x^1, y^1) and (x^2, y^2) be vertices of $FSTAB(G)$. Let $V^{\otimes} = x^1 \otimes x^2$ and $E^{\otimes} = \{(i,j) \in E : i,j \in V^{\otimes},\ y_{ij}^1 = y_{ij}^2 = 0\}$. Then (x^1, y^1) and (x^2, y^2) are adjacent in $FSTAB(G)$ if and only if $G^{\otimes}(V^{\otimes}, E^{\otimes})$ is a connected (bipartite) subgraph of G.*

Proof. First, let us prove that if $G^{\otimes}(V^{\otimes}, E^{\otimes})$ is connected, (x^1, y^1) and (x^2, y^2) are adjacent in $FSTAB(G)$. Remark that $E^{\otimes}$ contains only edges $(u,v) \in E$ such that:

(i) $u \in V^{01}$ and $v \in V^{01}$;

(ii) $u \in V^{0\frac{1}{2}}$ and $v \in V^{1\frac{1}{2}}$;

(iii) $u \in V^{\frac{1}{2}0}$ and $v \in V^{\frac{1}{2}1}$.

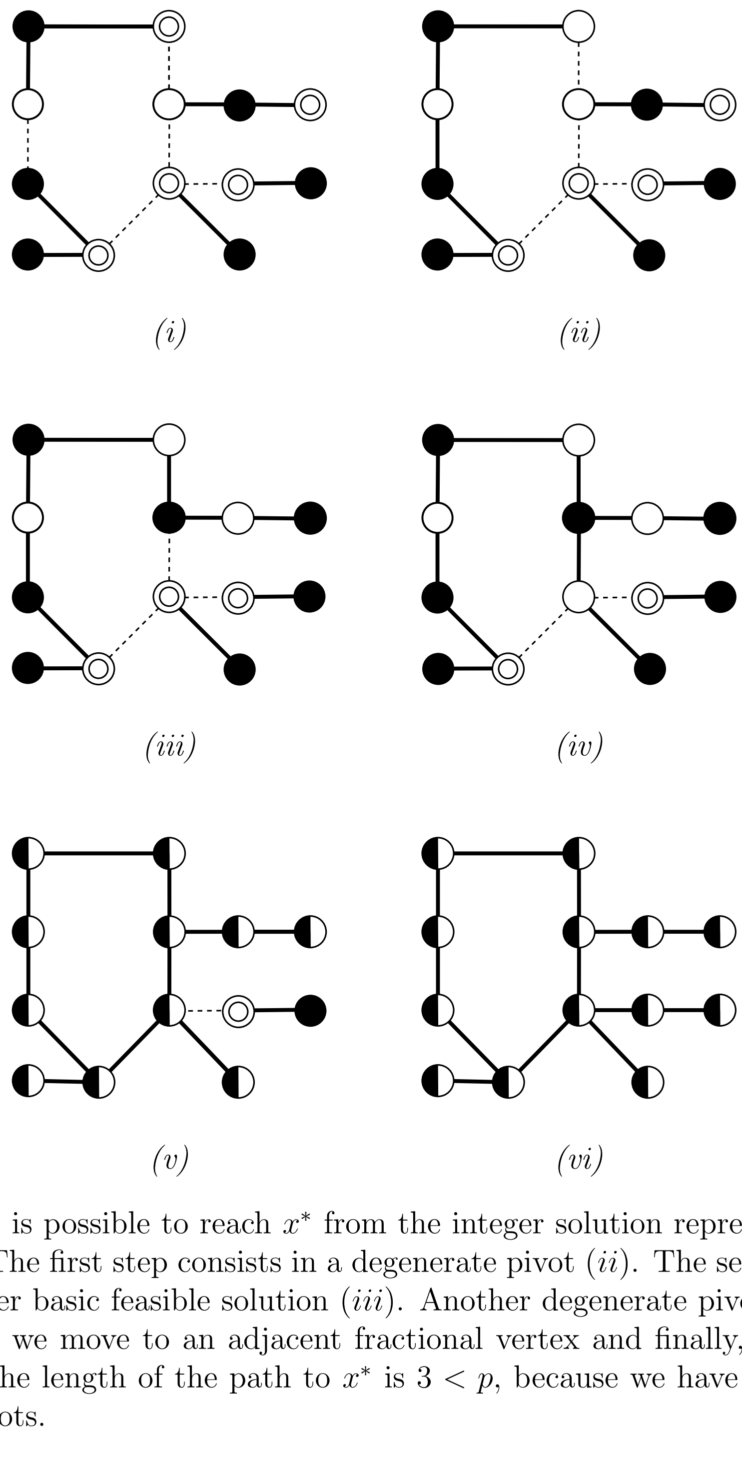

Figure 3.17: It is possible to reach x^* from the integer solution represented in (i) in $p = 5$ pivots. The first step consists in a degenerate pivot (ii). The second step leads to a new integer basic feasible solution (iii). Another degenerate pivot is performed in (iv). In (v) we move to an adjacent fractional vertex and finally, in (vi), to x^*. Remark that the length of the path to x^* is $3 < p$, because we have performed two degenerate pivots.

We claim that, if $G^{\otimes}$ is connected, then either $V^{\otimes} = V^{01} \cup V^{10}$ and edges of $E^{\otimes}$ are of type (i), or $V^{\otimes} = V^{0\frac{1}{2}} \cup V^{1\frac{1}{2}}$ and edges of $E^{\otimes}$ are of type (ii), or $V^{\otimes} = V^{\frac{1}{2}0} \cup V^{\frac{1}{2}1}$ and edges of $E^{\otimes}$ are of type (iii). By contradiction, suppose that $G^{\otimes}$ is connected, $V^{\frac{1}{2}0} \cup V^{\frac{1}{2}1} = \emptyset$, and that $V^{\otimes}$ contains $u \in V^{10} \cup V^{01}$ and $v \in V^{0\frac{1}{2}} \cup V^{1\frac{1}{2}}$. Then, any path connecting u and v should traverse an edge (s, t) such that $s \in V^{10}$ and $t \in V^{0\frac{1}{2}}$, contradicting the fact that $y^2_{st} = 0$. If $V^{0\frac{1}{2}} \cup V^{1\frac{1}{2}} = \emptyset$, and $V^{\otimes}$ contains $u \in V^{10} \cup V^{01}$ and $v \in V^{\frac{1}{2}0} \cup V^{\frac{1}{2}1}$, any path connecting u and v traverses an edge (s, t) such that $s \in V^{01}$ and $t \in V^{\frac{1}{2}0}$, contradicting the fact that $y^1_{st} = 0$. Finally, if $V^{01} \cup V^{10} = \emptyset$, and $V^{\otimes}$ contains $u \in V^{0\frac{1}{2}} \cup V^{1\frac{1}{2}}$ and $v \in V^{\frac{1}{2}0} \cup V^{\frac{1}{2}1}$, any path connecting u and v traverses an edge (s, t) such that $s \in V^{0\frac{1}{2}}$ and $t \in V^{\frac{1}{2}0}$, contradicting the fact that $y^1_{st} = y^2_{st} = 0$.

Suppose now that $G^{\otimes}$ has only edges of type (i). Note that for any basis B^1 associated to x^1, in G_{B^1} the only edges connecting a node $u \in V^{\otimes}$ to a node $v \in V \setminus V^{\otimes}$ are such that $u \in V^{10}$ and $v \in V^{00}$. If $V^{\otimes} = \{u\}$, in G_{B^1} there exists exactly one edge (u, v) belonging to the cutset $(V^{\otimes}, V \setminus V^{\otimes})$, as G by assumption does not contain singletons. By pivoting in (u, v) and pivoting out u yields an adjacent basis associated to x^2. This proves that x^1 and x^2 are adjacent on $FSTAB(G)$.

If $|V^{\otimes}| > 1$, we define a new basis $\bar{B}^1$ for x^1, and a pivot operation to a a basis of x^2. Denote by $E(G_{B^1})$ the edge set of the basic subgraph G_{B^1} and define

$$\bar{E} = \big(E(G_{B^1}) \cap \Gamma(V \setminus V^{\otimes})\big) \cup T,$$

where T is a spanning tree of $G^{\otimes}(V^{\otimes}, E^{\otimes})$.

We essentially break any tree component $C_i(V_i, E_i)$ of G_{B^1} which contains both nodes of $V^{\otimes}$ and of $V \setminus V^{\otimes}$, by removing the edges (u^i, v^i) with $u^i \in V^{\otimes}$ and $v^i \in V \setminus V^{\otimes}$. For each tree component that is generated, we can set v^i as its nonbasic node. Then, we replace $G_{B^1}[V^{\otimes}]$ with the tree $C^{\otimes}(V^{\otimes}, T)$, and we set $\tau(C^{\otimes}) = z$, with $z \in V^{01}$ (observe that such a node exists, because $|V^{\otimes}| > 1$). This defines a basic subgraph $G_{\bar{B}^1}$ satisfying the conditions of Theorem 2.2.2. Pivoting in z and pivoting out a node $z' \in V^{10}$ yields the adjacent basis $B^2 = \bar{B}^1 \setminus \{z'\} \cup \{z\}$ associated to x^2. This proves that x^1 and x^2 are adjacent on $FSTAB(G)$.

Suppose that $G^{\otimes}$ has only edges of type (ii). Assume w.l.o.g. that the integral support of x^1 contains that of x^2 ($I(x^1) \supset I(x^2)$). Note that for any basis B^1 associated to

x^1, in G_{B^1} the only edges (u,v) in the cutset of $(V^{\otimes}, V \setminus V^{\otimes})$ connect a node $u \in V^{1\frac{1}{2}}$ to a node $v \in V^{00}$. We claim that, if $V^{\otimes} = \{u\}$, then $u \in V^{0\frac{1}{2}}$. In fact, suppose by contradiction that $u \in V^{1\frac{1}{2}}$. By feasibility of x^1, it follows that u is not connected to any node in $V^{\frac{1}{2}\frac{1}{2}}$. This contradicts the fact that x^2 is a vertex, because $x^2_u = \frac{1}{2}$ and u cannot be included in a 1-tree with the other $\frac{1}{2}$-valued nodes of x^2. This proves that $u \in V^{0\frac{1}{2}}$. Then, in G_{B^1} there exists exactly one edge (u,v) in the cutset $(V^{\otimes}, V \setminus V^{\otimes})$, as G by assumption does not contain singletons. Moreover, for any basis B^2 of x^2, in G_{B^2} u is connected to a node $w \in V^{\frac{1}{2}\frac{1}{2}}$. By pivoting in (u,v) and pivoting out (u,w), we obtain a basis B^2 associated to x^2. This proves that x^1 and x^2 are adjacent on $FSTAB(G)$.

If $|V^{\otimes}| > 1$, we define a new basis $\bar{B}^1$ for x^1, and a pivot operation to a a basis of x^2. Define

$$\bar{E} = \big(E(G_{B^1}) \cap \Gamma(V \setminus V^{\otimes})\big) \cup T,$$

where T is a spanning tree of $G^{\otimes}(V^{\otimes}, E^{\otimes})$. Again, we remove from any tree $C_i(V_i, E_i)$ of G_{B^1} the edges (u^i, v^i) with $u^i \in V^{\otimes}$ and $v^i \in V \setminus V^{\otimes}$, if any. For each tree component that is generated, we can set v^i as its nonbasic node. Then, we replace $G_{B^1}[V^{\otimes}]$ with the tree $C^{\otimes}(V^{\otimes}, T^{\otimes})$, and we set $\tau(C^{\otimes}) = z$, with $z \in V^{0\frac{1}{2}}$ (observe that such a node exists, because $|V^{\otimes}| > 1$). This defines a basic subgraph $G_{\bar{B}^1}$ satisfying the conditions of Theorem 2.2.2. If $G[V^{\otimes}]$ is not bipartite, there exists an edge (u,t) with $u, t \in V^{0\frac{1}{2}}$. By pivoting in z and pivoting out (u,t), we get a basis associated to x^2. If, conversely, $G[V^{\otimes}]$ is bipartite, there exists an edge (u,w) with $u \in V^{0\frac{1}{2}}$ and $w \in V^{\frac{1}{2}\frac{1}{2}}$. By pivoting in z and pivoting out (u,w), we obtain a basis associated to x^2. This proves that x^1 and x^2 are adjacent on $FSTAB(G)$.

Finally, the case where G has only edges of type (iii) is symmetrical to case (ii).

To prove the converse, i.e. that if (x^1, y^1) and (x^2, y^2) are adjacent in $FSTAB(G)$ then $G^{\otimes}(V^{\otimes}, E^{\otimes})$ is connected, suppose that B^1 and B^2 are adjacent bases of $FSTAB(G)$ associated to (x^1, y^1) and (x^2, y^2), respectively. Recall that all the pivots described in Section 3.2 modify x in one of the following ways:

(i) swap the zeros and ones of x on a subset of nodes;

(ii) assign integer values to a subset of fractional components of x;

(iii) assign fractional values to a subset of integer components of x.

Moreover, by Remark 3.2.1, nodes in $V^{\otimes}$ induce connected subgraphs both of G_{B^1} and of G_{B^2}. Therefore, $G^{\otimes}$ is connected. □

Remark 3.4.1. *Suppose that G consists of k connected components $G^1, \ldots, G^k$ and consider $FSTAB(G^i)$, $i = 1, \ldots, k$. Observe that $FSTAB(G) = FSTAB(G^1) \times \cdots \times FSTAB(G^k)$. Given two vertices x and y of $FSTAB(G)$, express them according to the above decomposition as*

$$x = \begin{pmatrix} x^1 \\ \vdots \\ x^k \end{pmatrix}, \qquad y = \begin{pmatrix} y^1 \\ \vdots \\ y^k \end{pmatrix},$$

where $x^i, y^i \in FSTAB(G^i)$, $i = 1, \ldots, k$. Then, x and y are adjacent on $FSTAB(G)$ if and only if x^j and y^j are adjacent on $FSTAB(G^j)$ for some $j \in \{1, \ldots, k\}$ and $x^i = y^i \ \forall\, i \neq j$.

3.5 The diameter of the Fractional Stable Set Polytope

In this section we prove that the Hirsch conjecture is true for $FSTAB(G)$. The Hirsch Conjecture (1957) stated that a d-dimensional polytope with f facets cannot have (combinatorial) diameter greater than $f - d$, i.e. any two vertices of the polytope can be connected to each other by a path of at most $f - d$ edges. The conjecture was first disproven for unbounded polyhedra [50], then recently also for bounded ones [67]; it was instead proven to be true for $(0,1)$-polytopes [58]. In our case $f - d = m$.

In the following, we first assume G to be connected and we next generalize our results to graphs composed by several connected components. To prove that the Hirsch conjecture is true for $FSTAB(G)$, we have to show that the combinatorial diameter of $FSTAB(G)$ is at most $|E| = m$. In other words, we have to prove that from an arbitrary vertex x^A of $FSTAB(G)$, it is possible to reach any other vertex x^Z through a sequence of solutions $\{x^t\}_{t=1,\ldots,T}$, such that x^t is a basic feasible solution of $FSTAB(G)$ for all $t = 1, \ldots, T$, $x^T = x^Z$ and $T \leq m$. Basically, at each step the current vertex is transformed into an adjacent one. Observe that V^{01}, V^{10}, $V^{\frac{1}{2}1}$, $V^{1\frac{1}{2}}$

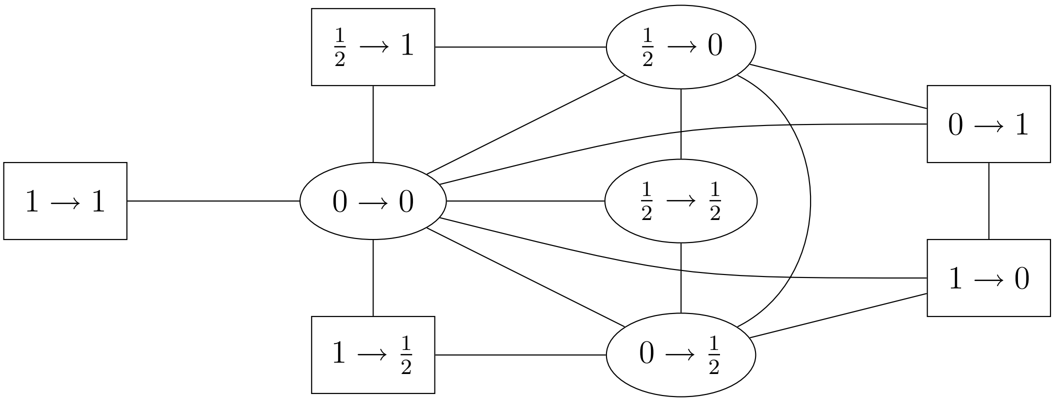

Figure 3.18: Potential connections in G are represented, according to the the partition of V into sets V^{ij}, $i, j \in \{0, \frac{1}{2}, 1\}$. Sets included in a square are stable sets, while there can be connections between nodes belonging to the sets represented inside circles.

and V^{11} are stable sets, because their nodes are 1-valued either in x^A or in x^Z. Note also that there is no edge of G connecting V^{ij} and V^{hk}, if $i + h > 1$ or $j + k > 1$. In Figure 3.18 we represent all potential connections in G, according to the the partition of V into sets V^{ij}, $i, j \in \{0, \frac{1}{2}, 1\}$. Sets included in a square are stable sets, while there can be connections between nodes belonging to the sets represented inside circles.

To guarantee feasibility, we establish precedence relations among transformations involving nodes of V^{ij}, $i, j \in \{0, \frac{1}{2}, 1\}$. In Figure 3.19, each arrow goes from a set V^{ij} to a set V^{hk}, where $i + k > 1$. The directed arc (V^{ij}, V^{hk}) indicates that, for each $(u, v) \in E$ with $u \in V^{ij}$ and $v \in V^{hk}$, if $x^t_v = k$, then $x^t_u = j$. In other words, the value of x_v should not be set to k before that the value of x_u is set to j. To guarantee that each feasible solution x^t is a vertex, we show that it admits an underlying basic subgraph. If x^t is integer, this is trivially true: the incidence vector of every stable set is a vertex of $FSTAB(G)$. If x^t is fractional, we need to define a basic subgraph of G associated to x^t, where all fractional nodes belong to 1-tree components.

Let us first introduce some intermediate results.

Lemma 3.5.1. *Let $G(V, E)$ be a graph and x be a vertex of $FSTAB(G)$. Then each node indexing a $\frac{1}{2}$-valued component of x is connected in G to another node indexing a $\frac{1}{2}$-valued component of x.*

Proof. By contradiction, assume that there is a node $v \in V$ such that $x_v = \frac{1}{2}$ and

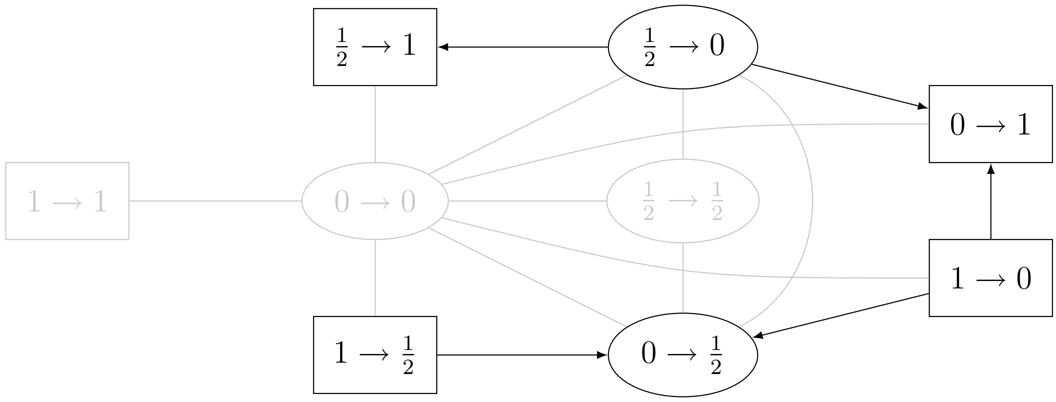

Figure 3.19: Precedence relations among transformations involving nodes of V^{ij} $i, j \in \{0, \frac{1}{2}, 1\}$. Each arrow goes from a set V^{ij} to a set V^{hk}, where $i + k > 1$.

each edge $(u, v) \in E$ is such that $x_u \in \{0, 1\}$. Then, in any basic subgraph associated to x, v does not belong to any 1-tree component, which contradicts the hypothesis that x is a basic feasible solution of $FSTAB(G)$. $\square$

Lemma 3.5.2. *Given a connected graph $G(V, E)$, let x^1 be a vertex of $FSTAB(G)$ and $x^2 \in \{0, \frac{1}{2}, 1\}^{|V|}$. Given a partition of V into $(T, V \setminus T)$, let*

$$x^1 = \begin{pmatrix} x^1_T \\ x^1_{V \setminus T} \end{pmatrix}, \qquad x^2 = \begin{pmatrix} x^2_T \\ x^2_{V \setminus T} \end{pmatrix},$$

and suppose that:

(i) $x^1_{V \setminus T} = x^2_{V \setminus T}$*;*

(ii) *$G[T]$ does not contain singletons and x^2_T is a vertex of $FSTAB(G[T])$;*

(iii) $x^1_i = 0 \;\; \forall \;\; i \in N(T)$.

Then, x^2 is a vertex of $FSTAB(G)$.

Proof. (ii) and (iii) imply that x^2 is feasible for $FSTAB(G)$. To prove that x^2 is a vertex, we show that it admits an underlying basic subgraph. Let B_1 and $\bar{B}_2$ be the bases associated to x^1 and x^2_T, respectively, and denote by G_{B_1} and $G_{\bar{B}_2}$ the corresponding basic subgraphs. Note that $FSTAB(G[T])$ is well defined, as

$G[T]$ does not contain singletons, and recall that x^2_T is a vertex of $FSTAB(G[T])$, implying that $G_{\bar{B}_2}$ is defined only on nodes of T. To define a basis B_2 associated to x^2, we will conveniently append to the basic subgraph $G_{\bar{B}_2}$ some extra tree and 1-tree components defined on the nodes of $V \setminus T$, in such a way to obtain a basic subgraph G_{B_2} spanning all the node set V. Precisely, we will prove that the spanning subgraph

$$G_{B_2} = G_{B_1}[V \setminus T] \cup G_{\bar{B}_2}$$

is a basic subgraph associated to x^2. To this purpose, we first show that all nodes that are $\frac{1}{2}$-valued in x^2 belong to a 1-tree of G_{B_2}, and then that each tree component of G_{B_2} has a nonbasic node.

(iii) implies that each edge in the cutset defined by the partition $(T, V \setminus T)$ is 0-valued if and only if it connects a 1-valued node of T to a 0-valued node of $V \setminus T$. This observation is valid for both x^1 and x^2 as, by (i), they coincide on $V \setminus T$. As a consequence, each 1-tree of G_{B_1} is such that all its nodes belong either to T or to $V \setminus T$. Therefore, in $G_{B_1}[V \setminus T]$, all nodes that are $\frac{1}{2}$-valued in x^2 belong to a 1-tree. Moreover, nodes of T that are $\frac{1}{2}$-valued in x^2 also define 1-tree components, because $G_{\bar{B}_2}$ is a basic subgraph associated to x^2_T.

Consider now a tree component of G_{B_2}. By construction, it belongs either to $G_{\bar{B}_2}$ or to $G_{B_1}[V \setminus T]$. In the first case, the tree clearly has a nonbasic node, because $G_{\bar{B}_2}$ is a basic subgraph. In the second case, we have to show that no tree of $G_{B_1}[V \setminus T]$ is a singleton that is 1-valued in x^2. This follows from the fact that G is connected, implying that each 1-valued node of $V \setminus T$ must be connected to a 0-valued node of $V \setminus T$ with respect to both solutions x^1 and x^2 (in fact, nodes in $N(T)$ are 0-valued, therefore 1-valued nodes in $V \setminus T$ are not adjacent to nodes in T). This implies that G_{B_2} satisfies the conditions of Theorem 2.2.2, proving that x^2 is a vertex. □

Theorem 3.5.1. *Let $G(V, E)$ be a connected graph. Then, the combinatorial diameter of $FSTAB(G)$ is at most $|V| = n$.*

Proof. We will show that it is possible to go from an arbitrary vertex x^A of $FSTAB(G)$ to another vertex x^Z traversing at most n edges of $FSTAB(G)$. To this purpose, we perform four blocks of transitions, each block consisting in a sequence of adjacent vertices of $FSTAB(G)$.

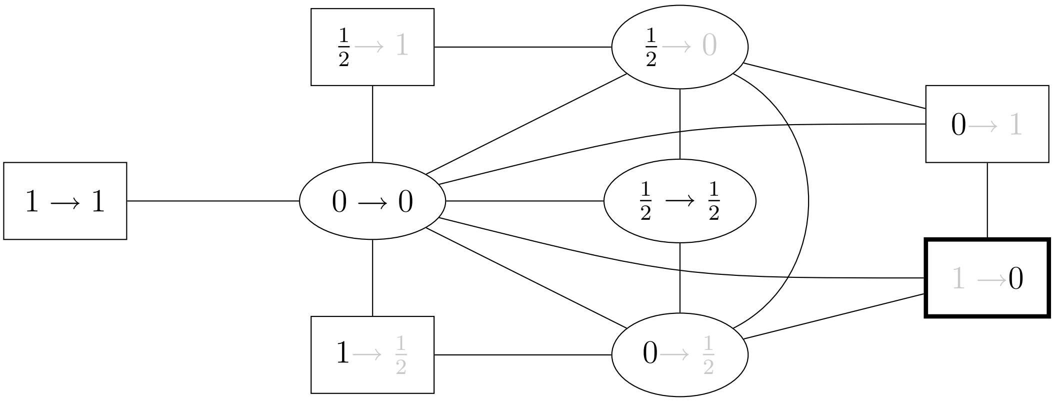

Figure 3.20: The first block of transitions yields vertex x^B.

In the first block of transitions we switch to zero, one by one, nodes of V^{10}. Clearly no precedence relation is violated (see Fig. 3.19), implying that each point of the sequence $x^A = v^0, \ldots, v^{|V^{10}|} = x^B$ is feasible. We also need to show that each point of the sequence is a basic solution. Define $W = V^{\frac{1}{2}1} \cup V^{\frac{1}{2}0} \cup V^{\frac{1}{2}\frac{1}{2}}$. First, for each point v^k, with $k = 1, \ldots, |V^{10}|$, we can construct a suitable spanning forest of rooted trees in $G[V \setminus W]$. This can always be done, because it cannot be the case that we have singletons that should be 1-valued in x^A, and therefore in v^k. To define a basic subgraph for every point of the sequence, we can then complete such forest with the 1-trees of a basic subgraph associated to x^A, that span the nodes of W. This proves that $v^1, \ldots, v^{|V^{10}|}$ are vertices of $FSTAB(G)$. By Theorem 3.4.1, all consecutive points of the above sequence are adjacent. Finally, we obtain a basic graph associated to

$$x_j^B = \begin{cases} 0 & j \in V^{10}, \\ x_j^A & j \in V \setminus V^{10}, \end{cases}$$

$j \in V$ (see Fig. 3.20).

In the second block of transitions we change the values of nodes of $V^{1\frac{1}{2}} \cup V^{0\frac{1}{2}}$. In order to respect the precedence relations illustrated in Fig. 3.19 and preserve feasibility, whenever a node of $V^{0\frac{1}{2}}$ is adjacent to a node of $V^{1\frac{1}{2}}$ we change their values simultaneously, keeping tight the corresponding edge constraint. Define $Y = V^{0\frac{1}{2}} \cup V^{1\frac{1}{2}} \cup V^{\frac{1}{2}\frac{1}{2}} \cup V^{\frac{1}{2}0} \cup V^{\frac{1}{2}1}$ and partition x^B accordingly. Consider the vector

$$x_j^C = \begin{cases} \frac{1}{2} & j \in Y, \\ x_j^B & j \in V \setminus Y, \end{cases}$$

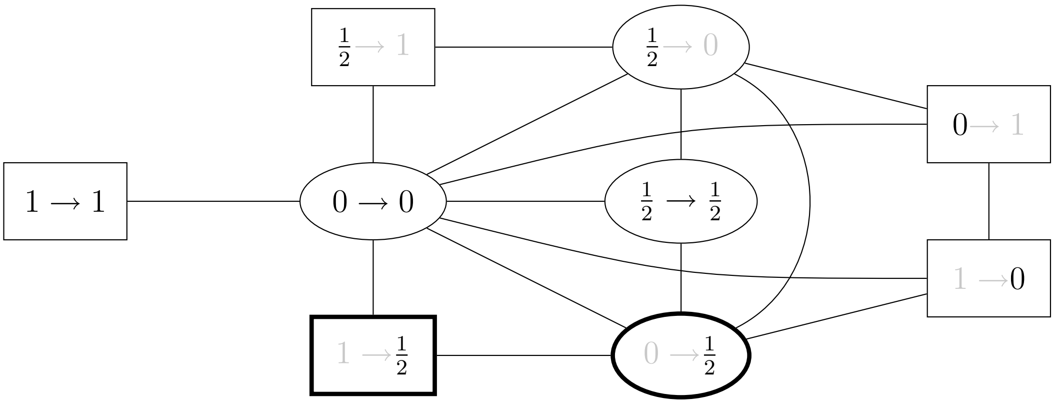

Figure 3.21: The second block of transitions yields vertex x^C.

$j \in V$. By Corollary 3.3.1, we can generate in $G[Y]$ a sequence $x^B_Y = y^0, \ldots, y^p = x^C_Y$ of adjacent vertices of $FSTAB(G[Y])$, where p is the number of nonbasic nodes of a basis associated to x^B_Y, implying $p \leq |V^{0\frac{1}{2}}|$. We can then lift all the points of such sequence to vertices of $FSTAB(G)$ by setting $y^k_{V \setminus Y} = x^B_{V \setminus Y}$, $k = 0, \ldots, p$. We next show that the consecutive points of this lifted sequence satisfy the hypothesis of Lemma 3.5.2. Indeed, they all coincide on $V \setminus Y$ and, by Lemma 3.5.1 $G[Y]$ does not contain singletons. Moreover, $y^k_j = 0$ for all $j \in N(Y)$ and $k = 0, \ldots, p$ (see Fig. 3.18). The lifted sequence defines a path of length at most $|V^{0\frac{1}{2}}|$ from x^B to x^C, along edges of $FSTAB(G)$ (see Fig. 3.21).

In the third block of transitions we change the values of nodes in $V^{\frac{1}{2}1} \cup V^{\frac{1}{2}0}$. As before, to preserve feasibility we respect the precedence relations illustrated in Fig. 3.19: whenever a node of $V^{\frac{1}{2}1}$ is adjacent to a node of $V^{\frac{1}{2}0}$ we change their values simultaneously, and keep tight the corresponding edge constraint. We want to define a path from x^C to

$$x^D_j = \begin{cases} x^Z & j \in Y, \\ x^C_j & j \in V \setminus Y, \end{cases}$$

$j \in V$. Again, we can apply Corollary 3.3.1 to generate a sequence $x^C_Y = z^0, \ldots, z^q = x^D_Y$ of adjacent vertices of $FSTAB(G[Y])$, where $q \leq |V^{\frac{1}{2}0}|$, and we can lift the points of such sequence to vertices of $FSTAB(G)$ by setting $z^k_{V \setminus Y} = x^C_{V \setminus Y}$, $k = 0, \ldots, q$ (Fig. 3.22).

Finally, in the last block of transitions, we switch to one, in succession, nodes of V^{01}, obtaining the sequence $x^D = u^0, \ldots, u^{|V^{01}|} = x^Z$. In doing so, we do not violate

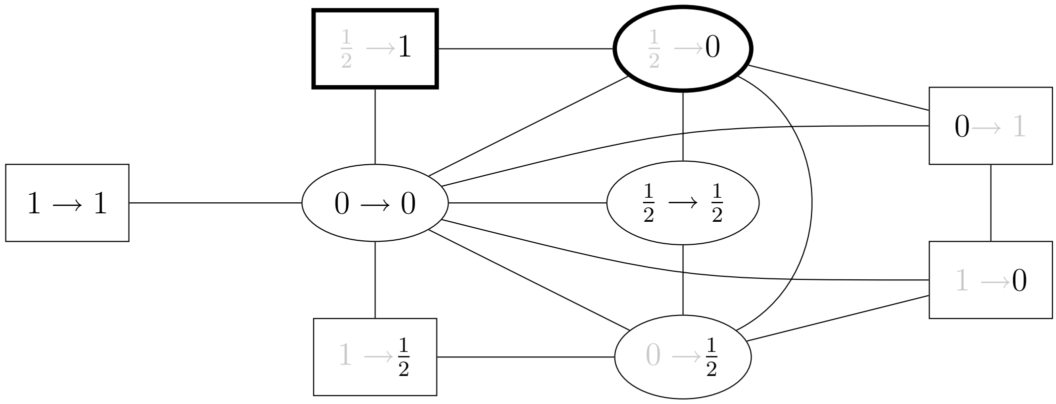

Figure 3.22: The third block of transitions yields vertex x^C.

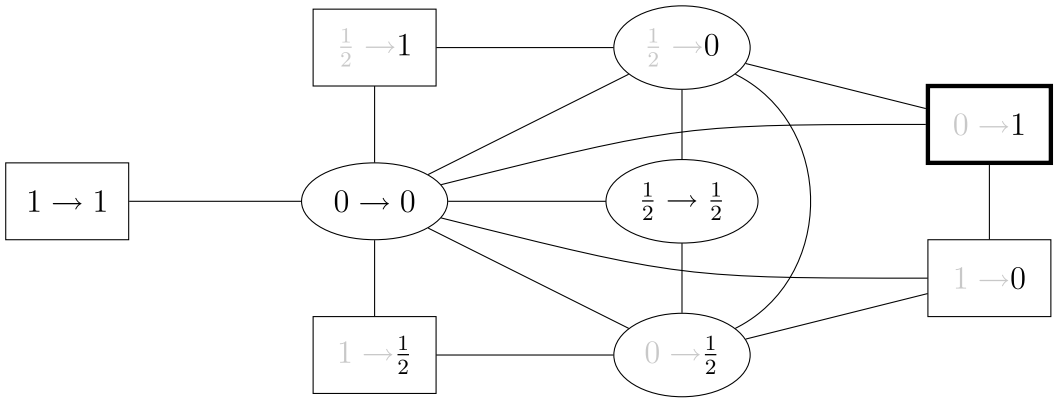

Figure 3.23: The fourth block of transitions yields vertex x^D.

any precedence relation, because all nodes of V^{10} and $V^{\frac{1}{2}0}$ have already been set to zero. Each intermediate point of this sequence is a vertex, as we can define a basic subgraph associated to it. To this purpose, define $Z = V^{1\frac{1}{2}} \cup V^{0\frac{1}{2}} \cup V^{\frac{1}{2}\frac{1}{2}}$. First, for each point u^k, with $k = 1, \ldots, |V^{01}|$, we can construct a suitable spanning forest of rooted trees in $G[V \setminus Z]$. This is possible because x^Z is a vertex, implying that in $G[V \setminus Z]$ there are no singletons that are 1-valued in u^k. Then, to define a basic subgraph for every point of the sequence, we complete such forest with the 1-trees of a basic subgraph associated to x^Z, spanning this way also the nodes of Z. By Theorem 3.4.1, all consecutive points of the above sequence are adjacent. An illustration is given in Fig. 3.23.

Summing up the lengths of the sequences defined above, we obtain

$$|V^{10}| + p + q + |V^{01}| \leq |V^{10}| + |V^{0\frac{1}{2}}| + |V^{\frac{1}{2}0}| + |V^{01}| \leq n,$$

which proves the theorem. □

Corollary 3.5.1. *Let* $G(V, E)$ *be a graph. The combinatorial diameter of* $FSTAB(G)$ *is at most* $\min\{n, m\}$*, where* $n = |V|$ *and* $m = |E|$*.*

Proof. For each bipartite component G^i of G, $FSTAB(G^i)$ is a (0,1)-polytope, and its combinatorial diameter is at most $min\{m^i, n^i\}$, where m^i and n^i are the number of edges and nodes of G^i, respectively [58]. For each connected non-bipartite component G^j of G apply Theorem 3.5.1 to $FSTAB(G^j)$. The claim directly follows from Remark 3.4.1. □

Corollary 3.5.2. *The Hirsch conjecture is true for* $FSTAB(G)$*.*

3.6 A simplex-like algorithm for the Stable Set Problem

In this section we propose an algorithm for the stable set problem, which exploits the characterization of adjacency given in the previous sections.

Balas and Padberg [7, 8, 9] developed a column generating primal simplex algorithm for the set partitioning problem, which is a generalization of the set packing problem (SP). The algorithm produces a sequence of integer solutions converging to an optimal solution of the problem and is based on the characterization of adjacency defined in Theorems 3.1.1 and 3.1.2: starting from an integral basic feasible solution, there exists a sequence of pivots leading to an integer optimum, such that all of the intermediate solutions are integral, the reduced costs of the pivot columns are positive, and the number of pivots equals the number of nonbasic variables in the initial solution that are 1-valued in the specified integer optimum. A Balas-Padberg pivot may have a pivot element with negative sign. Precisely, the algorithm performs non-degenerate primal simplex pivots on +1 entries as long as this is possible. When this cannot be continued, degenerate pivots are performed on positive or negative entries, as long as they decrease total dual infeasibility. When neither type of pivoting can be continued, a column generating procedure is used to produce a composite column defining an edge of LSP which connects the current vertex to a better one, or to establish the absence of any improving vertex.

Ikura and Nemhauser [48], showed that there exists a sequence of simplex pivots from an arbitrary integer vertex of $FSTAB(G)$ to the optimal one, such that each pivot column has positive reduced weight and each pivot element equals +1. The number of the pivots of this sequence corresponds to the number of nonbasic nodes in a basis of the starting solution, which are 1-valued in the optimal solution. The pivot sequence is defined constructively and it requires the knowledge of an optimal solution.

Again Ikura and Nemhauser developed an efficient primal "simplex-like" algorithm based on a graphical interpretation of pivots for solving (FSTAB) on bipartite graphs [47]. For the cardinality problem, their method has a number of pivots which is bounded by n^2, while the running time is $O(n^4)$, with $n = |V|$. For general integer weights a scaling technique is used and the bounds are increased by a factor equal to the logarithm of the largest weight.

Armstrong and Jin [4] gave later an algorithm to solve the weighted vertex packing problem on a bipartite graph via strong spanning trees. The strong spanning tree structure makes their algorithm pseudo-polynomial and improves the complexity of Ikura and Nemhauser's algorithm.

In this section we propose a primal "simplex-like" algorithm based on a graphical interpretation of simplex pivots for connected non-bipartite graphs. The algorithm explores integer solutions, starting from the empty solution, and tries to augment the current stable set at each iteration. Clearly, the basis associated to the starting solution is composed by $|V|$ singletons. Moreover, as we refrain from pivoting to fractional solutions, i.e. we preserve integrality at each step, for all solutions of the sequence, the associated basic subgraph is composed only by trees.

Next, we outline our "simplex-like" algorithm.

3.6.1 Reduced costs

We define reduced costs of a basic integer solution according to [47].

Given $B \in \mathcal{B}$, and one of its tree components $C_i(V_i, E_i)$, we define $W(C_i) = \{j \in$

V_i: $d(\tau(C_i), j)$ on C_i is even} and $B(C_i) = \{j \in V_i\colon d(\tau(C_i), j)$ on C_i is odd}. Note that $W(C_i)$ and $B(C_i)$ correspond to 0 and 1 valued nodes, respectively, in the basic solution associated to B. For each (nonbasic) edge (u, v) of a tree component C_i, define the *branch* B_v as the subtree obtained by removing (u, v) from C_i, which does not contain $\tau(C_i)$. We assume w.l.o.g. that $v \in V(B_v)$. We define branch B_v to be *black* if $d(\tau(C_i, v)$ is odd, i.e. if v is 1-valued in the basic solution associated to B; we define B_v to be *white* if $d(\tau(C_i, v)$ is even, i.e. if v is 0-valued in the basic solution associated to B. Given a branch B_v of a tree C_i, we define $W(B_v) = \{j \in V(B_v)\colon j \in W(C_i)\}$ and $B(B_v) = \{j \in V(B_v)\colon j \in B(C_i)\}$. For each nonbasic variable, reduced costs are computed as follows:

- the reduced cost associated to a nonbasic node, i.e. to the root of a tree component C_i of G_B, is equal to $W(C_i) - B(C_i)$;

- the reduced cost associated to a nonbasic edge (u, v) of a tree component C_i, is equal to $W(B_v) - B(B_v)$ if B_v is a black branch;

- the reduced cost associated to a nonbasic edge (u, v) of a tree component C_i, is equal to $B(B_v) - W(B_v)$ if B_v is a white branch.

3.6.2 non-degenerate pivots

In our algorithm, we perform the following non-degenerate pivots.

Suppose that the variable entering the basis is a node, i.e. the root of a tree component C_i of G_B. If C_i is a singleton, i.e. $V(C_i) = \{\tau(C_i)\}$, we merge C_i to another tree component C_j according to pivot (*ii*) of Proposition 3.2.8. The leaving variable is chosen arbitrarily between those originating basic feasible solutions (all of them yield the same improvement). An illustration is given in Fig. 3.24 (*i*). If C_i is not a singleton, we change the root of the tree, as described in pivot (*i*) of Proposition 3.2.8 (Fig. 3.24 (*ii*)). The new root is chosen between the neighbors of $\tau(C_i)$ in C_i. Note that in both cases, the new basic feasible solution is such that: nodes of C_i that were 0-valued in B are now 1-valued; nodes of C_i that were 1-valued in B are now 0-valued. Remark that pivots (*iii*) and (*iv*) of Proposition 3.2.8 don't preserve integrality of the current solution, and therefore we refrain from performing them.

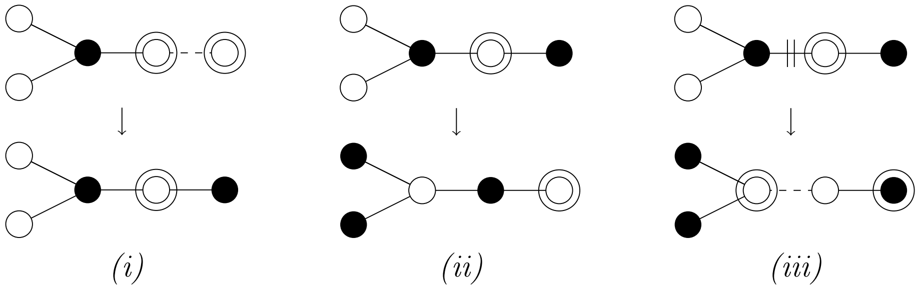

Figure 3.24: Non-degenerate pivots implemented in the simplex-like algorithm.

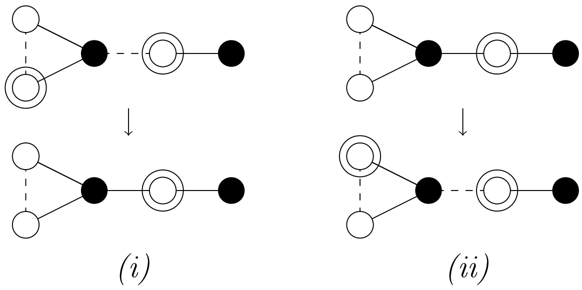

Figure 3.25: Degenerate pivots implemented in the simplex-like algorithm.

Suppose now that the variable entering the basis is an edge (u, v) of tree C_i. If its removal generates a white branch, all pivots described in Section 3.2.2 are trivially infeasible (see Proposition 3.2.5), and therefore we discard them. Suppose now that the removal of (u, v) generates a black branch B_v. In this case, we cut the branch B_v from C_i by performing pivot (i) of Proposition 3.2.6. As the leaving variable, we choose to pivot out v, which is set to be the root of B_v (Fig. 3.24 (iii)). Remark that pivots (iv) and (v) of Proposition 3.2.6 don't preserve integrality of the current solution, and therefore we refrain from performing them. We also discard pivots (ii) and (iii), which do preserve integrality, because these pivots yield bases that we can anyway reach by combining one of our feasible pivots with a degenerate one.

3.6.3 Degenerate pivots

In our algorithm, we perform the following degenerate pivots.

Suppose that the entering variable is a node, i.e. the root of a tree component C_i of G_B. We merge C_i to another tree component C_j according to pivot (ii) of Proposition

3.2.7. The leaving variable is chosen arbitrarily between the 0-valued slacks yielding a degenerate pivot. An illustration is given in Fig. 3.25 (i).

If the entering variable is an edge (u, v) of tree C_i, we cut the corresponding branch B_v according to pivot (iii) of Proposition 3.2.4. The root of B_v is chosen arbitrarily between the neighbors of v in B_v, which are all 0-valued in the original feasible basic solution associated to B (Fig. 3.25 (ii)).

Note that pivots (i) and (ii) of Proposition 3.2.4 can be obtained by the combination of two the operations described above.

3.6.4 Alternative schemes

We can define alternative schemes of our simplex-like algorithm, according to different strategies for choosing the entering and the leaving variables.

For the entering variable, at each iteration we can select either a random variable with positive reduced cost, or the variable with highest reduced cost. In this latter case, we can break possible ties by choosing between the variables with highest reduced cost, either a random one, or that with the smallest index. The leaving variable is always chosen according to the strategy described in Sections 3.6.2 and 3.6.3.

When non-degenerate pivots are no longer possible (because they all yield infeasible bases) we can choose to perform a degenerate pivot. When we perform degenerate pivots, there is no improvement of the objective function and cycling can occur. We can attempt to decrease the occurrence of cycling by perturbing costs. Another possibility consists in disabling some degenerate pivots, as for instance the one where the root of a tree enters the basis and the corresponding tree is merged to another one through a 0-valued edge. In this case, we avoid merging operations between trees and we only allow the degenerate pivot operations which consist in breaking tree components.

The algorithm terminates either when no more pivot is possible, or when an iteration limit is reached. We can also set an iteration limit for the maximum number of degenerate pivots that are performed consecutively (i.e. without any improvement of the objective function).

3.6.5 Computational results

Our simplex-like algorithm for the stable set problem is a primal method, as it mantains integrality and primal feasibility, and uses adjacency properties of $FSTAB(G)$ to drive the solution towards optimality. This is not explicitly done by adding cuts (e.g. odd cycle inequalities), but by avoiding fractional vertices whenever an odd cycle is detected. The flavour of the algorithm is therefore similar to that of *primal cutting plane* algorithms, that are also based on the primal simplex method and were developed in the 1960s by Ben-Israel [1] and Charnes and Young [65, 66]. In fact, the primal approach consists in starting from a feasible solution and improving it through a series of augmentation steps, until the IP optimum is reached.

While a lot of effort has been devoted to *dual fractional* cutting plane algorithms, which form the basis of the well-known branch-and-cut method, few research has been developed on primal methods. In [53] Letchford and Lodi argue for a re-examination of these primal methods and describe a primal algorithm for 0-1 programs giving some interesting computational results. In [32] Firla at al. investigate the approach of *integer pivoting* and discuss algorithmic issues related to the design of an augmentation algorithm for 0-1 programs. Interestingly, they also report computational experience on the max-clique problems that are collected in the DIMACS Challenge library[1]. To solve these instances, they perform at each iteration a non-degenerate integer pivot, and they stop whenever this is not possible. In many cases, optimality of the returned solution can be proven by dual arguments and by performing a fixing of the nonbasic variables by means of a bound analysis.

We also performed preliminary computational experiments on the DIMACS instances, that we modeled as stable set problems on the respective complement graphs, according to the edge formulation. Because of degeneracy and, as a consequence, of cycling phenomena, the current implementation of our algorithm also returns a heuristic solution, i.e. a stable set that is not necessarily optimal. We have tested different schemes of our algorithm, according to the variations described in section 3.6.4. We

[1] Second DIMACS Implementation Challenge, `http://dimacs.rutgers.edu/Challenges/`

report results in Tables 3.1, 3.2 and 3.3. In Scheme 1 we choose as the entering variable the nonbasic variable with the highest reduced cost, and we break possible ties by choosing a random variable between those with the same reduced cost. In Scheme 2 we choose as the entering variable a random nonbasic variable with positive reduced cost. In Scheme 3 the entering variable is again the one with highest reduced cost, but when a degenerate pivot has to be performed, a perturbation of costs is applied. For Schemes 1, 2 and 3, we allow at most 100 consecutive degenerate pivots, i.e. 100 iterations without any improvement of the objective function. In Scheme 4 we disable the degenerate pivot where the entering variable is the root of a tree and we therefore avoid cycling. The algorithm stops when no more pivot is possible. All computation times are given in CPU seconds on a Intel Core i5 with 2.3 GHz. The results obtained by these four different schemes of the algorithm are comparable, except for few cases. For example, for the instances of the class MANN, the first two schemes achieve better performances, probably due to the fact that degenerate pivots are useful to further improve the current solution when non-degenerate pivots are no longer possible. This leads to the following considerations. As an heuristic method, Scheme 4 is probably preferable, as it avoids cycling and runs in a shorter time. Though, the results reported in Tables 3.1, 3.2 and 3.3 are not competitive with state-of-the-art heuristics [62, 64, 44], due to the stalling phenomena that we haven't yet treated. We therefore expect that, provided that an anti-cycling rule is implemented, the algorithm can be competitive as an exact method of solution, due to the fact that each pivot step is not performed algebraically but, more efficiently, as a simple combinatorial operation, and because we can avoid to pivot to fractional solutions.

This promising behavior is highlighted by the experiments conducted on a subset of the DIMACS instances, the ones reported in the computational experiments of [32]. As a comparison, we transcribe in Table 3.4 the results of [32]. On this testset, both algorithms reach optimality on all the instances, [32] providing in addition an optimality certificate on the starred instances. Observe that our computing times are extremely short for such small instances.

	Scheme 1		Scheme 2		Scheme 3		Scheme 4	
	obj	time	obj	time	obj	time	obj	time
brock200_1	12	0.015	13	0.009	13	0.021	13	0.001
brock200_2	7	0.023	7	0.020	7	0.010	7	0.001
brock200_3	9	0.017	9	0.013	11	0.011	11	0.002
brock200_4	11	0.010	12	0.012	12	0.012	12	0.002
brock400_1	18	0.029	18	0.020	15	0.025	15	0.004
brock400_2	18	0.023	18	0.025	17	0.019	17	0.003
brock400_3	18	0.032	15	0.023	17	0.015	17	0.004
brock400_4	17	0.036	14	0.018	17	0.018	17	0.004
brock800_1	14	0.087	15	0.089	14	0.041	14	0.012
brock800_2	13	0.090	14	0.090	14	0.039	14	0.011
brock800_3	12	0.117	16	0.069	14	0.048	14	0.015
brock800_4	14	0.109	12	0.116	13	0.042	13	0.012
c-fat200-1	10	0.013	10	0.007	12	0.006	12	0.001
c-fat200-2	22	0.006	22	0.005	24	0.008	24	0.002
c-fat200-5	57	0.006	57	0.005	58	0.007	58	0.002
c-fat500-1	12	0.039	12	0.032	14	0.017	14	0.004
c-fat500-2	24	0.038	24	0.031	26	0.022	26	0.005
c-fat500-5	62	0.028	62	0.023	64	0.024	64	0.007
C125.9	27	0.006	27	0.004	26	0.005	26	0.001
C250.9	32	0.014	31	0.009	30	0.010	30	0.002
C500.9	37	0.024	36	0.026	33	0.023	33	0.005
C1000.9	47	0.084	44	0.062	44	0.051	44	0.021
C2000.5	10	0.662	11	0.560	11	0.152	11	0.078
C2000.9	53	0.244	50	0.201	52	0.139	52	0.068
C4000.5	11	2.715	12	3.077	12	0.519	12	0.309

Table 3.1: Computational results for four different schemes of our simplex-like algoritm (I).

	Scheme 1		Scheme 2		Scheme 3		Scheme 4	
	obj	time	obj	time	obj	time	obj	time
DSJC500.5	10	0.082	8	0.022	9	0.026	9	0.005
DSJC1000.5	10	0.236	7	0.241	9	0.051	9	0.018
gen200_p0.9_44	29	0.006	28	0.008	30	0.007	30	0.001
gen200_p0.9_55	28	0.006	30	0.008	30	0.009	30	0.002
gen400_p0.9_55	37	0.019	35	0.022	34	0.016	34	0.003
gen400_p0.9_65	39	0.022	37	0.017	35	0.017	35	0.004
gen400_p0.9_75	40	0.023	35	0.015	38	0.017	38	0.003
hamming6-2	32	0.002	32	0.001	32	0.002	32	0.001
hamming6-4	4	0.003	4	0.002	4	0.005	4	0.000
hamming8-2	128	0.019	128	0.010	128	0.014	128	0.005
hamming8-4	7	0.019	7	0.011	16	0.011	16	0.002
hamming10-2	238	0.133	251	0.082	512	0.116	512	0.076
hamming10-4	17	0.116	17	0.094	32	0.084	32	0.026
johnson8-2-4	4	0.002	4	0.001	4	0.001	4	0.000
johnson8-4-4	8	0.003	9	0.004	14	0.004	14	0.000
johnson16-2-4	8	0.005	8	0.003	8	0.005	8	0.000
johnson32-2-4	16	0.026	16	0.026	16	0.025	16	0.004
keller4	7	0.008	9	0.007	7	0.006	7	0.001
keller5	15	0.091	15	0.062	15	0.050	15	0.009
keller6	35	0.868	28	0.796	31	0.494	31	0.156
MANN_a9	16	0.002	16	0.001	16	0.003	9	0.000
MANN_a27	118	0.019	121	0.015	98	0.020	27	0.002
MANN_a45	336	0.096	333	0.088	267	0.134	45	0.018
MANN_a81	1085	0.964	1080	0.653	266	0.340	81	0.154
p_hat300-1	5	0.035	5	0.024	6	0.013	6	0.002
p_hat300-2	11	0.029	11	0.021	12	0.011	12	0.002
p_hat300-3	21	0.014	19	0.018	18	0.011	18	0.002
p_hat500-1	5	0.141	5	0.070	6	0.019	6	0.005
p_hat500-2	18	0.044	19	0.028	21	0.021	21	0.005
p_hat500-3	20	0.049	29	0.021	29	0.021	29	0.006
p_hat700-1	6	0.128	6	0.122	5	0.034	5	0.007
p_hat700-2	21	0.082	22	0.086	18	0.034	18	0.010
p_hat700-3	36	0.059	35	0.052	30	0.030	30	0.012
p_hat1000-1	8	0.276	8	0.246	6	0.051	6	0.016
p_hat1000-2	22	0.107	26	0.195	24	0.058	24	0.021
p_hat1000-3	35	0.118	36	0.093	29	0.054	29	0.019
p_hat1500-1	5	0.410	7	0.237	6	0.094	6	0.036
p_hat1500-2	24	0.278	22	0.212	23	0.097	23	0.040
p_hat1500-3	42	0.222	33	0.164	49	0.097	49	0.053

Table 3.2: Computational results for four different schemes of our simplex-like algoritm (II).

	Scheme 1		Scheme 2		Scheme 3		Scheme 4	
	obj	time [s]	obj	time [s]	obj	time [s]	obj	time [s]
san200_0.7_1	16	0.009	16	0.009	15	0.008	15	0.001
san200_0.7_2	14	0.010	14	0.008	12	0.007	12	0.001
san200_0.9_1	28	0.007	33	0.006	47	0.009	43	0.001
san200_0.9_2	35	0.006	38	0.007	36	0.008	36	0.001
san200_0.9_3	27	0.008	27	0.010	21	0.009	21	0.001
san400_0.5_1	7	0.028	7	0.025	7	0.016	7	0.003
san400_0.7_1	21	0.028	21	0.026	21	0.018	21	0.004
san400_0.7_2	15	0.033	15	0.027	15	0.016	15	0.003
san400_0.7_3	13	0.031	10	0.031	10	0.014	10	0.003
san400_0.9_1	33	0.015	51	0.030	35	0.016	35	0.003
san1000	7	0.259	7	0.233	7	0.061	7	0.018
sanr200_0.7	12	0.007	11	0.008	13	0.008	13	0.001
sanr200_0.9	27	0.009	27	0.006	29	0.009	29	0.001
sanr400_0.5	9	0.057	9	0.054	8	0.015	8	0.003
sanr400_0.7	15	0.034	14	0.021	15	0.017	15	0.004

Table 3.3: Computational results for four different schemes of our simplex-like algoritm (III).

	Clique Size	Firla et al. time [s]	Scheme 4 time[s]
hamming6-2	32	*0.6	0.001
hamming6-4	4	3.4	0.000
hamming8-2	128	*12.3	0.005
johnson8-2-4	4	*0.4	0.000
johnson8-4-4	14	10.2	0.000
johnson16-2-4	8	*6.0	0.000
johnson32-2-4	16	*191.2	0.004
c-fat200-1	12	*112.2	0.001
c-fat200-2	24	*272.2	0.002
c-fat200-5	58	*864.5	0.002
c-fat500-1	14	*2300.8	0.004
c-fat500-2	26	*5336.3	0.005
c-fat500-5	64	*29314.5	0.007

Table 3.4: Computational results on a subset of the DIMACS instances.

Chapter 4

How tight is the corner relaxation? Insights gained from the stable set problem

[1]

Consider a Mixed-Integer Linear Program (MILP) in standard form:

$$\begin{aligned} \min \quad & c^T x \\ \text{s.t.} \quad & Ax = b \\ & x \geq 0 \\ & x_i \in \mathbb{Z} \quad \forall\, i \in I, \end{aligned} \tag{MILP}$$

where A is a $m \times n$ rational matrix with full row rank m, $c \in \mathbb{Q}^n, b \in \mathbb{Q}^m$ and $I \subseteq \{1, \dots, n\}$ is the subset of variables that are integer constrained. The Linear Programming (LP) relaxation of (MILP) is the problem:

[1]This is a work developed with Gérard Cornuéjols (CMU) and Giacomo Nannicini (SUTD) when the author of the thesis was visiting Tepper School of Business (CMU) and their support is gratefully acknowledged.

$$\begin{aligned} \min \quad & c^T x \\ \text{s.t.} \quad & Ax = b \\ & x \geq 0. \end{aligned} \tag{LP}$$

In Section 1.4 we have discussed some exact methods for the solution of (MILP). In particular, we have briefly outlined the Branch-and-Cut method, which combines both cutting planes and enumeration techniques. The cutting planes component of Branch-and-Cut generates cuts that are valid for (MILP), which are then added to (LP). Most general purpose cutting planes, such as Gomory mixed-integer [39] and mixed-integer rounding [59] cuts, are valid for the *corner relaxation* of (MILP), introduced by Gomory [40]. Studying the strength of the corner relaxation is therefore of both theoretical and practical interest. Given a basis B of (LP), the corner relaxation is the convex hull of the integer points of the problem obtained from (MILP) by dropping non-negativity on the basic variables. If non-negativity is dropped on the strictly positive basic variables only, we call the convex hull of the resulting set of points *strict corner relaxation* (there is no standard terminology for this relaxation in the literature). If (LP) has primal degeneracy, the strict corner relaxation can be stronger than the corner relaxation.

In this chapter we study the corner and other related relaxations in the particular case of the edge formulation of the stable set problem. This is an important combinatorial optimization problem, and stable set type constraints appear in the MILP formulation of many real-world problems. We give a precise characterization of the bounds arising from four different relaxations for this particular combinatorial problem. The reason for choosing this problem is that it is one of the very few where the structure of the bases is well understood, allowing a tight analysis of the relaxations.

Given a graph $G(V, E)$ with $|V| = n$ nodes, it is known that, under mild assumptions, the linear programming relaxation (FSTAB) of the edge formulation of (STAB) for the maximum cardinality stable set problem has value $n/2$. For the most common random graph models and for $n \to \infty$, these assumptions hold with probability 1 [42, 43]. The results proven in this chapter can be summarized as follows. We show that, for a connected graph G, if the graph admits a perfect matching or a nearly perfect matching, there exists an optimal basis B of (FSTAB) such that the associated corner relaxation gives a bound of $\lfloor n/2 \rfloor$. If the nodes of the graph can be partitioned into cliques of size at least 3, the split closure [23] yields a bound of $n/3$. If all cliques in the partition have size 3, the same bound can also be obtained from a corner relaxation associated with an optimal basis. We show that in some cases, generating cutting planes from a corner relaxation and adding them to (FSTAB) significantly improves the corner relaxation bound. Finally, we show that the strict corner relaxation yields the optimal value of (STAB).

In Section 4.1 we introduce the *corner relaxation* and some basic results concerning the *corner polyhedron*. In Section 4.2 we discuss the empirical results on the strength of the corner and other related relaxations obtained in [35]. In Section 4.3 we describe our main results on the strength of different relaxations of (STAB). In Section 4.4 we present some crucial results, due to Campelo and Cornuéjols [17], concerning the complete description of the corner polyhedron and of the intersection closure. In Section 4.5 we prove that, in oder to compute the bounds yielded by the relaxations considered in this chapter, it is possible to restrict to the subgraph induced by the nodes indexing fractional variables of a given optimum to (FSTAB). In Sections 4.6, 4.7, 4.8, 4.9 we study the bounds arising from the corner and other related relaxations, which we introduce next.

4.1 The Corner Relaxation

In this section we present some basic notions concerning the corner relaxation and the corner polyhedron, originally introduced by Gomory [40] and Gomory and Johnson [41]. For further readings on this topic and recent developments in multi-row cuts, we recommend [21]. Consider the mixed integer linear set of (MILP) and a feasible basis B of (LP). Let $N = \{1, \ldots, n\} \setminus B$ index the nonbasic variables. It is possible to express the system $Ax = b$ in tableau form, in order to rewrite basic variables in terms of nonbasic variables as

$$x_i = \bar{b}_i - \sum_{j \in N} \bar{a}_{ij} x_j \qquad i \in B, \tag{4.1}$$

where feasibility of B implies that $\bar{b}_i \geq 0$, for each $i \in B$. The basic feasible solution $\bar{x}$ associated to B is then $\bar{x}_i = \bar{b}_i$, $i \in B$, $\bar{x}_j = 0$, $j \in N$. If $\bar{b}_i \in \mathbb{Z}\ \forall\ i \in B \cap I$, $\bar{x}$ is feasible for (MILP). If, instead, any of the integrality constraints is violated by $\bar{x}$, it is possible to *separate* $\bar{x}$, i.e. to determine an inequality that is valid for (MILP), which is violated by $\bar{x}$. The *corner relaxation*, introduced by Gomory in [40], is obtained from MILP by dropping non-negativity constraints on the basic variables x_i, $\forall\, i \in B$. If some of the basic variables are not integer constrained, i.e. $B \setminus I \neq \emptyset$, these variables only appear in one equation of type (4.1). Therefore, it is possible to drop constraints $x_i = \bar{b}_i - \sum_{j \in N} \bar{a}_{ij} x_j$ for each basic continuous variable x_i, with $i \in B \setminus I$. For the sake of simplicity, it is usually assumed that all the basic variables are integer constrained, i.e. $I \supseteq B$. Therefore, the corner relaxation of (MILP) is

$$\begin{array}{lll} \min & c^T x & \\ \text{s.t.} & x_i = \bar{b}_i - \sum_{j \in N} \bar{a}_{ij} x_j & i \in B \\ & x_i \geq 0 & i \in N \\ & x_i \in \mathbb{Z} & i \in I. \end{array} \tag{4.2}$$

The convex hull of the feasible solutions to (4.2) is the *corner polyhedron* relative to the basis B, denoted by corner(B). Any inequality that is valid for the corner polyhedron corner(B) is also valid for the integer hull of feasible solution to (MILP).

Consider now the linear relaxation of (4.2), and denote its feasible region by $P(B)$. $P(B)$ is a cone pointed in $\bar{x}$ and such that its extreme rays are the vectors satisfying at equality all but one non-negativity constraints. The extreme rays of $P(B)$ are linearly independent, implying that $P(B)$ has dimension $|N|$ and that the affine hull of $P(B)$ is defined by the equations (4.1) [21].

Lemma 4.1.1. *[21] If the affine hull of $P(B)$ contains a point $x \in \mathbb{R}^n$ such that $x_i \in \mathbb{Z}$ for each $i \in I$, then corner(B) is an $|N|$-dimensional polyhedron. Otherwise, corner(B) is empty.*

In Figure 4.1 are represented: the convex hull of the feasible solutions of a MILP (in red); the objective function of the MILP (dashed lines); the feasible region of the LP relaxation of the MILP (in black); the optimum of the LP relaxation of the MILP (the star); the cone $P(B)$ (in blue); the feasible points of the corner relaxation (4.2) (green dots); the corner polyhedron (in green).

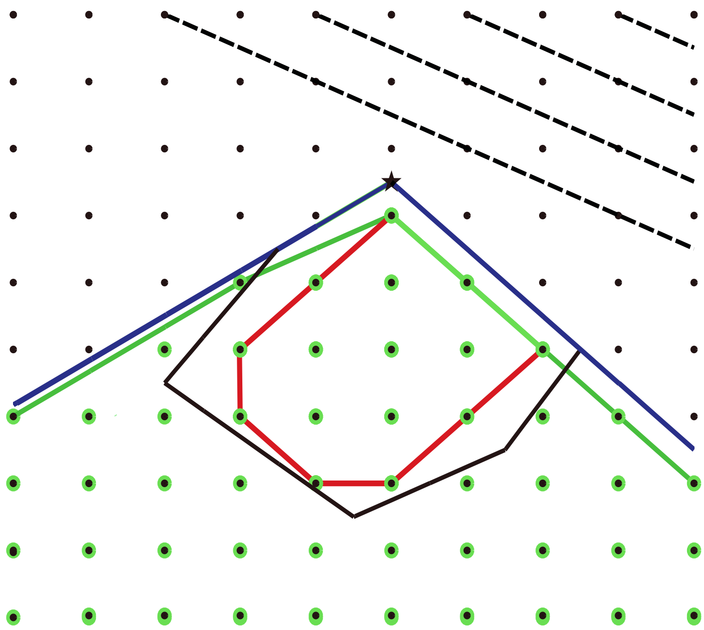

Figure 4.1: The corner polyhedron of the given MILP is the convex hull of green points.

4.2 How tight is the Corner Relaxation?

Fischetti and Monaci [35] empirically study the strength of the corner relaxation, strict corner relaxation and other related polyhedra on a set of benchmark MILP instances. They compare the objective value of the integer optimum of (MILP) with the bounds given by the strict corner relaxation, the corner relaxation associated with an optimal basis of (LP), (LP) alone, and (LP) strengthened by one round of cutting planes from an optimal basis. They conclude that:

- For problems with binary variables, the corner relaxation is often a weak approximation of (MILP).
- The strict corner relaxation gives on average 50% better bounds (in relative terms) than the corner relaxation.
- The conclusion that the corner relaxation is often a weak approximation of (MILP) is mitigated by the fact that, in practice, cutting planes are added to (LP) and this often gives much better bounds.

4.3 Main Results

Consider a simple graph $G(V,E)$, where V and E are the sets of n nodes and m edges of G, respectively. We a assume that G does not contain isolated nodes (recall Assumption 1.5.1). In Section 1.5 we have introduced the stable set problem which consists, in its unweighted version, in finding the maximum cardinality subset of nodes which are not pairwise adjacent. A natural formulation for the stable set problem is the edge formulation. In fact, a vector $x \in \{0,1\}^n$ is the incidence vector of a stable set of G if and only if it satisfies $x_u + x_v \leq 1$, for all $(u,v) \in E$. By introducing a slack variable for each edge constraint, the edge formulation of the stable set problem can be written in the form

$$\begin{aligned} \alpha(G) = \max \quad & \sum_{i=1}^{n} x_i \\ \text{s.t.} \quad & x_i + x_j + y_{ij} = 1 \qquad \forall\,(i,j) \in E \\ & y_{ij} \geq 0 \qquad \forall\,(i,j) \in E \\ & x_i \in \{0,1\} \qquad \forall\, i \in V. \end{aligned} \tag{STAB}$$

The *stable set polytope*, which we have denoted by $STAB(G)$, is the convex hull of the incidence vectors of stable sets of G, which correspond to feasible solutions of (STAB). For the MILP (STAB), under the assumption that G does not contain singletons, the LP relaxation has the form

$$\begin{aligned} \alpha_{\text{FSTAB}} = \max \quad & \sum_{i=1}^{n} x_i \\ \text{s.t.} \quad & x_i + x_j + y_{ij} = 1 \qquad \forall\,(i,j) \in E \\ & x_i \geq 0 \qquad \forall\, i \in V \\ & y_{ij} \geq 0 \qquad \forall\,(i,j) \in E, \end{aligned} \tag{FSTAB}$$

whose feasible set will be denoted $FSTAB(G)$.

In Chapter 2 we have extensively discussed structural properties of the bases of $FSTAB(G)$. In particular, we have presented a graphic characterization of bases of $FSTAB(G)$. Precisely, a one-to-one correspondence exists, between bases of $FSTAB(G)$ and special pseudoforests of G, composed by rooted trees and 1-trees with an odd cycle.

We have denoted by A the edge-node incidence matrix of G and by $\mathcal{B}$ the set of all bases of the constraint matrix $[A \quad I]$. The rows of (FSTAB) are linearly independent, implying that a basis consists of m columns. It follows that a cobasis (the set indexing out-of-basis variables) is composed by n columns. We have denoted by B an element of $\mathcal{B}$ and by N the resulting nonbasic submatrix (we use the same symbols to denote the sets of indices of a basis and of a cobasis, respectively). The variables can be then partitioned according to each basis $B \in \mathcal{B}$, as $x = \begin{pmatrix} x_B \\ x_N \end{pmatrix}$ and $y = \begin{pmatrix} y_B \\ y_N \end{pmatrix}$. Discarding non-negativity constraints on the basic variables, we get a relaxation of (STAB). The convex hull of the resulting set of feasible solutions corresponds to the *corner polyhedron* associated with basis B, denoted by corner(B). If the basic solution associated with basis B is not integral, then it does not belong to corner(B), and a valid inequality for corner(B) can be generated, such that the fractional solution is cut off. It has been shown [17] that all valid inequalities necessary to describe corner(B) can be derived from one row of the simplex tableau associated to basis B as Chvátal-Gomory cuts.

Define now the *intersection closure* as the intersection of the corner polyhedra associated to all bases and denote it by int$(\mathcal{B})$, namely

$$\text{int}(\mathcal{B}) = \bigcap_{B \in \mathcal{B}} \text{corner}(B). \tag{4.3}$$

It has been proven [17] that, for the stable set formulation (STAB), the set int$(\mathcal{B})$ and the split, Chvátal and $\{0, \frac{1}{2}\}$-Chvátal closures are all identical.

We address two additional relaxations of (STAB). The first one, that we call *strict corner relaxation*, is obtained from (STAB) by relaxing non-negativity constraints on those variables that are strictly positive in an optimal solution x^* of (FSTAB). The convex hull of the feasible points of the strict corner relaxation is the *strict corner polyhedron*, denoted by strict(x^*). The second relaxation is defined by intersecting corner(B) and $FSTAB(G)$ for a given $B \in \mathcal{B}$, and we denote it by LP$\cap$corner(B). The reason for studying this relaxation is that LP$\cap$corner(B) corresponds to strengthening (FSTAB) with cutting planes valid for corner(B), and is therefore highly relevant in practice.

In this chapter, we estimate and compare the bounds obtained by optimizing over corner(B), int($\mathcal{B}$), strict(x^*) and LP∩corner(B). In other words, we study the following problems:

$$\alpha_{\text{corner}(B)} = \max\{\mathbf{1}^T x \colon x \in \text{corner}(B)\}, \qquad (\text{corner}(B))$$

$$\alpha_{\text{int}(\mathcal{B})} = \max\{\mathbf{1}^T x \colon x \in \text{int}(\mathcal{B})\}, \qquad (\text{int}(\mathcal{B}))$$

$$\alpha_{\text{strict}(x^*)} = \max\{\mathbf{1}^T x \colon x \in \text{strict}(x^*)\}, \qquad (\text{strict}(x^*))$$

$$\alpha_{\text{LP}\cap\text{corner}(B)} = \max\{\mathbf{1}^T x \colon x \in FSTAB(G) \cap \text{corner}(B)\}. \qquad (\text{LP}\cap\text{corner}(B))$$

In Section 1.7 we have presented some fundamental properties of the linear relaxation (FSTAB) arising from the edge formulation (STAB). One of them asserts that in basic feasible solutions to (FSTAB), variables must be $(0, \frac{1}{2}, 1)$-valued, see Theorem 1.7.1. For an optimal solution x^* to (FSTAB) , we define $P = \{i \in V \colon x_i^* = 1\}$, $Q = \{i \in V \colon x_i^* = \frac{1}{2}\}$, $p = |P|$ and $q = |Q|$. Therefore $\alpha_{\text{FSTAB}} = p + \frac{q}{2}$. Define $G[Q]$ as the subgraph of G induced by nodes of Q.

Our main results are stated in the following theorems.

Theorem 4.3.1. *If $G[Q]$ is connected and admits a perfect or nearly perfect matching, then there exists an optimal basis B associated to x^* such that $\alpha_{corner(B)} = p + \left\lfloor \frac{q}{2} \right\rfloor$.*

Theorem 4.3.2. *Optimizing over strict(x^*) yields the same optimal value as optimizing over the original integer problem, namely $\alpha_{strict(x^*)} = \alpha(G)$.*

Theorem 4.3.3. *If the nodes of Q can be partitioned into cliques of size at least 3, $\alpha_{int(\mathcal{B})} = p + \frac{q}{3}$. If all cliques of the partition have size exactly 3, there exists an optimal basis B associated to x^* such that $\alpha_{corner(B)} = p + \frac{q}{3}$.*

Theorem 4.3.4. *For an optimal basis B associated to x^*, the difference between $\alpha_{corner(B)}$ and $\alpha_{LP\cap corner(B)}$ is at most $\frac{q}{8}$, and there are graphs for which this bound is tight.*

4.4 Complete description of the corner polyhedron and of the intersection closure

The characterization of the bases of $FSTAB(G)$ that we have described in Chapter 2 lays the foundation for the results that we present in this section, due mainly to Campelo and Cornuéjols [17]. These results concern the complete description of the corner polyhedron and of the intersection closure. Given a basis $B \in \mathcal{B}$, consider the associated basic subgraph G_B. The description of the corner polyhedron is obtained from the linear relaxation (FSTAB) of (STAB) by adding an odd cycle inequality for each 1-tree component of G_B. Recall that, given a 1-tree component C_i of G_B, we denote by $\kappa(C_i)$ its unique odd cycle.

Theorem 4.4.1. *(Campelo and Cornuéjols [17]) For every $B \in \mathcal{B}$, the corner polyhedron of (STAB) associated to B is*

$$corner(B) = \left\{ (x, y) \in \mathbb{R}^{n+m} \colon Ax + y = \mathbf{1}, x_N \geq 0, y_N \geq 0, \sum_{e \in \kappa(C_i)} y_e \geq 1, i \in I_1 \right\}. \tag{4.4}$$

Note that in the above description of corner(B), the odd cycle inequalities (1.3) are expressed in terms of the edge-slack variables y.

Let us denote by $\mathcal{B}_+ = \{B \in \mathcal{B} \colon B^{-1}\mathbf{1} \geq \mathbf{0}\}$ the set of feasible bases and by int($\mathcal{B}_+$) the intersection of the corner polyhedra associated to all feasible bases, that is

$$\text{int}(\mathcal{B}_+) = \bigcap_{B \in \mathcal{B}_+} \text{corner}(B), \tag{4.5}$$

and define $\mathcal{C}$ as the set of all the induced odd cycles of $G(V, E)$.

Theorem 4.4.2. *(Campelo and Cornuéjols [17]) int($\mathcal{B}$) = int($\mathcal{B}_+$) = $\bar{S}(G)$, where*

$$\bar{S}(G) = \left\{ (x, y) \in \mathbb{R}^{n+m}_+ \colon Ax + y = 1, \sum_{e \in \mathcal{C}} y_e \geq 1, \forall\, C \in \mathcal{C} \right\}. \tag{4.6}$$

Theorem 4.4.3. *(Campelo and Cornuéjols [17]) For the stable set formulation (STAB), the set $\bar{S}(G)$, the split closure, the Chvátal closure, the $\{0, \frac{1}{2}\}$-Chvátal closure, int($\mathcal{B}$)*

and int($\mathcal{B}_+$) are all identical.

4.5 Restriction to the fractional minor

In this section we show that, in order to prove Theorems 4.3.1-4.3.4, it is sufficient to consider the case where the optimal solution to (FSTAB) is $x_i^* = \frac{1}{2} \quad \forall\ i = 1, \ldots, n$. Given the linear relaxation (FSTAB) of (STAB), if an optimal solution has some 0-1 components we know that, by Theorem 1.7.3, it is possible to fix those components to their values. Therefore, we can restrict our attention to the minor obtained by contracting 0-valued variables and by deleting 1-valued ones. In fact, our approach consists in considering first the subgraph induced by fractional nodes and then extending our results to the original graph.

In the next lemma we show that dropping non-negativity constraints on all x variables from (STAB) does not affect the optimal value when $x_i^* = \frac{1}{2} \quad \forall\ i = 1, \ldots, n$ is an optimal solution of (FSTAB).

Lemma 4.5.1. *Given a graph $G(V, E)$, suppose that $x_i^* = \frac{1}{2}, i = 1, \ldots, n$ is optimal for (FSTAB). Define (NSTAB) as the problem obtained from (STAB) by dropping non-negativity on the x variables. Then:*

- (i) *(NSTAB) has an optimal 0-1 solution;*
- (ii) *if x^* is the unique optimal solution to (FSTAB), all optimal solutions to (NSTAB) are 0-1.*

Proof. For simplicity, we write (NSTAB) as:

$$\begin{aligned} \max \quad & \sum_{i=1}^{n} x_i \\ \text{s.t.} \quad & x_i + x_j \le 1 \qquad \forall\ (i, j) \in E \\ & x_i \in \mathbb{Z}^n \qquad \forall\ i \in V. \end{aligned} \tag{NSTAB}$$

This avoids dealing with the y variables.

First, note that (NSTAB) has a feasible solution (the 0 vector). Second, observe that (NSTAB) is bounded because x^* is optimal for its LP relaxation (this follows from the fact that we remove from (FSTAB) only constraints that are not tight at x^*, i.e. non-negativity on the x variables). Therefore (NSTAB) has an optimal solution.

(i): Assume that $x_i^* = 1/2, i = 1, \ldots, n$ is optimal for (FSTAB) but not necessarily unique. For any feasible solution $\hat{x}$ to (NSTAB), define $S_-(\hat{x}) : \{i \in V : \hat{x}_i < 0\}, S_+(\hat{x}) : \{i \in V : \hat{x}_i > 1\}$. Observe that every node in $S_+(\hat{x})$ can only be adjacent to nodes in $S_-(\hat{x})$. Therefore the incidence vector of $S_+(\hat{x})$ defines a stable set of G. By Theorem 1.7.4, this implies $|S_-(\hat{x})| \geq |S_+(\hat{x})|$.

Let $\bar{x}$ be an optimal solution to (NSTAB). If $S_-(\bar{x}) = \emptyset$, we are done. Define $\Delta(\bar{x}) = \min_{i \in V}\{\bar{x}_i\}$. Note that $\Delta(\bar{x}) \leq -1$. Construct a solution $\tilde{x}$ as:

$$\tilde{x}_k = \begin{cases} \bar{x}_k & \text{for} \quad k \in V \setminus (S_-(\bar{x}) \cup S_+(\bar{x})) \\ \bar{x}_k - 1 & \text{for} \quad k \in V \cap S_+(\bar{x}) \\ \bar{x}_k + 1 & \text{for} \quad k \in V \cap S_-(\bar{x}). \end{cases}$$

We show that $\tilde{x}$ satisfies all the edge constraints. It suffices to prove that increasing by 1 a variable x_i with $i \in S_-(\bar{x})$ does not yield constraint violations. Observe that $\tilde{x}_i \leq 0$. Let j be a node adjacent to i. Either:

- $j \in S_+(\bar{x})$ and $\tilde{x}_j = \bar{x}_j - 1$, or
- $j \in S_-(\bar{x})$ and $\tilde{x}_j \leq 0$, or
- $j \in V \setminus (S_-(\bar{x}) \cup S_+(\bar{x}))$ and $\tilde{x}_j = \bar{x}_j \leq 1$.

In all cases, $\tilde{x}_i + \tilde{x}_j \leq 1$. Therefore $\tilde{x}$ is feasible for (NSTAB) and $\Delta(\tilde{x}) = \Delta(\bar{x}) + 1$. The objective value of $\tilde{x}$ is $\sum_{i \in V} \tilde{x}_i = \sum_{i \in V} \bar{x}_i + |S_-(\bar{x})| - |S_+(\bar{x})| \geq \sum_{i \in V} \bar{x}_i$, so $\tilde{x}$ is optimal. We can iterate this construction from $\tilde{x}$ until we obtain an optimal solution x' with $\Delta(x') = 0$, i.e., $S_-(x') = \emptyset$. This implies that x' has 0-1 components.

(ii): Observe that if $x_i^* = 1/2, i = 1, \ldots, n$ is the unique optimum of (FSTAB), by Theorem 1.7.4 $|S_-(\hat{x})| > |S_+(\hat{x})|$. Let $\bar{x}$ be an optimal solution to (NSTAB) and suppose $S_-(\bar{x}) \neq \emptyset$. Construct $\tilde{x}$ as shown above. $\tilde{x}$ has cost $\sum_{i \in V} \bar{x}_i + |S_-(\bar{x})| - |S_+(\bar{x})| > \sum_{i \in V} \bar{x}_i$. This contradicts optimality of $\bar{x}$, therefore $S_-(\bar{x}) = \emptyset$, i.e. $\bar{x}$ is 0-1. □

We now show that, to prove Theorems 4.3.1-4.3.4, it is sufficient to restrict our attention to the case where the optimum of (FSTAB) is $x^*_i = 1/2, i = 1, \ldots, n$.

Given an optimal solution x^* to (FSTAB), let $V^0 = \{i \in V : x^*_i = 0\}, V^{\frac{1}{2}} = \{i \in V : x^*_i = \frac{1}{2}\}, V^1 = \{i \in V : x^*_i = 1\}$. Define $E^{00} = \{(i,j) \in E : i,j \in V^0\}, E^{0\frac{1}{2}} = \{(i,j) \in E : i \in V^0, j \in V^{\frac{1}{2}}\}, E^{01} = \{(i,j) \in E : i \in V^0, j \in V^1\}, E^{\frac{1}{2}\frac{1}{2}} = \{(i,j) \in E : i,j \in V^{\frac{1}{2}}\}$ (the graph being undirected, the edges are unordered pairs). By Theorem 1.7.1, $V^0, V^{\frac{1}{2}}, V^1$ define a partition of V. Since there can be no edge between V^1 and $V^{\frac{1}{2}} \cup V^1$, it follows that $E^{00}, E^{0\frac{1}{2}}, E^{01}, E^{\frac{1}{2}\frac{1}{2}}$ is a partition of E. We consider two induced subgraphs of G: $G^{\frac{1}{2}}$ induced by $V^{\frac{1}{2}}$, and G^{01} induced by $V^0 \cup V^1$. We show that for all relaxations of (STAB) studied in this chapter, if we are able to compute a bound on $G^{\frac{1}{2}}$, we can generalize its value to G by simply adding $|V^1| = p$.

Theorem 4.5.1. *Let x^* be the optimal solution to (FSTAB) and let B be an optimal basis associated to x^*. Partition x^* according to $V^{\frac{1}{2}}$ and $V^0 \cup V^1$ as $(x^{*\frac{1}{2}}, x^{*01})$. Define $B^{01} = B \cap (V^0 \cup V^1 \cup E^{00} \cup E^{01})$ and $B^{\frac{1}{2}} = B \cap (V^{\frac{1}{2}} \cup E^{\frac{1}{2}\frac{1}{2}})$. Let $G^{01} = G[V^0 \cup V^1]$ and $G^{\frac{1}{2}} = G[V^{\frac{1}{2}}]$. Then:*

(i) *if $\tilde{B}^{\frac{1}{2}}$ is an optimal basis associated to $x^{*\frac{1}{2}}$ for* $\max\{\mathbf{1}^T x \colon x \in FSTAB(G^{\frac{1}{2}})\}$ *and $\tilde{B} = B^{01} \cup \tilde{B}^{\frac{1}{2}} \cup E^{0\frac{1}{2}}$, then:*

1. *$\tilde{B}$ is a basis of (FSTAB),*
2. $\max\{\mathbf{1}^T x \colon x \in corner(\tilde{B})\} = p + \max\{\mathbf{1}^T x \colon x \in corner(\tilde{B}^{\frac{1}{2}})\}$,
3. $\max\{\mathbf{1}^T x \colon x \in FSTAB(G) \cap corner(\tilde{B})\} = p + \max\{\mathbf{1}^T x \colon x \in FSTAB(G^{\frac{1}{2}}) \cap corner(\tilde{B}^{\frac{1}{2}})\}$;

(ii) *if $\mathcal{B}^{\frac{1}{2}}$ is the set of all bases of* $\max\{\mathbf{1}^T x \colon x \in FSTAB(G^{\frac{1}{2}})\}$, *then:* $\max\{\mathbf{1}^T x \colon x \in int(\mathcal{B})\} = p + \max\{\mathbf{1}^T x \colon x \in int(\mathcal{B}^{\frac{1}{2}})\}$;

(iii) $\max\{\mathbf{1}^T x \colon x \in strict(x^*)\} = p + \max\{\mathbf{1}^T x \colon x \in strict(x^{*\frac{1}{2}})\}$.

Proof. First, observe that the constraints corresponding to edges in $E^{0\frac{1}{2}}$ are not tight at x^*. Therefore they can be relaxed without affecting optimality of x^* for (FSTAB). This implies that $\max\{\mathbf{1}^T x \colon x \in FSTAB(G)\} = \max\{\mathbf{1}^T x \colon x \in FSTAB(G^{01})\} + \max\{\mathbf{1}^T x \colon x \in FSTAB(G^{\frac{1}{2}})\}$ and x^{*01} is optimal on $FSTAB(G^{01})$. Since x^{*01} is in $STAB(G^{01})$, it is an optimal stable set in G^{01} and $\max\{\mathbf{1}^T x \colon x \in S(G^{01})\} = \max\{\mathbf{1}^T x \colon x \in FSTAB(G^{01})\}$.

Let $\mathcal{B}^{01}$ be the set of all bases of $\max\{\mathbf{1}^T x \colon x \in FSTAB(G^{01})\}$. We have the chains:

$$\begin{aligned}\max\{\mathbf{1}^T x \colon x \in STAB(G^{01})\} &\leq \max\{\mathbf{1}^T x \colon x \in \text{strict}(x^{*01})\} \leq \\ \max\{\mathbf{1}^T x \colon x \in \text{corner}(B^{01})\} &\leq \max\{\mathbf{1}^T x \colon x \in FSTAB(G^{01})\} = \\ \max\{\mathbf{1}^T x \colon x \in STAB(G^{01})\}\end{aligned}$$

and

$$\begin{aligned}\max\{\mathbf{1}^T x \colon x \in STAB(G^{01})\} &\leq \max\{\mathbf{1}^T x \colon x \in \text{int}(\mathcal{B}^{01})\} \leq \\ \max\{\mathbf{1}^T x \colon x \in FSTAB(G^{01}) \cap \text{corner}(B^{01})\} &\leq \max\{\mathbf{1}^T x \colon x \in FSTAB(G^{01})\} = \\ \max\{\mathbf{1}^T x \colon x \in STAB(G^{01})\},\end{aligned}$$

which imply that x^{*01} is optimal for all the relaxations discussed above on G^{01}, with cost $|V^1| = p$.

(i). $\tilde{B}$ has m elements, and the subgraph $G_{\tilde{B}}$ corresponds to the union of $G[V_0 \cup V_1]$ and $G_{\tilde{B}^{\frac{1}{2}}}$. Therefore, by Theorems 2.2.2 and 2.2.3, $\tilde{B}$ is a basis of $FSTAB(G)$, which proves (i)-1.

Observe that the y variables corresponding to the constraints $E^{0\frac{1}{2}}$ are basic in $\tilde{B}$. Therefore, they become free variables in $\text{corner}(\tilde{B})$ and the constraints $E^{0\frac{1}{2}}$ can be dropped. Since there are no constraints linking G^{01} and $G^{\frac{1}{2}}$ in $\text{corner}(\tilde{B})$, we have that $\max\{\mathbf{1}^T x \colon x \in \text{corner}(\tilde{B})\} = \max\{\mathbf{1}^T x \colon x \in \text{corner}(B^{01})\} + \max\{\mathbf{1}^T x \colon x \in \text{corner}(\tilde{B}^{\frac{1}{2}})\} = p + \max\{\mathbf{1}^T x \colon x \in \text{corner}(\tilde{B}^{\frac{1}{2}})\}$. This proves (i)-2.

For (i)-3, we note that $\max\{\mathbf{1}^T x \colon x \in FSTAB(G) \cap \text{corner}(\tilde{B})\} \leq \max\{\mathbf{1}^T x \colon x \in FSTAB(G^{01}) \cap \text{corner}(B^{01})\} + \max\{\mathbf{1}^T x \colon x \in FSTAB(G^{\frac{1}{2}}) \cap \text{corner}(\tilde{B}^{\frac{1}{2}})\}$ because by optimizing separately over $FSTAB(G^{01}) \cap \text{corner}(B^{01})$ and $FSTAB(G^{\frac{1}{2}}) \cap \text{corner}(\tilde{B}^{\frac{1}{2}})$ we are relaxing the edge constraints $E^{0\frac{1}{2}}$ that are present in FSTAB$\cap\text{corner}(\tilde{B})$. Observe that any optimal solution to $\max\{\mathbf{1}^T x \colon x \in FSTAB(G^{\frac{1}{2}}) \cap \text{corner}(\tilde{B}^{\frac{1}{2}})\}$ has components in $[0,1]$. Pick any such solution $\tilde{x}^{\frac{1}{2}}$. Define $\hat{x}$ as:

$$\hat{x}_i = \begin{cases} \tilde{x}_i^{\frac{1}{2}} & \text{for} \quad i \in V^{\frac{1}{2}} \\ x_i^* & \text{for} \quad i \in V^0 \cup V^1. \end{cases} \tag{4.7}$$

Clearly $\hat{x}$ satisfies the constraints of $FSTAB(G^{01}) \cap \text{corner}(B^{01})$ and $FSTAB(G^{\frac{1}{2}}) \cap \text{corner}(\tilde{B}^{\frac{1}{2}})$. Additionally, it satisfies the edge constraints $E^{0\frac{1}{2}}$ because the variables corresponding to nodes in V^0 have value 0 and those in $V^{\frac{1}{2}}$ have value in $[0,1]$. Thus,

$\hat{x}$ is feasible for $FSTAB(G) \cap \text{corner}(\tilde{B})$ with cost $\max\{\mathbf{1}^T x\colon x \in FSTAB(G^{01}) \cap \text{corner}(B^{01})\} + \max\{\mathbf{1}^T x\colon x \in FSTAB(G^{\frac{1}{2}}) \cap \text{corner}(\tilde{B}^{\frac{1}{2}})\} = p + \max\{\mathbf{1}^T x\colon x \in FSTAB(G^{\frac{1}{2}}) \cap \text{corner}(\tilde{B}^{\frac{1}{2}})\}$, and therefore optimal.

(ii). Recall the description of $\text{int}(\mathcal{B})$ given in Theorem 4.4.2. Observe that $\max\{\mathbf{1}^T x\colon x \in \text{int}(\mathcal{B})\} \leq \max\{\mathbf{1}^T x\colon x \in \text{int}(\mathcal{B}^{01})\} + \max\{\mathbf{1}^T x\colon x \in \text{int}(\mathcal{B}^{\frac{1}{2}})\}$ because by optimizing separately over $\text{int}(\mathcal{B}^{01})$ and $\text{int}(\mathcal{B}^{\frac{1}{2}})$ we are relaxing some of the constraints that define $\text{int}(\mathcal{B})$, namely: the edge constraints $E^{0\frac{1}{2}}$, and the odd cycle inequalities involving at least one edge in $E^{0\frac{1}{2}}$. Let $\tilde{x}^{\frac{1}{2}}$ be an optimal solution to $\max\{\mathbf{1}^T x\colon x \in \text{int}(\mathcal{B}^{\frac{1}{2}})\}$. Define $\hat{x}$ as in (4.7). By construction, $\hat{x}$ satisfies the constraints of $\text{int}(\mathcal{B}^{01})$ and $\text{int}(\mathcal{B}^{\frac{1}{2}})$. Since $\hat{x}_i = 0 \quad \forall\, i \in V_0$, it also satisfies all the edge constraints $E^{0\frac{1}{2}}$ ($\tilde{x}^{\frac{1}{2}}$ has components in $[0,1]$) and any odd cycle inequality involving at least one edge in $E^{0\frac{1}{2}}$ (for any such cycle κ with $2k+1$ edges, no more than $2k$ nodes are in $V^{\frac{1}{2}}$; since they form a chain, the nodes in the cycle add up to at most k). Thus, $\hat{x}$ is feasible for $\text{int}(\mathcal{B})$ with cost $\max\{\mathbf{1}^T x\colon x \in \text{int}(\mathcal{B}^{01})\} + \max\{\mathbf{1}^T x\colon x \in \text{int}(\mathcal{B}^{\frac{1}{2}})\} = p + \max\{\mathbf{1}^T x\colon x \in \text{int}(\mathcal{B}^{\frac{1}{2}})\}$, and therefore optimal.

(iii). We have $\max\{\mathbf{1}^T x\colon x \in \text{strict}(x^*)\} \leq \max\{\mathbf{1}^T x\colon x \in \text{strict}(x^{*01})\} + \max\{\mathbf{1}^T x\colon x \in \text{strict}(x^{*\frac{1}{2}})\}$ since the edge constraints $E^{0\frac{1}{2}}$ are relaxed when optimizing separately over $\text{strict}(x^{*01})$ and $\text{strict}(x^{*\frac{1}{2}})$. By Lemma 4.5.1, there exists an optimal 0-1 solution to $\max\{\mathbf{1}^T x\colon x \in \text{strict}(x^{*\frac{1}{2}})\}$. Let $\tilde{x}^{\frac{1}{2}}$ be such a solution, and define $\hat{x}$ as in (4.7). Observe that $\hat{x}$ is 0-1 and satisfies all the edge constraints, including those in $E^{0\frac{1}{2}}$ because $x_i = 0 \quad \forall\, i \in V^0$. This implies that $\hat{x}$ is feasible for $\text{strict}(x^*)$ with cost $\max\{\mathbf{1}^T x\colon x \in \text{strict}(x^{*01})\} + \max\{\mathbf{1}^T x\colon x \in \text{strict}(x^{*\frac{1}{2}})\} = p + \max\{\mathbf{1}^T x\colon x \in \text{strict}(x^{*\frac{1}{2}})\}$, and therefore optimal. □

By Theorem 4.5.1, the bound provided by a relaxation on $G^{\frac{1}{2}}$ is sufficient to characterize the bound by the same kind of relaxation on G. In particular, for the corner relaxation and LP∩corner we can take any basis of $FSTAB(G^{\frac{1}{2}})$, and there always exists a basis of $FSTAB(G)$ for which the generalization of the bound on $G^{\frac{1}{2}}$ is valid.

4.6 Optimizing over the corner relaxation

We assume that $x_i^* = \frac{1}{2} \;\; \forall\, i = 1, \dots, n$ is an optimal solution to (FSTAB). Thus $z_{\text{LP}} = \frac{n}{2}$. If $m > n$, there are many bases associated to vertex x^*, which may yield different corner relaxations. We show that the strength of these relaxations can be significantly different. We prove that if the graph is connected and its maximum matching has size $\lfloor \frac{n}{2} \rfloor$, there exists an optimal basis associated to x^* yielding a bound of $\lfloor \frac{n}{2} \rfloor$, i.e. a weak bound improvement over (FSTAB). On the other hand, if the graph can be partitioned into triangles, we show that there is also a basis providing the much stronger bound of $\frac{n}{3}$. In the classical random graph model where edges occur independently with a fixed probability p, both of the above conditions hold almost surely (i.e. with probability going to 1 as the number of nodes n increases) [15] when n is a multiple of 3. This implies that almost all graphs have both a weak corner relaxation with bound $\lfloor \frac{n}{2} \rfloor$ and a much stronger one with bound of the order of $\frac{n}{3}$.

For each basis $B \in \mathcal{B}$ associated to x^*, all x variables are positive and belong to B. In the corner polyhedron we drop the non-negativity constraints on variables y_{ij} such that $(i, j) \in E_B$. This corresponds to removing the redundant constraints of type $x_i + x_j + y_{ij} = 1$ for each $(i, j) \in E_B$. Thus, the corner polyhedron associated to B is the convex hull of the points satisfying

$$\begin{aligned} & x_i + x_j + y_{ij} = 1 && \forall\, (i, j) \in E_N \\ & y_{ij} \geq 0 && \forall\, (i, j) \in E_N \\ & x_i \in \mathbb{Z}^n && \forall\, i \in V. \end{aligned} \tag{4.8}$$

Using the graphic characterization of the bases described in Chapter 2, we show that any basis B associated to x^* has, in general, an associated graph G_B with $k \geq 1$ connected components, each one representing a 1-tree.

Lemma 4.6.1. *Any basis B associated to vertex $x_i^* = \frac{1}{2} \;\; \forall\, i = 1, \dots, n$ is such that all connected components $C_1, \dots, C_k, \;\; k \geq 1$ are 1-trees.*

Proof. By contradiction, suppose this is not the case, that is, there exists at least one connected component $C_i(V_i, E_i)$ which is a tree. Then, by Theorem 2.2.3, some components of x^* would have binary values, precisely $x_j^* \in \{0, 1\} \; \forall\, j \in V_i$. □

Consider now the linear relaxation of (4.8). The feasible set of such problem corresponds to the cone $P(B)$ defined in Section 4.1, for our particular case. Precisely, $P(B)$ has the form:

$$P(B) = \{(x, y) \colon Ax + y = 1, y_N \geq 0\} . \tag{4.9}$$

For a general MILP, Lemma 4.1.1 establishes that if $P(B)$ contains a point satisfying the integrality constraints of (MILP), then any basis B of the linear relaxation (LP) is optimal if and only if corner(B) has an optimal solution. For sake of completeness, we prove a specialization of this result to the edge formulation of the stable set problem. An example is given in Figure 4.2.

Lemma 4.6.2. *Suppose $x^*_i = \frac{1}{2}$ $\forall\, i = 1, \ldots, n$ is an optimal solution of (FSTAB). Any basis B associated to x^* is optimal for (FSTAB) if and only if corner(B) has an optimal solution.*

Proof. If B is an optimal basis of (FSTAB) associated to x^*, all the reduced costs of nonbasic variables are nonpositive. The objective function can be rewritten in terms of the nonbasic variables as $\frac{n}{2} + \max \sum_{(i,j) \in E_N} c'_{ij} y_{ij}$, where for all $(i, j) \in E_N$, c'_{ij} is the reduced cost of nonbasic variable y_{ij}. Because $c'_{ij} \leq 0$ $\forall (i, j) \in E \cap N$ and $y_{ij} \geq 0$ $\forall (i, j) \in E_N$, optimizing over (4.8) is not unbounded. As (4.8) is non-empty, corner(B) has an optimal solution. Assume now that problem (corner(B)) has an optimal solution and, by contradiction, suppose $c'_{uv} > 0$ for some $(u, v) \in E_N$. It is possible to increase variable y_{uv} by a positive integer M, without modifying any of the other (nonbasic) y variables, in such a way that all x_i are integer (because all x_i's are basic and unrestricted in sign). This would yield an improvement of the objective function equal to Mc'_{uv}, showing that (corner(B)) is unbounded for $M \to \infty$. □

Definition 4.6.1. *A bipartite graph $B(U, V, W)$ is* balanced *if* $|U| = |V|$.

Definition 4.6.2. *A bipartite graph $B(U, V, W)$ is* nearly balanced *if* $|U| - |V| = \pm 1$.

Definition 4.6.3. *We define a 1-tree component $C_i(V_i, E_i)$ to be* unbalanced *if it admits a stable set $P_i \subseteq V_i$ such that $2|P_i| \geq |V_i| + 1$. Otherwise, we say that it is* balanced.

Lemma 4.6.3. *Let B be a basis of (FSTAB) associated to x^*, and let G_B be the corresponding basic subgraph. B is an optimal basis of (FSTAB) if and only if x^* is an optimal solution of (FSTAB) over G_B.*

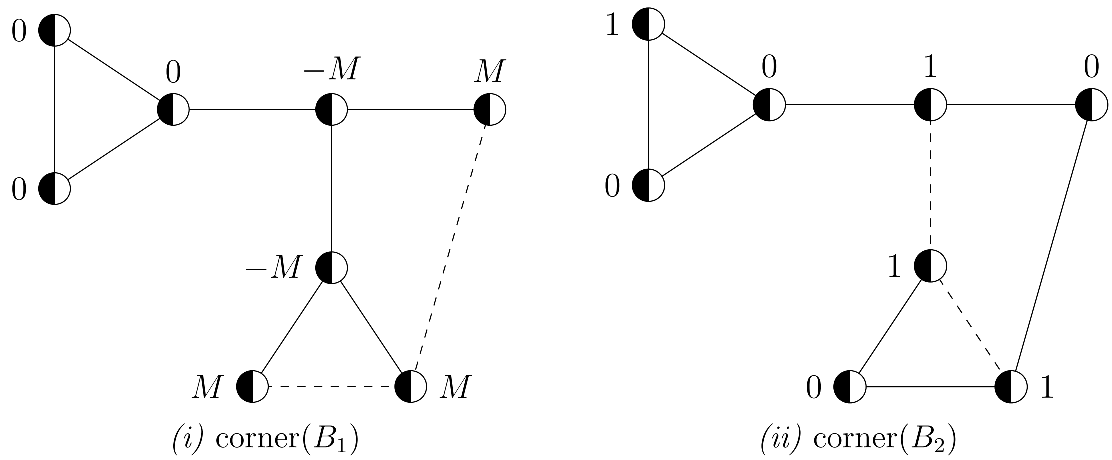

Figure 4.2: The given graph admits $x_i^* = \frac{1}{2}\ \forall\ i \in V$ as the unique optimum of (FSTAB). (i) and (ii) represent two bases associated to x^*. However, B_1 is not an optimal basis: for $M \in \mathbb{Z}_+$ the assignment depicted satisfies the constraints of corner(B_1). This implies that corner(B_1) is unbounded. On the other hand, B_2 is an optimal basis associated to x^* and correspondingly corner(B_2) has a finite optimum. An optimal solution of corner(B_2) is the 0-1 solution represented in (ii).

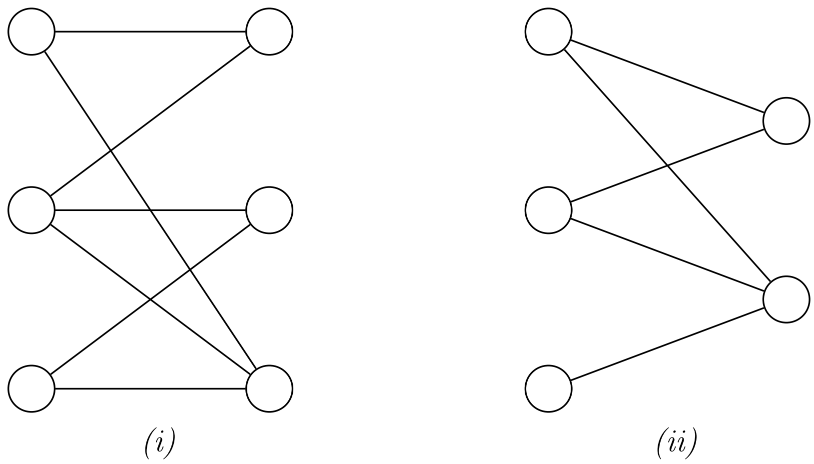

Figure 4.3: A balanced bipartite graph (i) and a nearly balanced one (ii).

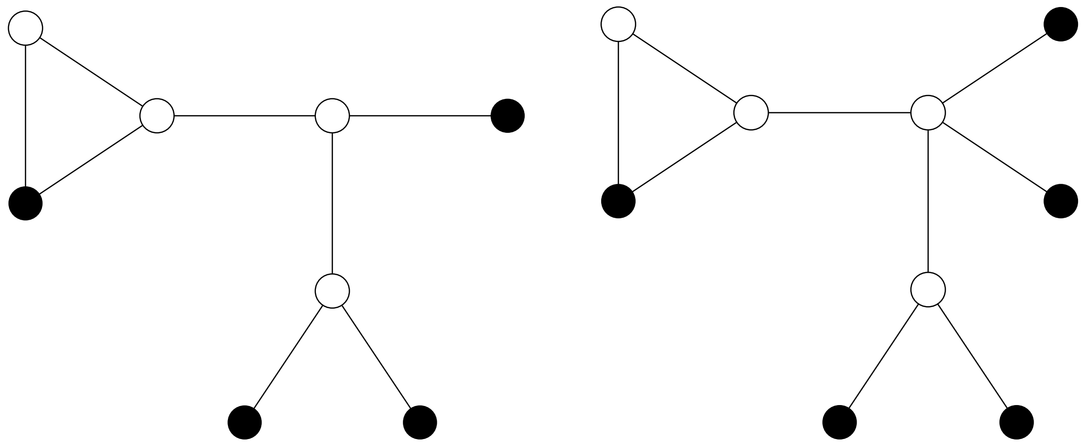

Figure 4.4: A balanced 1-tree (i) and an unbalanced one (ii).

Proof. Suppose B is an optimal basis of (FSTAB). Relaxing non-negativity constraints of basic variables of $FSTAB(G)$ corresponds to optimizing on $FSTAB(G_B)$. Because B is an optimal basis, dropping the constraints associated to basic variables does not affect optimality of x^*. For the converse, suppose x^* is an optimal solution on $FSTAB(G_B)$. By Lemma 4.5.1, (NSTAB) admits a 0-1 optimal solution on G_B. Remark that the latter problem is exactly corner(B). This implies that corner(B) has an optimal solution and therefore, by Lemma 4.6.2, B is an optimal basis of (FSTAB). □

As an example, consider again the bases represented in Fig. 4.2. Basis B_1 is not optimal and correspondingly $x_i^* = \frac{1}{2}\ \forall\ i \in V$ is not an optimal solution of corner(B_1). Indeed, the solution that assigns values 1, 0 and $\frac{1}{2}$ to the nodes that in Fig. 4.2 (i) take values M, $-M$ and 0, respectively, is an optimum of corner(B_1).

Lemma 4.6.4. *Suppose B is an optimal basis of (LP) associated to x^*. Then all its components C_i are balanced.*

Proof. By Lemma 4.6.3, optimizing over $FSTAB(G_B)$ gives an upper bound of $\frac{n}{2}$ for (STAB) over G_B. This implies that the maximum stable set of G_B has size at most $\lfloor \frac{n}{2} \rfloor$. □

Note that the condition stated in Lemma 4.6.4 is only necessary. For example, basis B_1 of Fig.4.2 (i) yields a balanced 1-tree but is not an optimal basis.

Lemma 4.6.5. *If $C(V, E)$ is a balanced 1-tree containing an odd cycle, then there exists an edge $(u_j, u_{j+1}) \in \kappa(C)$ such that $C'(V, E \setminus (u_j, u_{j+1}))$ is a bipartite graph which is nearly balanced if $|V|$ is odd, or balanced if $|V|$ is even.*

Proof. Suppose that $|\kappa(C)| = k$ and let $u_1, u_2, \dots, u_k, u_{k+1} = u_1$ be the nodes of $\kappa(C)$. By contradiction, suppose that for all $j = 1, \dots, k$ the removal of edge (u_j, u_{j+1}) yields the tree $T_j(V_j^+, V_j^-, E \setminus (u_j, u_{j+1}))$, such that $|V_j^+| \geq |V_j^-| + 2$. As there is a path of even length connecting u_j and u_{j+1}, they both belong to the same side of the partition. They cannot both belong to V_j^-, otherwise V_j^+ would be a stable set of C such that $2|V_j^+| \geq |V_j^+| + |V_j^-| + 2 = |V| + 2$ and therefore C would be unbalanced. Thus, $u_j, u_{j+1} \in V_j^+$.

We first prove that the inequality $|V_j^+| \geq |V_j^-| + 2$ cannot hold strictly. Suppose otherwise. The stable set $P = V_j^+ \setminus \{u_j\}$ would be such that $2|P| \geq |V| + 1$, implying again that C is unbalanced. Thus, it can be only $|V_j^+| = |V_j^-| + 2$. If $|V|$ is odd, this is not possible and the first part of the statement is proven.

Consider therefore the case where $|V|$ is even. Observe that C can be partitioned into k branches $B(u_j)$, one departing from each node of the odd cycle u_j, $j = 1, \dots, k$ (Fig 4.5 (i)). By contradiction, suppose that for all $j = 1, \dots, k$ removing (u_j, u_{j+1}) yields a tree T_j whose bipartition satisfies $|V_j^+| = |V_j^-| + 2$. Remark that, when we remove two consecutive edges (u_{j-1}, u_j) and (u_j, u_{j+1}) the corresponding trees T_{j-1} and T_j are such that: $V_{j-1}^+ \cap B(u_j) = V_j^+ \cap B(u_j)$ and $V_{j-1}^- \cap B(u_j) = V_j^- \cap B(u_j)$; $V_{j-1}^+ \setminus B(u_j) = V_j^- \setminus B(u_j)$ and $V_{j-1}^- \setminus B(u_j) = V_j^+ \setminus B(u_j)$ (Fig 4.5 (ii) and (iii)). Following this remark and recalling that $2 = |V_j^+| - |V_j^-| = |V_{j-1}^+| - |V_{j-1}^-|$, by elementary algebraic manipulations it follows that $|B(u_j) \cap V_j^+| = |B(u_j) \cap V_j^-| + 2$ (Fig 4.5 (iv)). Therefore, one can build a stable set P of C by selecting in each branch $B(u_j)$, all nodes in $B(u_j) \cap V_j^+$, and by including or excluding u_j depending on j being even or odd, i.e.

$$P = \left(\bigcup_{j=1,\dots,k} V_j^+ \right) \setminus \left(\bigcup_{\substack{j=1,\dots,k \\ j \text{ odd}}} u_j \right).$$

The corresponding stable set has therefore size $\frac{|V|+k-1}{2} \geq \frac{|V|}{2} + 1$, contradicting the hypothesis that C is a balanced 1-tree. $\square$

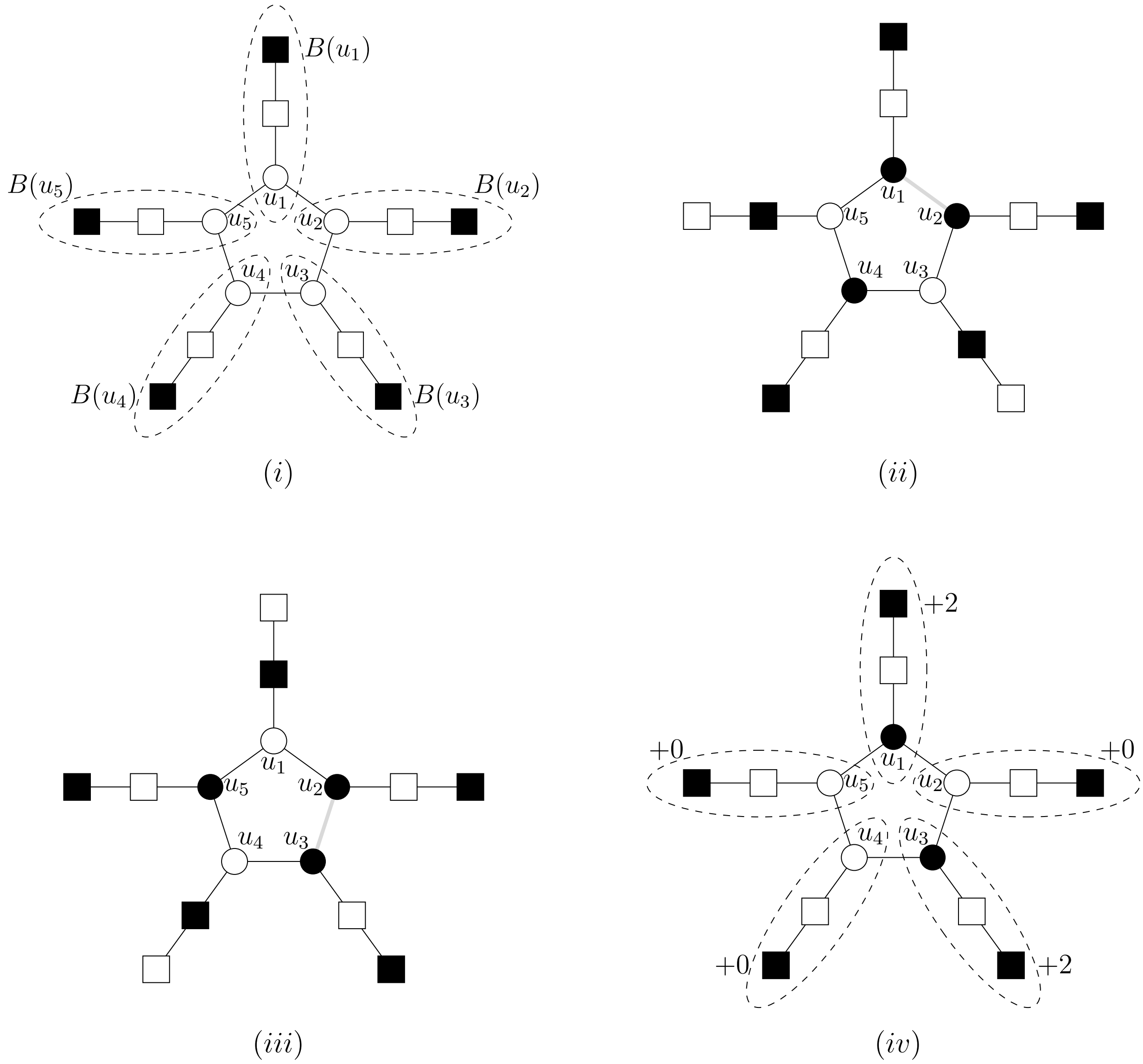

Figure 4.5: The nodes of the 1-tree can be partitioned into branches departing from each node of the odd cycle (*i*). Each branch is a tree, therefore a bipartite graph, whose sides are indicated by a black and a white square. Removing two consecutive edges of the odd cycle modifies the bipartition of the so-obtained trees as represented in (*ii*) and (*iii*). Supposing, by contradiction, that the removal of each edge of the cycle yields an unbalanced tree contradicts the hypothesis that the 1-tree is balanced (*iv*).

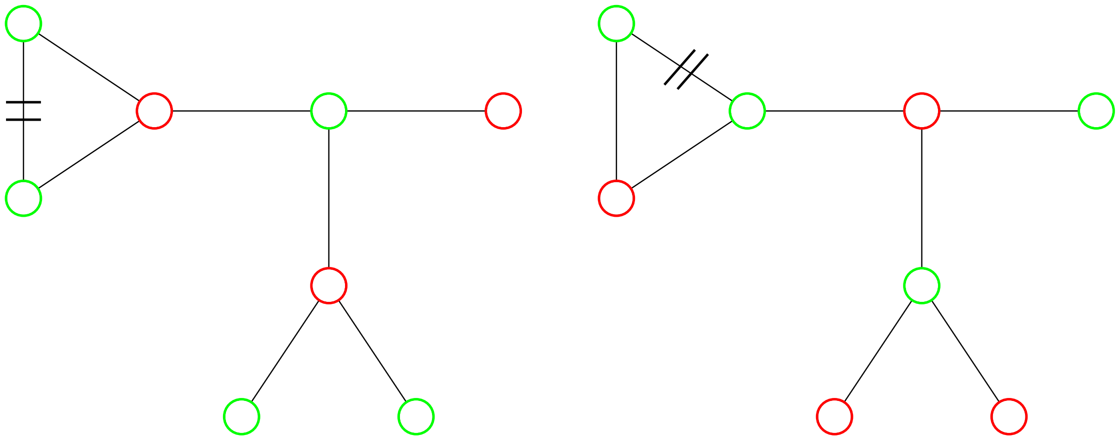

Figure 4.6: There always exists an edge belonging to the odd cycle of a balanced 1-tree which, removed, yields a balanced tree.

An illustration of Lemma 4.6.5 is given in Fig. 4.6.

Lemma 4.6.6. *Let $B \in \mathcal{B}$ be an optimal basis associated to x^* and consider any 1-tree component $C_i(V_i, E_i)$ of G_B. There exists an edge $(u_i, v_i) \in \kappa(C_i)$ which, if removed, yields a tree with all of its stable sets P_i satisfying $|P_i| \leq \left\lceil \frac{|V_i|}{2} \right\rceil$.*

Proof. By Lemma 4.6.4, C_i is balanced. By Lemma 4.6.5, we can remove an edge $(u_i, v_i) \in \kappa(C_i)$ in order to obtain a bipartite graph, which is balanced or nearly balanced depending on the parity of $|V_i|$. Let us denote by V_i^+ and V_i^- the two sets of the bipartition and suppose w.l.o.g. $|V_i^+| = \left\lceil \frac{|V_i|}{2} \right\rceil$ and $|V_i^-| = \left\lfloor \frac{|V_i|}{2} \right\rfloor$ (Fig. 4.7 (i)).

By contradiction, suppose that there exists a stable set P_i of $C_i'(V_i, E_i \setminus (u_i, v_i))$ such that $|P_i| > \left\lceil \frac{|V_i|}{2} \right\rceil$. Consider $P_i \cap V_i^+$ and $P_i \cap V_i^-$. Clearly neither V_i^+, nor P_i are stable sets of C_i, because C_i is a balanced 1-tree. This implies that $u_i, v_i \in P_i \cap V_i^+$.

Because B is optimal, by Lemma 4.6.3, x^* is optimal for the linear relaxation of (STAB) on G_B. A feasible solution $\tilde{x}$ for $FSTAB(G_B)$ can be obtained as

$$\tilde{x}_j = \begin{cases} 0 & \text{for} \quad j \in V_i^+ \setminus P_i \\ 1 & \text{for} \quad j \in P_i \cap V_i^- \\ \frac{1}{2} & \text{for} \quad j \in (V_i^- \setminus P_i) \cup (P_i \cap V_i^+) \end{cases}$$

Because $|V_i^- \cap P_i| > |V_i^+ \setminus P_i|$, for this latter solution $\sum_{i=1}^n \tilde{x}_i \geq \frac{|V_i|+1}{2}$ which contradicts optimality of x^* on $FSTAB(G_B)$ (Fig. 4.7 (ii)). □

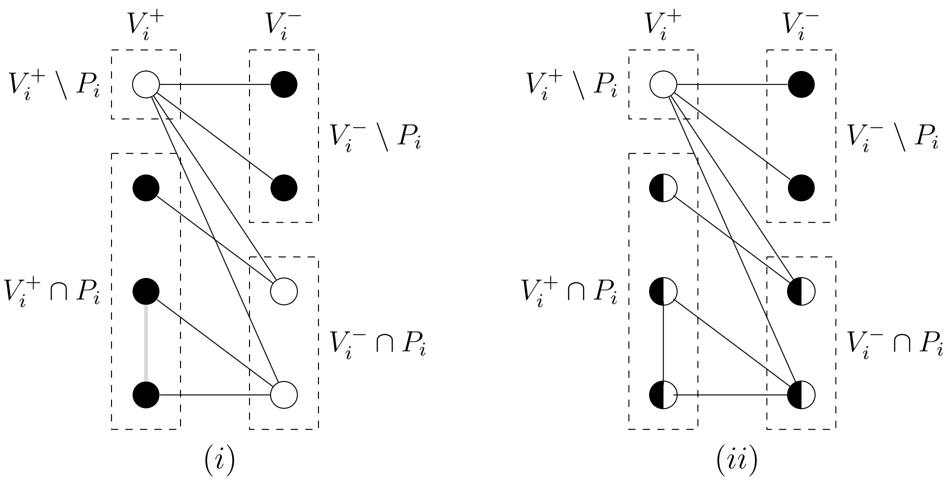

Figure 4.7: An illustration of the proof of Lemma 4.6.6.

Theorem 4.6.1. *Consider (STAB) and its linear relaxation (FSTAB). Suppose* $x_i^* = \frac{1}{2} \;\; \forall\, i = 1, \ldots, n$ *is an optimal solution of (FSTAB) and B is an optimal basis associated to* x^**, composed by k 1-tree components* $C_i(V_i, E_i)$*,* $i = 1, \ldots, k$*. Then,* $z_{corner(B)} = \frac{n-k_o}{2}$*, where* $k_o \le k$ *is the number of odd components among* C_i*,* $i = 1, \ldots, k$*.*

Proof. For all $i = 1, \ldots, k$ define

$$W_i = \{(x, y) \in \mathbb{Z}^{|V_i|} \times \mathbb{R}_+^{|E_i|} \colon x_u + x_v + y_{uv} = 1 \;\; \forall\, (u, v) \in E_i\}$$

and partition (x, y) according to components C_i into $\{(x^i, y^i)\}$, $i = 1, \ldots, k$. Problem (corner(B)) can be split into k independent problems, one for each connected component of G_B, because $z_{\text{corner}(B)} = \sum_{i=1}^k z(C_i)$, where

$$z(C_i) = \max\{\mathbf{1}^T x^i \colon (x^i, y^i) \in W_i\} \tag{4.10}$$

By optimality of B and Lemma 4.5.1, (4.10) admits an optimal $(0,1)$-valued solution (x^i, y^i), such that x^i is the incidence vector of a stable set P_i of nodes in C_i. By Lemma 4.6.4, because B is an optimal basis, C_i is balanced and therefore $|P_i| = z(C_i) \le \left\lfloor \frac{|V_i|}{2} \right\rfloor$. By Lemma 4.6.5, there exists an edge $(u_i, v_i) \in \kappa(C_i)$ such that $C_i'(V_i, E_i \setminus (u_i, v_i))$ is a bipartite graph which is balanced or nearly balanced, depending on $|V_i|$ being even or odd, respectively. Moreover, by Lemma 4.6.6, C_i' is such that all its stable sets have size at most $\left\lceil \frac{|V_i|}{2} \right\rceil$.

This implies that a maximum stable set of C_i corresponds to the side of the bipartition of C_i' with has cardinality $\left\lfloor \frac{|V_i|}{2} \right\rfloor$ and does not contain (u_i, v_i). Therefore, there always exists in C_i a stable set of cardinality $\left\lfloor \frac{|V_i|}{2} \right\rfloor$, implying $z(C_i) = \left\lfloor \frac{|V_i|}{2} \right\rfloor$, which completes the proof of the theorem. □

Next, we show that if the graph is connected and its maximum matching has size $\left\lfloor \frac{n}{2} \right\rfloor$, there always exists an optimal basis that has only one connected component.

Lemma 4.6.7. *Let G be a connected graph on n nodes. Consider (STAB) and its linear relaxation (FSTAB). Suppose $x_i^* = \frac{1}{2}$ $\forall i = 1, \ldots, n$ is an optimal solution of (FSTAB). There exists an optimal basis B associated to x^* which has only one connected component if and only if G admits a perfect matching, if n is even, or a nearly perfect matching, if n is odd.*

Proof. Let us first show that the condition is sufficient.

If n is even and G admits a perfect matching, it is possible to incrementally build a spanning tree of G, such that all edges of the perfect matching belong to the tree. This can be done by adding to the tree, at each iteration, first an edge of the matching, and then an edge of the cutset separating the nodes in the tree from the nodes outside the tree. By construction, a maximum stable set in this spanning tree has size $\frac{n}{2}$. Moreover, because the tree is bipartite, both sides of the bipartition correspond to maximum stable sets of the spanning tree. Now, recall that G admits x^* as an optimal solution of (FSTAB). This implies that G cannot be bipartite, hence there exists an edge between two nodes on the same side of the bipartition. Adding this edge to the spanning tree yields a 1-tree with an odd cycle. It follows that an optimal solution of (FSTAB) on the 1-tree is x^*, proving that the 1-tree corresponds to an optimal basis.

If n is odd and G admits a nearly perfect matching, consider $G-v$, the graph obtained from G by removing node v and all its incident edges, where v is the only exposed node of the matching. Remark that $G - v$ may not be connected. Because $G - v$ admits a perfect matching, every connected component has an even number of nodes. Applying the same procedure described for the case of n even, it is possible to build a forest spanning $G - v$, in such a way that all the edges of the nearly perfect matching belong to the forest. The maximum stable set in this forest has cardinality $\frac{n-1}{2}$. It is then possible to connect v to the forest, in order to obtain a spanning tree of G, whose maximum stable set has at most cardinality $\frac{n+1}{2}$. In this case, a maximum stable set of the tree is given by the side of its bipartition which contains v. Recalling that x^* is an optimal solution of (FSTAB), it follows that there cannot exist a stable set of size $\frac{n+1}{2}$. This implies that there exists an edge between two nodes in the side of the bipartition that contains v. We can add this edge to the tree, in order to obtain a 1-tree with an odd cycle, such that the optimal solution of (FSTAB) on the 1-tree has value $\frac{n}{2}$.

In order to show the converse recall that, by Lemma 1.6.1, for a bipartite graph B of n nodes, $n = \nu(B) + \alpha(B)$, where $\nu(B)$ and $\alpha(B)$ are the size of the maximum matching and of the maximum stable set in B, respectively. By Lemma 4.6.6, given an optimal basis associated to x^*, which consists of a unique 1-tree, it is possible to remove an edge of the odd cycle of the 1-tree, in order to obtain a tree such that its maximum stable set has cardinality $\lceil \frac{n}{2} \rceil$. This implies that in the same tree, the maximum matching has cardinality $\lfloor \frac{n}{2} \rfloor$. Therefore, G has a perfect matching, if n is even, or a nearly perfect matching, if n is odd. □

An illustration of Lemma 4.6.7 is given in Fig. 4.8: the graph represented in (i) does not admit a perfect matching and there does not exist an optimal basis composed by a unique 1-tree component; for instance, the 1-tree depicted in (ii) corresponds to a basis that is not optimal (the corresponding corner relaxation is unbounded); the graph represented in (iii) has a perfect matching and an optimal basis composed by a unique 1-tree component is depicted in (iv).

Theorem 4.6.2. *Let G be a connected graph on n nodes. Consider (STAB) and its linear relaxation (FSTAB). Suppose $x_i^* = \frac{1}{2}$ $\forall\, i = 1, \ldots, n$ is an optimal solution of (FSTAB). There exists an optimal basis B associated to x^* such that $z_{corner(B)} = \lfloor \frac{n}{2} \rfloor$ if and only if G admits a perfect matching, if n is even, or a nearly perfect matching, if n is odd.*

Proof. By Lemma 4.6.7 there exists an optimal basis associated to x^*, which is composed by a unique 1-tree if and only if G admits a perfect matching, if n is even, or a nearly perfect matching, if n is odd. Under this assumption, applying Theorem 4.6.1 with $k = 1$, it follows that $z_{\text{corner}(B)} = \frac{n-1}{2}$, if n is odd, and $z_{\text{corner}(B)} = \frac{n}{2}$ if n is even. □

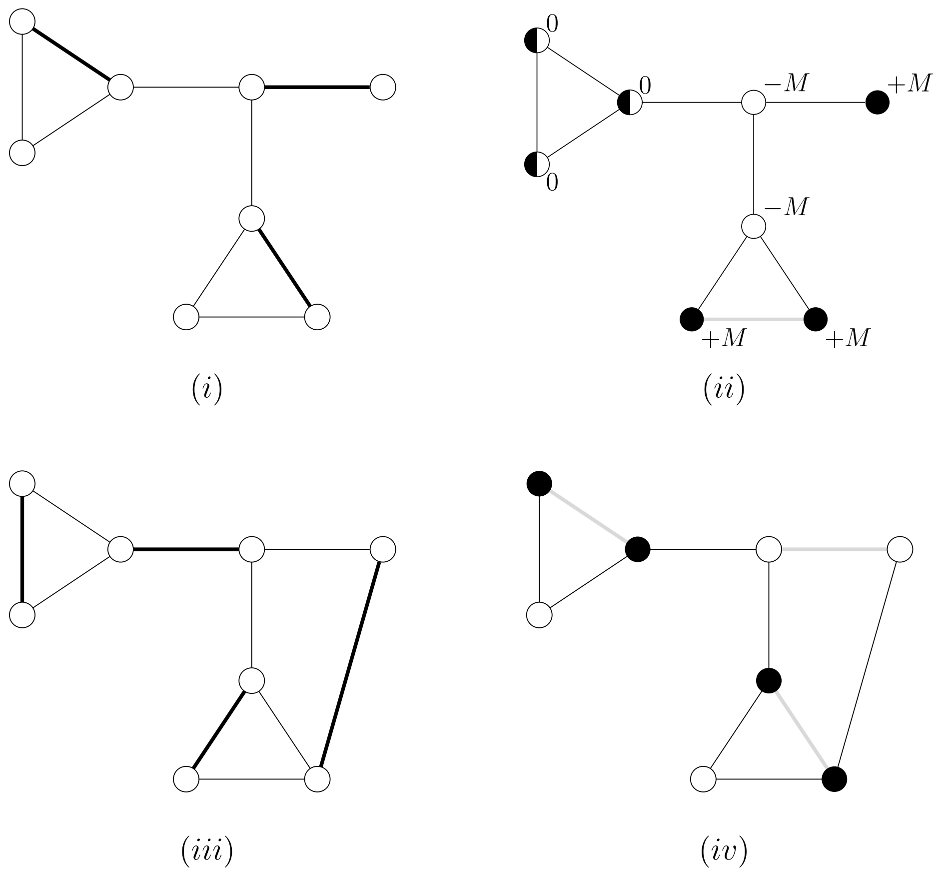

Figure 4.8: An illustration of Lemma 4.6.7.

Theorems 4.5.1 and 4.6.2 imply Theorem 4.3.1.

Theorem 4.6.2 highlights the unlucky possibility where a basis yields an extremely weak corner relaxation. On the other hand, there may be the chance of choosing a basis which provides a much stronger corner relaxation, as shown in the next theorem.

Theorem 4.6.3. *Suppose that there exists a partition of V into triangles, i.e., cliques of size 3. Then there is an optimal basis B associated to x^* such that $z_{corner(B)} = \frac{n}{3}$.*

Proof. Such a basis has $\frac{n}{3}$ connected 1-tree components corresponding to the partition into triangles. By Theorem 4.6.1, because $k_o = \frac{n}{3}$, $z_{\text{corner}(B)} = \frac{n}{3}$. □

A sufficient condition for V to be partitioned into triangles is established by [26] and amounts to requiring that the minimum node degree is at least $\frac{2}{3}n$. A random graph $G(n,p)$ almost surely has such a partition whenever $n = 3k$ and $p \geq O(\frac{1}{n^{0.6}})$ [51].

4.7 Optimizing over the intersection closure

Because bounds from corner relaxations can be significantly different, instead of relying on a single basis, it may be advantageous to consider the intersection of the corner polyhedra associated to all bases.

We now study problem (int($\mathcal{B}$)). By Theorem 4.4.2, (int($\mathcal{B}$)) can be expressed as

$$
\begin{aligned}
z_{\text{int}(\mathcal{B})} = \max \quad & \sum_{i=1}^{n} x_i \\
\text{s.t.} \quad & x_i + x_j + y_{ij} = 1 && \forall\ (i,j) \in E \\
& y_{ij} \geq 0 && \forall\ (i,j) \in E \\
& \textstyle\sum_{(i,j)\in C} y_{ij} \geq 1 && \forall C \in \mathcal{C}.
\end{aligned}
\tag{4.11}
$$

Proposition 4.7.1. *$z_{int(\mathcal{B})} \geq \frac{n}{3}$.*

Proof. Consider vector $x'_i = \frac{1}{3}\ \forall\ i \in V$, $y_{ij} = \frac{1}{3}\ \forall (i,j) \in E$. We want to prove feasibility of x'. For every induced odd cycle $C \in \mathcal{C}$, denote by $l(C)$ the length of the cycle. For every $C \in \mathcal{C}$ the corresponding odd cycle constraint is satisfied:

$$\sum_{(i,j)\in C} y_{ij} = \frac{l(C)}{3} \geq 1,$$

where the last inequality follows by $l(C) \geq 3$ $\forall C \in \mathcal{C}$. All the other constraints are trivially satisfied by x', and this implies that $\sum_{i=1}^{n} x'_i = \frac{n}{3}$ is a lower bound for $z_{\text{int}(\mathcal{B})}$. □

We now state a sufficient condition for $z_{\text{int}(\mathcal{B})}$ to be $\frac{n}{3}$.

Theorem 4.7.1. *Assume that there exists a partition of V into cliques of size at least 3. Then $z_{int(\mathcal{B})} = \frac{n}{3}$.*

Proof. Assume $G(V,E)$ can be partitioned in h cliques $\{Q_i\}$, $i = \{1,\dots,h\}$. Denote by $V(Q_i)$ the set of nodes in Q_i and define the size of every clique as $s(Q_i) = |V(Q_i)|$. Note that every clique Q_i of size at least 3 contains exactly $\binom{s(Q_i)}{3}$ triangles, and each node is in $\binom{s(Q_i)-1}{2}$ triangles. Remark also that every odd cycle inequality of type

$$\sum_{(i,j)\in C} y_{ij} \geq 1, \quad C \in \mathcal{C},$$

can be rewritten in term of the x variables as

$$\sum_{i\in C} x_i \leq \frac{l(C)-1}{2}, \quad C \in \mathcal{C}.$$

Consequently, for each $i \in \{1,\dots,h\}$, summing up all triangle inequalities on clique Q_i yields the valid inequality

$$\binom{s(Q_i)-1}{2} \sum_{i\in V(Q_i)} x_i \leq \binom{s(Q_i)}{3},$$

which implies

$$\sum_{i\in V(Q_i)} x_i \leq \frac{\binom{s(Q_i)}{3}}{\binom{s(Q_i)-1}{2}} = \frac{s(Q_i)}{3}. \tag{4.12}$$

Summing up inequalities (4.12) over the cliques in the partition, we get

$$\sum_{i=1}^{n} x_i \leq \frac{1}{3}\sum_{i=1}^{h} s(Q_i) = \frac{n}{3}.$$

By Proposition (4.7.1), $z_{\text{int}(\mathcal{B})} \geq \frac{n}{3}$. The two results imply $z_{\text{int}(\mathcal{B})} = \frac{n}{3}$. □

Theorems 4.5.1, 4.6.3 and 4.7.1 imply Theorem 4.3.3.

4.8 Optimizing over the strict corner relaxation

Assume that $x_i^* = \frac{1}{2}\ \ \forall\ i = 1, \ldots, n$ is an optimal solution to (FSTAB). If $m > n$ there are many optimal bases associated with x^*. Let B be one of these bases. In this section we study the strict corner relaxation of (STAB), obtained by relaxing non-negativity of the strictly positive basic variables. The strict corner is a tighter relaxation than the corner relaxation, because the latter relaxes non-negativity of all the basic variables, i.e., also degenerate basic variables. Note that the strict corner relaxation does not depend on the choice of B, since all degenerate bases associated with x^* have the same non-degenerate basic variables.

Observe that all the edge constraints are tight at x^*. Therefore, problem (strict(x^*)) reads:

$$\begin{aligned} \max\ & \sum_{i=1}^{n} x_i \\ \text{s.t.}\ & x_i + x_j + y_{ij} = 1 && \forall\ (i,j) \in E \\ & y_{ij} \geq 0 && \forall\ (i,j) \in E \\ & x_i \in \mathbb{Z}^n && \forall\ i \in V. \end{aligned} \tag{STR}$$

The main result of this section consists in showing that (STAB) and strict(x^*) have the same optimal value.

Theorem 4.8.1. *If $x_i^* = \frac{1}{2}\ \ \forall\ i = 1, \ldots, n$ is the optimum of (FSTAB), $z_{STAB} = z_{strict(x^*)}$.*

Proof. $z_{\text{STAB}} \leq z_{\text{strict}(x^*)}$ because (STR) is a relaxation of (STAB). By Lemma 4.5.1 (i), (STR) has an optimal solution that is 0-1. This solution is feasible for (STAB). Therefore $z_{\text{STAB}} \geq z_{\text{strict}(x^*)}$. □

Together with Theorem 4.5.1, this implies Theorem 4.3.2.

Even though $z_{\text{strict}(x^*)} = z_{\text{STAB}}$, optimal solutions to (STR) are not always feasible for (STAB) when (FSTAB) has alternate optimal solutions. However, when x^* is the unique optimal solution to (STAB), the following holds.

Theorem 4.8.2. *Suppose that $x_i^* = \frac{1}{2}\ \forall\, i = 1, \ldots, n$ is the unique optimal solution to (FSTAB). Then the optimal solution to (STR) is 0-1.*

Proof. Follows immediately from Lemma 4.5.1 (ii). □

4.9 Strengthening the LP relaxation with the description of the corner polyhedron

In this section we study the strength of (LP∩corner(B)) for an optimal basis B of (FSTAB) associated to $x_i^* = \frac{1}{2}\ \forall\, i = 1, \ldots, n$.

Theorem 4.9.1. *Given graph $G(V, E)$, let B be an optimal basis of (FSTAB) associated to $x_i^* = \frac{1}{2}\ \forall\, i = 1, \ldots, n$. Suppose that B is composed by k 1-tree components $C_i(V_i, E_i)$, $i = 1, \ldots, k$. Then $\frac{n-k}{2} \leq z_{LP\cap corner(B)}$. If G is the complete graph on n nodes, $z_{LP\cap corner(B)} = \frac{n-k}{2}$.*

Proof. We start by proving $\frac{n-k}{2} \leq z_{\text{LP}\cap\text{corner}(B)}$. By Theorem 4.4.1, the intersection of (FSTAB) and corner(B) is given by (FSTAB) plus the odd cycle inequalities of (4.4). Therefore we can express problem (LP∩corner(B)) on G in terms of the x variables as:

$$\begin{aligned} z_{\text{LP}\cap\text{corner}(B)} = \max \quad & \sum_{j=1}^{n} x_j & & \\ \text{s.t.} \quad & x_i + x_j \leq 1 & & \forall\, (i,j) \in E \\ & x_j \geq 0 & & \forall\, j \in V \\ & \textstyle\sum_{j\in\kappa(C_i)} x_j \leq \frac{|\kappa(C_i)|-1}{2} & & i = 1, \ldots, k. \end{aligned} \tag{4.13}$$

Observe that problem

$$
\begin{array}{rlll}
z_{\text{clique}} = \max & \sum_{j=1}^{n} x_j & & \\
\text{s.t.} & x_i + x_j \leq 1 & \forall\,(i,j) \in E & \\
& x_i + x_j \leq 1 & \forall\,(i,j) \notin E & \\
& x_j \geq 0 & \forall\, j \in V & \\
& \sum_{j \in \kappa(C_i)} x_j \leq \frac{|\kappa(C_i)|-1}{2} & i = 1,\dots,k, &
\end{array}
\tag{4.14}
$$

is obtained from (4.13) by adding constraints relative to the edges of the complement graph. This implies $z_{\text{clique}} \leq z_{\text{LP}\cap\text{corner}(B)}$. Remark also that, letting $B' = B \cup \{(i,j) \notin E\}$, (4.14) corresponds to problem (LP$\cap$corner(B')) on K_n, the clique defined on the node set V. This is because $N' = N$ indexes the same 1-tree components C_i, $i = 1,\dots,k$. We first show $z_{\text{clique}} \leq \frac{n-k}{2}$. Partition V into subsets $K = \{j \in V : \exists\, i \in \{1,\dots,k\}$ with $j \in \kappa(C_i)\}$ and $V \setminus K$. The objective function of (4.14) can be rewritten as $\sum_{j\in K} x_j + \sum_{j \in V\setminus K} x_j$. Summing up constraints $x_i + x_j \leq 1$ for all edges of K_n with both ends in $V \setminus K$ we obtain $\sum_{j\in V\setminus K} x_j \leq \frac{|V\setminus K|}{2}$. Similarly, the odd cycle inequalities imply

$$
\sum_{j\in K} x_j = \sum_{i=1}^{k} \sum_{j\in\kappa(C_i)} x_j \leq \sum_{i=1}^{k} \frac{|\kappa(C_i)|-1}{2} = \frac{|K|-k}{2}.
$$

Therefore,

$$
\sum_{j=1}^{n} x_j \leq \frac{|V\setminus K| + |K| - k}{2} = \frac{n-k}{2}.
$$

It remains to prove that a feasible solution of (4.14) with value $\frac{n-k}{2}$ exists. Such a solution can be easily constructed by arbitrarily choosing one node for each odd cycle $\kappa(C_i)$ $i = 1,\dots,k$ and assigning 0 to the corresponding x variables, while setting all remaining x variables to $\frac{1}{2}$.

The second statement follows directly from the fact that, when G is itself a clique, problems (4.13) and (4.14) coincide and $B' = B$. □

Theorem 4.9.2. *Given an optimal basis B associated to $x_i^* = \frac{1}{2}\ \forall\, i = 1,\dots,n$, the difference between $z_{corner(B)}$ and $z_{LP\cap corner(B)}$ can be at most $n/8$, and there are graphs for which this bound is tight.*

Proof. By Theorem 4.6.1, $z_{\text{corner}(B)} = \frac{n-k_o}{2}$. By Theorem 4.9.1, $z_{\text{LP}\cap\text{corner}(B)} \geq \frac{n-k}{2}$. It follows that the greatest gap between $z_{\text{corner}(B)}$ and $z_{\text{LP}\cap\text{corner}(B)}$ can occur when $(k - k_o)$ is maximized. Because the maximum number of even 1-tree components in a basis is at most $\left\lfloor \frac{n}{4} \right\rfloor$ the theorem follows. For a clique K_n with n multiple of 4, we can find a basis with exactly $\frac{n}{4}$ even 1-tree components. In this case, by Theorem 4.9.1, $z_{\text{corner}(B)} - z_{\text{LP}\cap\text{corner}(B)} = \frac{n}{8}$. □

By Theorems 4.5.1 and 4.9.2, Theorem 4.3.4 is proven.

Chapter 5

A concave heuristic for the stable set problem

Heuristic algorithms are very useful not only to be applied stand-alone, when determining the optimal solution of a problem is prohibitive from a computational point of view. Indeed, also in the framework of exact methods, heuristics can enhance efficiency by providing good incumbent solutions. For example, heuristic methods can be combined with branch-and-bound and branch-and-cut methods to facilitate pruning of subtrees or intensifying reduced-cost fixing.

The literature shows a very rich variety of heuristics for 0-1 MILPs, including the *pivot-and-complement* method of Balas and Martin [6], the tabu-search-based method of Løkkentagen and Glover [54], *OCTANE* of Balas et al. [5], the *local branching* by Fischetti and Lodi [34], the *relaxation induced neighborhood search* (RINS) by Danna et al. [27], the *pivot, cut and dive* heuristic by Eckstein and Nediak [30], and the *feasibility pump* heuristic by Fischetti and Lodi [33]. For a review of the state of the art heuristics for mixed integer and binary programs we recommend [2] and [19]. We will describe in more detail the *pivot, cut and dive* [30], and the *feasibility pump* heuristics [33] in Section 5.2.

In this chapter we present a concave reformulation for set covering problems, where integrality constraints are dropped and the original linear objective function is replaced by a concave one, penalizing fractional values. When a local integer optimum of the concave problem is generated, a heuristic solution of the original problem has been found. This task can be accomplished quite efficiently by means of the Frank-Wolfe algorithm. The choice of a suitable parametric concave function allows us to regulate the smoothness of the objective function and to achieve sparseness of the local optimum. When applied to the edge formulation of the stable set problem, additional properties of local optima can be established. Namely, if the parameter of the objective function belongs to a certain range, binary valued variables of the local optimum can be fixed, allowing for a dimensionality reduction of the problem.

In Section 5.1 we describe some properties of concave programming problems and we introduce the Frank-Wolfe method; in Section 5.2 we present a couple of heuristics for 0-1 mixed integer programs that exploit, directly or indirectly, a concave merit function penalizing fractional solutions; in Section 5.3 we propose a concave heuristic for set covering problems and in Section 5.4 we apply it to the stable set problem. In Section 5.5 we establish a new fixing theorem for variables that are integer valued in local optima and finally, in Section 5.6, we present some computational results.

5.1 Concave programming and the Frank-Wolfe algorithm

In this section we introduce concave programming problems and the Frank-Wolfe algorithm. We refer to [46] and [12] for further reading. A *concave programming* problem consists in the minimization of a concave function over a convex set (or, equivalently, in the maximization of a convex function over a convex set):

$$\begin{aligned} \min \quad & f(x) \\ \text{s.t.} \quad & x \in D, \end{aligned} \tag{5.1}$$

where $D \subset \mathbb{R}^n$ is non-empty, closed and convex and where $f\colon A \to \mathbb{R}$ is concave on a suitable set $A \subset \mathbb{R}^n$ containing D. Standard optimization techniques can fail in determining a global optimum of (5.1), because of the existence of local minima that are not global. In the literature, such optimization problems are referred as *multiextremal global optimization problems.* Concave programming problems have, however, some special properties that make them easier to handle than general multiextremal global optimization problems. In particular, if f is concave, its global minimum over D is attained at an extreme point of D (see Theorem I.1 in [46]). It has been shown that concave minimization problems are NP-hard, even in some special cases, as, for example, the minimization of a concave quadratic function over a hypercube. In [46] an overview of practical applications of concave minimization problems is presented. Furthermore, the relationships between concave programming and integer programming are investigated; in fact, it is shown that there exists a $\mu_0 \in \mathbb{R}$ such that for all $\mu \geq \mu_0$ problems

$$\begin{array}{ll} \min & c^T x \\ \text{s.t.} & Ax \leq b \\ & x \in \{0,1\}^n \end{array} \qquad\qquad \begin{array}{ll} \min & c^T x + \mu x^T(\mathbf{1} - x) \\ \text{s.t.} & Ax \leq b \\ & x \in [0,1]^n \end{array}$$

are equivalent.

One of the most well-known iterative methods of nonlinear programming to solve constrained problems of type

$$\begin{array}{ll} \min & f(x) \\ \text{s.t.} & x \in X, \end{array}$$

where f is continuously differentiable and $X \subseteq \mathbb{R}^n$ is non-empty, compact and convex, is the *Frank-Wolfe method*, also called the *conditional gradient method.* The algorithm, introduced in 1956 [37], was originally proposed for problems with a nonlinear quadratic objective function.

Starting from a feasible vector x^0, the method generates a sequence of feasible points $\{x^k\}$ according to $x^{k+1} = x^k + \alpha^k d^k$, where $d^k = (\bar{x}^k - x^k)$ is a feasible direction at x^k, which is also descent, i. e. $\nabla f(x^k)^T(\bar{x}^k - x^k) < 0$, and the stepsize α^k is positive and such that $x^k + \alpha^k d^k \in X$. At each iteration, $\bar{x}^k$ is obtained as a solution of the linear program

$$\begin{array}{ll} \min & \nabla f(x^k)^T(x - x^k) \\ \text{s.t.} & x \in X. \end{array} \tag{5.2}$$

X is assumed to be compact, which guarantees that (5.2) has a solution. Intuitively, $\bar{x}^k$ is the furthest point of X along the negative gradient direction. The algorithm generates points of the sequence by searching along descent directions. In this sense, it can be viewed as a constrained version of the unconstrained descent algorithms, like the gradient method. If the stepsize is chosen according to suitable rules like, for example, the *Armijo rule* or the *limited minimization rule*, every limit point of the sequence $\{x^k\}$ generated by the Frank-Wolfe method is a stationary point x^* of (5.2), i.e. a point satisfying $\nabla f(x^*)^T(x - x^*) \geq 0 \; \forall \; x \in X$ [12].

In the special case where X is a non-empty polyhedral set and f is a continuously differentiable concave function bounded below on X, the Frank-Wolfe algorithm with unitary stepsize is guaranteed to converge in a finite number of iterations to a vertex that is a stationary point of the problem [56]. Moreover, it is possible to choose a random starting point $x^0 \in \mathbb{R}^n$, which may even not belong to X. The same convergence properties of the algorithm can be extended to the case where X is a polyhedral set in R^n that does not contain a line and f is a concave non-differentiable function bounded below on X [57].

5.2 Concave heuristics for 0-1 mixed integer programming

In [30] Eckstein and Nediak presented a heuristic for 0-1 mixed integer programming based on the use of a concave merit function to measure integrality of solutions. The concave merit function is zero at integer-feasible points and positive elsewhere in the unit cube. The key layer of the method presented by Eckstein and Nediak consists in using either individual pivots or Frank-Wolfe blocks of pivots to reduce the value of the merit function, while trying to avoid excessive deterioration of the original objective function. Pivots are selected on the base of local gradient information, leading to local optimal solutions of the merit function over the feasible set. When a local optimal solution is reached, that is not integer-feasible, a phase of *probing* is performed, which consists in explicitly testing a list of possible pivots. If an adjacent vertex is not found, which improves the merit function without excessively deteriorating the original objective function, a *convexity cut* (or *intersection cut*) violated by the current vertex is computed. Finally, if probing fails, and the resulting convexity cut is too shallow, a recursive depth-first diving technique is applied, in order to fix a possibly large group of variables simultaneously. When this operation yields an infeasible problem, it is possible to backtrack and apply a complementary vertex cut.

Consider a Mixed-Integer Linear Program (MILP) of the form:

$$\begin{aligned} \min \quad & c^T x \\ \text{s.t.} \quad & Ax = b \\ & l \leq x \leq u \\ & x_i \in \{0,1\} \quad \forall\, i \in I, \end{aligned}$$

where A is a $m \times n$ matrix, $I \subseteq \{1,\ldots,n\}$ is the set of integer variables and $l_i, u_i \in \{0,1\}$ $\forall\ i \in I$. Dropping integrality constraints on variables in I yields the linear programming (LP) relaxation of the MILP. Integer infeasibility can be measured by means of a concave merit function defined as follows. Let $\phi_i \colon \mathbb{R} \to \mathbb{R},\ i \in I$ be a concave, continuously differentiable function such that $\phi(0) = \phi(1) = 0$ and $\phi(x) > 0\ \forall\ x \in (0,1)$. The class of functions used by Eckstein and Nediak in their experiments depends on a parameter $\alpha \in (0,1)$ and is defined as:

$$\phi_\alpha(x) = 1 - \begin{cases} \left(\dfrac{x-\alpha}{\alpha}\right)^2, & x \leq \alpha \\ \left(\dfrac{x-\alpha}{1-\alpha}\right)^2, & x \geq \alpha \end{cases}$$

Note that the function attains its maximum value of 1 at $x = \alpha$.

Define function ψ as:

$$\psi(x) = \sum_{i \in I} \phi_i(x_i).$$

Eckstein and Nediak propose two rounding procedures, the *ratio* and the *sum* methods, to reach integer feasibility and preserve a good value of the objective function. In the ratio method, simplex pivots are performed, where reduced costs are computed on the base of local gradient information. First, the algorithm tries to select a pivot that decreases the merit function and that does not increase the objective. If no such pivot is possible, the pivot corresponding to least possible increase of the objective function is chosen. In the sum method, integrality and objective improvement are simultaneously taken into account by defining the merit function

$$\hat{\psi}(x) = \psi(x) + wc^T x,$$

where $w > 0$ weights the original objective function. The merit function $\hat{\psi}$ is minimized over the polyhedral set defined by the feasible region of the LP relaxation.

In this case, either individual simplex pivots can be performed, or the Frank-Wolfe algorithm can be applied to perform block pivots, which saves time in highly degenerate problems. Eckstein and Nediak observed that the feasibility pump heuristic of Fischetti, Glover and Lodi [33] is strongly related to the Frank-Wolfe algorithm.

The feasibility pump heuristic is a scheme for finding a feasible solution to general MILPs, that performs quite successfully in finding heuristics even for hard instances, and is currently implemented in many optimization solvers, both commercial and open-source. The method rounds a sequence of fractional solutions of the LP relaxation, until an integer feasible solution is possibly found. More precisely, the heuristic is applied to a MILP of the form $min\{c^T x \colon Ax \geq b,\ x_j \text{ integer } \forall\, j \in I\}$, where A is a $m \times n$ matrix and $I \subseteq \{1, \dots, n\}$ is the set of integer constrained variables, and to the polyhedron $P = \{x \colon Ax \geq b\}$ associated to the LP relaxation of the MILP. At each iteration (called *pumping cycle*) a LP-feasible solution, that is, a solution $x^* \in P$, is rounded to an integer solution (with respect to variables in I) $\tilde{x}$, where $\tilde{x}_j = [x^*_j]$ if $j \in I$ and $\tilde{x}_j = x^*_j$ otherwise, and $[\,\cdot\,]$ represents the rounding to the nearest integer value. A new fractional point x^* is then obtained from $\tilde{x}$ as a point minimizing $\Delta(x, \tilde{x}) = \sum_{j \in I} |x_j - \tilde{x}_j|$ over P. The procedure terminates if $x^* = \tilde{x}$, meaning that x^* is integer-feasible, or if a time or iteration limit is reached. Intuitively, starting from a LP-feasible point, two (hopefully) convergent trajectories of points x^* and $\tilde{x}$ are generated, such that points x^* are LP-feasible but may not be integer feasible, while points $\tilde{x}$ are integer, but may be not LP-feasible. Further improvements of the basic version of the feasibility pump heuristic have been found in [36] and [3].

Eckstein and Nediak remarked that the feasibility pump heuristic corresponds in the binary case to applying the Frank-Wolfe method to the minimization of the non-smooth merit function $\hat{\psi}$, with $w = 0$ and $\psi = \sum_{i=1}^{n} \phi(x_i)$, where $\phi(x) = min\{x, 1 - x\}$. Note that $\phi(x)$ is a concave function that is non-differentiable in $x = \frac{1}{2}$. For a concave function $f \colon \mathbb{R}^n \to \mathbb{R}$ the supergradient $\partial f(x)$ of f at x is a vector in $\mathbb{R}^n$ satisfying

$$f(y) - f(x) \leq \partial f(x)^T (x - y)$$

It is then possible to define the supergradient $\partial\psi(x)$ of ψ, in the following way: the i-th component of $\partial\psi(x)$ is set to 1 if $x_i < \frac{1}{2}$, or to -1 if $x_i \geq \frac{1}{2}$. Starting from a point x^0, at each iteration k of the Frank-Wolfe procedure, a point $x^{k+1} \in P$ is determined, such that

$$\begin{aligned} x^{k+1} &= \operatorname*{argmin}_{x \in P} \sum_{i:\, x_i^k < \frac{1}{2}} (x_i - x_i^k) - \sum_{i:\, x_i^k \geq \frac{1}{2}} (x_i - x_i^k) \\ &= \operatorname*{argmin}_{x \in P} \sum_{i:\, x_i^k < \frac{1}{2}} x_i - \sum_{i:\, x_i^k \geq \frac{1}{2}} x_i, \end{aligned}$$

where, in the linear objective function, the costs correspond exactly to the components of the supergradient of ψ evaluated in x^k. This corresponds to a pumping cycle of the feasibility pump heuristic, where a new fractional point $x^{k+1} \in P$ is determined according to the rounding $[x^k]$ of x^k, as described above. Precisely:

$$\begin{aligned} x^{k+1} &= \operatorname*{argmin}_{x \in P} \sum_{i:\, [x_i^k]=0} x_i + \sum_{i:\, [x_i^k]=1} (1 - x_i) \\ &= \operatorname*{argmin}_{x \in P} \sum_{i:\, x_i^k < \frac{1}{2}} x_i + \sum_{i:\, x_i^k \geq \frac{1}{2}} (1 - x_i) \\ &= \operatorname*{argmin}_{x \in P} \sum_{i:\, x_i^k < \frac{1}{2}} x_i - \sum_{i:\, x_i^k \geq \frac{1}{2}} x_i. \end{aligned}$$

5.3 A concave heuristic for Set Covering

Consider now the *set covering problem*

$$\begin{aligned} \min \quad & c^T x \\ \text{s.t.} \quad & Ax \geq \mathbf{1}_m \\ & x \in \{0,1\}^n, \end{aligned} \tag{SC}$$

where A is an $(m \times n)$ matrix with $a_{ij} \in \{0,1\}$ and $c \in \mathbb{R}^n_+$. The linear relaxation of (SC) is

$$\begin{aligned} \min \quad & c^T x \\ \text{s.t.} \quad & Ax \geq \mathbf{1}_m \\ & 0 \leq x_i \leq 1 \quad i = 1, \ldots, n. \end{aligned} \tag{LSC}$$

We propose the following concave reformulation for (SC), which is inspired by the concave approximation of the step function described in [56]

$$\begin{array}{lll} \min & \sum_{i=1}^{n} c_i(1 - e^{-\alpha x_i}) & \\ \text{s.t.} & Ax \geq \mathbf{1}_m & \\ & 0 \leq x_i \leq 1 & i = 1, \dots, n, \end{array} \tag{CSC}$$

where $\alpha > 0$ is a parameter. The motivation for replacing the original objective function by $\psi(x) = \sum_{i=1}^{n} c_i(1 - e^{-\alpha x_i})$, lies in the fact that f penalizes fractional values simultaneously taking into account weights c_i, $i = 1, \dots, n$ of the original objective function, see Fig. 5.1. Roughly speaking, the concave objective function ψ

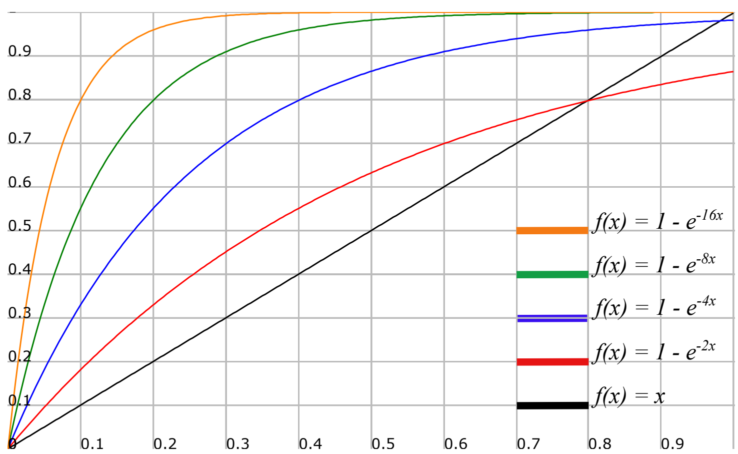

Figure 5.1: The concave function $\phi(x) = 1 - e^{-\alpha x}$, plotted for different values of parameter α.

discourages components of x from taking fractional values: as long as the parameter α is sufficiently large $(1 - e^{-\alpha x_i})$ tends to 1 for any $x_i \in (0, 1)$. As a consequence, in the objective function fractional components of x tend to be weighted as if they were 1-valued, while they contribute less in covering the constraints of the problem, possibly requiring that other components are fixed to positive values.

It is possible to determine a local minimum of (CSC) by applying the Frank-Wolfe method with unitary stepsize, starting from a random point $x^0 \in \mathbb{R}^n$. Rounding up the vector returned by the Frank-Wolfe algorithm immediately provides an heuristic for the original problem.

5.4 Application to the stable set problem

We are interested in applying the concave heuristic to the stable set problem (STAB). Given a graph $G(V, E)$, a *node cover* of G is a set $C \subseteq V$ such that each edge is incident with at least one node of C. Suppose we are given a weight c_i for each node $i \in V$. The *minimum weight node covering problem* consists in finding the node cover with minimum total weight and can be formulated as

$$\begin{array}{rll} \min & c^T y & \\ \text{s.t.} & y_i + y_j \geq 1 & (i,j) \in E \\ & y_i \in \{0,1\} & i \in V. \end{array} \tag{NC}$$

The linear relaxation of (NC) is

$$\begin{array}{rll} \min & c^T y & \\ \text{s.t.} & y_i + y_j \geq 1 & (i,j) \in E \\ & 0 \leq y_i \leq 1 & i \in V. \end{array} \tag{LNC}$$

Observe that y is a feasible solution of (LNC) (resp. of (NC)) if and only if $x = \mathbf{1} - y$ is a feasible solution of (FSTAB) (resp. of (STAB)). Therefore, vertices of (LNC) are also half-integer, i.e. $(0, \frac{1}{2}, 1)$-valued, and those components that are $(0,1)$-valued in an optimal solution of (LNC) can be fixed (recall Theorem 1.7.3).

As a consequence, finding a concave heuristic of (NC) by determining a local minimum $\bar{y}$ of

$$\begin{array}{rll} \min & \sum_{i=1}^n c_i(1 - e^{-\alpha y_i}) & \\ \text{s.t.} & y_i + y_j \geq 1 & (i,j) \in E \\ & 0 \leq y_i \leq 1 & i \in V. \end{array} \tag{CNC}$$

immediately provides us with an heuristic solution $\bar{x}$ of (STAB), namely $\bar{x} = 1 - \lceil \bar{y} \rceil$. This task can be equivalently accomplished by finding a local maximum of the concave problem

$$\begin{array}{rll} \max & \sum_{i=1}^n c_i e^{-\alpha(1-x_i)} & \\ \text{s.t.} & x_i + x_j \leq 1 & (i,j) \in E \\ & 0 \leq x_i \leq 1 & i \in V, \end{array} \tag{CSTAB}$$

which, rounded down, generates a feasible solution of $STAB(G)$.

Recall that the constraints $x_i \leq 1 \ \forall \ i \in V$ do not appear in the definition of $FSTAB(G)$, as G by assumption does not contain isolated nodes, implying that these constraints are redundant. Remark also that we can drop constraints $y_i \leq 1 \ \forall \ i \in V$ in (LNC), in order to obtain an equivalent minimization problem, i.e. a problem with the same optimal solution:

$$\begin{aligned} \min \quad & c^T y & & \\ \text{s.t.} \quad & y_i + y_j \geq 1 & & (i,j) \in E \\ & y_i \geq 0 & & i \in V. \end{aligned} \tag{LNC'}$$

In the literature this latter formulation of the node covering problem is very common, due to the fact that, if $c_i = 1 \ \forall \ i \in V$, the dual linear program of (LNC') is the maximum matching problem [55], [68]. The feasible region of (LNC) is strictly contained in that of (LNC'). In other words, there exists some feasible solution $\tilde{y}$ of (LNC') with some of its components strictly greater to 1, implying that $\tilde{x} = \mathbf{1} - \tilde{y}$ is not feasible for (FSTAB). Nevertheless, none of such feasible solutions is a vertex of the polyhedron defined by (LNC'), as we clarify in the next proposition.

Proposition 5.4.1. *$\tilde{y}$ is a vertex of $P = \{y \in \mathbb{R}_+^{|V|} : y_i + y_j \geq 1 (i,j) \in E\}$ if and only if $\tilde{x} = \mathbf{1} - \tilde{y}$ is a vertex of $FSTAB(G)$.*

Proof. First, we show that if $\tilde{y}$ is a vertex of P, then $\tilde{y} \leq \mathbf{1}$. By contradiction, suppose that $\tilde{y}$ is a vertex of P and that $\tilde{y}_u = 1 + \epsilon$, $\epsilon > 0$, for some $u \in V$. Define y^- and y^+ as $y_v^- = y_v^+ = \tilde{y}_v \ \forall \ v \neq u$, $y_u^- = \tilde{y}_u - \epsilon$, and $y_u^+ = \tilde{y}_u + \epsilon$. Clearly, $y^-, y^+ \in P$ and $\tilde{y}$ can be obtained as a convex combination of y^- and y^+, contradicting the fact that $\tilde{y}$ is a vertex of P. Therefore $vert(P) = vert(P \cap [0,1]^{|V|})$. Because $P \cap [0,1]^{|V|}$ is an affine transformation of $FSTAB(G)$, the thesis directly follows. □

As a consequence, we can equivalently compute our heuristic by applying the Frank-Wolfe algorithm to the following concave formulation of (NC)

$$\begin{aligned} \min \quad & \textstyle\sum_{i=1}^n c_i (1 - e^{-\alpha y_i}) & & \\ \text{s.t.} \quad & y_i + y_j \geq 1 & & (i,j) \in E \\ & y_i \geq 0 & & i \in V. \end{aligned} \tag{CNC'}$$

5.5 Fixing integer variables of local optima

In this section we prove that, under suitable assumptions, it is possible to fix the components of x that are integer valued in a critical point x^* of (CSTAB). In the following we assume G to be connected and $c_i \in \mathbb{Z}$, $c_i > 0 \ \forall \ i \in V$. We denote by $f\colon \mathbb{R}^{|V|} \to \mathbb{R}$ the convex, continuously differentiable function $f(x) = \sum_{i=1}^{n} c_i e^{-\alpha(1-x_i)}$ and, for any $P \subseteq V$, we define $c(P) = \sum_{i\in P} c_i$. Suppose x^* is a $(0, \frac{1}{2}, 1)$-valued vertex of $FSTAB(G)$ and define $S = \{j \in V\colon x_j^* = 1\}$ and $N(S) = \{j \in V\colon x_j^* = 0\}$.

Given a stable set S of G, define

$$\beta(S) = \min \{c(S) + 1, \lfloor \omega^*(G[N(S)]) \rfloor\}, \tag{5.3}$$

where $\omega^*(G[N(S)])$ is the optimal value of $(FSTAB)$ on the subgraph of G induced by nodes in $N(S)$, i.e. $\omega^*(G[N(S)]) = \max \left\{\sum_{i\in N(S)} c_i x_i \colon x \in FSTAB(G[N(S)])\right\}$.

Theorem 5.5.1. *Suppose x^* is a vertex of (CSTAB) such that $|S| > 0$ and*

$$\nabla f(x^*)^T (x - x^*) \leq 0 \qquad \forall \ x \in FSTAB(G). \tag{5.4}$$

If $0 < \alpha < -\ln \dfrac{\beta(S) - 1}{\beta(S)}$, then there exists a maximum stable set of G containing S.

Proof. First, observe that the condition on α is well defined, as the argument of the logarithm is always nonnegative. This follows from the fact that $S \neq \emptyset$, implying $N(S) \neq \emptyset$, as the graph is connected.

By Theorem 1.7.2, we only need to show that S is a maximum weight stable set in $G[S \cup N(S)]$, the subgraph of G induced by $S \cup N(S)$. Let us suppose by contradiction that there exists a stable set S' of $G[S \cup N(S)]$ such that $c(S') \geq c(S) + 1$. To prove the claim, we contradict condition (5.4). Recall that S' can be expressed in terms of S and an augmenting subset $I \subseteq N(S)$, such that I is a stable set of G and

$$S' = S \cup I \setminus S(I),$$

where $S(I) = S \cap N(I)$. Consequently, $c(S) + 1 \leq c(S') = c(S) + c(I) - c(S(I))$ implies

$$c(I) \geq c(S(I)) + 1.$$

Define now vector $x' \in \mathbb{R}^n$ as follows:

$$x'_j := \begin{cases} \frac{1}{2} & j \in I \cup S(I) \\ x^*_j & \text{otherwise,} \end{cases}$$

$\forall\, j \in V$. Note that nodes u with $x^*_u = 0$ and $x'_u = \frac{1}{2}$ are such that $u \in I$. Because $x'_v = \frac{1}{2} \;\forall\, v \in S(I)$, it follows that x' is a feasible solution of $FSTAB(G)$. Furthermore, x' contradicts condition (5.4), as

$$\begin{aligned} \nabla f(x^*)^T(x' - x^*) &= \alpha \left[\sum_{i \in I} c_i \left(e^{-\alpha} \left(\frac{1}{2} - 0 \right) \right) + \sum_{i \in S(I)} c_i \left(\frac{1}{2} - 1 \right) \right] \\ &= \frac{\alpha}{2} \left[e^{-\alpha} c(I) - c(S(I)) \right] > 0, \end{aligned} \tag{5.5}$$

where the last inequality is implied by condition (5.5.1). In fact

$$e^{-\alpha} c(I) - c(S(I)) > 0 \;\text{ if and only if }\; \alpha < -\ln \frac{c(S(I))}{c(I)} \tag{5.6}$$

and the latter inequality is satisfied because $\alpha < -\ln \dfrac{\beta(S) - 1}{\beta(S)}$ and either $\beta(S) = c(S) + 1$, implying

$$\frac{c(S(I))}{c(I)} \le \frac{c(S(I))}{c(S(I)) + 1} \le \frac{c(S)}{c(S) + 1},$$

or $\beta(S) = \lfloor \omega^*(G[N(S)]) \rfloor$, and therefore

$$\frac{c(S(I))}{c(I)} \le \frac{c(I) - 1}{c(I)} \le \frac{\lfloor \omega^*(G[N(S)]) \rfloor - 1}{\lfloor \omega^*(G[N(S)]) \rfloor}.$$

Hence, condition (5.6) is satisfied, contradicting that x^* is critical point. Therefore, S is a maximum weight stable set in $G[S \cup N(S)]$ and, by Theorem 1.7.2, we can fix to 1 the components of x in S, and to 0 those that are in $N(S)$. □

Remark 5.5.1. *To prove Theorem 5.5.1 we exploited Theorem 1.7.2. Note that the condition of such theorem is only sufficient, implying that there may exist a maximum weight stable set of G containing S, although S is not a maximum weight stable set in $G[S \cup N(S)]$.*

Remark 5.5.2. *Given vectors x^* and $\bar{x}_i = \frac{1}{2} \;\forall\, i \in V$, and defined $S = \{j \in V : x^*_j = 1\}$, suppose that S is a maximum weight stable set in $G[S \cup N(S)]$. Then $(\bar{x} - x^*)$ is*

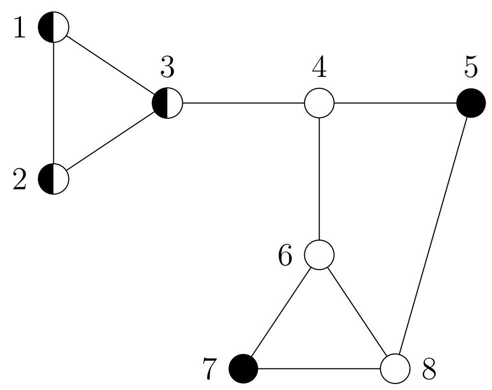

Figure 5.2: An illustration of Example 5.5.1

.

an ascent feasible direction if

$$\nabla f(x^*)^T(\bar{x} - x^*) = \frac{\alpha}{2}\left[e^{-\alpha}c(N(S)) - c(S)\right] > 0,$$

which implies $\alpha < -\ln\frac{c(S)}{c(N(S))}$. *Roughly speaking, if* α *is sufficiently large, i.e.* $\alpha \geq -\ln\frac{c(S)}{c(N(S))}$, x^* *does not fall into the basin of attraction of* $\bar{x}$. *Note that, by convexity of* f, $\nabla f(x^*)^T(\bar{x}-x^*) > 0$ *implies* $f(\bar{x}) - f(x^*) > 0$. *The analogous condition for the linear relaxation (FSTAB) of (STAB) is* $\frac{c(S \cup N(S))}{2} - c(S) > 0$, *implying* $c(N(S)) > c(S)$, *that is satisfied if and only if* $\bar{x}$ *is the unique optimal solution of (FSTAB) (see Theorem 1.7.4).*

Example 5.5.1. *Consider* $G(V, E)$ *with* $V = \{1, 2, 3, 4, 5, 6, 7, 8\}$ *and* $E = \{(1, 2), (1, 3), (2, 3), (3, 4), (4, 5), (4, 6), (5, 8), (6, 7), (6, 8), (7, 8)\}$. *Suppose that* $c_i = 1 \; \forall \; i \in V$. *For* $\alpha \geq -\ln\frac{2}{3}$, *vector* $x_5^* = x_7^* = 1$, $x_4^* = x_6^* = x_8^* = 0$, $x_1^* = x_2^* = x_3^* = \frac{1}{2}$ *is a local maximum of (CSTAB), as*

$$\nabla f(x^*)^T(x - x^*) = \alpha\left[x_5 + x_7 + e^{-\frac{\alpha}{2}}(x_1 + x_2 + x_3) + e^{-\alpha}(x_4 + x_6 + x_8)\right] \leq 0,$$

$\forall \; x \in FSTAB(G)$. *In this case* $\omega^*(G[4, 6, 8]) = 2$, *therefore* $\beta = 2$, *and we can fix variables that are (0,1)-valued in* x^*, *for any* $-\ln\frac{2}{3} \leq \alpha < -\ln\frac{1}{2}$. *Remark that* $\bar{x}_i = \frac{1}{2} \; \forall \; i \in V$ *is the unique optimum of (FSTAB).*

5.6 Computational results

We have performed computational experiments on the DIMACS benchmark instances[1] to test the heuristic described in Sections 5.3 and 5.4. In these experiments, we perform a progressive fixing of blocks of variables, until we obtain an integer feasible solution. To fix each block, we start from an initial value of parameter α and we run the Frank-Wolfe heuristic 20 times. We then fix the integer variables of the most integral local maximum found so far. If no variable can be fixed, we increase α by a factor and repeat. If we have performed three consecutive blocks of fixing without increasing α, we decrease its value by a factor. This corresponds to an automatic tuning of α, which is necessary because the problems of the test set are very different in their nature and structure. We repeat this procedure 100 times, and we finally choose the best heuristic. For each instance, the initial value of α is chosen as the lowest such that some variables can be fixed in a run of the Frank-Wolfe algorithm.

The results are shown in Table 5.1, where we report the objective value of the best heuristic solution found by the algorithm, the overall number of blocks of fixing, and the overall time (in seconds) needed to run the algorithm. The experiments have been performed on a Intel Core i7 at 3.47 GHz. The algorithm shows some weaknesses, especially in terms of efficiency. In fact, not only the results are often not comparable with state-of-the-art metaheuristics [62, 64, 44], but also the time required to compute the best heuristic solution is significantly high, due to the multistart implementation and because we are not exploiting the combinatorial structure of the problem.

Though, on some difficult instances, as the MANN instances, the algorithm performs quite well, as it can achieve good feasible solutions in a reasonable amount of time. This may be due to the fact that these max clique instances are more dense than the others, and therefore probably less degenerate. We have also tested a different scheme of the algorithm, where the heuristic fixing is not performed till all the components are integer, but it is terminated when the dimension of the problem has been reduced to a tractable size, in order to solve this smaller instance as a MILP. In this setting, we can reach the optimal solutions over all the MANN instances. Note that on some of the hamming instances the maximum stable set is in fact computed in the preprocessing step, consisting in solving (FSTAB) and fixing the integer valued variables of its

[1] Second DIMACS Implementation Challenge, `http://dimacs.rutgers.edu/Challenges/`

	Obj	nb	time[s]		Obj	nb	time[s]
brock200_1	19	491	735	p_hat1000-1	9	324	124023
brock200_2	9	225	302	p_hat1000-2	31	418	36621
brock200_3	13	276	503	p_hat1000-3	53	588	125331
brock200_4	15	239	176	p_hat1500-1	9	353	386713
brock400_1	21	288	1228	p_hat1500-2	43	527	461292
brock400_2	20	261	751	p_hat1500-3	68	673	549517
brock400_3	20	342	1755	p_hat300-1	8	226	3741
brock400_4	20	301	1285	p_hat300-1	8	226	3741
brock800_1	17	354	33088	p_hat300-2	20	326	4244
brock800_2	16	365	33916	p_hat300-3	31	415	996
brock800_3	17	358	21075	p_hat500-1	8	254	12815
brock800_4	16	376	28044	p_hat500-2	30	316	9108
c-fat200-1	12	204	1243	p_hat500-3	40	415	5248
c-fat200-2	24	178	587	p_hat700-1	8	315	38488
c-fat200-5	58	177	556	p_hat700-2	35	349	19183
c-fat500-10	124	290	44051	p_hat700-3	49	434	18460
c-fat500-1	14	3652	16	san1000	8	249	72717
c-fat500-2	26	2786	463462	san200_0.7_1	16	281	478
c-fat500-5	64	471	107199	san200_0.7_2	12	358	175
hamming10-2	512	0	1	san200_0.9_1	70	307	42
hamming10-4	32	722	72208	san200_0.9_2	48	691	213
hamming6-2	32	0	0	san200_0.9_3	34	735	152
hamming6-4	4	141	13	san400_0.5_1	8	232	3228
hamming8-2	128	0	0	san400_0.7_1	21	258	2687
hamming8-4	15	302	415	san400_0.7_2	16	271	2350
johnson16-2-4	8	374	34	san400_0.7_3	15	311	2903
johnson32-2-4	16	707	1150	san400_0.9_1	53	427	2287
johnson8-2-4	4	176	1	sanr200_0.7	15	243	221
johnson8-4-4	14	337	12	sanr200_0.9	38	672	129
keller4	11	309	100	sanr400_0.5	10	266	4704
keller5	21	527	21443	sanr400_0.7	18	369	5493
keller6	-	-	-				
MANN_a27	126	8699	185				
MANN_a45	343	24234	1547				
MANN_a81	1098	71087	16204				
MANN_a9	16	565	1				

Table 5.1: Computational results on the DIMACS instances.

optimal solution.

Chapter 6

Conclusions and perspectives

This dissertation has focused on the edge formulation of the stable set problem, investigating some polyhedral aspects concerning the polytope arising from the linear relaxation of such formulation. This polytope, which we have called *fractional stable set polytope*, is a very weak approximation of the stable set polytope, i.e. the convex hull of the incidence vectors of stable sets of the graph. However, it seems to have a special geometrical structure, that allowed us to characterize its bases and vertices.

In particular, we have established a necessary and sufficient condition for two (possibly fractional) vertices to be adjacent, extending a condition for vertex adjacency on the stable set polytope, due to Chvátal. Our graphic characterization of simplex pivots was also crucial to prove a bound on the diameter of the fractional stable set polytope, equal to the number of nodes of the input graph. A direct implication is that the Hirsch conjecture holds for our fractional polytope. Another byproduct of the graphic characterization of bases is that we could easily design a simplex-like algorithm that generates a sequence of integer vertices of the polytope without using cutting plane techniques.

In addition, the structural properties outlined above let us gain insight on issues related to the strength of the corner and other related relaxations in the context of mixed integer linear programming. With respect to the MILP arising from the edge formulation of the stable set problem, we have proven that the corner relaxation can yield a very weak bound if the input graph admits a perfect or a nearly perfect matching; on the other hand, some related relaxations can be significantly stronger than the corner relaxation.

Finally, we have presented a concave reformulation of the stable set problem, where fractionality is penalized by means of a concave merit function. Exploiting the structure of our polytope, we could derive a condition for fixing variables that are integer valued in a local optimum of the concave problem, which extends a well-known condition due to Nemhauser and Trotter.

In the following sections we point out some observations and directions for future research about the main topics of the thesis.

6.1 Vertex adjacency and the Hirsch conjecture

Chvátal's condition about vertex adjacency on the stable set polytope directly implies that the Hirsch conjecture holds for this 0-1 polytope. In fact, the incidence vector of any stable set is a vertex of the stable set polytope. Therefore, if the symmetric difference between two stable sets induces a subgraph with several connected components, sequentially inverting the assignment of zeros and ones on each of them would generate a succession of integer vertices of the polytope.

Concerning the fractional stable set polytope, our definition of generalized symmetric difference does not maintain this property, because a feasible $\{0, \frac{1}{2}, 1\}$-valued feasible solution is not necessarily a vertex. This is why, in the proof of the Hirsch conjecture, we needed to make sure that any intermediate feasible solution generated on the path between two arbitrary vertices is still a vertex. A natural question arising at this point is whether there exists an alternative definition of generalized symmetric difference extending the properties of the symmetric difference in the 0-1 case. This would probably facilitate the proof of the Hirsch conjecture, and could even help in characterizing more precisely the distance between two arbitrary vertices, and hence the exact diameter of the polytope.

Currently, we are also exploring alternative and more straightforward ways of proving the bound of $|V|$ for the diameter of the fractional stable set polytope, and it seems very likely that there are different paths in order to prove this result. We also suspect that the length of diameter of the polytope depends somehow on the size of the maximum stable set in the graph: the furthest vertices is the polytope could in fact be the empty solution and the maximum stable set. A future perspective consists therefore in answering these questions, and in possibly validating our current intuitions.

6.2 A simplex-like algorithm for the stable set problem

As discussed in Section 3.6, our simplex-like algorithm is affected by the problem of cycling, that can occur due the high degree of degeneracy of the problem. For few instances of the DIMACS library we could almost instantaneously determine the optimal solution by starting from the empty stable set and augmenting it at each iteration through a non-degenerate pivot, in a primal fashion. The efficiency of our primal algorithm relies on the fact that each simplex pivot is performed as a simple graph transformation. This indicates that it would be significantly more interesting to extend the current heuristic algorithm to an exact method of solution. To this purpose, it will be crucial to devise a smart anti-cycling rule, as it is sometimes needed to perform degenerate pivots to reach some improving neighboring vertices that are not adjacent to the current basis.

6.3 The strength of the corner relaxation

Our results confirm the empirical study of Fischetti and Monaci [35]. They lead to the following observations. The corner relaxation can be a very weak approximation of the integer hull. Using cuts from multiple bases of the fractional stable set polytope can greatly improve over using a single basis; for this line of research, see e.g. [28]. Degeneracy plays a major role. The stable set problem is highly degenerate, and the difference in the bounds given by corner relaxations from two different optimal bases can be arbitrarily large. Furthermore, the strict corner relaxation can be much stronger than corner relaxations. Although generating cutting planes from the strict corner relaxation is difficult, this is another indication that, in the presence of LP degeneracy, exploiting multiple degenerate bases for cut generation could give significantly better bounds than working with just a single basis. Finally, the strength of the corner relaxation is not always a good indicator of the strength of the cutting planes that can be obtained from it, when these cuts are added to the LP relaxation. A future direction of research consists in extending these results to the weighted case.

6.4 A concave reformulation of the stable set problem

The current implementation of the concave heuristic described in Chapter 5 does not seem very promising, especially for efficiency issues. Current state-of-the-art metaheuristics are extremely fast in computing good solutions. Indeed, the week point of our algorithm is that we don't exploit the combinatorial structure of the problem and that we solve it as a general linear problem. Probably an analogous concave reformulation, applied to stronger formulations of the stable set problem would yield better results. It would also be interesting to test the condition for fixing stated in Theorem 5.5.1. This condition generalizes the well-known result of Nemhauser and Trotter, but it may be still not likely to generate local optima satisfying the requirements of Theorem 5.5.1.

Bibliography

[1] Ben-Israel A. and Charnes A. On some problems od diophantine programming. *Cahiers du Centre d'Etude de Recherche Operationelle*, 4:215–280, 1962.

[2] Tobias Achterberg. *Constrained Integer Programming*. PhD thesis, ZIB, Berlin, 2007.

[3] Tobias Achterberg and Timo Berthold. Improving the feasibility pump. *Discrete Optimizaton*, 4:2007, 2005.

[4] Ronald D. Armstrong and Zhiying Jin. Strongly polynomial simplex algorithm for bipartite vertex packing. *Discrete Applied Mathematics*, 64(2):97 – 103, 1996.

[5] Egon Balas, Sebastián Ceria, Milind Dawande, Francois Margot, and Gábor Pataki. Octane: A new heuristic for pure 0-1 programs. *Oper. Res.*, 49:207–225, March 2001.

[6] Egon Balas and C.H. Martin. Pivot-and-complement: A heuristic for 0-1 programming. *Management Science*, 26(1):86–96, 1980.

[7] Egon Balas and Manfred W. Padberg. On the set-covering problem. *Operations Research*, 20(6):1152–1161, Nov.-Dec. 1972.

[8] Egon Balas and Manfred W. Padberg. On the set-covering problem: II. an algorithm for set partitioning. *Operations Research*, 23(1):74–90, Jan.-Feb. 1975.

[9] Egon Balas and Manfred W. Padberg. Set partitioning: A survey. *SIAM Review*, 18(4), October 1976.

[10] M. L. Balinski. *Labelling to obtain a maximum matching*. Number 4 in Series in Probability and Statistics. University of North Carolina Press, 1969.

[11] M. L. Balinski. On maximum matching, minimum covering and their connections. In HW Kuhn, editor, *Proceedings of the Princeton symposium on mathematical programming*, pages 303–312, 1970.

[12] Dimitri P. Bertsekas. *Nonlinear Programming.* Athena Scientific, second edition, 1995.

[13] D. Bertsimas and J.N. Tsitsiklis. *Introduction to linear optimization.* Athena Scientific series in optimization and neural computation. Athena Scientific, 1997.

[14] Robert E. Bixby. Solving real-world linear programs: A decade and more of progress. *Operations Research*, 50:3–15, 2002.

[15] B. Bollobás. *Random Graphs.* Cambridge University Press, Cambridge, 2001.

[16] J. A. Bondy and U. S. R Murty. *Graph theory with applications.* The Macmillan Press Ltd., 1976.

[17] Manuel Campêlo and Gérard Cornuéjols. Stable sets, corner polyhedra and the Chvátal closure. *Operations Research Letters*, 37(6):375–378, 2009.

[18] Manuel Campêlo and Gérard Cornuéjols. Stable sets, corner polyhedra and the Chvátal closure, 2009. Extended Version available on http://integer.tepper.cmu.edu/.

[19] John W. Chinneck. *Feasibility and Infeasibility in Optimization: Algorithms and Computational Methods*, volume 118 of *International Series in Operations Research and Management Science.* Springer, 2007.

[20] V. Chvátal. On certain polytopes associated with graphs. *Journal of Combinatorial Theory, Series B*, 18(2):138 – 154, 1975.

[21] Michele Conforti, Gérard Cornuéjols, and Giacomo Zambelli. Corner polyhedron and intersection cuts. *Surveys in Operations Research and Management Science*, 16:105–120, 2011.

[22] Stephen A. Cook. The complexity of theorem-proving procedures. In *Proceedings of the third annual ACM symposium on Theory of computing*, STOC '71, pages 151–158, New York, NY, USA, 1971. ACM.

[23] W. Cook, R. Kannan, and A. Schrijver. Chvátal closures for mixed integer programming problems. *Mathematical Programming*, 47:155–174, 1990.

[24] Gérard Cornuéjols. Valid inequalities for mixed integer linear programs. *Mathematical Programming B*, 112:2008, 2006.

[25] Gérard Cornuéjols. Revival of the gomory cuts in the 1990's. *Annals of Operations Research*, 149:63–66, 2007. 10.1007/s10479-006-0100-1.

[26] K. Corrádi and A. Hajnal. On the maximal number of independent circuits in a graph. *Acta Mathematica Hungarica*, 14:423–439, 1963.

[27] Emilie Danna, Edward Rothberg, and Claude Le Pape. Exploring relaxation induced neighborhoods to improve mip solutions. *Mathematical Programming*, 102:71–90, 2005. 10.1007/s10107-004-0518-7.

[28] S. Dash and M. Goycoolea. A heuristic to generate rank-1 GMI cuts. *Mathematical Programming C*, 2:231–257, 2010.

[29] R. Diestel. *Graph theory*. Graduate texts in mathematics. Springer, 2006.

[30] Jonathan Eckstein and Mikhail Nediak. Pivot, cut, and dive: a heuristic for 0-1 mixed integer programming. *Journal of Heuristics*, 13:471–503, 2007. 10.1007/s10732-007-9021-7.

[31] J. Edmonds. Covers and packings in a family of sets. *Bull. Amer. Math Soc.*, (68):494–499, 1962.

[32] Robert T. Firla, Utz-Uwe Haus, Matthias Köppe, Bianca Spille, and Robert Weismantel. Integer pivoting revisited, 2001.

[33] Matteo Fischetti, Fred Glover, and Andrea Lodi. The feasibility pump. *Mathematical Programming*, 104:2005, 2005.

[34] Matteo Fischetti and Andrea Lodi. Local branching. *Mathematical Programming*, 98:23–47, 2003. 10.1007/s10107-003-0395-5.

[35] Matteo Fischetti and Michele Monaci. How tight is the corner relaxation? *Discrete Optimizaton*, 5(2):262–269, 2008.

[36] Matteo Fischetti and Domenico Salvagnin. Feasibility pump 2.0. *Mathematical Programming Computation*, 1:201–222, 2009. 10.1007/s12532-009-0007-3.

[37] Marguerite Frank and Philip Wolfe. An algorithm for quadratic programming. *Naval Research Logistics Quarterly*, 3(1-2):95–110, 1956.

[38] R. E. Gomory. Outline of an algorithm for integer solutions to linear programs. *Bull. Amer. Math Soc.*, 64:275–278, 1958.

[39] R. E. Gomory. An algorithm for integer solutions to linear programs. In P. Wolfe, editor, *Recent Advances in Mathematical Programming*, pages 269–302. McGraw-Hill, New York, 1963.

[40] Ralph E. Gomory. Some polyhedra related to combinatorial problems. *Linear Algebra and its Applications*, 2:451–558, 1969.

[41] Ralph E. Gomory and Ellis L. Johnson. Some continuous functions related to corner polyhedra. *Mathematical Programming*, 3:23–85, 1972. 10.1007/BF01584976.

[42] G. R. Grimmett. An exact threshold theorem for random graphs and the node-packing problem. *J. Comb. Theory Ser. B*, 40:187–195, April 1986.

[43] G. R. Grimmett and W. R. Pulleyblank. Random near-regular graphs and the node packing problem. *Operations Research Letters*, 4(4):169 – 174, 1985.

[44] Andrea Grosso, Marco Locatelli, and Wayne Pullan. Simple ingredients leading to very efficient heuristics for the maximum clique problem. *Journal of Heuristics*, 14:587–612, 2008. 10.1007/s10732-007-9055-x.

[45] M. Grötschel, L. Lovász, and A. Schrijver. *Geometric algorithms and combinatorial optimization*. Algorithms and combinatorics. Springer-Verlag, 1988.

[46] R. Horst and H. Tuy. *Global Optimization (Deterministic Approaches)*. Springer-Verlag, Berlin, New York, third edition, 1996.

[47] Yoshiro Ikura and George L. Nemhauser. An efficient primal simplex algorithm for maximum weighted vertex packing on bipartite graphs. In *Bonn Workshop on Combinatorial Optimization*, volume 66 of *North-Holland Mathematics Studies*, pages 149 – 168. North-Holland, 1982.

[48] Yoshiro Ikura and George L. Nemhauser. Simplex pivots on the set packing polytope. *Mathematical Programming*, 33:123–138, 1985.

[49] Richard M. Karp. Reducibility among combinatorial problems. In R. E. Miller and J. W. Thatcher, editors, *Complexity of Computer Computations*, pages 85–103. Plenum Press, 1972.

[50] Victor Klee and David Walkup. The d-step conjecture for polyhedra of dimension $d < 6$. *Acta Mathematica*, 117:53–78, 1967.

[51] Michael Krivelevich. Triangle factors in random graphs. *Journal of Combinatorics, Probability and Computing*, 6:337–347, 1997.

[52] A. H. Land and A. G. Doig. An automatic method for solving discrete programming problems. *Econometrica*, 28:497–520, 1960.

[53] A.N Letchford and Andrea Lodi. Primal cutting plane algorithms revisited. *Mathematical Methods of Operational Research*, 56(1):67–81, 2002.

[54] Arne Lokketangen and Fred Glover. Solving zero-one mixed integer programming problems using tabu search. *European Journal of Operational Research*, 106(2-3):624 – 658, 1998.

[55] L. Lovász and M.D. Plummer, editors. *Matching Theory*, volume 121 of *North-Holland Mathematics Studies.* North-Holland, 1986.

[56] O. L. Mangasarian. Machine learning via polyhedral concave minimization. *Applied Mathematics and Parallel Computing*, pages 175–188, 1996.

[57] O. L. Mangasarian. Solution of general linear complementarity problems via nondifferentiable concave minimization. *Acta Mathematica Vietnamica*, 22:393–402, 1997.

[58] Denis Naddef. The Hirsch conjecture is true for (0, 1)-polytopes. *Mathematical Programming*, 45:109–110, 1989.

[59] G. Nemhauser and L. Wolsey. A recursive procedure for generating all cuts for 0-1 mixed integer programs. *Mathematical Programming*, 46:379–390, 1990.

[60] G. L. Nemhauser and L. E. Trotter. Properties of vertex packing and independence system polyhedra. *Mathematical Programming*, 6:48–61, 1974.

[61] G. L. Nemhauser and L. E. Trotter. Vertex packings: Structural properties and algorithms. *Mathematical Programming*, 8:232–248, 1975.

[62] Patric R.J. Östergård and Patric R. J. A fast algorithm for the maximum clique problem. *Discrete Applied Mathematics*, 120:197–207, 2002.

[63] Manfred W. Padberg. On the facial structure of set packing polyhedra. *Mathematical Programming*, 5:199–215, 1973.

[64] Wayne Pullan and Holger H. Hoos. Dynamic local search for the maximum clique problem. *J. Artif. Int. Res.*, 25:159–185, February 2006.

[65] Young R.D. A primal (all-integer) integer programming algorithm. *Journal of Research of the National Bureau of Standards*, 69B(3):213–250, 1965.

[66] Young R.D. A simplified primal (all-integer) integer programming algorithm. *Operations Research*, 16:750–782, 1968.

[67] Francisco Santos. A counterexample to the Hirsch conjecture, 2011. arXiv:1006.2814v2.

[68] Alexandre Schrijver. *Combinatorial Optimization: Polyhedra and Efficiency*, volume 24 of *Algorithms and Combinatorics*. Springer-Verlag, 2003.

[69] L.E. Trotter. *Solution characteristics and algorithms for the vertex packing problem*. Cornell University, May, 1973.

[70] V. A. Trubin. On a method of solution of integer linear programming problems of a special kind. *Soviet Math. Dokl.*, 10:1544–1596, 1969.