London School of Economics and Political Science

A generalization of totally unimodular and network matrices

PhD thesis

by

BALÁZS KOTNYEK

London, 2002.

ProQuest LLC
789 East Eisenhower Parkway
P.O. Box 1346
Ann Arbor, MI 48106-1346

Tisztaság – félegészség
Hungarian proverb (almost)

Always seek the general and never quite trust it
Joseph Epstein

Abstract

In this thesis we discuss possible generalizations of totally unimodular and network matrices. Our purpose is to introduce new classes of matrices that preserve the advantageous properties of these well-known matrices. In particular, our focus is on the polyhedral consequences of totally unimodular matrices, namely we look for matrices that can ensure vertices that are scalable to an integral vector by an integer k. We argue that simply generalizing the determinantal structure of totally unimodular matrices does not suffice to achieve this goal and one has to extend the range of values the inverses of submatrices can contain. To this end, we define k-regular matrices. We show that k-regularity is a proper generalization of total unimodularity in polyhedral terms, as it guarantees the scalability of vertices. Moreover, we prove that the k-regularity of a matrix is necessary and sufficient for substituting mod-k cuts for rank-1 Chvátal-Gomory cuts.

In the second part of the thesis we introduce binet matrices, an extension of network matrices to bidirected graphs. We provide an algorithm to calculate the columns of a binet matrix using the underlying graphical structure. Using this method, we prove some results about binet matrices and demonstrate that several interesting classes of matrices are binet. We show that binet matrices are 2-regular, therefore they provide half-integral vertices for a polyhedron with a binet constraint matrix and integral right hand side vector. We also prove that optimization on such a polyhedron can be carried out very efficiently, as there exists an extension of the network simplex method for binet matrices. Furthermore, the integer optimization with binet matrices is equivalent to solving a matching problem.

We also describe the connection of k-regular and binet matrices to other parts of combinatorial optimization, notably to matroid theory and regular vectorspaces.

Acknowledgements

First of all, I would like to thank Dr. Gautam Appa, my supervisor, for his guidance and support. His enthusiasm for research and personal attention was essential for accomplishing this work.

The Operational Research Department of the LSE was a very pleasant place to work. This is in great part due to Brenda Mowlam, who is always helpful to deal with any problem, and my fellow doctoral students, who have become real friends. This is especially true for Yiannis Mourtos, who was always there to discuss any half-baked ideas or just cheer me up.

I am also indebted to several people for fruitful discussions and encouragement. In particular, I am grateful to R. Chandrashekhar, Andras Frank, Jan van den Heuvel and R. Ravi.

Finally, I should like to acknowledge the financial support of the London School of Economics, the Overseas Research Scholarship (ORS) of the United Kingdom and the Discrete Optimization Network (DONET) of the European Union.

Contents

1 Introduction 7

1.1 The structure of the thesis 10

1.2 Preliminaries 11

1.2.1 Graphs 11

1.2.2 Numbers, vectors, matrices 12

1.2.3 Polyhedra 15

2 Generalizations of total unimodularity 17

2.1 Definitions and basic facts 18

2.2 Matrix operations 20

2.3 Necessary and sufficient conditions 23

3 Polyhedra with k-regular matrices 25

3.1 Integral polyhedra 26

3.1.1 1-regular matrices 26

3.1.2 k-regular matrices 28

3.2 Balancedness and bicolorability 29

3.3 Total dual half-integrality 30

3.4 Chvátal-Gomory cuts 33

4 Bidirected graphs 36

4.1 Basic notions 37

4.2 Determinants 42

5 Binet matrices 46

5.1 Definition and graphical representation 47

5.2 Operations on binet matrices 53

5.3 Necessary conditions 58

5.4 Towards recognition 64

6 Examples of binet and 2-regular non-binet matrices **69**
6.1 Examples 69
6.1.1 Node-edge incidence matrices 70
6.1.2 Inverse of a basis 70
6.1.3 Network matrices 71
6.1.4 Interval matrices and their generalizations 72
6.1.5 Totally unimodular but not network matrices 74
6.1.6 Minimally non-totally unimodular matrices 74
6.2 Counterexamples 76
6.3 Matrices with at most three non-zeros per row 78

7 Polyhedral results about binet matrices **81**
7.1 Half-integral polyhedra 82
7.2 Integral polyhedra 83
7.3 The Chvátal rank of binet matrices 87

8 Optimization with binet matrices **91**
8.1 Continuous binet optimization 92
8.2 Integer binet optimization 93

9 Generalized networks and the binet simplex method **96**
9.1 Simplex method for network problems 97
9.1.1 Simplex method 97
9.1.2 Network simplex method 99
9.2 Simplex method for generalized networks 101
9.2.1 Generalized networks 102
9.2.2 Generalized network simplex method 106

10 General graphs and their matroids **112**
10.1 Generalized graphs 113
10.1.1 Signed graphs 114
10.1.2 Gain graphs 115
10.1.3 Biased graphs 116
10.2 Matroids 116
10.2.1 Relevant matroid terminology 117
10.2.2 Bias matroid 120
10.2.3 Lift matroids 125

11 Connection to the row spaces of matrices **127**

11.1 Definitions 128

11.2 From k-adic vectorspaces to k-regular matrices 130

11.3 Matroids of subspaces 133

12 Conclusions **136**

Chapter 1

Introduction

Totally unimodular matrices play a central role in combinatorial optimization and integer programming because they ensure integrality. In this thesis we examine matrices that can guarantee half-integrality.

The link that connects totally unimodular matrices to integral polyhedra is Hoffman and Kruskal's [44] famous theorem which states that for integral A matrices, the polyhedron $P(A,b) = \{x \mid Ax \leq b,\ x \geq 0\}$ is integral for all integral right hand side vectors b, if an only if A is totally unimodular. In terms of integer programming, totally unimodular matrices are the integral matrices for which $max\{cx \mid Ax \leq b,\ x \geq 0\}$ has integral optimal solution for any c and any integral b. Total unimodularity thus settles the question of integrality in the sense that if we wish to decide if an integral matrix will give integral optimal solutions for all objective vectors and any integral right hand side, then we only have to check if the matrix is totally unimodular or not.

There are conditions, however, that take us beyond total unimodularity. For example, what if the matrix in question is not integral? Or what can we say about matrices that ensure integral optimal solutions for only a special set of right hand sides? These questions are not independent. If A is rational, then one can find a nonnegative integer k, such that if we multiply every row of A by k, we get an integral matrix, kA. But then instead of inequalities $Ax \leq b$, we have $kAx \leq kb$ and we deal with polyhedra that are required to be integral for only special b' vectors, namely for those whose elements are integer multiples of k. For example, if $k = 2$, so the elements of A are halves of integers, then we are to characterize integral matrices A' for which $\{x \mid A'x \leq b',\ x \geq 0\}$ is integral for all *even* vectors b'. Or equivalently, we examine matrices that provide half-integral vertices for polyhedra with integral right hand sides. In the approach followed in this work we combine the two questions above and examine rational matrices that ensure integral vertices for polyhedra in which the right hand side vector has elements that are integer multiples of a positive integer k.

Half-integral vertices arise naturally when A is the incidence matrix of an undirected graph. It is well-known that the incidence matrix of a directed graph is totally unimodular, and Heller and

Tompkins [40] showed that the incidence matrix of an undirected graph is totally unimodular if and only if the graph is bipartite. In fact, if an undirected graph G is not bipartite, then it contains an odd cycle, and the determinant of the related submatrix of its incidence matrix A is ± 2. Furthermore, it is not difficult to prove that if G is a connected undirected graph, then

$$\text{for each submatrix } R \text{ of } A,\ det(R) \in \{0, \pm 1, \pm 2\}. \tag{1.1}$$

This property then easily implies that $P(A, b)$ is integral for even b vectors. The converse, however, is not true. For example, $P(A, b)$ is clearly integral for $A = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}$ and any even b, but $det(A) = 4$. Let us relax then the condition on determinants in (1.1) so that this example is not ruled out.

$$\text{For each submatrix } R \text{ of } A, det(R) \in \{0, \pm 2^r, r \in \mathbb{N}\}. \tag{1.2}$$

This property seems to be a sufficiently wide extension of total unimodularity, one that involves the incidence matrix of any undirected or directed graph. It would be natural to claim that if a matrix A has property (1.2), then $P(A, b)$ is integral for any even vector b. Such a claim would be *false*, however, as evidenced by matrix $A = [4]$ and vector $b = 2$.

So neither property (1.1) nor property (1.2) is able to capture the real nature of matrices that provide integral vertices for any even right hand side vector. We will show that the property that accomplishes this task is:

$$\text{for each non-singular submatrix } R \text{ of } A,\ 2R^{-1} \text{ is integral.} \tag{1.3}$$

Let us give now two definitions that extend properties (1.2) and (1.3) for any positive integer k.

Definition. A matrix is called *totally k-modular* if for all of its square submatrices R, the determinant $det(R) \in \{0, \pm k^r, r \in \mathbb{N}\}$.

Definition. A rational matrix is called *k-regular* if for each of its non-singular square submatrices R, kR^{-1} is integral.

Both definitions generalize total unimodularity, but as argued above for the case of $k = 2$, and proved later for general values of k in the thesis, k-regularity is the property that takes over the role of total unimodularity in the theory of rational matrices that ensures integral vertices for polyhedra with special right hand sides. One of our most important new results states this.

Theorem. *A rational matrix A is k-regular, if and only if the polyhedron* $\{x \mid Ax \leq kb,\ x \geq 0\}$ *is integral for any integral vector b.*

Having generalized total unimodularity by k-regularity, one can ask for interesting k-regular matrices. Something similar to network matrices, which are totally unimodular and involve all 'nice'

examples of totally unimodular matrices, such as the node-edge incidence matrices of directed graphs or interval matrices. In the second part of the thesis we show that for 2-regularity, the most important case of k-regularity, there is a natural extension of network matrices, which we call the binet matrices. Binet matrices are 2-regular and involve a wide range of 'nice' 2-regular matrices. For example, the node-edge incidence matrix of an undirected graph is a binet matrix. Furthermore, just as every totally unimodular matrix is 2-regular, every network matrix is binet. And there are totally unimodular matrices that are not network matrices but we will show that they are binet matrices.

Network matrices are derived from directed graphs. Binet matrices, on the other hand, are defined on a common generalization of directed and undirected graphs, namely the bidirected graphs introduced by Edmonds [23]. The result about the 2-regularity of the incidence matrix of an undirected graph can be extended to the more general context of bidirected graphs. In fact, bidirected graphs provide the most general graphical environment in which 2-regularity can be discussed. We define binet matrices algebraicaly, using node-edge incidence matrices of bidirected graphs, but we also show that there is a graphical derivation of binet matrices from the underlying bidirected graph, enhancing the parallelism with network matrices.

Network matrices are the most important totally unimodular matrices not only because they involve all the 'nice' examples. They are also desirable for their practical advantages and theoretical significance. On the practical side, if the constraint matrix of a linear program is a network matrix, then besides the fact that the total unimodularity of network matrices guarantees integral optimal solutions in case of integral right hand sides, the graphical structure underlying network matrices also offers a very efficient algorithm to find these optimal solutions. This algorithm is the network simplex method that adapts the techniques of the simplex method to directed graphs. The simplex method can also be adapted to bidirected graphs, giving rise to the binet simplex method, an algorithm that can efficiently find the optimal solution of linear programs with a binet constraint matrix.

The theoretical significance of network matrices is related to the decomposition theory of totally unimodular matrices, due to Seymour [58]. This theory claims that every totally unimodular matrix is built up by simple operations from network matrices and two further totally unimodular matrices that are not network. We will show that the two exceptional matrices are binet, as are all network matrices, so the decomposition theory of totally unimodular matrices can be rephrased so that every totally unimodular matrix is built up by simple operations from binet matrices.

The decomposition of totally unimodular matrices is a consequence of the decomposition of regular matroids. Totally unimodular matrices are strongly connected to regular matroids, just as network matrices are to graphic matroids. In this thesis, although our focus is on matrices, we also discuss the matroidal connections of binet matrices. Basically, binet matrices represent signed graphic matroids, in much the same way as network matrices represent graphic matroids. On the other hand, we do not give a matroidal representation of 2-regular matrices, nor a characterization of signed graphic matroids similar to that of graphic matroids. This gap in matroidal equivalence

will probably delay the discovery of a recognition algorithm for 2-regular or binet matrices. In fact, the only device we possess to prove that a matrix is 2-regular (besides checking the inverse of each submatrix) is to show that the matrix is binet. Thus, the current status of recognizing 2-regular matrices is in par with the theory of totally unimodular matrices after the identification of network matrices but prior to the discovery of the decomposition theory that led to a recognition algorithm for totally unimodular matrices.

In the remainder of this chapter we first give an overview of the thesis, then provide a background to the research chapters by listing the necessary definitions and results.

1.1 The structure of the thesis

The thesis can be divided into two parts. The first deals with the generalization of total unimodularity, the second discusses the generalization of network matrices.

Total k-modularity and k-regularity, the extensions of total unimodularity, are defined in Chapter 2, where their connection and basic properties are also described. Chapter 3 is about the polyhedral implications of k-regularity. This is the chapter where the choice of k-regularity as the proper generalization of total unimodularity is justified. We extend the polyhedral integrality results about totally unimodular matrices to rational and integral k-regular matrices in that chapter.

Chapter 4 is about bidirected graphs. This chapter does not contain new results, it serves as a foundation for further parts.

We embark on generalizing network matrices in Chapter 5, where we define binet matrices, describe in great detail the graphical method to derive them from the underlying bidirected graphs, and prove some fundamental properties of binet matrices. Chapter 5 is a focal point in the thesis, every chapter after it exploits the results given there. The next chapter, for instance, gives examples of binet matrices by delineating the underlying bidirected graphs, and relies on the graphical algorithm described in Chapter 5 to derive the matrices form the graphs. We show in Chapter 6 that network matrices and other interesting totally unimodular and not totally unimodular matrices are binet.

Chapter 7 is about the polyhedral results related to binet matrices. We prove that binet matrices are 2-regular, linking thus the two parts of the thesis. We also present a characterization of binet matrices that are totally unimodular. The 2-regularity of binet matrices means that a linear program with binet constraint matrix and integral right hand side has half-integral optimal solutions. Chapters 8 and 9 offer methods to find the optimal solutions. In Chapter 8, we mention general-purpose methods for linear programs with binet constraint matrix (for short, binet optimization problems), but mainly focus on integer programs. It turns out that finding integer solution to a binet optimization problem is equivalent to solving a matching problem, so a strongly polynomial algorithm exists for this task. Chapter 9 concentrates on the binet simplex method. We argue that such a method exists, since bidirected graphs can be considered to be special cases of generalized networks and the existing

generalized network simplex method can be adapted to bidirected graphs.

Chapter 10 deals with matroids. To exhibit the connection of binet matrices to existing special classes of matroids, we introduce some further generalizations of graphs and define the matroids on these graphs in a more general form. That chapter mainly contains known results. Our contribution is to show how binet matrices fit into existing theories, so widening their applicability. This is also true for Chapter 11, which describes a different concept to generalize totally unimodular matrices. This generalization, due to Lee [48], deals with the linear vectorspaces arising as row spaces of matrices. We demonstrate the differences of this concept from ours, which concentrates on matrices. Chapter 11 belongs rather to the first part of the thesis, which is about k-regularity, than the second, which concerns binet matrices. We put it at the end of the thesis because it is not an essential part of the dissertation, only a very important related issue.

1.2 Preliminaries

Here we define the notions used throughout the thesis, and give some known results that are going to be built upon later. The notations we use are standard. Similarly, the results listed here can be found in any textbook on combinatorial optimization. A reader not familiar with the content of this section can consult, for example, Nemhauser and Wolsey [51] or Schrijver [55]. Later, in due course, we will give the definitions of further concepts that appear only in parts of the dissertation. For example, we describe basic matroid theory in Section 10.2.

1.2.1 Graphs

$G(V, E)$ denotes a graph with node set V and edge set E. Edges can be directed, undirected or bidirected. Bidirected graphs are defined in Chapter 4, where we also redefine many of the general graphic structures mentioned here. E can contain *loops*, i.e., edges whose end-nodes coincide. We will also employ *loose edges*, i.e., virtual edges that have no end-nodes. A directed edge has one *tail* (the node where the edge comes from) and one *head* (where it goes). An undirected graph can be considered as a graph in which each node has two heads and no tails. An edge connecting nodes u and v is denoted by (u, v). If it is a directed edge, then this notation implies that u is the tail and v is the head of the edge.

We will use the standard names for special subgraphs. So we will speak about walks, paths (meaning simple paths), trees, parallel edges, etc. A *walk* is a sequence of edges such that every edge is incident to the ones standing before and after it in the sequence. (In Figure 5.2(ii), page 49, edges $s, r_1, r_4, r_3, r_2, r_1$ form a walk.) A *path* is a walk that does not cross itself, i.e., it does not use the same node twice, except maybe the first node. (As e.g., s, r_1, r_4 in Figure 5.2(ii).) A graph is *connected* if there is a path between any two nodes. By a *cycle* we mean a closed path, and not

a closed walk. For example, the first graph in Figure 5.2 is a cycle, the other two graphs are not. A *tree* is a connected graph which does not contain a cycle. A *theta graph* is the union of three internally node-disjoint paths between the same pair of nodes (see Figure 1.1). In all these subgraphs

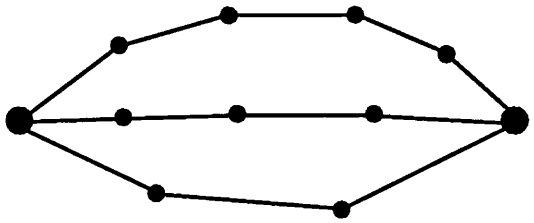


Figure 1.1: A theta graph.

the directions of the edges are not relevant. When they are, we speak about *directed paths, directed cycles*, etc.. If G is not connected, then it has more than one connected components, or in short, *components*. An edge is a *cut-edge*, if it separates two parts of the graph, i.e., after deleting a cut-edge the graph has one more connected component. We call a connected subgraph that contains exactly one cycle a *1-tree*. The name follows [2], but the same or very similar structures have appeared in the literature with several different names. A very similar concept is called an 'augmented tree' in [2], and a 'quasitree' in [50]. Zaslavsky [73] used the name 'unicycle' for a 1-tree. Note that Held and Karp [39] used the term 1-tree for a similar, less general structure. We use the name 1-tree to express the fact that such a subgraph contains a tree plus exactly one additional edge. Note that the number of nodes in a 1-tree equals the number of edges. A subgraph *spanned* by an edge set contains the edges in the set and their end-nodes. A subgraph is *spanning*, if the graph spanned by its edges contains all nodes of the graph. A node is *isolated* if it is not incident to any edge.

We will use the standard graph operations, such as deletion or contraction of edges. *Contraction* can be imagined as shrinking the edge until it has zero length, and its end-nodes become one node.

The *node-edge incidence matrix* of a graph has its rows and columns associated with the nodes and edges of the graph. The non-zeros in a column associated with edge e stand in the rows that correspond to the end-nodes of e. Heads get positive signs, tails get negative signs.

1.2.2 Numbers, vectors, matrices

As always, $\mathbb{Z}$, $\mathbb{Q}$ and $\mathbb{R}$ denote the set of integer, rational, and real numbers. $\mathbb{N}$ contains the nonnegative integers. The set of positive real and integer numbers are $\mathbb{R}_+$ and $\mathbb{Z}_+$, respectively. The greatest integer smaller than $x \in \mathbb{R}$ is denoted by $\lfloor x \rfloor$.

Vectors and matrices whose elements are integers are called *integral*. That is, integral m-dimensional vectors are those in $\mathbb{Z}^m$, an integral matrix of size $m \times n$ is in $\mathbb{Z}^{m \times n}$. Similarly, *rational* vectors and matrices have elements from $\mathbb{Q}$. The set of m-dimensional vectors whose elements are

all integer multiples of a rational r is denoted by $r\mathbb{Z}^m$. If $r = ½$, then we use the term *half-integral*. That is, half-integral vectors and matrices have elements that are integer multiples of ½. If k is integer, then for any element a of $k\mathbb{Z}$ it is true that $k|a$, i.e., k divides a or a is divisible by k. If x is an integral vector, then the greatest integer that divides each element of x is its *greatest common divisor*, denoted as $\gcd(x)$. The *characteristic vector* $x \in \{0,1\}^m$ of a set $S \subseteq \{1,2,\ldots,m\}$ is defined as

$$x_i = \begin{cases} 1 & i \in S \\ 0 & i \notin S \end{cases}$$

For a matrix A with row set I and column set J, $(A)_{ij}$ or A_{ij} or a_{ij} denotes the element in row $i \in I$ and column $j \in J$. This fact can also be expressed as $A = [a_{ij}]_{i\in I}^{j\in J}$. If A is a matrix, then kA and A/k denote the matrix obtained by multiplying or, respectively, dividing all elements by k. Thus, A is a half-integral matrix, if and only if $2A$ is integral.

Vectors that contain only zeros are called *all-zero*, and denoted by $\mathbf{0}$. Every element of an *all-one* vector is 1. We use the notation $\mathbf{1}$ for all-one vectors. In matrix notation, $[A, B]$ denotes a matrix in which A and B stand next to each other, and not one in which A is above B. To get simpler notations for matrices, we sometimes leave blank the cells that contain 0 elements. If we do not want to specify a part of a matrix or vector, we use the sign $*$. So

$$A = \begin{bmatrix} a & \\ * & b \end{bmatrix}$$

is a 2×2 matrix which has a and b in its diagonal, zero above its diagonal and an unspecified element below it. In other words, A is a *lower triangular* matrix. Note that instead of elements, we can define a matrix with its submatrices. Thus,

$$A = \begin{bmatrix} A_1 & \\ & A_2 \end{bmatrix}$$

is a matrix that has submatrices A_1 and A_2 in the upper left and bottom right corners, respectively, and zeros outside them. This structure of A is its *decomposition* to components A_1 and A_2. If A_1 and A_2 are square submatrices, then they are *blocks* of A, and A is *block diagonal*.

The unit matrix is denoted by I. We do not specify its size, it is usually clear from the context. If not, we will use the notation I_m for unit matrices of size m.

Vectors $x_1, \ldots, x_n$ are *linearly independent*, if there is no $\lambda = (\lambda_1 \ldots, \lambda_n)$ such that $\lambda \neq \mathbf{0}$ and $\sum \lambda_i x_i = 0$. The determinant of a square matrix A is denoted as $det(A)$. A is *non-singular*, if $det(A) \neq 0$. In this case, the columns of A are linearly independent vectors, and A is invertible. The *cofactor* of element A_{ij}, $cof(A_{ij})$, is the determinant of the submatrix achieved by deleting row j and column i from the square matrix A and scaled by $(-1)^{i+j}$. The *rank* of a matrix A, $rank(A)$, is the size of the largest non-singular square submatrix in A. A matrix is of *full row rank*, if its rank

equals the number of its rows, or equivalently, if its row vectors are linearly independent. A *basis* of a full row rank matrix A is a non-singular square submatrix of size $rank(A)$. The rank of the node-edge incidence matrix A of a connected directed graph G on m nodes is $m-1$. Moreover, by deleting any row, A can be made a full row rank matrix A'. The bases of A' correspond to spanning trees of G.

Pivoting on an element α of a matrix A means the following transformation:

$$A = \begin{bmatrix} \alpha & c \\ b & D \end{bmatrix} \longrightarrow \bar{A} = \begin{bmatrix} 1/\alpha & c/\alpha \\ -b/\alpha & D - \frac{1}{\alpha}bc \end{bmatrix}, \tag{1.4}$$

where α is a non-zero element, b is a column vector, c is a row vector, and D is a matrix. Pivoting is equivalent to a series of row operations executed on $[A, I]$, which is the same as premultiplying $[A, I]$ by the the inverse of a basis. That is, if $A = [R, S]$ and R is a basis of A, then consecutive pivoting on the diagonal elements of R converts $[R, S, I]$ to $[I, R^{-1}S, R^{-1}]$ (up to column permutations). This phenomenon can also be expressed as a 'change in the basis', namely changing the roles of I and R.

A matrix is called *totally unimodular*, if each of its submatrices has determinant $0, +1$ or -1. The following lemma can be proved by *Cramer's rule*, which determines the element of the inverse of a matrix. Namely, it states that $(A^{-1})_{ij} = cof(A_{ij})/det(A)$.

Lemma 1.1. *For every non-singular submatrix R of a totally unimodular matrix, R^{-1} is integral.*

A full row rank matrix is *unimodular*, if each of its bases has determinant ± 1.

Lemma 1.2. *A matrix A is totally unimodular if and only if $[A, I]$ is unimodular.*

Totally unimodular matrices play a central role in this thesis. We give two theorems that characterize totally unimodular matrices.

Theorem 1.3. (Ghouila-Houri [34]) *A $0, \pm 1$ matrix is totally unimodular, if and only if for each collection of columns or rows, there exists a scaling of the selected columns or rows by ± 1 such that the sum of the scaled columns or rows is a vector of $0, \pm 1$ elements.*

Theorem 1.4. (Gomory, see Camion [10]) *If A is a $0, \pm 1$ matrix, then it is either totally unimodular or has a submatrix with determinant ± 2.*

Examples of totally unimodular matrices include the incidence matrices of directed graphs and network matrices. Pivoting on a network matrix yields another network matrix. Similarly, the pivoted version of a totally unimodular matrix is still totally unimodular.

1.2.3 Polyhedra

A set P of vectors in $\mathbb{R}^n$ is a *polyhedron* if $P = \{x \mid Ax \leq b\}$ for an $m \times n$ matrix A and m-dimensional vector b. If A is a rational matrix and b is a rational vector, then P is a *rational* polyhedron. A *polytope* is a bounded polyhedron. Equivalently, a polytope P can be defined as the *convex hull* of a finite set F of vectors, $P = conv(F)$. The vectors of a polyhedron are also called its points. An *extreme point* or *vertex* of a polyhedron $P = \{x \mid Ax \leq b\}$ is a point determined by n linearly independent equations from $Ax = b$. Every extreme point of P can arise as an optimal solution of $max\{cx \mid x \in P\}$ for a suitably chosen c.

If P has at least one vertex (in which case P is called *pointed*), then P is called *integral*, if all of its vertices are integral. One can also define integrality of polyhedra without vertices, as follows. A non-empty subset F of polyhedron $P = \{x \mid Ax \leq b\}$ is a *minimal face* if $F = \{x \mid A'x = b'\}$ for some subsystem $A'x \leq b'$ of $Ax \leq b$. Therefore, a vertex is always a minimal face, consisting of one point. Furthermore, $P = \{x \mid Ax \leq b\}$ is pointed, if and only if $rank(A) = n$. Now, a polyhedron is *integral*, if each of its minimal faces contains an integral point. This is consistent with the previous definition given above for pointed polyhedra.

We call a polyhedron *half-integral*, if each of its minimal faces contains a half-integral point. The extreme points of a pointed half-integral polyhedron are half-integral. An integral polyhedron P provides integral optimal solutions for $max\{cx \mid x \in P\}$ for any c. Similarly, if P is half-integral, then the optimal solutions are half-integral.

Totally unimodular matrices are important, because they lead to integral polyhedra. The following results illustrate this statement.

Theorem 1.5. *Let A be a totally unimodular matrix, and b be an integral vector. Then the polyhedron $P = \{x \mid Ax \leq b\}$ is integral.*

We give a sketch of the proof for the next lemma here, because later we will refer to it.

Lemma 1.6. *Let A be an integral matrix and b an integral vector. Then polyhedron $P = \{x \mid Ax \leq b, x \geq 0\}$ is integral, if and only if polyhedron $Q = \{z \mid [A, I]z = b, z \geq 0\}$ is integral.*

Proof: Let us first suppose that P is integral and z^* is a vertex of Q. Then $z^* = (x^*, u^*) \geq 0$ and $Ax^* + u^* = b$. Thus, $x^* \in P$ and it is a vertex of P. So x^* is integral, and because A and b are integral, u^* and z^* are integral.

Suppose now that Q is integral and x^* is a vertex of P. Let $u^* = b - Ax^*$ and $z^* = (x^*, u^*)$. Then z^* is a vertex of Q, so it is integral, hence x^* is integral too. □

Theorem 1.7. *Let A be an integral full row rank matrix. Then the polyhedron $P = \{x \mid Ax = b, x \geq 0\}$ is integral for each integral vector b, if and only if A is unimodular.*

The next theorem, which is fundamental in the field, is a consequence of Lemma 1.2, Lemma 1.6 and Theorem 1.7.

Theorem 1.8. (Hoffmann and Kruskal [44]) *Let A be an integral matrix. Then the polyhedron $P = \{x \mid Ax \leq b, x \geq 0\}$ is integral for each integral vector b, if and only if A is totally unimodular.*

Further bounds on x and Ax can be handled similarly, as it is easy to see that A is totally unimodular if and only if $[A^T, -A^T, I, -I]^T$ is such.

Theorem 1.9. *Let A be an integral matrix. Then the polyhedron $P = \{x \mid a \leq Ax \leq b, c \leq x \leq d\}$ is integral for all integral vectors a, b, c and d, if and only if A is totally unimodular.*

Chapter 2

Generalizations of total unimodularity

Our aim is to generalize totally unimodular matrices so that most of their advantages are preserved. A totally unimodular matrix has two important attributes that establish its agreeable properties: it is integral and any of its non-singular submatrices has an integral inverse. Both attributes are consequences of the requirement on the subdeterminants of a totally unimodular matrix. We will see that if the requirement is relaxed, then this advantageous co-existence breaks down. If we extend the possible values for subdeterminants, then the inverses of submatrices are not integral. On the other hand, if we want all inverses to be integral, or scalable to integral, then the integrality of the matrix cannot be guaranteed. Thus we give two different generalizations of totally unimodular matrices: total k-modularity prescribes specific values for subdeterminants, while k-regularity requires the inverses of subdeterminats to be scalable to integral.

One can argue about which generalization is more appropriate. Total k-modularity is the natural generalization, allowing powers of an integer k for subdeterminants, and it keeps the integrality of a matrix, but it fails to provide similar consequences for polyhedra as does total unimodularity. In fact, we will show in the next chapter that it is k-regularity, which preserves the integrality of the inverses up to a multiplication by k, that inherits the properties of totally unimodular matrices that make them so important in combinatorial optimization. We also show in this chapter that for integral matrices and special but important values of k, total k-modularity is implied by k-regularity. That is why most of this chapter is about k-regular matrices. In our treatment, total k-modularity appears mainly as an advantageous property of special k-regular matrices.

Matrices with conditions on subdeterminants have been studied. The name of total k-modularity comes from Appa [3]. Whittle [68] called totally 2-modular matrices *2-matrices*. For the same set of matrices, Zaslavsky used the name *totally dyadic*. Lee [48] also dealt with subdeterminants.

We present more about his results in Chapter 11. Matrices with special inverses, to the best of our knowledge, first appeared in Appa [3]. In that work, all matrices are assumed to be integral, too strong a requirement for k-regularity.

We start the chapter with the definitions of total k-modularity and k-regularity. Section 2.1 also contains the basic results about totally k-modular and k-regular matrices. In Section 2.2 we deal with operations that preserve the k-regularity of a matrix. It will be apparent in this section that there is a substantial difference between integral and non-integral k-regular matrices. The last section of this chapter is about conditions that are necessary or sufficient for the total k-modularity or k-regularity of a matrix.

2.1 Definitions and basic facts

A matrix is *totally unimodular* if all of its square submatrices have determinant 0, 1 or -1. An immediate generalization is to allow other values for subdeterminants. In what follows, let k be a positive integer.

Definition 2.1. A matrix is called *totally k-modular*, if for all of its square submatrices R, the determinant $det(R) \in \{0, \pm k^r, r \in \mathbb{N}\}$.

Obviously, a totally k-modular matrix is integral. The other generalization we give is based on Lemma 1.1, which claims that if a matrix is totally unimodular, then the inverses of its submatrices are integral.

Definition 2.2. A rational matrix is called *k-regular*, if for each of its non-singular square submatrices R, kR^{-1} is integral.

Now, Lemma 1.1 can be rephrased so that if A is totally unimodular, then it is 1-regular. The set of 1-regular matrices, however, is wider than that of totally unimodular matrices as a 1-regular matrix is not necessarily integral. For example, $A = [1/2]$ is a 1-regular matrix. In Section 6.1.2 we give further non-integral, 1-regular matrices. On the other hand, it is easy to see that for integral matrices total unimodularity is equivalent to 1-regularity.

Lemma 2.3. *An integral matrix is 1-regular if and only if it is totally unimodular.*

Appa extended this result for special but important values of k.

Theorem 2.4. (Appa [3]) *Suppose $k = 1$ or k is a prime number. If an integral matrix is k-regular, then it is totally k-modular.*

Proof: Let A be an integral, k-regular matrix. Then for any nonsingular submatrix R of A, kR^{-1} is integral. Now $|det(kR^{-1})| \cdot |det(R)| = k^m$ if R is an $m \times m$ matrix. But kR^{-1} and R are integral matrices, so $det(kR^{-1})$ and $det(R)$ are integers. Then, for either $k = 1$ or k is a prime number, $det(R) \in \{\pm k^r, r \in \mathbb{N}\}$. □

With the same proof, the following similar theorem can be proved. It holds for more possible values of k, but its claim is weaker.

Theorem 2.5. *Suppose that $k = p^r$ where $p = 1$ or p is a prime number and $r \in \mathbb{N}$. If an integral matrix is k-regular, then it is totally p-modular.*

The converse of Theorem 2.4 is not true. The following matrices are all totally 2-modular but none of them is 2-regular.

$$A = [4] \quad \text{with } A^{-1} = [1/4] \tag{2.1}$$

$$A = \begin{bmatrix} 1 & 2 \\ 2 & 0 \end{bmatrix} \text{ with } A^{-1} = \begin{bmatrix} 0 & 1/2 \\ 1/2 & -1/4 \end{bmatrix} \tag{2.2}$$

$$A = \begin{bmatrix} 1 & -1 & 1 \\ 1 & 1 & 1 \\ 1 & 0 & -1 \end{bmatrix} \text{ with } A^{-1} = \begin{bmatrix} 1/4 & 1/4 & 1/2 \\ -1/2 & 1/2 & 0 \\ 1/4 & 1/4 & -1/2 \end{bmatrix} \tag{2.3}$$

$$A = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 \end{bmatrix} \text{ with } A^{-1} = \begin{bmatrix} 1/2 & -1/2 & 1/2 & 0 & 0 & 0 \\ 1/2 & 1/2 & -1/2 & 0 & 0 & 0 \\ -1/2 & 1/2 & 1/2 & 0 & 0 & 0 \\ -1/4 & 1/4 & -1/4 & 1/2 & -1/2 & 1/2 \\ -1/4 & 1/4 & -1/4 & 1/2 & 1/2 & -1/2 \\ 1/4 & -1/4 & 1/4 & -1/2 & 1/2 & 1/2 \end{bmatrix} \tag{2.4}$$

Observe that all these matrices are 4-regular. This fact suggests the following result, which is valid for any integer k.

Theorem 2.6. *For any totally k-modular matrix A, there exists an $r \in \mathbb{N}$ such that A is k^r-regular.*

Proof: Let $r = max\{q : |det(R)| = k^q \text{ if } R \text{ is a submatrix of } A\}$. The elements of the inverse of a non-singular R are computed as $(R^{-1})_{ij} = cof(R_{ij})/det(R)$, where $cof(R_{ij})$ is the cofactor of element R_{ij} in R, i.e., the determinant of a submatrix of A. Thus $(R^{-1})_{ij} = \pm k^s$ for an appropriate integer s, and we can be sure that $s \geq -r$. It follows then that $k^r R_{ij}$ is integral. □

Note that if $r = 0$, so every subdeterminant of A is 0 or ± 1, then we get Lemma 1.1. For further references, we state Theorem 2.6 for the case where $r = 1$.

Lemma 2.7. *If for each non-singular square submatrix R of a matrix A, $det(R) \in \{\pm 1, \pm k\}$, then A is k-regular.*

The converse of this lemma, which would require that a k-regular matrix has only subdeterminant $0, \pm 1, \pm k$, is not true, as $2I_2$, where I_2 is the 2×2 identity matrix, or the following $0, \pm 1$ matrix

show.

$$B = \begin{bmatrix} -1 & -1 & 1 \\ 1 & -1 & 1 \\ -1 & 1 & 1 \end{bmatrix} \text{ with } B^{-1} = \begin{bmatrix} -½ & ½ & 0 \\ -½ & 0 & ½ \\ 0 & ½ & ½ \end{bmatrix} \tag{2.5}$$

In both examples the determinant is 4 though the matrices are 2-regular. Thus, allowing higher powers of k in Definition 2.1, as opposed to the case in Lemma 2.7, has two advantages. It provides us with a necessary condition of k-regularity for an important class of possible values of k, namely if k is a power of a prime. On the other hand, this definition does not rule out decomposable matrices.

In this thesis we mainly focus on 2-regular matrices. Their importance among k-regular matrices is explained by Theorem 1.4. It claims that if a $0, \pm 1$ matrix is not totally unimodular, then it has a submatrix with determinant ± 2. Take a k-regular matrix A with elements $0, \pm 1$. If it is not totally unimodular (i.e, 1-regular), then it has a minimal submatrix B with determinant ± 2. That is, every subdeterminant of B is $0, \pm 1$. It follows then that the inverse of B has half-integral but not integral elements. This fact implies the following observation.

Lemma 2.8. *If a matrix with* $0, \pm 1$ *elements is* k*-regular, then either* $k = 1$, *or* $2|k$.

2.2 Matrix operations

Total unimodularity of a matrix is maintained under several operations, e.g. transposing, taking submatrices or pivoting. Some of these operations preserve k-regularity too, as simple checking of the defining condition shows.

Lemma 2.9. *Let* A *be* k*-regular. Then the following matrices are also* k*-regular:*
(a) the transpose of A,
(b) any submatrix of A,
(c) the matrix obtained by multiplying a row or column of A *by* -1,
(d) the matrix obtained by interchanging two rows or columns of A,
(e) the matrix obtained by duplicating a row or column of A,
(f) the matrix obtained by dividing a row or column of A *by a non-zero integer.*

Proof: We prove only part (f). The other parts are even more trivial. If one divides a row of A by a non-zero integer d, then the inverse of any submatrix B of A is either unchanged, if B does not contain the divided row, or a column of B^{-1} is multiplied by d. So if kB^{-1} is integral, then the new inverse is also integral. □

Matrix A is totally unimodular if and only if $[A, I]$ is unimodular, i.e., each of its bases has determinant ± 1. This equivalence can be carried over for integral k-regular matrices.

Lemma 2.10. *Let A be an integral matrix. Then the following statements are equivalent.*
(a) A is k-regular.
(b) $[A, I]$ is k-regular.
(c) For each basis T of $[A, I]$ the matrix kT^{-1} is integral.

Proof: Let T be a basis of $[A, I]$. Then up to row permutations

$$T = \begin{bmatrix} S & I \\ R & 0 \end{bmatrix} \text{ and } T^{-1} = \begin{bmatrix} 0 & R^{-1} \\ I & -SR^{-1} \end{bmatrix},$$

where S and R are submatrices of A, R is a non-singular square matrix and I is an identity matrix of appropriate size (it may be empty, in which case $T = R$). If kT^{-1} is integral, then obviously kR^{-1} is integral. If kR^{-1} is integral, then, since S is integral, kT^{-1} is also integral. The rest of the proof is an easy consequence of the fact that any submatrix of a k-regular matrix is also k-regular. □

It is necessary to require the integrality of A in the lemma above, because for example if $A = \begin{bmatrix} 1 \\ \frac{1}{2} \end{bmatrix}$, then A is 1-regular, but $[A, I]$ is not, as

$$\begin{bmatrix} 1 & 0 \\ \frac{1}{2} & 1 \end{bmatrix}^{-1} = \begin{bmatrix} 1 & 0 \\ -\frac{1}{2} & 1 \end{bmatrix}.$$

Another operation which preserves the k-regularity of an integral matrix is pivoting.

Lemma 2.11. *Let A be an integral k-regular matrix. Then matrix $\tilde{A}$ obtained by pivoting on a non-zero element is also k-regular.*

Proof: Let us suppose that A has m rows. Pivoting is equivalent to row operations on $[A, I]$. That is, by using the notations of (1.4), first divide the first row of $[A, I]$ by α, getting A'. By Lemma 2.9(f) and Lemma 2.10, A' is k-regular. Then subtract b_i times the first row of A' from the i^{th} row $(i = 2, \ldots, m)$, obtaining $\tilde{A}$. Matrices A' and $\tilde{A}$ have the following form:

$$A' = \begin{bmatrix} 1 & c/\alpha & 1/\alpha & 0 \\ b & D & 0 & I \end{bmatrix}, \quad \tilde{A} = \begin{bmatrix} 1 & c/\alpha & 1/\alpha & 0 \\ 0 & D - \frac{1}{\alpha}bc & -b/\alpha & I \end{bmatrix}.$$

Notice that $\tilde{A}$ equals $[\bar{A}, I]$ up to column exchanges. Thus, by Lemma 2.10, $\tilde{A}$ is k-regular if kT^{-1} is integral for each basis T of $\tilde{A}$. We show that this is true.

Let T be a basis of $\tilde{A}$. Then there is a basis T' of A' such that

$$T = \begin{bmatrix} 1 & 0 \\ -b & I \end{bmatrix} \cdot T' \text{ and then } T^{-1} = (T')^{-1} \cdot \begin{bmatrix} 1 & 0 \\ b & I \end{bmatrix}.$$

A' is k-regular, so $k(T')^{-1}$ is integral. Column vector b is integral, hence kT^{-1} is also integral. □

It is a well-known fact, and easily follows from the proof above, that pivoting on a matrix is equivalent to multiplying the matrix by the inverse of a basis. We make use of this parallelism in the next lemma.

Lemma 2.12. *Let A be an integral, k-regular, full row rank matrix, R be a basis of it and $A = [R, S]$. Then*
(a) R^{-1} is k-regular,
(b) $R^{-1}S$ is k-regular,
(c) $R^{-1}A$ is k-regular.

Proof: The matrix $R^{-1}[A, I] = [I, R^{-1}S, R^{-1}]$ can be obtained from $[A, I]$ by consecutive pivoting. Then, by Lemma 2.11 and Lemma 2.10, all claims follow. □

As the proofs of Lemma 2.11 and Lemma 2.12 strongly used Lemma 2.10, it is to be expected that they do not remain valid for rational matrices. For example, $A = [½]$ is 1-regular but $\bar{A} = A^{-1} = [2]$ is not 1-regular.

This example emphasises the fact that rational matrices behave quite differently from integral ones. However, the examples given so far of rational 1-regular matrices could be obtained trivially from totally unimodular matrices by dividing a row by 2. Lemma 2.9(f) then ensures their 1-regularity. This is not always the case. For example, the 1-regular matrix $\begin{bmatrix} ½ & ½ \\ -½ & ½ \end{bmatrix}$ cannot be achieved in this way, because the only candidate, $\begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix}$ is not totally unimodular. But it is 2-regular, which leads to the following result. Recall that A/s denotes the matrix obtained from A by dividing each of its elements by s.

Lemma 2.13. *Let A be a rational matrix and s and k be positive integers. Then A/s is k-regular if and only if A is sk-regular.*

Proof: Let R be a non-singular square submatrix of A. $(R/s)^{-1} = sR^{-1}$ so $k(R/s)^{-1}$ is integral if and only if skR^{-1} is such. □

One direction of this lemma means that dividing the *whole* matrix makes it 'stronger', k-regular. It is necessary to require that all elements of A are divided. If this is not the case, then we cannot ensure k-regularity, only a weaker version, as in Lemma 2.9(f), where we stated that if we divide some rows or columns of A by s, then we can preserve its sk-regularity. The other direction of Lemma 2.13 is about multiplication of a matrix with a positive integer. It is easy to show that in this case, as opposed to division, there is no difference between multiplying the whole matrix or only some rows or columns – both make it 'weaker'.

2.3 Necessary and sufficient conditions

Since it became clear that totally unimodular matrices play an important role in the theory of integer programming several necessary and sufficient conditions for a matrix being totally unimodular have been found. A comprehensive list of these results can be found in e.g., [55]. We also stated some of these conditions in Section 1.2.2. Unfortunately, these characterizations cannot be directly extended to k-regular matrices because they make use of the determinental definition of totally unimodular matrices, which does not remain valid for k-regular matrices. Nevertheless, necessary conditions can be given. One of them extends the following result of Chandrasekaran. Recall that $\gcd(x)$ denotes the greatest common divisor of the elements of the integral vector x.

Theorem 2.14. (Chandrasekaran [13]) *Matrix A is totally unimodular if and only if for each non-singular square submatrix R of A and for each non-zero $0, \pm 1$ vector y, $\gcd(Ry) = 1$.*

The necessary condition for integral k-regular matrices is the following:

Theorem 2.15. *If A is an integral k-regular matrix, then for any non-singular square submatrix R of A and for each non-zero $0, \pm 1$ vector y, $\gcd(Ry)|k$.*

Proof: A is k-regular, so kR^{-1} is integral. Then using the notation $l = \gcd(Ry)$,

$$\frac{1}{l}ky = R^{-1}R\frac{1}{l}ky = kR^{-1}\frac{1}{l}Ry$$

is integral. This implies that $l|k$. □

This condition is not sufficient, as shown for $k = 2$ by matrix A of (2.2), which is not 2-regular even though $\gcd(Ry)|2$ for any submatrix R and non-zero $0, \pm 1$ vector y. For total k-modularity, however, a similar condition is sufficient.

Theorem 2.16. *If for each non-singular square submatrix R of matrix A and for each non-zero vector y whose elements are of $\{0, \pm k^r, r \in \mathbb{N}\}$, $\gcd(Ry)$ is a power of k, then A is totally k-modular.*

Proof: We prove the theorem by induction on the size of R. If $R = [\alpha]$ is a 1×1 non-singular matrix, then for $y = 1$ the condition of the theorem claims that $det(R) = \gcd(Ry) = \alpha$ is a power of k. Now let R be a non-singular square submatrix of A such that for all real submatrices S of R, $det(S) \in \{0, \pm k^r, r \in \mathbb{N}\}$. Then $Rd = [det(R), 0, \ldots, 0]^T$, where d is a vector made up from the cofactors of R corresponding to its first row, hence its elements are of the set $\{0, \pm k^r,\ r \in \mathbb{N}\}$. The condition of the theorem claims that then $\gcd(Rd) = \pm det(R)$ is a power of k, so the total k-modularity of A follows. □

If we assume that k is prime, then the condition becomes necessary as well.

Theorem 2.17. *If k is a prime number (or 1) and A is a totally k-modular matrix, then for each non-singular submatrix R of A and for each non-zero vector y whose elements are of $\{0, \pm k^r, r \in \mathbb{N}\}$, $\gcd(Ry)$ is a power of k.*

Proof: Let us use the shorthand notation $l = \gcd(Ry)$. The vector $b = Ry/l$ is integral, and the only solution of the linear system $Rx = b$ is $x = y/l$. Then, by Cramer's rule, $x_i = det(D)/det(R)$, where D is an integral matrix made up from b and all but the i^{th} columns of R. Thus, $l | y_i det(R)$ and because i can be chosen so that $y_i, det(R) \in \{\pm k^r, r \in \mathbb{N}\}$ and k is prime, l is a power of k. □

Chapter 3

Polyhedra with k-regular matrices

Hoffman and Kruskal's theorem (Theorem 1.8) fully characterizes integral matrices which lead to an integral polyhedron for all integral right hand side vectors. Our purpose in generalizing totally unimodular matrices was to extend this result. One can take two approaches for extension. The first asks what kind of rational matrices provide the same polyhedral property. We will show that these are exactly the 1-regular matrices. The other approach for extending Theorem 1.8 is to consider only special right hand sides, namely vectors which are integer multiples of k, and require the integrality of the polyhedra for only these vectors. This approach leads to k-regular matrices. These results justify our statement in the previous chapter which claimed that k-regularity is the appropriate generalization of total unimodularity as it retains some important polyhedral implications. We discuss these two extensions in Section 3.1.

Polyhedra with special right hand side vectors have been extensively studied. The best known of these results deals with balanced matrices, which can guarantee integral polyhedra for all-one right hand sides. In Section 3.2, we examine the relationship of 2-regular and balanced matrices. Our results are negative – neither of these two notions implies the other. We discuss balanced matrices in this thesis only to demonstrate that 2-regular matrices represent a really new set of matrices.

Another important notion connected to integral polyhedra is total dual integrality, which deals with fixed b vectors and examines the integral solutions of the inequality system $Ax \leq b$. In Section 3.3, we extend the definition of total dual integrality to cover half-integral solutions.

Finally, we discuss Chvátal-Gomory cuts. If a polyhedron $\{x \mid Ax \leq b,\ x \geq 0\}$ is not integral, then its integer hull can be achieved by applying cuts. We show in Section 3.4 that it suffices to use very special cuts if matrix A is k-regular.

3.1 Integral polyhedra

In Section 1.2.3 we listed the important results that relate totally unimodular matrices to integral polyhedra. The key property of totally unimodular matrices, which makes it possible to prove those results, is the integrality of the inverse of any of its non-singular submatrices. The definition of 1-regularity requires just this property of a rational matrix, therefore it is no suprise that most of the results that hold for totally unimodular matrices can be proved for 1-regular matrices too, using the same ideas. We present these results in Section 3.1.1.

In Section 3.1.2 we treat k-regular matrices. By a simple device based on Lemma 2.13, we adapt the results about 1-regular matrices to k-regular ones. We prove that k-regular matrices are exactly the rational A matrices for which polyhedron $P = \{x \mid Ax \leq b,\ x \geq 0\}$ is integral for all right hand side vectors b that are divisible by k. In other words, if b is integral then for any vertex v of P, kv is an integral vector. This result is the most important one in this chapter, and one of the main new results of the whole thesis.

3.1.1 1-regular matrices

Theorem 3.1. *Let A be a 1-regular matrix, and b an integral vector. Then the polyhedron $P = \{x \mid Ax \leq b\}$ is integral.*

Proof: Let $F = \{x \mid A'x = b'\}$ be a minimal face of P with a full row rank matrix A'. Then it can be assumed that $A' = [R, S]$ where R is a basis. Thus, $\left(R^{-1}b', 0\right)^T \in F$ and $R^{-1}b'$ is an integral vector, so P is integral. □

We showed in Section 2.2 that Lemma 1.2 cannot be extended to rational matrices. In fact, we gave a 1-regular matrix A for which $[A, I]$ was not 1-regular. As a consequence, we can state only half of Lemma 1.6 for rational matrices. In the proof of the other direction, we exploited the integrality of the matrix.

Lemma 3.2. *Let A be a 1-regular matrix and b an integral vector. Then polyhedron $P = \{x \mid Ax \leq b,\ x \geq 0\}$ is integral, if polyhedron $Q = \{z \mid [A, I]z = b,\ z \geq 0\}$ is integral.*

The analogue of Theorem 1.7, however, is true. The proof we give here is a slight modification of Veinott and Dantzig's [67] proof.

Theorem 3.3. *Let A be a rational full row rank matrix. Then the polyhedron $P = \{x \mid Ax = b,\ x \geq 0\}$ is integral for each integral vector b, if and only if each basis of A has integral inverse.*

Proof: Let us suppose first that each basis of A has an integral inverse, and x^* is a vertex of P. Then the columns of A corresponding to non-zero components of x^* are linearly independent, so we can extend them to a basis B. The non-zero part of x^* equals $B^{-1}b$, which is integral, thus x^* is integral.

Suppose now that $\{x \mid Ax = b,\ x \geq 0\}$ is integral for each integral vector b. Let B be a basis of A. We will show that $B^{-1}e_i$ is an integral vector for each unit vector e_i, which implies the integrality of B^{-1}. Since B is a rational matrix, there exists an integral y such that By is integral and $z = y + B^{-1}e_i \geq 0$. Then $b = Bz$ is integral. Let z' arise from z by adding zero-components corresponding to the colums that are not in B so as to obtain $Az' = b$. Then z' is a vertex of P, so it is integral. Therefore z and $B^{-1}e_i = z - y$ are integral, too. □

Hoffman and Kruskal's Theorem 1.8, is a consequence of Lemma 1.2, Lemma 1.6 and Theorem 1.7. We showed above that the lemmas do not hold for 1-regular matrices, so the rational version of Theorem 1.8 cannot be achieved through them. It can, however, be proved directly.

Theorem 3.4. *Let A be a rational matrix. Then the polyhedron $P = \{x \mid Ax \leq b,\ x \geq 0\}$ is integral for each integral vector b, if and only if A is 1-regular.*

Proof: The theorem can be proved in much the same way as Theorem 3.3 above. Here we outline only the few differences from that proof.

When proving that P is integral for all b vectors if A is 1-regular, observe that each vertex of P is related to a non-singular square submatrix R of A such that $x^* = (R^{-1}b^R, 0)$, where b^R consists of the components of b corresponding to the rows of R.

To see that R^{-1} is integral, generate y and z and define b^R as Rz. For $z' = (z, 0)$, extend b^R to integral vector b so that $Az' \leq b$. □

Theorem 1.9, where lower and upper bounds on x and Ax are present, has a strong practical importance. Unfortunately, it does not remain valid for rational matrices, as we cannot ensure the 1-regularity of $\begin{bmatrix} A \\ I \end{bmatrix}$ if A is a non-integral 1-regular matrix. For example, $(x_1, x_2) = (\frac{1}{2}, 1)$ is a non-integral vertex of the polyhedron defined by the following inequality system

$$\begin{aligned} 0 \leq\ & x_1 + \tfrac{1}{2}x_2 \leq 1 \\ 0 \leq\ & x_1, x_2 \leq 1 \end{aligned}$$

though the matrix $[1, \frac{1}{2}]$ is 1-regular. Hence, the 1-regularity of A is not sufficient to guarantee the integrality of polyhedron $\{x \mid a \leq Ax \leq b, l \leq x \leq u\}$ for all integral vectors l, u, a, b. But is is necessary.

Theorem 3.5. *Let A be a rational matrix. If for all integral vectors l, u, a and b the polyhedron $\{x \mid a \leq Ax \leq b,\ l \leq x \leq u\}$ is integral, then A is 1-regular.*

Proof: If $\{x \mid a \leq Ax \leq b,\ l \leq x \leq u\}$ is integral for all integral vectors l, u, a and b, then $\{x \mid Ax \leq b,\ x \geq 0\}$ is integral for all b, so by Theorem 3.4, A is 1-regular. □

If A is integral, then the converse of Theorem 3.5 is also true, but it is exactly Theorem 1.9, as integral 1-regular matrices are totally unimodular, by Lemma 2.3.

3.1.2 k-regular matrices

Recall that $k\mathbb{Z}^m$ denotes the set of m-dimensional vectors with elements that are integer multiples of k. Moreover, A/k is the matrix obtained from A by dividing every element by k. Lemma 2.13 claims that

$$A \text{ is } k\text{-regular if and only if } A/k \text{ is 1-regular.} \tag{3.1}$$

The proofs of the following theorems rely on the fact that one can divide every element in a linear inequality system by a positive number, leaving the solution space unchanged, i.e.,

$$\{x \mid Ax \leq b\} = \{x \mid (A/k) \leq (b/k)\} \text{ for a } k > 0 \tag{3.2}$$

Using (3.1) and (3.2), we can extend the results given in the previous section for 1-regular matrices to k-regular ones. In the proofs, replace A with A/k and b with b/k. A is a matrix of size $m \times n$ in all theorems.

Theorem 3.6. *Let A be a k-regular matrix, and $b \in k\mathbb{Z}^m$. Then the polyhedron $\{x \mid Ax \leq b\}$ is integral.*

Theorem 3.7. *Let A be a rational full row rank matrix. Then the polyhedron $\{x \mid Ax = b,\ x \geq 0\}$ is integral for each vector $b \in k\mathbb{Z}^m$, if and only if for each basis B of A, kB^{-1} is integral.*

Theorem 3.8. *Let A be a rational matrix. Then the polyhedron $\{x \mid Ax \leq b,\ x \geq 0\}$ is integral for each vector $b \in k\mathbb{Z}^m$, if and only if A is k-regular.*

Theorem 3.9. *Let A be a rational matrix. If the polyhedron $\{x \mid a \leq Ax \leq b,\ l \leq x \leq u\}$ is integral for all vectors $a, b \in k\mathbb{Z}^m$ and $l, u \in k\mathbb{Z}^n$, then A is k-regular.*

Theorem 3.8 is a central result of the thesis. It extends Hoffman and Kruskal's characterization of totally unimodular matrices to k-regular matrices. From the opposite point of view, it describes rational matrices that provide integral polyhedra for any right hand side vector that is an integer multiple of k.

Just as for non-integral 1-regular matrices, the converse of Theorem 3.9 does not hold if A is not integral. For integral matrices, however, one can prove the reverse direction too.

Theorem 3.10. *Let A be an integral matrix. Then the polyhedron $\{x \mid a \leq Ax \leq b,\ l \leq x \leq u\}$ is integral for all vectors $a, b \in k\mathbb{Z}^m$ and $l, u \in k\mathbb{Z}^n$, if and only if A is k-regular.*

Proof: We have only to show that if A is k-regular, then the polyhedron is integral. But this easily follows from Lemma 2.9 and 2.10, which imply that $[A, -A, I, -I]^T$ is k-regular, if A is such. □

In the remainder of the thesis we focus on 2-regular matrices. An alternative way of stating Theorem 3.8 is that 2-regular matrices are the rational matrices that ensure half-integral polyhedra.

Corollary 3.11. *The rational matrix A is 2-regular, if and only if polyhedron $\{x \mid Ax \leq b,\ x \geq 0\}$ has half-integral vertices for all integral vectors b.*

3.2 Balancedness and bicolorability

This short section contains only counterexamples. We show that 2-regularity is not equivalent to balancedness or bicolorability. Our purpose is to show that 2-regular matrices are not disguised versions of matrices belonging to a known class. In other words, k-regularity is a new generalization of total unimodularity.

A 0,1 matrix is *balanced* if it does not contain a square submatrix of odd order with exactly two 1's in each row and column. Balanced matrices were introduced by Berge [7]. For example, a balanced matrix cannot have a submatrix

$$A_1 = \begin{bmatrix} 1 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix} \tag{3.3}$$

Each totally unimodular 0,1 matrix is balanced, but not conversely. The following matrix is balanced, but not totally unimodular.

$$A_2 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix} \text{ with } A_2^{-1} = \begin{bmatrix} -½ & ½ & ½ & ½ \\ ½ & ½ & -½ & -½ \\ ½ & -½ & ½ & -½ \\ ½ & -½ & -½ & ½ \end{bmatrix} \tag{3.4}$$

Note that in Section 6.1.6 we show that A_2 is 2-regular.

Balanced matrices are important because of the following result.

Theorem 3.12. (Berge [7]) *A 0,1 matrix A is balanced if and only if for every submatrix B of A, the polyhedron $\{x \mid Bx = \mathbf{1},\ x \geq 0\}$ is integral.*

The proof of this theorem can be found in [55].

Berge [7] introduced another notion for 0,1 matrices. A 0,1 matrix is *bicolorable* if its columns can be partioned into blue and red columns in such a way that every row with two or more 1's contains a 1 in a blue column and a 1 in a red column. Bicolorability of a matrix is strongly connected to its balancedness.

Theorem 3.13. (Berge [7]) *A 0,1 matrix A is balanced if and only if every submatrix of A is bicolorable.*

Thus balancedness implies bicolorability. Does 2-regularity imply either of them? The answer is in the negative. Matrix A_1 of (3.3) is 2-regular, but it is obviously not balanced, and it is easy to show that it is not bicolorable either. The opposite direction is also blocked. Matrix

$$A_3 = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \end{bmatrix}$$

is balanced, hence bicolorable, but not 2-regular, as its inverse contains quarters (e.g., the elements in rows 5 to 8 and columns 1 to 4 of the inverse are all $\pm\frac{1}{4}$). Figure 3.1 illustrates the relation between 2-regular, balanced and totally unimodular 0,1 matrices. Note that Truemper [63] and Conforti and Cornuéjols [15] generalized balancedness and bicolorability for $0, \pm 1$ matrices. The relation illustrated in Figure 3.1 holds in this more general setting as well.

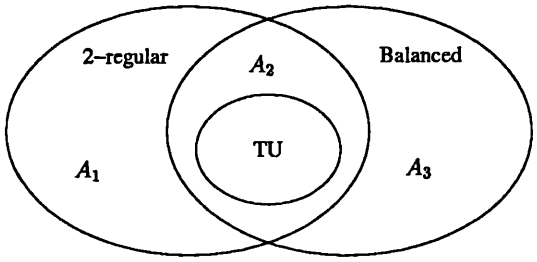


Figure 3.1: The relation between 2-regular, balanced and totally unimodular (TU) matrices

3.3 Total dual half-integrality

If a matrix A is totally unimodular, then the polyhedron $\{x \mid Ax \leq b\}$ is integral for all integral vectors b. If we need integrality for only a given b, and not all the integral ones, then total unimodularity

is too strong a requirement. For example,

$$A = \begin{bmatrix} 1 & 1 \\ 1 & 0 \\ 1 & -1 \end{bmatrix} \tag{3.5}$$

is not a totally unimodular matrix, but the only extreme point of $P = \{x \mid Ax \leq \mathbf{0}\}$ is $(0,0)$, so P is an integral polyhedron. Thus, we need new notions if we are to examine the integrality of polyhedron $P(A,b) = \{x \mid Ax \leq b\}$ for a specific b vector. The notion that proved to be the most useful for this is *total dual integrality.*

Definition 3.14. (Edmonds and Giles [24]) A system of rational inequalities $Ax \leq b$ is called *totally dual integral* (TDI) if for all integral c such that $max\{cx \mid Ax \leq b\}$ is finite, the dual problem $min\{yb \mid yA = c, y \geq 0\}$ has an integral optimal solution.

Note that the definition is about the inequality system, not the polyhedron it defines. For instance,

$$\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} x \leq \begin{pmatrix} 0 \\ 0 \end{pmatrix} \tag{3.6}$$

determines the same polyhedron as $\{x \mid Ax \leq \mathbf{0}\}$ with A of (3.5), but while $Ax \leq \mathbf{0}$ is TDI, the system of (3.6) is not.

Obviously, if A is totally unimodular, then $Ax \leq b$ is TDI for any rational vector b. TDI systems are important, because they define integral polyhedra. The proof of this fact relies on the following result of Edmonds and Giles [24] and Hoffman [43].

Theorem 3.15. *The polyhedron $\{x \mid Ax \leq b\}$ is integral if and only if $max\{cx \mid Ax \leq b\}$ is integral for all integral c for which the maximum is finite.*

In other words, for the existence of an integral optimal solution of $max\{cx \mid Ax \leq b\}$ for any c, it is enough to show that the maximum itself is integral for all integral choices of c.

Now the following theorem is easy.

Theorem 3.16. *If $Ax \leq b$ is TDI and b is integral, then the polyhedron $P(A,b) = \{x \mid Ax \leq b\}$ is integral.*

Note that one can specify the totally dual integral requirement for polyhedra of special format. For example, the system $Ax \leq b$, $x \geq 0$ is TDI, if and only if the dual $min\{yb \mid yA \geq c,\ y \geq 0\}$ has an integral optimum for each integral vector c with finite minimum. More about TDI systems and proofs of the theorems given above can be found in the usual sources, [51] and [55].

In this work we are more interested in half-integral polyhedra than integral ones. The definition of total dual integrality can be changed to accommodate half-integrality.

Definition 3.17. A rational system of inequalities $Ax \leq b$ is *totally dual half-integral* (TDHI) if for all integral c such that $max\{cx \mid Ax \leq b\}$ is finite, the dual problem $min\{yb \mid yA = c,\ y \geq 0\}$ has a half-integral optimal solution.

Theorem 3.18. *If $Ax \leq b$ is TDHI and b is integral, then the polyhedron $P(A,b) = \{x \mid Ax \leq b\}$ is half-integral.*

Proof: $P(A,b)$ is half-integral if and only if $P(A,2b)$ is integral, which is equivalent, by Theorem 3.15, to the fact that for all integral c the optimum in $max\{cx \mid Ax \leq 2b\}$ is integral. This is satisfied if the dual $min\{2yb \mid yA = c,\ y \geq 0\}$ has a half-integral optimal solution, exactly what is required for the total dual half-integrality of $Ax \leq b$. □

Just as with totally unimodular matrices and total dual integrality, if A is 2-regular, then $Ax \leq b$ is TDHI for any rational b. However, the constraint matrix in

$$\begin{bmatrix} 1 & 3 \\ 2 & 0 \\ 1 & 1 \end{bmatrix} x \leq \begin{pmatrix} 2 \\ 1 \\ 1 \end{pmatrix} \tag{3.7}$$

is not 2-regular, but the system is TDHI. To see this, take the dual problem with an integral vector $c = (c_1, c_2)$:

$$\begin{array}{rl} \min & 2y_1 + y_2 + y_3 \\ \text{subject to} & y_1 + 2y_2 + y_3 = c_1 \\ & 3y_1 + y_3 = c_2 \\ & y_1, y_2, y_3 \geq 0 \end{array} \tag{3.8}$$

The basic solutions of the dual are

$$y^{(1)} = \left(0, \frac{c_1 - c_2}{2}, c_2\right), y^{(2)} = \left(\frac{c_2 - c_1}{2}, 0, \frac{3c_1 - c_2}{2}\right), y^{(3)} = \left(\frac{c_2}{3}, \frac{3c_1 - c_2}{6}, 0\right)$$

All three have the same objective value $(c_2 + c_1)/2$. Furthermore, $y^{(1)}$ is feasible if $0 \leq c_2 \leq c_1$, $y^{(2)}$ is feasible if $c_1 \leq c_2 \leq 3c_1$, and $y^{(3)}$ is feasible if $0 \leq c_2 \leq 3c_1$. Therefore, if there is a feasible solution of (3.8), then there is a half-integral optimal solution. Note that the system

$$\begin{bmatrix} 1 & 3 \\ 2 & 0 \end{bmatrix} x \leq \begin{pmatrix} 2 \\ 1 \end{pmatrix}$$

defines the same polyhedron as (3.7), but this latter system is not TDHI, because the only basic solution, $y^{(3)}$, is not necessarily half-integral.

3.4 Chvátal-Gomory cuts

An important problem of integer programming is to find the integer hull P_I of a polyhedron P, i.e, the convex hull of the integer points in P:

$$P_I = conv\{P \cap \mathbb{Z}^n\}$$

Theorem 1.8 can be stated in the following way.

Corollary 3.19. *An integral matrix A is totally unimodular, if and only if $P = P_I$ for $P = \{x \mid Ax \leq b,\ x \geq 0\}$ and any integral b.*

In several theoretical and practical problems, however, the constraint matrix A is not totally unimodular and the right hand side vector b is such that $P \neq P_I$. To tackle these cases and find integer solutions, different methods have been developed. One of the most studied approaches is the cutting plane method, pioneered by Gomory in [37]. The basic concept of the cutting plane method is the *Chvátal-Gomory (CG) cut*, defined as follows. Given an integral $m \times n$ matrix A and an integral vector b, a CG-cut of the polyhedron $P = \{x \mid Ax \leq b\}$ is an inequality of the form

$$\lambda^T Ax \leq \lfloor \lambda^T b \rfloor \text{ where } \lambda \in \mathbb{R}_+^m \text{ and } \lambda^T A \in \mathbb{Z}^n.$$

Note that it is enough to require that $\lambda \in [0,1)^m$ because if we replace λ by $\lambda - \lfloor \lambda \rfloor$, then we get a stronger CG-cut.

The *rank-1 closure* of P is defined as the intersection of P with the half-spaces induced by all possible undominated CG-cuts:

$$P_1 = \{x \in P \mid \lambda^T Ax \leq \lfloor \lambda^T b \rfloor \text{ for } \lambda \in [0,1)^m,\ \lambda^T A \in \mathbb{Z}^n\}$$

Caprara and Fischetti [11] introduced *half-integral Chvátal-Gomory cuts* and the corresponding closure:

$$P_{1/2} = \{x \in P \mid \mu^T Ax \leq \lfloor \mu^T b \rfloor \text{ for } \mu \in \{0, 1/2\}^m,\ \mu^T A \in \mathbb{Z}^n\}$$

Obviously, $P_I \subseteq P_1 \subseteq P_{1/2} \subseteq P$. It is also known that $P = P_1$ holds if and only if $P = P_I$. This does not remain valid for half-integral cuts, for example $P = P_{1/2}$ for any $b \in 2\mathbb{Z}^m$ but this does not imply that $P = P_I$. It is worth mentioning that Caprara and Fischetti [11] showed that $P_{1/2}$ can replace P_I in Corollary 3.19, i.e., a $0, \pm 1$ matrix A is totally unimodular if and only if $P = P_{1/2}$ for $P = \{x \mid Ax \leq b,\ x \geq 0\}$ and all $b \in \mathbb{Z}^m$.

In this section we will examine the case where $P_1 = P_{1/2}$. This occurs, for example, when A is

an integral $m \times n$ matrix satisfying

$$\sum_{i=1}^{m} |a_{ij}| \leq 2 \quad \text{for } j = 1, \ldots, n,$$

according to Edmonds [22] and Edmonds and Johnson [25]. In this case $P_{1/2} = P_1 = P_I$. Gerards and Schrijver dealt with the transpose, i.e., integral A matrices of size $n \times m$ satisfying

$$\sum_{j=1}^{m} |a_{ij}| \leq 2 \quad \text{for } i = 1, \ldots, n, \tag{3.9}$$

and proved that $P_{1/2} = P_1$ for these matrices too. In this case, however, P_1 is not necessarily equal to P_I. (In Section 7.3 we will discuss the characterization of matrices that satisfy (3.9) and ensure $P_1 = P_I$.) Here we show that it is the 2-regularity of the matrices in these examples that ensures $P_{1/2} = P_1$.

Theorem 3.20. *Let A be a $m \times n$ integral matrix and $P = \{x \mid Ax \leq b,\ x \geq 0\}$. A is 2-regular if and only if $P_{1/2} = P_1$ for all $b \in \mathbb{Z}^m$.*

Proof: Let us suppose first that $P_1 = P_{1/2}$ for all $b \in \mathbb{Z}^m$, and let $b \in 2\mathbb{Z}^m$. Then $P = P_{1/2} = P_1$ which implies $P = P_I$ for all such b. By Theorem 3.8 this means that A is 2-regular.

Assume now that A is 2-regular. We will use the notations $\hat{A} = \begin{bmatrix} A \\ -I \end{bmatrix}$ and $\hat{b} = \begin{bmatrix} b \\ 0 \end{bmatrix}$. Obviously, $P = \{x \mid \hat{A}x \leq \hat{b}\}$, and by Lemma 2.10, $\hat{A}$ is 2-regular. Let us take an arbitrary CG-cut of P with $\lambda \in [0,1)^{m+n}$. The maximum of the linear program $max\{\lambda^T \hat{A}x \mid \hat{A}x \leq \hat{b}\}$ is at most $\lambda^T \hat{b}$ so the dual problem $min\{\mu^T \hat{b} \mid \mu^T \hat{A} = \lambda^T \hat{A},\ \mu \geq 0\}$ has an optimal solution μ. Since $\lambda^T \hat{A}$ is integral, it follows from Corollary 3.11 that μ can be chosen to be half-integral. For this half-integral dual optimal solution $\mu^T \hat{A} = \lambda^T \hat{A}$, and $\mu^T \hat{b} \leq \lambda^T \hat{b}$ because λ is dual feasible, so the CG-cut $\lambda^T \hat{A}x \leq \lfloor \lambda^T \hat{b} \rfloor$ is dominated by $\mu^T \hat{A}x \leq \lfloor \mu^T \hat{b} \rfloor$. We can further stregthen the cut by replacing μ with $\mu - \lfloor \mu \rfloor$, keeping its half-integrality. We showed that any CG-cut is dominated by a half-integral cut, so $P_{1/2} \subseteq P_1$, which is equivalent to $P_{1/2} = P_1$. □

Caprara, Fischetti and Letchford [12] extended half-integral cuts to *mod-k cuts*. These are CG-cuts where

$$\lambda \in \left\{0, \frac{1}{k}, \ldots, \frac{k-1}{k}\right\}^m \tag{3.10}$$

The corresponding closure is defined as

$$P_{\text{mod-}k} = \{x \in P \mid \lambda^T Ax \leq \lfloor \lambda^T b \rfloor \text{ for } \lambda \text{ satisfying (3.10) and } \lambda^T A \in \mathbb{Z}^n\}$$

For applications of mod-k cuts, see [12]. In much the same way as Theorem 3.20, we can prove the more general result:

Theorem 3.21. *Let A be a $m \times n$ integral matrix and $P = \{x \mid Ax \leq b, x \geq 0\}$. A is k-regular if and only if $P_1 = P_{\text{mod-}k}$ for all $b \in \mathbb{Z}^m$.*

Chapter 4

Bidirected graphs

In this chapter we describe bidirected graphs. They provide a common generalizations of both directed and undirected graphs, containing ordinary edges, loops and half-edges. The edges can be directed in more possible ways than in a directed graph. In a directed graph, if an end-node of an edge is its tail, then the other end-node of the edge must be its head. Undirected graphs can be viewed as graphs in which each edge has two heads. In a bidirected graph, the end-nodes of the edges can be heads or tails, independently from each other.

The notions defined for undirected or directed graphs can be extended to bidirected graphs. For example, we define operations on bidirected graphs, such as the deletion of nodes, edges, or the contraction of an edge. More importantly, we introduce the incidence matrix of a bidirected graph. The node-edge incidence matrix of a simple undirected graph is a 0,1 matrix with exactly two 1's in each column. A directed graph is represented by a $0, \pm 1$ matrix in which each column has one $+1$ and one -1. If there are loops in the graphs, then they can be represented by columns with a single non-zero. The node-edge incidence matrix of a bidirected graph is a matrix with elements $0, \pm 1, \pm 2$ such that a column can have at most two non-zeros, and if it has two, then they are both ± 1. The operations on graphs have their parallel operations on incidence matrices.

Bidirected graphs serve as a background for introducing binet matrices in the next chapter, and incidence matrices provide the link between them. To this end, we examine the subdeterminants of an incidence matrix in Section 4.2. Our main interest is in the bases of an incidence matrix, that is why we give a characterization of non-singular submatrices of the node-edge incidence matrix of a bidirected graph. We also prove that incidence matrices of bidirected graphs are 2-regular, thus establishing the connection of the remainder of the thesis with what has been described in the previous chapters.

The content of this chapter is not new. Bidirected graphs were introduced by Edmonds in [23], and since then they have appeared in the literature several times. For example, Schrijver [56] gave a necessary and sufficient condition for the existence of an integer solution to a linear inequality system

$Ax \leq b$ in which A is the edge-node incidence matrix of a bidirected graph. Gerards and Schrijver [33] characterized bidirected graphs which lead to matrices with strong Chvátal rank 1. We will give more details and extensions of this latter result in Section 7.3. Bidirected graphs are strongly related to signed graphs, introduced by Harary [38]. Chapter 10 describes signed graphs and other graphs related to bidirected graphs.

The structure of non-singular submatrices of incidence matrices is also known. For example, in Chapter 9 we describe generalized networks and, in Lemma 9.2, we characterize the bases of their incidence matrices. We will see that the structure of the bases is very similar to what appears in this chapter. In fact, bidirected graphs can be considered to be special generalized networks, so results about generalized networks imply those about bidirected graphs.

4.1 Basic notions

A bidirected graph $G(V, E)$ on node set V and with edge set E can have four types of edges: a *link*, written $e{:}uv$, has two distinct end-nodes (u and v); the end-nodes of a *loop*, $e{:}uu$, coincide; a *half-edge*, $e{:}u$, has one end-node, the other end is not connected to any node; and a *loose edge*, $e{:}\emptyset$ has no end-nodes at all. The edges are signed with $+$ or $-$ at their end-nodes. That is, links can be signed with $+-$, $++$ or $--$; loops, as their end-nodes coincide, are either $++$ or $--$; half-edges have only one sign, $+$ or $-$. Links with different signs at their two end-nodes are called *directed edges*, all other edges are called *bidirected*. If a bidirected edge is signed with $++$, then we call it a *positive edge*, edges signed with $--$ are *negative*. If an edge is signed with $+$ at an end-node, then this node is an *in-node* or *head* of the edge. An end-node signed with $-$ is called the *out-node* or *tail* of the edge.

Figure 4.1 shows two possible graphical representations of the same bidirected graph. Heads and tails can be represented with arrows as in (i), or signs as in (ii). All edges other than e_1 are bidirected. Edge e_1 is a directed link, e_2 and e_3 are bidirected links, e_5 and e_6 are loops, e_4 and e_7 are half-edges. Loose edges are not depicted. In the remainder of the thesis we will use arrows as in Figure 4.1(i).

A *walk* in a bidirected graph is a sequence $(v_1, e_1, v_2, e_2, \ldots, e_{t-2}, v_{t-1}, e_{t-1}, v_t)$ where v_i and v_{i+1} are end-nodes of edge e_i ($i = 1, \ldots, t-1$), including the case where $v_i = v_{i+1}$ and e_i is a half-edge or a loop. If $v_1 = v_t$, then the walk is *closed*. If the walk consists of only links, and it does not cross itself, i.e. $v_i \neq v_j$ for $1 < i < t$, $1 \leq j \leq t$, $i \neq j$, then it is a *path*. A closed walk which does not cross itself is called a *cycle*. That is, a cycle can be a loop, a half-edge or a closed path. If the number of bidirected edges in a cycle is odd, then it is called an *odd cycle*, otherwise it is an *even cycle*. Obviously, a loop or a half-edge always makes an odd cycle. Two links forming a cycle are *parallel*. A bidirected graph is *connected*, if there is a path between any two nodes. A *tree* is a connected graph which does not contain a cycle. A connected graph containing exactly one cycle is called a *1-tree*. This name is justified by the fact that a 1-tree consists of a tree and one additional

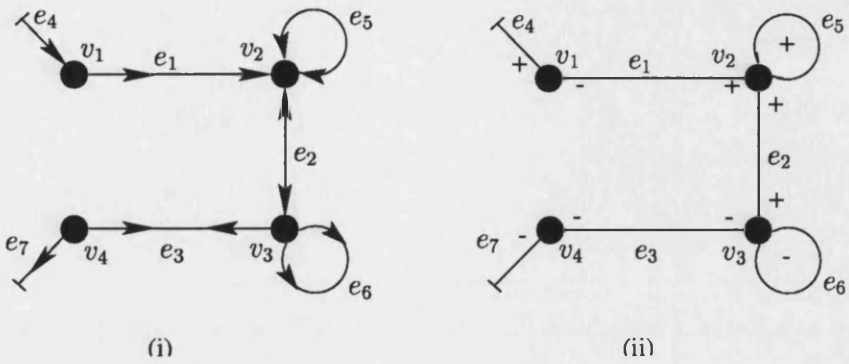


Figure 4.1: Possible graphical representations of a bidirected graph

edge. If the unique cycle in a 1-tree is odd, then we will call it an *odd 1-tree*.

One can straightforwardly define the *node-edge incidence matrix* $A(G)$ of a bidirected graph G. Let $a_e = (a_{ve} \mid v \in V)$ be a column of $A(G)$ corresponding to edge $e \in E$. The non-zero entries in this column, being ± 1 or ± 2, are in the rows corresponding to the end-nodes of e. The sign of the entry depends on whether the corresponding end-node is an in-node or an out-node of the edge. That is, the non-zeros in a_e are as follows:

if e:uv is a link, then $a_{ue} = \pm 1$, $a_{ve} = \pm 1$
if e:uu is a loop, then $a_{ue} = \pm 2$
if e:u is a half-edge, then $a_{ue} = \pm 1$.

If e:$\emptyset$ is a loose edge, then all the entries in the column a_e are zero. As an example, the incidence matrix of the bidirected graph depicted in Figure 4.1 is

$$A = \begin{array}{c|ccccccc} & e_1 & e_2 & e_3 & e_4 & e_5 & e_6 & e_7 \\ \hline v_1 & -1 & 0 & 0 & 1 & 0 & 0 & 0 \\ v_2 & 1 & 1 & 0 & 0 & 2 & 0 & 0 \\ v_3 & 0 & 1 & -1 & 0 & 0 & -2 & 0 \\ v_4 & 0 & 0 & -1 & 0 & 0 & 0 & -1 \end{array} \tag{4.1}$$

The edge-node incidence matrix of a bidirected graph is, of course, the transpose of the node-edge incidence matrix. When we write simply incidence matrix, then we mean the node-edge incidence matrix.

The relation of incidence matrices and bidirected graphs is two-way (or bidirectional). Given an $m \times n$ integral matrix $A = (a_{ij})$ satisfying

$$\sum_{i=1}^{m} |a_{ij}| \leq 2 \quad \text{for } j = 1, \ldots, n, \tag{4.2}$$

we can find a bidirected graph $G(A)$ with m nodes and n edges such that its node-edge incidence matrix is A. In other words, property (4.2) characterizes the node-edge incidence matrices of bidirected graphs. In what follows we identify the columns (rows) of $A(G)$ with the edges (nodes) of G, so that e can refer to either an edge of the graph or a column of the matrix.

Operations on $A(G)$ which maintain (4.2) can be translated to operations on bidirected graphs. Such operations are, for example, multiplying a row or a column with -1, or deleting a row or a column. It can be verified that the following transformation (which appeared in [33]) also maintains (4.2):

$$A = \begin{bmatrix} \alpha & c \\ b & D \end{bmatrix} \longrightarrow A' = D - \alpha bc \tag{4.3}$$

where α is a non-zero entry, b is a column vector, c is a row vector, and D is a submatrix of A. Note that if α is the only non-zero entry in the first column of A, then $b = 0$ so that $A' = D$. If α is not the only non-zero entry, then both α and the only other non-zero are ± 1. Thus αbc is a matrix with only one non-zero row equivalent to $\pm c$. Now $c_j = \pm 2$ implies $D_{ij} = 0$ for all i, because of (4.2), therefore A' is bound to satisfy (4.2).

We now give the graphical equivalents of the matrix operations. They are extensions of standard operations on directed or undirected graphs, such as edge or node deletion, and edge contraction, taking into account the signs of the edges.

When multiplying a column with -1 in A, we simply change the sign(s) at the end-node(s) of the corresponding edge. Directed edges remain directed with the opposite direction, positive edges become negative and vice versa. We call this operation *reversing the direction* of an edge. Multiplying a row with -1 means changing the signs of the incident edges at the node corresponding to the row. If the node was an in-node of an edge, then it becomes an out-node, and vice versa. Consequently, incident directed links become bidirected and bidirected links become directed. We call this operation *switching* at a node.

Column deletion easily translates to bidirected graphs, it is equivalent to deleting an edge from the graph. Deletion of a row corresponds to the removal of the related node together with the edge-ends incident to the node. That is, links connected to this node become half-edges, loops and half-edges located at the deleted node become loose edges. All other edges and nodes remain unchanged. We call this operation *deleting a node*. Figure 4.2 shows an example where node v_3 of the graph in Figure 4.1 is deleted.

Finally, the transformation defined by (4.3) translates to bidirected graphs as follows. Let us suppose that the first row and column of A in (4.3) correspond to node u and edge $e{:}uv$, respectively. Then, when applying the transformation on $G(A)$, the resulting graph, $G(A')$ has one less node and one less edge. We call this operation *contracting* edge e together with node u. For the different kinds of edges the operation has to be defined carefully, because the case when α is the only non-zero in column e (i.e. e is a loop or a half-edge) leads to $A' = D$, while e being a link leads to a more

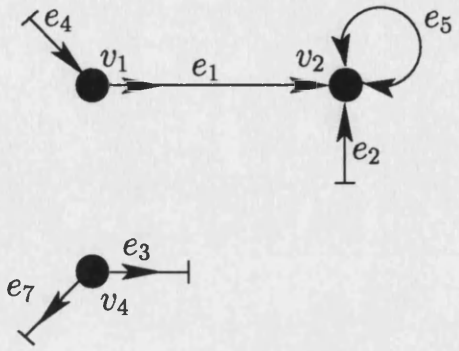


Figure 4.2: Deleting node v_3 from the graph in Figure 4.1

complex situation.

If e is a loop or a half-edge at u, then u and e are removed from the graph, the incident loops and half-edges at u other than e become loose edges, and the incident links lose one end-node and become half-edges. Thus, contracting a loop or a half-edge located at u differs from deleting node u in only one respect, namely that one less loose edge (corresponding to e) is created. As loose edges are not shown in diagrams in any case, there is no difference in the graphical representation of contracting a loop or a half-edge with u or deleting node u.

If e is a directed link, then we get an operation similar to the ordinary graph contraction. Here we remove the edge e and the end-node u. The edges connected to u in $G(A)$ will be connected to v in $G(A')$, directed edges parallel to e will become loose edges, while parallel bidirected edges in $G(A)$ will become loops at v in $G(A')$. Figure 4.3(i) shows the effect of contracting e_1 with node v_1 from the graph in Figure 4.1. Finally, if e is a bidirected link, then contracting it with u means first switching at u, and then contracting the now directed edge e, as described above. Figure 4.3(ii) shows the contraction of e_3 with v_3 from the graph in Figure 4.1. The incidence matrices corresponding to the graphs in Figure 4.3(i) and (ii) are as follows:

	e_2	e_3	e_4	e_5	e_6	e_7
v_2	1	0	1	2	0	0
v_3	1	-1	0	0	-2	0
v_4	0	-1	0	0	0	-1

and

	e_1	e_2	e_4	e_5	e_6	e_7
v_1	-1	0	1	0	0	0
v_2	1	1	0	2	0	0
v_4	0	-1	0	0	2	-1

These matrices can be obtained by applying (4.3) on matrix A of (4.1) with $\alpha = a_{11}$ and $\alpha = a_{33}$, respectively.

Remark 4.1. Contracting a bidirected link $e{:}uv$ with u and contracting it with v result in different graphs. But the two graphs differ only in the sign of the remaining end-node's row, which can be easily eliminated by a multiplication with -1, i.e. a switching at the node. For example, contracting

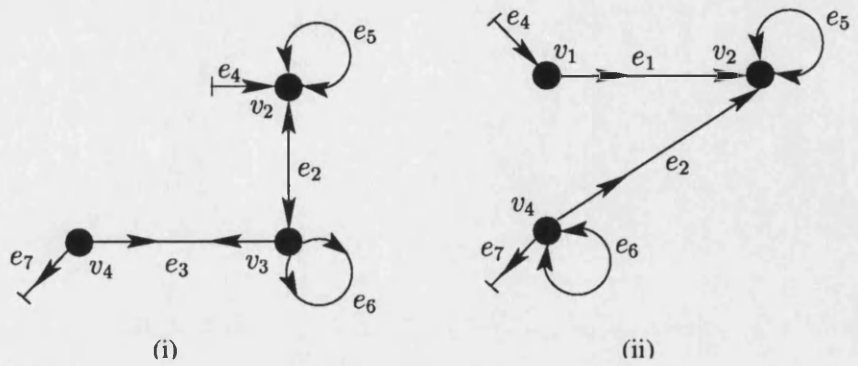
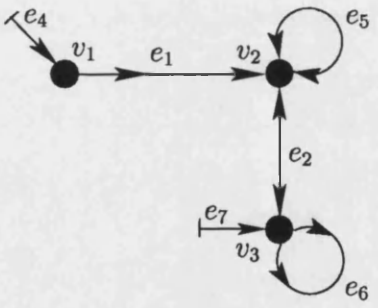


Figure 4.3: Contracting edge e_1 with node v_1 and edge e_3 with node v_3 in the graph in Figure 4.1

edge e_3 with v_4 from the graph in Figure 4.1 leads to the graph and incidence matrix shown in Figure 4.4. When it does not cause problems, e.g. the signs of the edges incident to u are not relevant, we will use the shorter 'contracting edge e' instead of the full 'contracting e with u'.



	e_1	e_2	e_4	e_5	e_6	e_7
v_1	-1	0	1	0	0	0
v_2	1	1	0	2	0	0
v_3	0	1	0	0	-2	1

Figure 4.4: Contracting edge e_3 with v_4 in the graph in Figure 4.1

Let R be a submatrix of the incidence matrix $A(G)$. It can be obtained by row and column deletions. By the analogous operations, a bidirected graph $G(R)$ can be obtained from G. It is clear that $G(R)$ does not depend on the order of the row and column deletions. This graph can be achieved by edge and node deletions, but strictly speaking it is not a subgraph of G, as it can contain half-edges and loose edges that are not present in the original edge set E. However, when it does not create confusion, we will call it a subgraph of G.

If $A(G)$ is not of full row rank, then by deleting rows it can be made a full row rank matrix. For the sake of simpler statements, and without loss of generality, in what follows we usually assume that the node-edge incidence matrix of a bidirected graph is always of full row rank.

A bidirected graph G is connected, if and only if its node-edge incidence matrix cannot be decomposed. Sometimes we will call a non-decomposable matrix $A(G)$ *connected* in this context. In what follows, we will mainly deal with connected bidirected graphs and incidence matrices. So

unless we specifically claim the contrary, we always assume bidirected graphs to be connected.

4.2 Determinants

In this section we deal with the determinants of incidence matrices of bidirected graphs. First we characterize non-singular submatrices by means of their graphical representation, then prove that the inverses of these submatrices are half-integral, i.e., the incidence matrices of bidirected graphs are 2-regular. This extends the well-known result about the total unimodularity of the incidence matrices of directed graphs.

Let us start with two propositions.

Proposition 4.2. *Let T be a square matrix of size n whose non-zero elements are $T_{ii} = 1$ for $i = 1, \ldots, n$; $T_{i+1,i} = -1$ for $i = 1, \ldots, n-1$ and $T_{1n} = \pm 1$. That is, T is of the following form*

$$T = \begin{bmatrix} 1 & & & \pm 1 \\ -1 & 1 & & \\ & -1 & \ddots & \\ & & \ddots & 1 \\ & & & -1 \quad 1 \end{bmatrix} \tag{4.4}$$

Then $det(T) = 0$ if $T_{1n} = -1$, and $det(T) = 2$ if $T_{1n} = 1$.

Proof: By cofactor expansion, $det(T) = 1 + (-1)^{n+1} T_{1n} (-1)^{n-1}$. □

Proposition 4.3. *Switchings at nodes do not change the parity of a cycle.*

Proof: Switchings at nodes that are not part of a cycle obviously do not have any effect on the cycle. If a cycle is a loop or a half-edge, then a switching at its node changes only the sign of the edge, the cycle remains odd. Otherwise, if the cycle consists of links, there are exactly two of them incident to any node of the cycle. Switching at such a node changes the incident directed links to bidirected ones and vice versa. Thus, if there were two bidirected or two directed links incident to the node, then the number of bidirected edges in the cycle changes by 2. If one incident link was directed and the other bidirected, then it remains so after switching. Therefore, the number of bidirected edges in the cycle changes by an even number at any switching, and the parity of the cycle remains the same. □

Lemma 4.4. *Any square submatrix R of the incidence matrix A of a bidirected graph G is non-singular if and only if each connected component of $G(R)$ is an odd 1-tree.*

Proof: If R is a 1×1 matrix, then the theorem is trivial as a loop or a half-edge is an odd 1-tree by definition. Let us assume that R is of size m, and the theorem is true for submatrices smaller than m.

If R is not connected, then it is non-singular if and only if it is decomposable to non-singular blocks, so the theorem follows by induction. Thus, let us assume that R is connected.

Since R is a square submatrix, $G(R)$ has as many edges as nodes. Because it is connected, it contains exactly one cycle, so it is a 1-tree. We now show that is must be odd, that is, its cycle is an odd cycle. Let R_1 be the submatrix of R corresponding to the cycle (i.e., its rows and columns correspond to the nodes and edges of the cycle), and R_2 be the submatrix of the non-cycle nodes and edges. That is, if the first rows and columns correspond to the nodes and edges of the cycle, then R has the following form:

$$R = \begin{bmatrix} R_1 & * \\ & R_2 \end{bmatrix}$$

$G(R_2)$ can be obtained from G by deleting the nodes and edges of the cycle, so all non-cycle edges incident to cycle nodes will be half-edges in $G(R_2)$, the other non-cycle edges remain unchanged. Consequently, each component of $G(R_2)$ is an odd 1-tree. As the size of any of these components is less then m, by induction the singularity of R depends on the singularity of R_1.

If $G(R_1)$ is a loop or a half-edge, then R_1 is non-singular. Else, by permuting its rows and columns and multiplying them by ± 1, R_1 can be transformed to R_1' which is of form (4.4). R_1' is non-singular if and only if R_1 is such.

By Proposition 4.2, R_1' is non-singular if and only if the element standing in its upper right hand corner is +1, i.e., if $G(R_1')$ contains exactly one bidirected edge. We achieved R' from R by permutations and multiplying some rows and columns by -1. Multiplying a row by -1 is equivalent to a switching, which keeps the parity of the cycle. Permutations or multipying columns with -1 do not change the number of bidirected edges. It follows that R is non-singular if its cycle contains an odd number of bidirected edges, and singular if it contains an even number of them. □

If a connected bidirected graph contains an odd cycle, then we can extend the cycle to a spanning odd 1-tree by adding new edges. That is, in this case we have a spanning submatrix R such that its incidence matrix is non-singular. The case where a bidirected graph does not contain an odd cycle is handled by the following lemma.

Lemma 4.5. *If there are no odd cycles in a bidirected graph G, then it can be transformed to a directed graph by switchings.*

Proof: First, there cannot be loops or half-edges in G, as they are odd cycles themselves. We can clearly suppose without loss of generality that G is connected. So there is a spanning tree in G. By switching on the nodes, the tree edges can be changed to directed links. If the graph now contains a bidirected link e, then the unique cycle formed by e and some tree-edges is an odd cycle, as it contains one bidirected edge, e itself. This contradicts the condition, taking into account that switchings preserve the parity of a cycle. □

Combining these observations, we have the following corollary.

Corollary 4.6. *Let G be a connected bidirected graph on m nodes and A be its node-edge incidence matrix. Then $rank(A)$ is either m or $m-1$, and $rank(A) = m-1$ if and only if G does not contain an odd cycle.*

We can state another easy consequence of Lemma 4.4.

Corollary 4.7. *Let A be a full row rank node-edge incidence matrix of a bidirected graph, and T a collection of linearly independent columns of A. Then each connected component of $G(T)$ is either an isolated node, or forms a tree or an odd 1-tree. Conversely, if any component of a subgraph $G(R)$ is either an isolated node or forms a tree or an odd 1-tree, then the columns of R are linearly independent.*

Proof: The proof in both directions relies on the fact that if the columns of T are linearly independent, then it can be extended to a basis R by adding columns of A. Then $G(T)$ arises from $G(R)$ by deleting edges. By Lemma 4.4, the connected components of $G(R)$ are odd 1-trees, so $G(T)$ can have only components listed in the statement. Note that isolated nodes correspond to all-zero rows of T. □

The following theorem establishes the connection of bidirected graphs to k-regularity.

Theorem 4.8. *The incidence matrix A of a bidirected graph is 2-regular.*

Proof: Let R be a non-singular submatrix of A. Clearly, it is enough to consider connected submatrices, because if R is decomposable into blocks and each block can be shown to have half-integral inverse, then the inverse of R is half-integral too. So let R be a connected, non-singular submatrix of size l. We will show that $det(R) = \pm 1$ or ± 2. Then Lemma 2.7 ensures that A is 2-regular.

As R is a submatrix of A, the total number of non-zero elements in R cannot exceed $2l$ because each column of A has at most two. Since R is non-decomposable and non-singular, the total number of non-zero elements in R must be at least $2l-1$. To see this consider the following bipartite graph, called the adjacency graph of R. It has $2l$ nodes – one for each row and column – and an edge joining a row node i to a column node j if and only if $R_{ij} \neq 0$. R is non-decomposable if and only if its adjacency graph is connected, i.e., it has at least $2l-1$ edges.

If R has $2l$ non-zero elements, then it can be transformed to the form of (4.4), so by Proposition 4.2, $|det(R)| = 2$.

Consider the case where R has $2l-1$ non-zero elements. Its adjacency graph is a tree graph. So the determinant of R can be obtained by expansion through a row or a column containing only one non-zero entry. It is fairly obvious that the absolute value of the determinant is the absolute value of the product of l non-zero elements of R. Since there is only one column that has less than two non-zeros, R contains at most one ± 2 element. Thus the product of any subset of the non-zero elements is ± 1 or ± 2. Hence $|det(R)| = 1$ or 2 and the theorem follows. □

As incidence matrices of bidirected graphs are exactly the matrices that satisfy (4.2), an alternative way of stating Theorem 4.8 is that if A is an integer matrix satisfying (4.2), then A is 2-regular. This fact has appeared in the literature several times. For example, it is mentioned in Gerards and Schrijver [33]. Hochbaum et al. [42] and Lee [48] prove that $det(R) = \pm 1$ or ± 2 for connected, non-singular submatrices, which is the main point of our proof.

Chapter 5

Binet matrices

This chapter is central in the thesis. It describes the generalization of network matrices for bidirected graphs.

Network matrices can be defined in two equivalent ways. The graphical definition starts with a connected directed graph with a given spanning tree in it. The rows and columns of the network matrix are associated with the tree and non-tree edges, respectively. For any non-tree edge s, we find the unique cycle (called the *fundamental cycle*) which contains s and some edges from the tree. The column of the network matrix corresponding to s will contain ± 1 in the rows of the tree edges in its fundamental cycle and 0 elsewhere. The signs of the non-zeros depend on the directions of the edges. If walking through the tree along the fundamental cycle starting at the tail of s, a tree edge lies in the same direction, it gets a positive sign, if it lies in the opposite direction, it gets a negative sign.

In the algebraic derivation of the network matrices, the incidence matrix A of the directed graph is used. To make it full row rank, an arbitrary row is deleted. Every basis in this full row rank matrix A' corresponds to a spanning tree in the graph. If basis R is associated with the given spanning tree, and we denote the remaining part of A' as S, then the network matrix equals $R^{-1}S$.

We apply these methods to bidirected graphs to get the bidirected analogue of network matrices, the *binet matrices*[1]. We define binet matrices in the algebraic way but, in parallel with network matrices, we also provide an algorithm to determine the columns of a binet matrix using its graphical representation. This algorithm will be used substantially in establishing the properties of binet matrices. Similarly to network matrices, if the graphical definition were not available, then the analysis of binet matrices would be more cumbersome. We will see examples in Section 5.2 that exhibit the difference between the elegant graphical ideas and the clumsy matrix explanations. Except for these illustrative points, we will use almost exclusively the graphical representation of binet matrices in

[1]The term binet is used here as a short form for *bi*directed *net*work, but by coincidence it also matches the name of Jacques Binet (1786-1856) who worked on the foundations of matrix theory and gave the rule of matrix multiplication.

the whole thesis. That is why we will spend a relatively large part of the chapter on explaining the graphical method of deriving binet matrices. Most of Section 5.1 is devoted to this task.

Section 5.2 deals with fundamental properties of binet matrices. We show that a wide range of operations on binet matrices maintain its binetness. We give necessary conditions for binet matrices in Section 5.3. Finally, we describe our attempts at finding a recognition algorithm for binet matrices in Section 5.4. The result contained in the last two sections might prove useful in yielding a characterization for binet matrices, a task which has not been accomplished.

5.1 Definition and graphical representation

Definition 5.1. Let A be a full row rank incidence matrix of a bidirected graph G, R be a basis of it and $A = [R, S]$. The matrix $B = R^{-1}S$ is called a *binet matrix.*

Subgraph $G(R)$ is called the *basis of the graph,* and its edges are called the *basic edges.* The edges of G that are not in the basis (i.e., those of $G(S)$) are the *non-basic edges.* By Lemma 4.4, a basis of a bidirected graph has odd 1-tree components. The unique cycles in the *basic components* are called *basic cycles.* We call bidirected graph $G(A)$ the *binet representation* of binet matrix B. When in a bidirected graph representing a binet matrix, the basic and non-basic edges are clearly indicated, then we call it a *binet graph.*

The same binet matrix may arise from different incidence matrices, i.e., it may have different binet representations. For example, the two binet graphs in Figure 5.1 give two possible representations of the following binet matrix:

$$B = \begin{array}{c|ccc} & s_1 & s_2 & s_3 \\ \hline r_1 & 1 & 0 & 1 \\ r_2 & 1 & 1 & 0 \\ r_3 & 0 & 1 & 1 \end{array} \tag{5.1}$$

Note that we identify the rows and columns of B with basic and non-basic edges, a technique we will use throughout this dissertation. The incidence matrices of the binet graphs in Figure 5.1 are as follows.

	r_1	r_2	r_3	s_1	s_2	s_3
v_1	1	0	0	1	0	1
v_2	0	1	0	1	1	0
v_3	0	0	1	0	1	1

and

	r_1	r_2	r_3	s_1	s_2	s_3
v_1	0	1	-1	1	0	-1
v_2	-1	1	1	0	2	0
v_3	1	0	0	1	0	1

That the graphs in Figure 5.1 really represent B can be checked by taking the inverse of the basis $R = [r_1, r_2, r_3]$ and multiplying it with $S = [s_1, s_2, s_3]$ in both incidence matrices. In what follows,

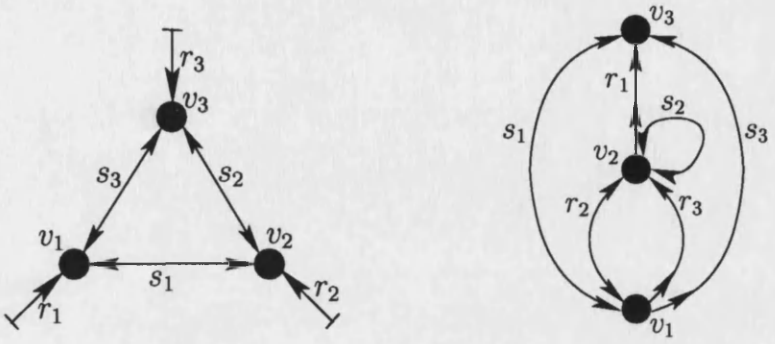


Figure 5.1: Different binet representations of binet matrix (5.1)

we give a graphical method to obtain a binet matrix from its binet graph. This spares us the need to take the inverse of the basis and makes handling binet matrices much easier.

Let A have m rows and n columns. Let s be a non-basic edge, in other words, column s of the node-edge incidence matrix is in S. Let the corresponding column of B be w, i.e. $w = R^{-1}s$. The column vectors of the $m \times m$ non-singular matrix R span an m-dimensional Eucledian vectorspace, $\mathbb{R}^m$. As $Rw = s$, column w represents the unique coordinates of vector s in this basis. Let $\{r_1, r_2, \ldots, r_t\}$ be the subset of columns of R where the coordinates are non-zero:

$$s = w(r_1)r_1 + w(r_2)r_2 + \cdots + w(r_t)r_t, \quad w(r_i) \neq 0, \ i = 1, \ldots, t \tag{5.2}$$

So by setting $s = r_0$ and defining $w(s) = w(r_0) = -1$, we get

$$\sum_{i=0}^{m} w(r_i)r_i = 0 \tag{5.3}$$

The columns of the matrix $R' = [s, r_1, r_2, \ldots, r_t]$ form a minimal dependent set in $\mathbb{R}^m$. That is, the columns of R' are dependent, and deleting any column from R', we end up with matrices the columns of which are linearly independent. This implies the following lemma, illustrated in Figure 5.2.

Lemma 5.2. *The graph $\widehat{G}(R')$ spanned by edges $s, r_1, \ldots, r_t$ falls in one of the following three categories.*

(i) It is an even cycle, or

(ii) it is a graph consisting of two node-disjoint odd cycles connected with a path which has no common node with the cycles except its end-nodes, or

(iii) it is a graph consisting of two odd cycles which have exactly one common node.

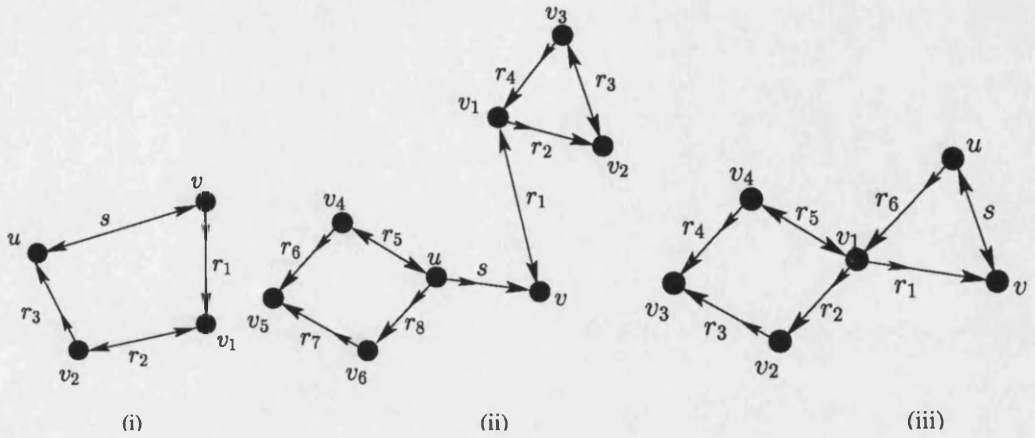


Figure 5.2: Examples of minimal dependent subgraphs

Proof: First note that $\widehat{G}(R')$ is connected. If it is not, then R' is decomposable, and s is spanned by the columns that are in the same submatrix as s. The other columns have zero coordinates, contradicting (5.2).

The columns of R' are minimally dependent, so according to Corollary 4.7, $\widehat{G}(R')$ is not a tree or an odd 1-tree, but every connected component of the bidirected graphs obtained from $\widehat{G}(R')$ by deleting any edge r is either a tree or an odd 1-tree. (Isolated edges are ruled out as $\widehat{G}(R')$ is spanned by edges.) Now let us have a closer look at the possible cases.

If the deleted edge r is a cut-edge in $\widehat{G}(R')$, i.e., the graph obtained by deleting it has two components, then both components must be an odd 1-tree, otherwise by redrawing r we would get a tree or an odd 1-tree. This corresponds to category (ii).

If r is not a cut-edge, i.e., the graph G' obtained by deleting r is connected, then G' is either a tree or an odd 1-tree. If it is a tree, then redrawing r the graph (which is $\widehat{G}(R')$) contains exactly one cycle, which cannot be odd. There cannot be any non-cycle edge in $\widehat{G}(R')$, because then deleting that edge would not result in a tree or odd 1-tree graph. So $\widehat{G}(R')$ is an even cycle, leading to category (i).

If r is not a cut-edge and G' is an odd 1-tree, then by redrawing r we get at least one new cycle. The new cycle must be odd, otherwise we could delete an edge from the old odd cycle, and the resulting graph would not be a tree or an odd 1-tree. So every cycle in $\widehat{G}(R')$ is odd. If there are two edge-disjoint cycles, then we have category (ii) or (iii). If the cycles are not edge-disjoint, then $\widehat{G}(R')$ can be viewed as a graph in which there are three internally node-disjoint paths connecting the same pair of nodes, that is a theta graph. (See Figure 1.1). It is easy to see that all three cycles in a theta graph cannot be odd, so this case is impossible, and the proof is completed. □

We will call graphs in categories (ii) and (iii) in Lemma 5.2 and illustrated in Figure 5.2, *hand-*

cuffs. In parallel with the term 'fundamental cycle' in case of network matrices, we will use the phrase *fundamental circuit of* s for $\widehat{G}(R')$. Thus, Lemma 5.2 claims that every fundamental circuit is either an even cycle or a handcuff.

Let s connect nodes u and v, and let the minimal closed walk going through all the edges of its fundamental circuit be $(u, s = f_0, v = n_0, f_1, n_1, f_2, n_2, \ldots, n_{q-1}, f_q, u)$, where $f_0, \ldots, f_q$ are edges and $n_0, \ldots, n_{q-1}$ are nodes. We will also refer to this walk as the fundamental circuit. Since the walk uses only the edges and nodes of $\widehat{G}(R')$ and it uses all of them at least once, this terminology causes no ambiguity.

For example, in Figure 5.2(i), q equals 3 and $f_1 = r_1$, $n_1 = v_1$, $f_2 = r_2$, $n_2 = v_2$, $f_3 = r_3$. In Figure 5.2(ii), $q = 10$ and $f_1 = r_1$, $n_1 = v_1$, $f_2 = r_2$, $n_2 = v_2$, $f_3 = r_3$, $n_3 = v_3$, $f_4 = r_4$, $n_4 = v_1$, $f_5 = r_1$, $n_5 = v$, $f_6 = s$, $n_6 = u$, $f_7 = r_5$, $n_7 = v_4$, $f_8 = r_6$, $n_8 = v_5$, $f_9 = r_7$, $n_9 = v_6$, $f_{10} = r_8$.

The minimality of the fundamental circuit ensures that any of its edges is traversed at most twice in the walk. When a particular edge (or node) is traversed twice, then two different edge (or node) labels in the walk refer to it. For example, in Figure 5.2(ii) n_1 and n_4 denote the same node, v_1 and edge r_1 is traversed twice, so both f_1 and f_5 refer to this edge. A node of the fundamental circuit is called an *even node* if it is an in-node of both or neither of the edges that stand next to it in the walk. Otherwise it is called an *odd node*. Note that while a node in $\widehat{G}(R')$ may have 3 or 4 incident edges, in the walk these edges get different labels so that exactly two edges from $\{f_0, \ldots, f_q\}$ are incident to it. This also means that a node of $\widehat{G}(R')$ may be represented by n_i and n_j in the walk with n_i being an odd node and n_j being an even node. For instance, node v_1 in Figure 5.2(ii) is an odd node when it first occurs in the list as n_1 but an even node in its second occurance as n_4.

Coefficients $w(r_i)$ $(i = 0, 1, \ldots, t)$ are non-zero weights corresponding to columns of R' so that their weighted sum is the zero vector (as shown in (5.3)). As rows of the incidence matrix are associated with the nodes of G and columns with the edges, $w(r_i)$'s can be viewed as non-zero weights assigned to the edges of the fundamental circuit so that at every node the sum of the weights signed by the signs of the edge-ends is zero. Knowing that $w(r_0) = -1$ and that two f_i edges are incident to every node in the fundamental circuit, this zero sum property provides us with a quick method for tracing the other coefficients. Basically, we first give weight $b(f_1)$ to f_1 so that vector $(b(f_1)f_1 - s)$ has 0 in the row corresponding to node v. Then, in the same way, we can determine $b(f_2)$, $b(f_3)$ and so on. Since different f_i's and n_i's could be associated with the same edge or node of the graph, values of $b(f_i)$'s must be adjusted to obtain the correct coefficients $w(r_i)$. For example, if f_j is a half-edge, then $n_{j-1} = n_j$ so $b(f_j)$ is counted twice, but in R' column f_j has ± 1 at this node. Furthermore, the $b(f_i)$ values of edges appearing twice in the fundamental circuit are summed. Algorithm 1 summarizes these considerations in a formal way. By applying these rules we get weights that satisfy (5.3). Since we know that w is unique, our rules provide us with the correct coefficients.

Instead of the successive method of determining the $b(f_i)$ values in Step 2 in Algorithm 1, an

Algorithm 1 Calculating the elements of a binet matrix

Step 1 $b(f_0) = -1$

Step 2 For $i = 1$ to q let

$$b(f_i) = \begin{cases} -b(f_{i-1}) & \text{if } n_{i-1} \text{ is an even node} \\ b(f_{i-1}) & \text{if } n_{i-1} \text{ is an odd node} \end{cases}$$

Step 3 If s is a half-edge, or a link which is traversed twice in the fundamental circuit (so it lies on the path connecting two odd cycles), then divide every weight by 2:

$$b(f_i) := \frac{b(f_i)}{2} \text{ for } i = 1, \ldots, q$$

Step 4 If f_i is a half-edge and $f_i \neq s$, then multiply $b(f_i)$ by 2:

$$b(f_i) := 2b(f_i)$$

Step 5 If basic edge r_i is traversed twice in the fundamental circuit, then let $w(r_i)$ be the sum of the values of its two appearences:

$$w(r_i) := b(f_{j_1}) + b(f_{j_2}) \text{ if } f_{j_1} = r_i \text{ and } f_{j_2} = r_i$$

Otherwise let $w(r_i)$ equal the value of its only appearence:

$$w(r_i) := b(f_j) \text{ if } f_j = r_i$$

alternative way can be given. It consists of first calculating the node-values $c(n_j)$ $(j = 0, 1, \ldots, q-1)$ defined as the sum of the signs of the incident f_i edges, i.e.:

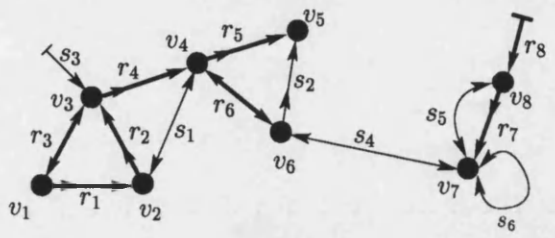
$$c(n_j) = \begin{cases} 0 & \text{if } n_j \text{ is an odd node} \\ +2 & \text{if } n_j \text{ is an in-node of both } f_i \text{ and } f_{i+1} \\ -2 & \text{if } n_j \text{ is an out-node of both } f_i \text{ and } f_{i+1} \end{cases} \tag{5.4}$$

Then for $i = 1, 2, \ldots, q$

$$b(f_i) = \begin{cases} 1 & \text{if } \sum_{j=0}^{i-1} c(n_j) \equiv 2 \pmod 4 \\ -1 & \text{if } \sum_{j=0}^{i-1} c(n_j) \equiv 0 \pmod 4 \end{cases}$$

As a summary, let us recap the steps of the graphical method for calculating the elements of a binet matrix. First identify the basis graph, and take a non-basic edge s. Find its fundamental circuit, which can be an even cycle or a handcuff. Assign -1 to s, then walk along the fundamental circuit and take note of the $b(f_i)$ value of each edge by negating the value at even nodes and keeping it at

odd nodes. If you follow the rules, then you arrive back to s with -1. Calculate the $w(r_i)$ values by taking into account the half-edges and the edges traversed twice. An edge traversed twice must get the same $b(f_i)$ value in both occurences, otherwise its corresponding $w(r_i)$ value would be 0. Finally, fill column s of the binet matrix with vector w: non-zeros in the rows of the basic edges in the fundamental circuit, zeros in other rows. Repeat this for each non-basic edge. Figure 5.3 shows a binet matrix achieved this way from the binet graph depicted next to it.



	s_1	s_2	s_3	s_4	s_5	s_6
r_1	1	1	½	-½	0	0
r_2	0	1	½	-½	0	0
r_3	1	1	½	-½	0	0
r_4	1	2	0	-1	0	0
r_5	0	1	0	0	0	0
r_6	0	-1	0	1	0	0
r_7	0	0	0	1	1	2
r_8	0	0	0	1	2	2

Figure 5.3: An example of a binet graph, and its binet matrix. The heavy edges $r_1, r_2, \ldots, r_8$ make up the basis.

The fundamental circuit of s_1 is an even cycle (v_4, $f_0 = s_1$, $n_0 = v_2$, $f_1 = r_1$, $n_1 = v_1$, $f_2 = r_3$, $n_2 = v_3$, $f_3 = r_4$, v_4). Nodes v_2 and v_4 are even nodes, nodes v_1 and v_3 are odd nodes. The non-zero elements of column s_1 in the binet matrix are calculated as:

$$w(r_1) = b(f_1) = 1,\ w(r_3) = b(f_2) = 1,\ w(r_4) = b(f_3) = 1.$$

The minimal dependent set including s_2 is a handcuff $\{s_2, r_6, r_4, r_2, r_1, r_3, r_5\}$. The fundamental circuit of s_2 is (v_5, $f_0 = s_2$, v_6, $f_1 = r_6$, v_4, $f_2 = r_4$, v_3, $f_3 = r_2$, v_2, $f_4 = r_1$, v_1, $f_5 = r_3$, v_3, $f_6 = r_4$, v_4, $f_7 = r_5$, v_5). The non-zero elements of column s_2 in the binet matrix are calculated as:

$$w(r_6) = b(f_1) = -1,\ w(r_4) = b(f_2) + b(f_6) = 2,\ w(r_2) = b(f_3) = 1,$$
$$w(r_1) = b(f_4) = 1,\ w(r_3) = b(f_5) = 1,\ w(r_5) = b(f_7) = 1.$$

The fundamental circuit of s_3 is a handcuff (v_3, $f_0 = s_3$, v_3, $f_1 = r_2$, v_2, $f_2 = r_1$, v_1, $f_3 = r_3$, v_3). As s_3 is a half-edge, the non-zero elements of column s_3 in the binet matrix are calculated as:

$$w(r_2) = \frac{b(f_1)}{2} = \frac{1}{2},\ w(r_1) = \frac{b(f_2)}{2} = \frac{1}{2},\ w(r_3) = \frac{b(f_3)}{2} = \frac{1}{2}.$$

The fundamental circuit of s_4 is a handcuff (v_7, $f_0 = s_4$, v_6, $f_1 = r_6$, v_4, $f_2 = r_4$, v_3, $f_3 = r_3$, v_1, $f_4 = r_1$, v_2, $f_5 = r_2$, v_3, $f_6 = r_4$, v_4, $f_7 = r_6$, v_6, $f_8 = s_4$, v_7, $f_9 = r_7$, v_8, $f_{10} = r_8$, v_8, $f_{11} = r_7$, v_7). As s_4 appears twice in the fundamental circuit and r_8 is a half-edge, the non-zero elements

of column s_4 in the binet matrix are calculated as:

$$w(r_6) = \frac{b(f_1)+b(f_7)}{2} = 1,\ w(r_4) = \frac{b(f_2)+b(f_6)}{2} = -1,$$
$$w(r_3) = \frac{b(f_3)}{2} = -\frac{1}{2},\ w(r_1) = \frac{b(f_4)}{2} = -\frac{1}{2},\ w(r_2) = \frac{b(f_5)}{2} = -\frac{1}{2},$$
$$w(r_7) = \frac{b(f_9)+b(f_{11})}{2} = 1,\ w(r_8) = \frac{2b(f_{10})}{2} = 1.$$

The fundamental circuit of s_5 is $(v_8,\ f_0 = s_5,\ v_7,\ f_1 = r_7,\ v_8,\ f_2 = r_8,\ v_8)$. The non-zero elements of column s_5 in the binet matrix are calculated as:

$$w(r_7) = b(f_1) = 1,\ \ w(r_8) = 2b(f_2) = 2.$$

The fundamental circuit of s_6 is $(v_7,\ f_0 = s_6,\ v_7,\ f_1 = r_7,\ v_8,\ f_2 = r_8,\ v_8,\ f_3 = r_7,\ v_7)$. The non-zero elements of column s_6 in the binet matrix are calculated as:

$$w(r_7) = b(f_1) + b(f_3) = 2,\ \ w(r_8) = 2b(f_2) = 2.$$

Note that if we use the algebraic method instead of the graphical one, we could get the 8×6 matrix of Figure 5.3 by inverting the 8×8 matrix $[r_1, r_2, \ldots, r_8]$ and taking $R^{-1}S$ where S is the 8×6 node-edge incidence matrix corresponding to the non-basic edges. Clearly, the graphical method is more elegant and compact. Furthermore, the graphical representation of a binet matrix helps us to prove results about binet matrices, which would be more cumbersome with the algebraic method. In the remaining sections we will give some of these results, where the elegant graphical proofs are much easier than the ones with matrices and their inverses.

5.2 Operations on binet matrices

In this section we give some operations that, when applied to a binet matrix, result in another binet matrix. We will give the proof in both graphical and algebric terms. It will be obvious that the graphical explanations are better, as they are more compact and easier to follow. This shows the power of the graphical algorithm to determine binet matrices.

We start with some trivial operations and then go on to more complex ones. Some other related results are also given. Let us start with a graphical operation which does not change the binet matrix.

Switching at a node of the binet graph keeps the parity of every node of the fundamental circuit (see Proposition 4.3), so clearly does not affect the calculations. The matrix operation equivalent to switching is multiplying a row of the node-edge incidence matrix A by -1. The effect of this change is easy to detect: a column in the inverse of the basis R and the corresponding row in the non-basic part S is multiplied by -1. So $R^{-1}S$ does not change.

Lemma 5.3. *Switching at a node of a binet graph keeps the binet matrix unchanged.*

Permuting rows or columns of a binet matrix obviously results in another binet matrix. It is just a bit more difficult to prove that a matrix achieved by multiplying a row or column of a binet matrix by -1 is also binet, as it is equivalent to reversving the direction of the corresponding, basic or non-basic, edge.

Lemma 5.4. *A binet matrix remains a binet matrix under the following operations:*

(a) permuting rows and columns,

(b) multiplying a row or a column by -1.

Lemma 5.5. *A binet matrix remains a binet matrix under the following operations:*

(a) deleting a column,

(b) repeating a column,

(c) adding a unit column.

Proof: If we delete a non-basic edge from the graph, its column is deleted from the binet matrix. Similarly, adding an identical copy of a non-basic edge to the graph results in a column repetition. Adding a new non-basic edge which is an identical copy of a basic edge means that the binet matrix of the extended graph contains a new unit column, having 1 in the row of the duplicated basis edge. □

Corollary 5.6. *B is a binet matrix if and only if $[B, I]$ is such.*

Lemma 5.5 lists column operations that maintain binet matrices. These operations can be applied to rows as well.

Lemma 5.7. *Let B be a binet matrix. Matrix B' obtained from B by the following operations is also a binet matrix.*

(a) deleting a row,

(b) repeating a row,

(c) adding a unit row.

Proof: Let us suppose that $B = R^{-1}S$, where $A = [R, S]$ is an incidence matrix of a bidirected graph. We give the basic (R') and non-basic (S') part of the new incidence matrix that defines B'. Furthermore, we present the related transformations on the binet graph.

We will use the following notations. Let V, E_R, and E_S be respectively the set of nodes, basic edges and non-basic edges in $G(A)$. If M is a matrix whose rows and columns are labelled by sets F and G, respectively, and $F_1 \subseteq F$, $G_1 \subseteq G$, then $(M)_{F_1}^{G_1}$ denotes the submatrix with row labels F_1 and column labels G_1. To simplify notations, if $F_1 = F$ or $G_1 = G$, then we do not put them in the corresponding subscript or superscript. So for example, $(M)_f$ denotes the row corresponding to

$f \in F$ and $(M)^g$ denotes the column corresponding to $g \in G$. Instead of $(M)^g_f$ we use the common M_{fg}.

(a) Deleting row r from B can be achieved by contracting the basic edge r from the binet graph. By enumerating all possible cases corresponding to the type of the edge and the parity of its end-nodes in the fundamental circuits, one can check that a new basis graph is created by the contraction, the fundamental circuits contain all their old edges except r, and the values corresponding to these edges are unchanged in the binet matrix. These implications are easier to establish for directed r, basically because then the sum of the node values defined in (5.4) is unchanged for each node. For bidirected r one can use the fact that contracting a bidirected link starts with a switching to make it directed, and switchings do not alter the binet matrix.

Algebraicly, let u be the removed end-node of r and v be the other end-node if it exists. Then $R' = (R)^{E_R-r}_{V-u} - R_{ur}(R)^r_{V-u}(R)^{E_R-r}_u$ and $S' = (S)_{V-u} - R_{ur}(R)^r_{V-u}(S)_u$. (See (4.3).) It is easily checked that $(R^{-1})^u_{E_R-r} = -R_{ur}R_{vr}(R^{-1})^v_{E_R-r}$ and $R'^{-1} = (R^{-1})^{V-u}_{E_R-r}$, therefore $R'^{-1}S' = (R^{-1})_{E_R-r}S = (B)_{E_R-r} = B'$.

(b) Let e be the row we wish to repeat. The new binet graph, $G(B')$ can be obtained from $G(B)$ by subdividing e into two new basic edges e and r by inserting a new odd node t, in the following way.

If e:uv (if it is a loop or a half-edge, then whenever we use v we mean u), then insert a new node t into the graph, connect e to t instead of v with the same sign(s) at its end-node(s), connect nodes t and v with a new directed link r and direct r so that t becomes an odd node (see Figure 5.4 for the cases of (i) a link, (ii) a loop and (iii) a half-edge).

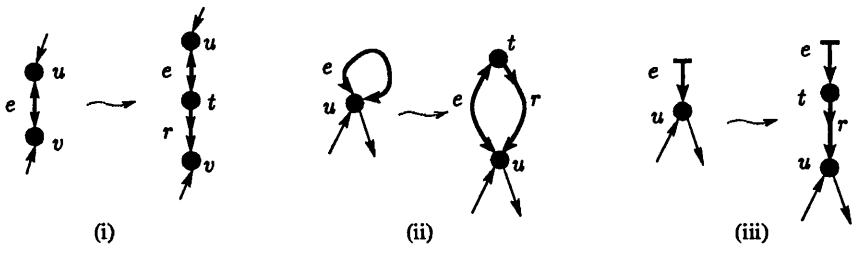


Figure 5.4: The graphical equivalent of repeating row e

Each fundamental circuit that goes through e will go through r as well, and one checks easily with reference to Algorithm 1 that the value of both $w(e)$ and $w(r)$ will be equal to $w(e)$ calculated in the old graph. Since t is an odd node, $c(t) = 0$ (see (5.4)), so all the other values in each fundamental circuit remain unchanged. Therefore the new row r of the binet matrix of the transformed graph equals row e of the original matrix and other rows are unchanged.

To explain the same operation algebraically, add a new column labelled r to R, and a new row labelled t to A. In the non-basic part S, the new row contains only zeros. The non-zero entries in the

new row and column of R' are the following:

$$R'_{te} = R'_{ur} = -R'_{tr} = \begin{cases} R_{ue} & \text{if } e \text{ is a link or a half-edge} \\ \frac{1}{2}R_{ue} & \text{if } e \text{ is a loop.} \end{cases}$$

Furthermore, one entry in R is changed: $R'_{ue} = 0$, if e is a link or a half-edge and $R'_{ue} = \frac{1}{2}R_{ue}$, if e is a loop. It can be readily verified that $(R'^{-1})^V_{E_R} = R^{-1}$ and $(R'^{-1})_r = (R^{-1})_e$, which implies that row r of B' equals row e of B.

(c) Let us suppose now that we would like to add a unit row with the only non-zero element in column $e{:}uv$. We can apply the same operation as in part (b). Suppose the non-basic edge e has end-nodes u and v. Then subdivide e with a new node t into edges e and r as shown in Figure 5.4. The new edge r will be a basic edge and e remains non-basic. Then r is only used in the fundamental circuit of e, that is why its row has zeros in the other columns. In column e, however, it has -1, as node t is an odd node. Multiplying the new unit row by -1, i.e., reversing the direction of r (see Lemma 5.4) we get a positive unit row.

The matrix transformation is very similar to the previous one. There is a new row t added to A, and a new column r to R. The new row and column have non-zeros

$$S'_{te} = R'_{ur} = -R'_{tr} = \begin{cases} S_{ue} & \text{if } e \text{ is a link or a half-edge} \\ \frac{1}{2}S_{ue} & \text{if } e \text{ is a loop.} \end{cases}$$

The only other change is that $S'_{ue} = 0$, if e is a link or a half-edge and $S'_{ue} = \frac{1}{2}S_{ue}$, if e is a loop. The inverse of R' contains R^{-1} as it is of the following form:

$$R'^{-1} = \begin{bmatrix} R^{-1} & (R^{-1})^u \\ & R'_{tr} \end{bmatrix}$$

where the last row and column are related to r and t. Thus the last row of B', corresponding to r, has a -1 in column t and zeros elsewhere. □

Parts (a) of Lemmas 5.5 and 5.7 together imply the most important result in this section:

Theorem 5.8. *Every submatrix of a binet matrix is binet.*

We already know that a binet matrix $[B, I]$ arises from a full row rank incidence matrix A by taking a basis and pre-multiplying A with its inverse. Alternatively, this can be viewed as pivoting on A, as defined in (1.4). We now show that these operations can also be applied to binet matrices.

Lemma 5.9. *Let B be a binet matrix and T be a basis of $[B, I]$. Then*

(a) $T^{-1}[B, I]$ is binet

(b) $\bar{B}$ obtained by pivoting on a non-zero element of B is binet.

Proof: (a) Let $B = R^{-1}S$, where $A = [S, R]$ is an incidence matrix of a bidirected graph. Then $[B, I] = R^{-1}A$. If T is a basis of $[B, I]$, then there is a basis U of A such that $T = R^{-1}U$. Then $T^{-1}[B, I] = U^{-1}R[R^{-1}S, I] = U^{-1}[S, R] = U^{-1}A$. With column permutations, $A = [W, U]$, so $U^{-1}A = [U^{-1}W, I] = [B', I]$, where B' is a binet matrix.

(b) Pivoting is equivalent to pre-multiplying $[B, I]$ with the inverse of matrix $T = \begin{bmatrix} \alpha & 0 \\ b & I \end{bmatrix}$ (using the notations of (1.4)). T is a basis of $[B, I]$, so $T^{-1}[B, I]$ and any matrix obtained from this by column deletion, e.g. $\bar{B}$, is binet.

Graphically, pivoting means changing the basis graph. If the first row and column of B correspond to basic edge r and non-basic edge s respectively, then in the binet graph representing $\bar{B}$, r is a non-basic edge and s is a basic edge. In other words, the new basis consists of $(R \setminus r) \cup s$. □

As stated above, binet matrices arise from incidence matrices by a series of pivoting. When applying basis exchanges, this translates to the following. We start with an artifical unit basis, i.e., the initial basis comprises positive half-edges at each node. In this basis, the incidence matrix is a binet matrix. When a pivot step is done, one of the artificial half-edges is exchanged with a normal basic edge, and the artifical edge disappears. When we have exchanged and removed all the artifical edges, the basis is R and the matrix has become B.

Switching on the binet graph keeps the binet matrix unchanged, which provides a method to find a special binet representation of a binet matrix.

Lemma 5.10. *There always exists a representation of a binet matrix where each connected component of the basis contains only one bidirected edge.*

Proof: Let binet matrix B be represented by the binet graph G. Consider a component of the basis. By switchings at the cycle nodes, all but one of the cycle edges can be made directed. Then by consecutive switchings at the ends of the bidirected non-cycle edges, we end up with a component where all the edges except one cycle edge are directed.

Algebraicaly it means the following. Let us suppose that the basis R is connected. We have shown in the proof of Lemma 4.4 that by multiplying the rows by -1, the submatrix of R that corresponds to the cycle can be transformed so that it contains only one bidirected edge (note that column scaling is not needed here). By further multiplications of the rows of R, the non-cycle basic edges of G can also be transformed to directed edges. So there exists a diagonal $0, \pm 1$ matrix E such that if $R' = ER$, then $G(R')$ contains only one bidirected edge. Then $B = R^{-1}S = R^{-1}E^{-1}ES = R'^{-1}S'$, where $S' = ES$ and R' is a basis of $[R', S'] = EA$. Thus B has a binet representation in which the basis R' contains only one bidirected edge. □

Remark 5.11. The transformation carried out in the proof of Lemma 5.10 changes only the directions of the edges in the binet graph, the structure remains unchanged. Futhermore, the only

bidirected edge in each component must be in its cycle, but we are free to choose which cycle edge remains bidirected.

Binet matrices are not necessarily integral matrices. It is easily seen, and follows from Algorithm 1, that binet matrices can have elements $0, \pm 1, \pm 2$ and $\pm ½$. The following lemma gives a necessary and sufficient condition for a binet matrix to be integral. Then we use the condition to prove that with a series of pivot steps any binet matrix can be made integral.

Lemma 5.12. *A binet matrix is integral if and only if it has a binet representation in which*
(a) every cycle in each connected component of its basis is a half-edge, or
(b) the basis is connected and the binet graph has no half-edges.

Proof: The lemma follows easily from the rules of calculating the elements of the matrix given in Algorithm 1, and the fact that G is connected. □

Remark 5.13. An alternative version of condition (b), where we forbid only non-basic half-edges is clearly equivalent, because if there is a basic half-edge, then we are in case (a).

Lemma 5.14. *Let B be a non-integral binet matrix with m rows. By at most $2m$ pivot operations B can be converted to an integral binet matrix*

Proof: Let B be a binet matrix with a non-integral element b_{ij} in row r_i and column s_j. Then $b_{ij} = \pm ½$ and, according to Algorithm 1 this can occur if and only if r_i is an edge of a cycle which is not a half-edge and s_j is a half-edge or lies on the connecting path between two basic components. By pivoting on b_{ij} we exchange basic edge r_i with non-basic edge s_j, as described in the proof of Lemma 5.9. After pivoting the basis contains more half-edges or fewer components, therefore after a finite number of pivots the basis satisfies the conditions of Lemma 5.12, and B is integral.

To determine the number of necessary pivots, let H denote the number of basic half-edges in a binet graph, and C be the number of basic components. If s_j is a half-edge, then after exchanging it with r_i, H is increased by 1 and C remains the same (as r_i cannot be a cut-edge because it is in a cycle and s_j cannot connect two unconnected components of the old basis because it is a half-edge). If s_j lies on the connecting path between two basic components, then C is decreased by 1 after pivoting and H is unchanged. Therefore, the value of $M = H - C$ is increased by 1 in each pivot steps. The maximal value for H is m, the number of basic edges. Moreover, C cannot be more than m. So M is certainly between $-m$ and m. It implies that the algorithm must terminate in at most $2m$ steps. □

5.3 Necessary conditions

We give two necessary conditions on integral binet matrices. The first claims that an integral binet matrix is the sum of two network matrices, while the second result is very similar to a well-known

characterization of totally unimodular matrices due to Ghouila-Houri [34] (see Theorem 1.3). Unfortunately, neither of the conditions are sufficient.

Theorem 5.15. *If B is an integral binet matrix, then there exist network matrices N_1 and N_2 such that*

(a) $B = N_1 + N_2$

(b) both $[N_1, N_2]$ *and* $\begin{bmatrix} N_1 \\ N_2 \end{bmatrix}$ *are network matrices*

Proof: Let binet graph $G = (V, R, S)$ with node set V, basic edges R, and non-basic edges S, represent B. By Lemma 5.10 and Lemma 5.12, G is one of the following two types.

I. All the basic cycles of G are half-edges, and they are the only bidirected edges in the basis.

II. There are no half-edges in G, the basis is connected and it contains only one bidirected edge.

We deal with the two cases separately. In both we introduce a way of constructing a spanning tree and two connected directed graphs giving network matrices that satisfy the conditions of the theorem.
Case I. We will use the graph in Figure 5.5 as an example.

Let us introduce a new node v_0 into G and connect each basic half-edge to it. That is, change the half-edges to links by declaring v_0 to be their second end-node. Their signs at v_0 should be chosen so that the links become directed. The basis now contains only directed links, and no cycles, so it is a directed tree T, spanning node set $V \cup v_0$. With the process described below, we construct connected directed graphs D_1 and D_2 so that they both contain T, and network matrices N_1 and N_2 based on this tree in graphs D_1 and D_2, respectively, that satisfy the conditions of the theorem. The idea is to replace each non-basic edge $s \in S$ in G by two directed non-tree edges, one in D_1 and one in D_2, such that the fundamental circuits of the directed edges (also denoted by s) together give the fundamental circuit of s in G. It might happen that one edge is enough, i.e., there is no edge in D_2 corresponding to s. Figure 5.6 illustrates the process on the graph of Figure 5.5. To keep a consistent setting, we will use the phrase 'loose edge' in this case, meaning that we add an edge that has no end-nodes. In directed graphs this means that the edge is not there, only virtually.

(a) If, s is a directed link (like s_1 in Figure 5.5), then let it remain the same in D_1 and associate a loose edge with it in D_2.

(b) If s is a half-edge (like s_2 in Figure 5.5), then connect it to v_0 in D_1 and direct it as described above. A loose edge is associated with it in D_2.

(c) If s:uv is a bidirected link or a loop (s_3 and s_4 are such in Figure 5.5), then imagine two new half-edges h_u and h_v incident to u and v and directed so that u (or v) is an out-node of h_u (h_v) if and only if it is an out-node of s. If s is a loop (i.e., $u = v$), then h_u and h_v are two identical half-edges. It is clear that the column of the incidence matrix of G that corresponds to s is the sum of the columns of the matrix that correspond to the related half-edges, and this remains true for binet

matrix B. Now apply the operation of part (b) to h_u and h_v, with the only change that for h_v we change the role of D_1 and D_2. So at the end, we have an edge in each of D_1 and D_2. They are the two new edges that correspond to s in G.

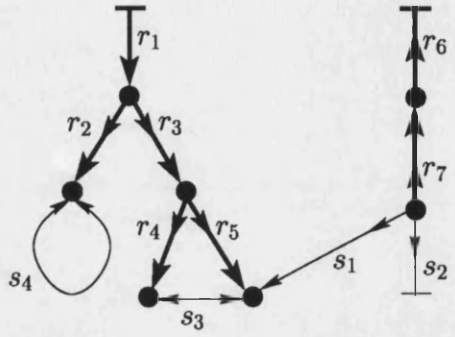


Figure 5.5: Binet graph B with only half-edge basic cycle edges

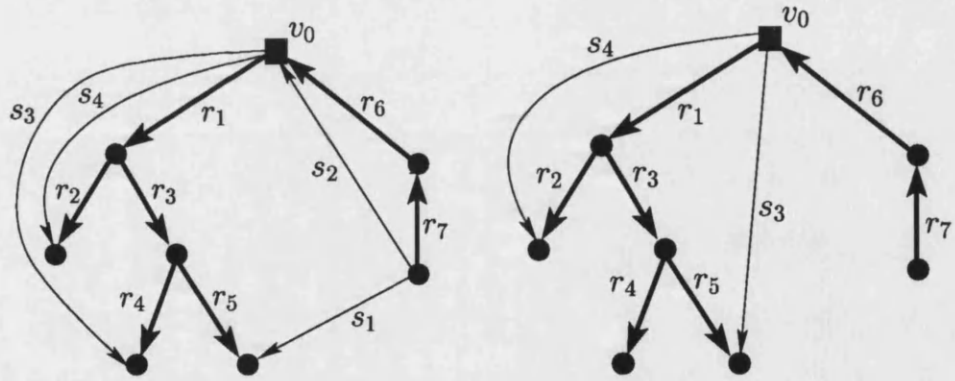


Figure 5.6: Directed graphs D_1 and D_2 made from binet graph B of Figure 5.5

It follows from the definition of binet and network matrices that column s of B equals the sum of columns s of N_1 and N_2 for any non-basic edge. We have derived every column of the binet matrix as the sum of two corresponding columns of network matrices, so $B = N_1 + N_2$.

$[N_1, N_2]$ is trivially a network matrix, as both N_1 and N_1 are based on T. As for $\begin{bmatrix} N_1 \\ N_2 \end{bmatrix}$, consider the following construction. Reverse the direction of every edge in D_2 (getting D_2'), unify the v_0 node in D_1 and D_2', as shown in Figure 5.7. As to non-tree edges, if there is a non-tree edge connecting u with v_0 in D_1 and another one connecting v with v_0 in D_2 and they correspond to the same non-basic edge of B, then we substitute them by a new edge between u and v in the unified graph, and keep the directions at u and v. The network matrix of the so defined unified directed graph is $\begin{bmatrix} N_1 \\ N_2 \end{bmatrix}$.

Case II. We will use the graph in Figure 5.8 as an example.

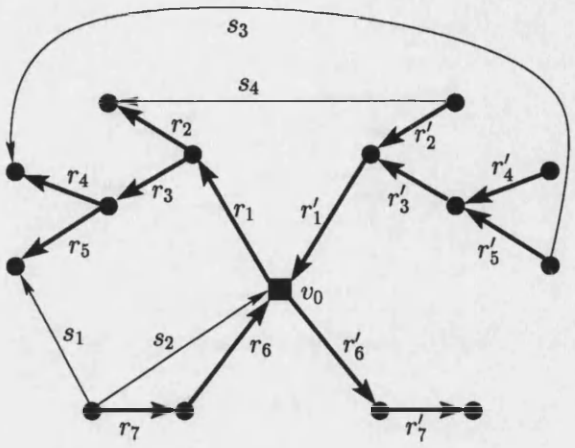


Figure 5.7: Unification of D_1 and D'_2

Let r_1 be the the only bidirected edge in G (so it must be in the cycle), and v_0 be an end-node of it. We 'open' the cycle at v_0, by splitting v_0 into two separate nodes v'_0 and v''_0 so that r_1 is incident to v'_0 and other basic edges incident to v_0 are incident to v''_0. Furthermore, we switch at v'_0 so that the basis becomes a directed tree T. Figure 5.9 shows this operation.

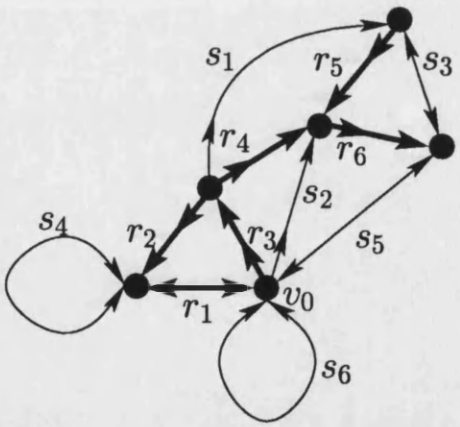


Figure 5.8: Binet graph B with a connected basis

We will construct directed graphs D_1 and D_2 such that they contain T as spanning tree, and their corresponding network matrices N_1 and N_2 satisfy the conditions of the theorem. Figure 5.9 illustrates. Again, we replace all non-basic edges $s \in S$ by two directed edges, one in D_1 and the other in D_2, so that the sum of the network matrix columns corresponding to the new edges (also denoted by s) is the column s of B.

- If s is a directed link and v_0 is not its end-node (as s_1 in Figure 5.8), then it remains the same in D_1 and becomes a loose edge in D_2.

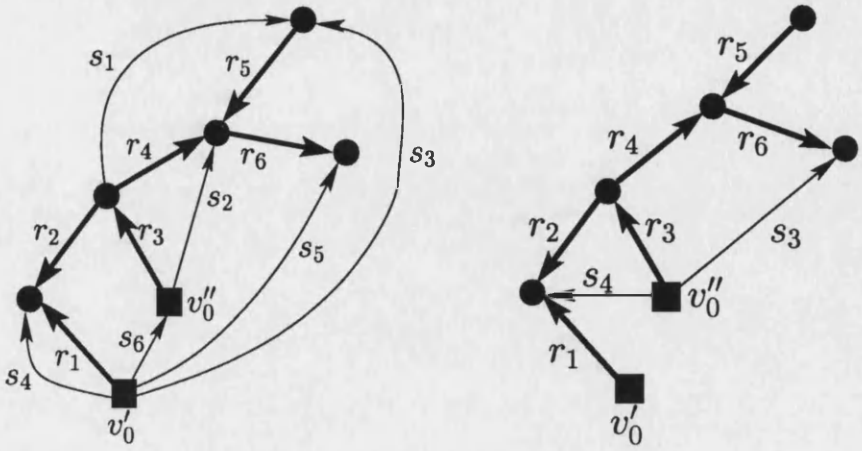


Figure 5.9: Directed graphs D_1 and D_2 made from binet graph B of Figure 5.8 by opening the cycle

- If s is a directed link and v_0 is its end-node (as for s_2 in Figure 5.8), then v_0'' is its end-node in D_1 and it is represented by a loose edge in D_2.
- If s is a bidirected edge connecting u and v where $u, v \neq v_0$ (but u might be equal to v) as in case of s_3 and s_4, then let s connect u and v_0' in D_1 and v and v_0'' in D_2, such that u (or v) is a tail of s in D_1 (or D_2) if and only if it is a tail of s in G.
- If $s{:}uv_0$ is a bidirected link incident to v_0 (as s_5), then s connects u and v_0' in D_1 with the above described direction at u, and s is a loose edge in D_2.
- If s is a loop at v_0 (as s_6), then let s connect v_0' and v_0'' in D_1 with the same direction at v_0' as s has at v_0 in G, and s is a loose edge in D_2.

We have listed all the possible cases, and in each case column s of B is equal to the sum of columns s of N_1 and N_2, so $B = N_1 + N_2$.

The network representations of $[N_1, N_2]$ and $\begin{bmatrix} N_1 \\ N_2 \end{bmatrix}$ are made in the same way as before, unifying v_0' of D_1 with v_0'' of D_2' in the latter case. Figure 5.10 gives the unified graph for our example.

□

As a corollary, another necessary condition can be derived. It is similar to Theorem 1.3, which claims that for each collection of columns of a totally unimodular matrix, there exists a scaling of the selected columns by ± 1 such that the sum of the scaled columns is a vector of $0, \pm 1$ elements, and the same is true for rows. Obviously, as this condition is also sufficient for total unimodularity, it cannot remain valid for 2-regular matrices. For integral binet matrices, however, we can provide something similar.

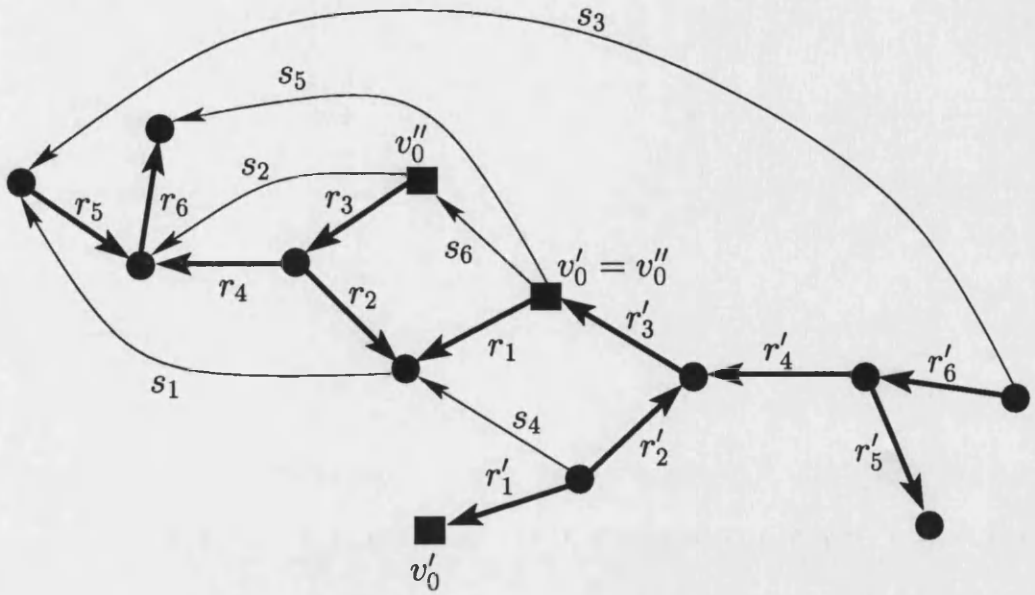


Figure 5.10: Unification of D_1 and D_2' in Case II.

Theorem 5.16. *For each collection of columns or rows of an integral binet matrix, there exists a scaling of the selected columns or rows by* ± 1 *such that the sum of the scaled columns or rows is a vector of* $\{0, \pm 1, \pm 2\}$ *elements.*

Proof: Since column (row) deletions preserves the binetness of matrix B, we can assume that every column (row) is selected.

1. Columns. By Theorem 5.15, there exist network matrices N_1 and N_2 such that $B = N_1 + N_2$ and $N = \begin{bmatrix} N_1 \\ N_2 \end{bmatrix}$ is also a network matrix. By Theorem 1.3, there is a ± 1 vector d such that Nd is a vector of $0, \pm 1$ elements, hence $Bd = (N_1 + N_2)d = N_1 d + N_2 d$ is a vector of $\{0, \pm 1, \pm 2\}$ elements. Thus vector d gives the good scaling.

2. Rows. Very similar using the fact that $[N_1, N_2]$ is a network matrix. □

Unfortunately, neither the condition of Theorem 5.15, nor the condition of Theorem 5.16 is sufficient for binet matrices. For example, for matrix B below, the network matrices N_1 and N_2 satisfy conditions (a) and (b) of Theorem 5.15. This implies that the rows and columns of B can be scaled as in Theorem 5.16. On the other hand, B is not a binet matrix, because it is not even 2-regular.

$$B = \begin{bmatrix} 2 & 1 \\ -1 & 1 \end{bmatrix}, \quad N_1 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad \text{and} \quad N_2 = \begin{bmatrix} 1 & 1 \\ -1 & 0 \end{bmatrix}. \tag{5.5}$$

5.4 Towards recognition

In this section we turn to the problem of recognizing binet matrices. This problem is in the complexity class $\mathcal{NP}$, because it is easy to verify that a matrix is binet, one only has to give a binet representation. It is a much more difficult question, and unsettled as yet, if the recognition problem is in $\mathcal{P}$ or not, i.e., if there is a polynomial algorithm that can decide if a given matrix is binet or not.

For network matrices, the parallel question is answered – several polynomial-time algorithms exist for recognition. The best-known one is probably Schrijver's method (see [51] and [55]), which adapts the matroidal ideas of Bixby and Cunningham [9] to matrices. The algorithm works by reducing the problem to a set of smaller problems, which can then be handled easily. The smaller problems consist of deciding if a matrix with at most two non-zeros per column is a network matrix or not. The reduction is done by identifying rows of the matrix that correspond to cut-edges of the spanning tree, and then carrying on with the smaller matrices associated with the components.

The ideas of this algorithm cannot be directly adapted to binet matrices because the basis of a binet graph, a forest of odd 1-trees is more complex than a tree. We have to admit that we have been unable to devise a recognition algorithm for binet matrices. However, we present here a simple method that is able to prove or disprove if a small matrix is binet. The approach is to formulate the problem as a mixed integer programming (MIP) problem, and then solve it. Of course, the method is not polynomial, as long as there is no polynomial algorithm invented to solve MIP problems.

Before embarking on the description of this method, we present some simple 'preprocessing' rules that simplifies our task. As Lemma 5.14 states, every rational binet matrix can be converted to an integral binet matrix by a finite number of pivot steps. The theorem also implies that even in the worst case we need only a specified number of these pivots to get the integral matrix, provided that the matrix we started with was binet. This means that if we have a method that can decide the binetness of integral matrices, then, preceded by the steps described in Algorithm 2, this method can be applied to rational matrices too.

Algorithm 2 Preprocessing steps for rational matrices

Input	An $m \times n$ matrix B
Step 1	Check the elements of B. If it has an element other than 0, ± 1, ± 2, or $\pm \frac{1}{2}$, then B is not binet.
Step 2	If B is integral, STOP. Otherwise, pivot on a $\pm \frac{1}{2}$ element.
Step 3	Repeat **Step 1** and **Step 2** at most $2m$ times. If the algorithm does not stop during these, then B is not binet.

So let us assume that we are given an $m \times n$ matrix B, and we want to decide if it is binet or not. B is binet if and only if there exist an $m \times m$ matrix R and an $m \times n$ matrix S such that they satisfy the following conditions:

(a) $RB = S$,

(b) R is a non-singular matrix,

Moreover, R and S are node-edge incidence matrices of bidirected graphs, i.e., we have the further conditions:

(c) R and S are integral matrices, and

(d) condition (4.2) holds for R and S.

Define index sets $I = \{1,\ldots,m\}$, $J = \{1,\ldots,m\}$ and $K = \{1,\ldots,n\}$. In our model, we have integer variables r_{ij} for all $(i,j) \in I \times J$, and integer variables s_{ik} for all $(i,k) \in I \times K$ to represent the unknown elements of matrices R and S. Moreover, let $B = [b_{jk}]_{j\in J}^{k\in K}$ be the given matrix, i.e., b_{jk} are data for this model. Condition (a) then takes the form

$$\sum_{j\in J} r_{ij} b_{jk} = s_{ik} \qquad \forall (i,k) \in I \times K$$

Variables r_{ij} and s_{ik} are integer but not nonnegative. So, following the standard technique, let us introduce the nonnegative variables r_{ij}^+, r_{ij}^- for all $(i,j) \in I \times J$, and s_{ik}^+, s_{ik}^- for all $(i,k) \in I \times K$, with which we wish to represent the positive and negative parts of r_{ij} and s_{ik}. Thus,

$$r_{ij} = r_{ij}^+ - r_{ij}^- \quad \text{and} \quad |r_{ij}| = r_{ij}^+ + r_{ij}^- \tag{5.6}$$

$$s_{ik} = s_{ik}^+ - s_{ik}^- \quad \text{and} \quad |s_{ik}| = s_{ik}^+ + s_{ik}^- \tag{5.7}$$

Condition (b), the non-singularity of R cannot be modelled with linear constraints. However, R certainly do not have all-zero rows or columns:

$$\sum_{j\in J} |r_{ij}| \geq 1 \text{ for all } i \in I, \text{ and } \sum_{i\in I} |r_{ij}| \geq 1 \text{ for all } j \in J$$

We can also assume that B has no all-zero column, therefore S cannot have one either.

$$\sum_{i\in I} |s_{ij}| \geq 1 \text{ for all } j \in J$$

So we look for a feasible solution in the set defined by the following constraints.

$$\sum_{j\in J} r_{ij} b_{jk} = s_{ik} \qquad \forall (i,k) \in I \times K \tag{5.8a}$$

$$r_{ij} = r_{ij}^+ - r_{ij}^- \qquad \forall (i,j) \in I \times J \tag{5.8b}$$

$$s_{ik} = s_{ik}^+ - s_{ik}^- \qquad \forall (i,k) \in I \times K \tag{5.8c}$$

$$1 \leq \sum_{i\in I} (r_{ij}^+ + r_{ij}^-) \leq 2 \qquad \forall j \in J \tag{5.8d}$$

$$\sum_{j\in J}(r_{ij}^+ + r_{ij}^-) \geq 1 \qquad \forall i \in I \tag{5.8e}$$

$$1 \leq \sum_{i\in I}(s_{ik}^+ + s_{ik}^-) \leq 2 \qquad \forall k \in K \tag{5.8f}$$

$$r_{ij}^+, r_{ij}^-, s_{ik}^+, s_{ik}^- \geq 0 \text{ integer} \qquad \forall i \in I, \forall j \in J, \forall k \in K \tag{5.8g}$$

Although we need only a feasible solution from this set to prove the binetness of B, to ensure that r_{ij}^+, r_{ij}^- and s_{ik}^+, s_{ik}^- are really the positive and negative parts of r_{ij} and s_{ik}, we use the objective function

$$\min \sum_{(i,j)\in I\times J}(r_{ij}^+ + r_{ij}^-) + \sum_{(i,k)\in I\times K}(s_{ik}^+ + s_{ik}^-) \tag{5.9}$$

We will see later that this might not be enough, and further devices are needed.

If there is no feasible solution to the mixed integer programming problem defined by constraints (5.8a)-(5.8g), then we can be sure that matrix B is not binet. However, if there is a solution, then it does not mean that B is binet, as it may happen that matrix R obtained from the solution is singular, so that $RB = S$ does not imply $R^{-1}S = B$. Consequently, in this form the method is more suitable for proving that a matrix is not binet, a theoretically interesting but practically not so important question. And even this works in practice only for small matrices, as the number of integer variables ($2m(m+n)$) is forbidding for large instances.

The model so far is applicable for any rational matrix. If B is integral, however, then there are further possibilities to strengthen the formulation. As mentioned earlier, this assumption is not restrictive, the recognition of non-integral binet matrices is not more difficult than that of integral ones. So let B be an integral matrix. By Lemmas 5.10 and 5.12 any possible binet representation of B is of the following two types:

Type I: Every basic cycle is a half-edge, and all the other basic edges are directed.

Type II: There are no half-edges in the binet graph, the basis is connected and there is only one bidirected edge in the basis.

These conditions imply further constraints in the MIP formulation.

In case of Type I, R cannot contain a loop, so each of its non-zero elements is ± 1. Hence

$$r_{ij}^+ + r_{ij}^- \leq 1 \qquad \forall (i,j) \in I \times J \tag{5.10}$$

Furthermore, if there are two non-zeros in a column of R, so this column corresponds to a link in the graph, then the link must be directed, i.e., the non-zeros are of opposite sign and their sum is zero. Basic half-edges, on the other hand, correspond to columns of R in which the sum of elements is ± 1. It can be assumed without loss of generality that all basic half-edges are positive, because we

can switch at any node of the binet graph without affecting B (see Lemma 5.3). So we have:

$$\sum_{i \in I} r_{ij} \leq 1 \qquad \forall j \in J \tag{5.11}$$

By adding constraints (5.10) and (5.11) to constraints (5.8a)-(5.8g), we get a model for matrices of Type I.

Assuming that B is of Type II, the graph cannot contain half-edges so (4.2) holds with equality for both R and S, therefore constraints (5.8d) and (5.8f) are replaced by

$$\sum_{i \in I} (r_{ij}^+ + r_{ij}^-) = 2 \qquad \forall j \in J \tag{5.8d'}$$

$$\sum_{i \in I} (s_{ik}^+ + s_{ik}^-) = 2 \qquad \forall k \in K \tag{5.8f'}$$

Observe that these constraints mean that any feasible solution now has the same value in objective fuction (5.9). Thus, the objective function will not ensure that at least one of r_{ij}^+ and r_{ij}^- (or s_{ik}^+ and s_{ik}^-) is zero as it should be with the positive and negative parts of r_{ij} (or s_{ik}). To overcome this, we can introduce new binary variables $\overline{r}_{ij} \in \{0,1\}$ and $\overline{s}_{ik} \in \{0,1\}$ for all $i \in I, j \in J$ and $k \in K$, and new constraints

$$r_{ij}^+ \leq 2\overline{r}_{ij} \qquad \forall (i,j) \in I \times J \tag{5.12a}$$

$$r_{ij}^- \leq 2 - 2\overline{r}_{ij} \qquad \forall (i,j) \in I \times J \tag{5.12b}$$

$$s_{ik}^+ \leq 2\overline{s}_{ik} \qquad \forall (i,k) \in I \times K \tag{5.12c}$$

$$s_{ik}^- \leq 2 - 2\overline{s}_{ik} \qquad \forall (i,k) \in I \times K \tag{5.12d}$$

These new constrains force that r_{ij}^+ and r_{ij}^- (and s_{ik}^+ and s_{ik}^-, respectively) cannot be non-zero at the same time, thus validating equations (5.6) (and (5.7), respectively).

Moreover, if B is of Type II, then R contains exactly one column that corresponds to a bidirected edge. The sum of the elements in this column is ± 2, whereas the sum of the elements in any other column is 0. If the bidirected edge is not positive (that is, the sum of the elements in its column is -2), then we can switch at its end-nodes, making it positive. Thus the following inequalities are also valid.

$$\sum_{(i,j) \in I \times J} r_{ij} = 2 \tag{5.13a}$$

$$0 \leq \sum_{i \in I} r_{ij} + \sum_{i \in I} r_{il} \leq 2 \qquad \forall j,l \in J, j \neq l \tag{5.13b}$$

We can also be sure that the number of -1 elements in R is $m - 1$, as each directed link is

associated with one, and the only bidirected link can be assumed to be positive:

$$\sum_{(i,j)\in I\times J} r_{ij}^{-} = m - 1 \tag{5.14}$$

With these new variables and new or modified constraints, we can formulate the recognition of integral binet matrices of Type II. Note, that the non-singularity of R still causes problems. That happened when we ran the model on $B = M(K_4)$ and $B = M(K_6)$ defined in Section 6.2. We got a feasible solution in which R was singular. For illustration, we give the R matrices obtained for $B = M(K_4)$ in the two types.

$$R = \begin{bmatrix} 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 1 & 1 & 0 & -1 \\ 0 & 1 & -1 & -1 & 1 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 \end{bmatrix} \quad \text{and} \quad R = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 1 & 1 & -1 & 0 \\ 1 & 0 & -1 & -1 & 0 & 1 \end{bmatrix}$$

In this singular case, further considerations might help. For example, one can try to show that the singular solution is unique. Or rule this bad solution out with new constraints.

However, as mentioned above, if there is no feasible solution (singular or non-singular) to either model, then the integral matrix B cannot fall in either Type I or Type II, and consequently, it cannot be binet. In this way, we could prove easily that the following matrix B or its transpose are not binet matrices. Note that B is 2-regular, as showed in [3] and in Section 6.2 in this work.

$$B = \begin{bmatrix} 1 & 1 & 1 & 1 \\ -1 & 1 & 1 & -1 \\ 1 & 1 & -1 & -1 \end{bmatrix}$$

Considering that our purpose with this formulation is to find a feasible solution of an integer programming model, in which there are conditions that cannot be formulated with linear constraint, or prove that such a solution does not exist, other methods, probably heuristic ones could be called in. On the other hand, we have not given up the hope of finding a combinatorial recognition algorithm.

Chapter 6

Examples of binet and 2-regular non-binet matrices

Here we give some examples of binet matrices, and also show that some interesting 2-regular matrices are not binet. The examples will show that the class of binet matrices is a real extension of that of network matrices as all network matrices are binet, but there are binet matrices that are not network. These latter matrices include totally unimodular and not totally unimodular ones. The totally unimodular but not network matrices we give are the special matrices appearing in the decomposition theory of totally unimodular matrices.

The counterexamples include matrices that are transposes of binet matrices, and we also give an example of a matrix which is 2-regular, but neither the matrix itself nor its transpose is binet. The techniques we employ to prove that these matrices are not binet are diverse. For example, in one instance we eliminate all the possible cases, while in another, a matrix cannot be binet because of a corollary of a theorem. We also use the MIP model, described in Section 5.4.

In Section 6.3 we show that binet matrices appear in special problems with at most three variables per row. This setting, extensively examined by Hochbaum [41], has several combinatorial and real life applications.

6.1 Examples

We start with some trivial examples, then show that some interesting totally unimodular and not totally unimodular matrices are binet. Recall that a binet matrix is defined as $R^{-1}S$, where matrix $[R, S]$ is the node-edge incidence matrix of a bidirected graph, and R is a basis.

6.1.1 Node-edge incidence matrices

If A is the node-edge incidence matrix of a bidirected graph, then it is binet because $[A, I]$ is also an incidence matrix of a bidirected graph, having a positive half-edge at every node. Taking $R = I$ and $S = A$, we get $R^{-1}S = A$.

6.1.2 Inverse of a basis

Let R be a basis of the incidence matrix A of a bidirected graph. The inverse of R is a binet matrix, since the matrix $[R, I]$ is the node-edge incidence matrix of a bidirected graph. Algorithm 1 provides us with a method of calculating elements of R^{-1}. In fact, a column of R^{-1} can be calculated by determining the fundamental circuit of a non-basic half-edge corresponding to a column of I. The following lemma makes use of this representation of the columns.

Lemma 6.1. *If R is a basis of the node-edge incidence matrix of a bidirected graph, then each row of the inverse of its every component is of the following two types:*

(a) contains only $\pm\frac{1}{2}$ entries,

(b) contains only $0, \pm 1$ entries.

Further, if the only cycle in the component is a half-edge, then its inverse is integral (has only rows of type (b)).

Proof: Let us suppose that the basis is connected as each component can be viewed separately with respect to taking inverse. Then $G(R)$ contains exactly one cycle, which is an odd cycle. As mentioned above, columns of R^{-1} are related to non-basic half-edges. The fundamental circuit of such a half-edge always contains two odd cycles (one of them is the half-edge itself, the other is the basic cycle) connected with a path, i.e. it is a handcuff illustrated in Figure 5.2(ii) and (iii). The basic cycle edges appear once in the fundamental circuit, while the non-cycle basic edges, if they lie on the connecting path, appear twice. According to Step 3 of Algorithm 1, all the $b(f_i)$ values are divided by 2. Therefore, if the odd cycle of $G(R)$ is not a half-edge, then coefficients $w(r_i)$ are $\pm\frac{1}{2}$ on the basic cycle edges, and ± 1 on the non-cycle basic edges for every non-basic half-edge. As each column of R^{-1} refers to a non-basic half-edge, the main part of the theorem is proven. In the special case where the unique cycle of the basis graph $G(R)$ is a half-edge, by Step 4 the only cycle edge row is multiplied by 2 so that all rows contain only $0, \pm 1$ entries. □

So R^{-1} is binet, and then 2-regular by Theorem 7.1. Furthermore, the inverse of R^{-1} is integral, which suggest that R^{-1} may be 1-regular. This suggestion can be easily verified.

Lemma 6.2. *If R is a basis of the node-edge incidence matrix of a bidirected graph, then R^{-1} is 1-regular.*

Proof: The matrix $[R, 2I]$ is a the node-edge incidence matrix of a bidirected graph, so $R^{-1}2I = 2R^{-1}$ is a binet matrix. Any binet matrix is 2-regular by Theorem 7.1, and according to Lemma 2.13, the half of a 2-regular matrix is 1-regular. □

Note that R^{-1} can easily be non-integral, in which case it represents an example of 1-regular but not totally unimodular matrices.

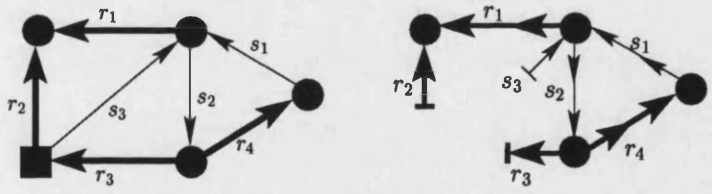
The question arising here is whether Lemma 6.2 is valid only for incidence matrices or true for any arbitrarily chosen integral 2-regular R matrix. The matrix

$$R = \begin{bmatrix} 1 & 2 \\ 0 & 1 \end{bmatrix} \text{ with inverse } R^{-1} = \begin{bmatrix} 1 & -2 \\ 0 & 1 \end{bmatrix}$$

answers this question in the negative, because R^{-1} has a submatrix whose inverse is not integral.

6.1.3 Network matrices

It is straightforward that every network matrix is a binet matrix. Recall that the deletion of a row from an incidence matrix is equivalent to the deletion of a node from the graph, so deleting a row from the incidence matrix of a directed graph results in a bidirected graph which contains directed links and half-edges. The spanning tree is replaced by a set of odd cycle components, comprising the basis of the binet graph. Figure 6.1 depicts the network and binet representation of a network matrix.



$N =$

	s_1	s_2	s_3
r_1	-1	1	-1
r_2	1	-1	1
r_3	1	-1	0
r_4	-1	0	0

Figure 6.1: The network and binet representations of the network matrix N.

Later, in Lemma 7.4 we will show that this kind of binet represenation is characteristic for network matrices. It does not mean, however, that this is the only possibility for representing network matrices with binet graphs.

Lemma 6.3. *There always exists a binet representation of any network matrix with at least two rows where the basis is connected, and the basic cycle is not a half-edge.*

Proof: Let a directed graph with a given spanning tree represent the network matrix. We show how to construct the required binet represenation. To this end, first insert a new bidirected basic link f into the graph so that it connects two nodes that are not connected with a tree edge. (Figure 6.2 illustrates the operation on the graph of Figure 6.1.) The tree extended with this new edge is a binet

basis graph in which the basic cycle contains at least three edges. Moreover, no fundamental circuit uses the new bidirected edge. So the binet matrix of this graph is equal to the original network matrix plus an all-zero row. If we delete the all-zero row, we get the original matrix. The equivalent graph operation, according to Lemma 5.7, is contracting edge f. Since the basic cycle contained at least three edges, after contraction the cycle has at least two edges, so it cannot be a half-edge. The basis is obviously connected. □

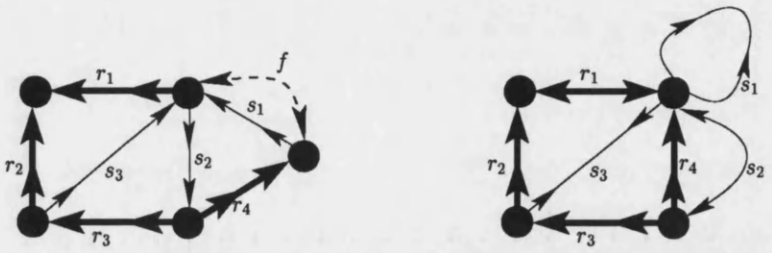


Figure 6.2: Inserting a bidirected edge f in a network and the graph obtained by contracting f

6.1.4 Interval matrices and their generalizations

Interval matrices [29] are well-known special network matrices, in which the spanning tree is a directed path. The non-tree edges then can be associated with subpaths. Obviously, the non-zero elements in each column of an interval matrix have the same sign, and there is a permutation of the rows such that the resulting matrix has consecutive non-zeros in each column. With appropriate scaling of the columns, the matrix can be made nonnegative, and then the columns can be considered as characteristic vectors of intervals on a line with finite segments. Hence the name interval matrix.

Lee [48] generalized interval matrices by defining *skew interval matrices*. In a skew interval matrix, the columns are of two types. There are columns which are characteristic vectors of intervals, just as in the case of interval matrices. The columns of the other type contain -1 in rows that correspond to segments to the left of an interval, 0 in the rows of the segments belonging to the interval, and 1 in rows whose segments are to the right of the interval. In the formal definition, we replace the segments by points, an obvious conversion.

Definition 6.4. (Lee [48]) Let $\mathcal{P}$ be a finite set of points on the real line, and let $\mathcal{I}$ and $\mathcal{J}$ be finite sets of intervals (defined by consecutive points) of the line. For each interval $I \in \mathcal{I} \cup \mathcal{J}$ define $L(I) = \{p \in \mathcal{P} \mid p < q, \forall q \in I\}$, and $R(I) = \{p \in \mathcal{P} \mid p > q, \forall q \in I\}$. The columns of a *skew interval matrix* A correspond to intervals in $\mathcal{I}$ and $\mathcal{J}$, its rows are associated with $\mathcal{P}$. The elements

of A are the following:

$$a_{pI} = \begin{cases} 1 & \text{if } p \in I & (I \in \mathcal{I}) \\ 0 & \text{if } p \notin I & (I \in \mathcal{I}) \\ -1 & \text{if } p \in L(I) & (I \in \mathcal{J}) \\ 0 & \text{if } p \in I & (I \in \mathcal{J}) \\ 1 & \text{if } p \in R(I) & (I \in \mathcal{J}) \end{cases}$$

Trivially, if $\mathcal{J} = \emptyset$, we get back the interval matrices (unsigned). Skew interval matrices, however, are not necessarily network matrices, and not even totally unimodular as the following matrix shows.

$$\begin{bmatrix} 1 & 0 & 0 & -1 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix}$$

A further step of generalization in Lee [48] is when zero-length intervals are also allowed. A matrix whose rows can be permuted and whose columns can be scaled by -1 to get a skew interval matrix with this generalized intervals is said to have the *skew consecutive property* (though 'cyclic consecutive property' would be a more descriptive name). That is, a matrix has the skew consecutive property if its rows can be premuted so that each column is of the form

$$\pm(0,0,\ldots,0,1,1,\ldots,1,0,0,\ldots,0)^T,$$
$$\text{or } \pm(-1,-1,\ldots,-1,0,0,\ldots,0,1,1,\ldots,1)^T,$$

where any of the continuous substrings may have zero length. Lee proved that a matrix with the skew consecutive property is 2-regular. Here we prove that they are binet.

Theorem 6.5. *If matrix B has the skew consecutive property, then it is binet.*

Proof: We give a binet representation. Let $r_1,\ldots,r_m$ and $s_1,\ldots,s_n$ denote the rows and columns of B. Let G be a binet graph in which edges $r_1,\ldots,r_m$ form the cycle $v_1, r_1, v_2, r_2, \ldots, r_m, v_m$ where $v_1 \ldots, v_m$ are the nodes of G. (See Figure 6.3 for an example.) Moreover, $s_1,\ldots,s_n$ are chords of the cycle, such that the two paths on the cycle connecting the end-nodes of a chord correspond to the rows of B with zeros and non-zeros, respectively. The cycle-edges are directed so that all nodes except v_1 have one incoming and one outgoing incident cycle-edges.

This setting implies that any fundamental circuit that goes through v_1 changes sign at this node, the other fundamental circuits have the same sign on all of its edges, representing the fact that the sign of elements in a column of B changes between the top and the bottom of the column. The exact directions of the edges can be easily adjusted to the signs of the non-zeros in the matrix. □

$B =$

	s_1	s_2	s_3	s_4	s_5
r_1	1	0	0	1	-1
r_2	1	-1	0	0	-1
r_3	1	-1	1	0	0
r_4	0	-1	1	-1	0
r_5	0	0	1	-1	1

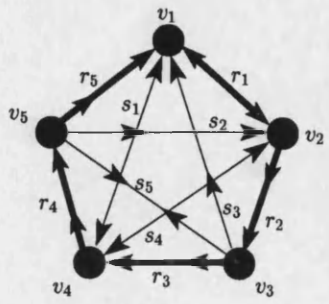


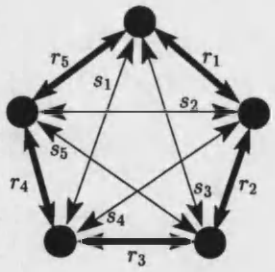
Figure 6.3: A matrix with the skew consecutive property, and its binet representation

6.1.5 Totally unimodular but not network matrices

Consider matrices B_1 and B_2 below. These are the two well-known (see e.g. [55]) totally unimodular matrices that play an important role in the decomposition theory of totally unimodular matrices, due to Seymour [58] (see Section 10.2.1). Neither they nor their transposes are network matrices. The graphs drawn next to them show that B_1 and B_2 are binet matrices. This implies that the binet representation of matrices is more powerful then the network representation.

$B_1 =$

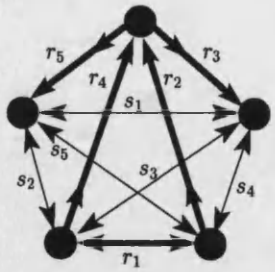
	s_1	s_2	s_3	s_4	s_5
r_1	1	0	0	1	-1
r_2	-1	1	0	0	1
r_3	1	-1	1	0	0
r_4	0	1	-1	1	0
r_5	0	0	1	-1	1



(6.1)

$B_2 =$

	s_1	s_2	s_3	s_4	s_5
r_1	1	1	1	1	1
r_2	1	1	1	0	0
r_3	1	0	1	1	0
r_4	1	0	0	1	1
r_5	1	1	0	0	1



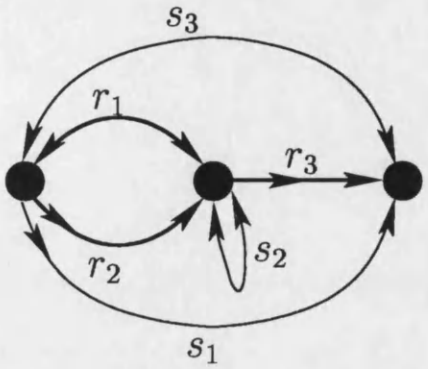
(6.2)

Note that up to row multiplications by -1, B_1 is the same as matrix B of Figure 6.3. No wonder that their binet representations are so similar.

6.1.6 Minimally non-totally unimodular matrices

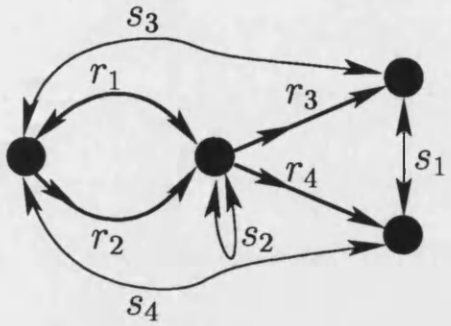
In [16], Cornuéjols gives four minimally non-totally unimodular 0, 1 matrices. To show the power of binet representations, we give the binet graphs of these matrices.

	s_1	s_2	s_3
r_1	0	1	1
r_2	1	1	0
r_3	1	0	1



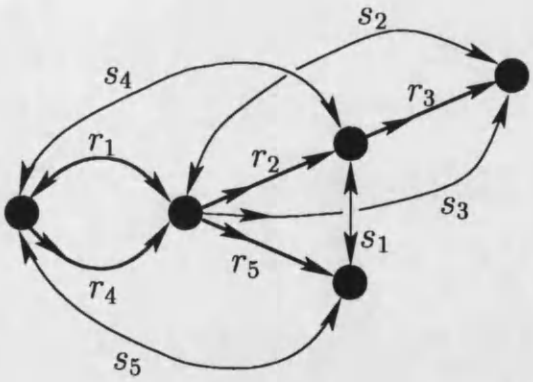
(6.3)

	s_1	s_2	s_3	s_4
r_1	1	1	1	1
r_2	1	1	0	0
r_3	1	0	1	0
r_4	1	0	0	1



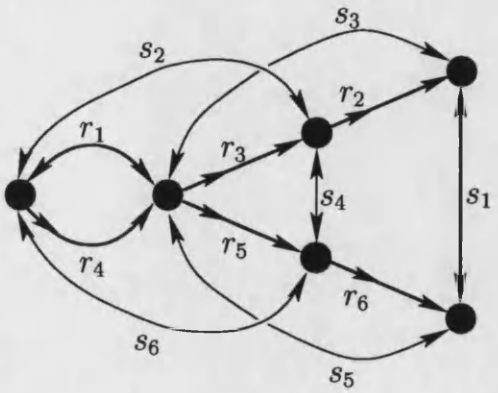
(6.4)

	s_1	s_2	s_3	s_4	s_5
r_1	1	1	0	1	1
r_2	1	1	1	1	0
r_3	0	1	1	0	0
r_4	1	1	0	0	0
r_5	1	0	0	0	1



(6.5)

	s_1	s_2	s_3	s_4	s_5	s_6
r_1	1	1	1	1	1	1
r_2	1	0	1	0	0	0
r_3	1	1	1	1	0	0
r_4	1	0	1	1	1	0
r_5	1	0	0	1	1	1
r_6	1	0	0	0	1	0



(6.6)

6.2 Counterexamples

The node-edge incidence matrix of a bidirected graph is binet. In this section we show that this is not necessarily true for the transpose, the edge-node incidence matrix of a bidirected graph. This result is not a surprise, as the transpose of a network matrix is not necessarily network either, so we should not expect the opposite for binet matrices. Nevertheless, our counterexamples are interesting because they also show that totally unimodular or 2-regular matrices are not necessarily binet.

As mentioned in Section 7.3, Gerards and Schrijver [33] gave a characterization of matrices that are edge-node incidence matrices of bidirected graphs and have strong Chvátal rank 1. The key matrix in their characterization is $M(K_4)$, the edge-node incidence matrix of K_4, the complete undirected graph on four points:

$$M(K_4) = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix}$$

The edge-node incidence matrices of bidirected graphs are exactly the integral matrices A (of size $m \times n$) that satisfy the transposed version of (4.2), namely,

$$\sum_{j=1}^{n} |a_{ij}| \leq 2 \quad \text{for } i = 1, \ldots, m \tag{6.7}$$

The characterization appearing in [33] is now the following.

Theorem 6.6. (Gerards and Schrijver) *An integral matrix satisfying* (6.7) *has strong Chvátal rank 1, if and only if it cannot be transformed to* $M(K_4)$ *by a series of following operations:*

(i) deleting or permuting rows or columns, or multiplying them by -1

(ii) replacing matrix $\begin{bmatrix} 1 & g \\ f & D \end{bmatrix}$ *by the matrix* $D - fg$.

The graphical equivalent of the matrix operations in Theorem 6.6 can also be given, following [33] and Section 4.1. Deleting a row or column of an edge-node incidence matrix is equivalent to deleting an edge or a node from the graph. Multiplying a row with -1 translates to reversing the direction of an edge, while multiplying a column with -1 corresponds to a switching. Operation (ii) has already appeared as (4.3), so it is the same as the contraction of an edge (note that (4.3) is symmetric to transposing). Thus, the edge-node incidence matrix of a bidirected graph has strong Chvátal rank 1, if and only if the graph cannot be transformed to K_4 by a series of edge and node deletions,

edge direction reversals, switchings and contractions. Theorem 7.9 states that every integral binet matrix has strong Chvátal rank 1. Combining it with Theorem 6.6 we get:

Theorem 6.7. *If a bidirected graph can be transformed to K_4 by a series of edge and node deletions, edge direction reversals, switchings and contractions, then its edge-node incidence matrix is not binet.*

Theorem 6.7 implies that the edge-node incidence matrix of any graph that has K_4 as a subgraph is not binet. For example, $M(K_6)$, the edge-node incidence matrix of the complete undirected graph on six nodes is such. The following lemma shows that we cannot sign the elements of $M(K_6)$ to make it binet. In other words, the edge-node incidence matrix of a bidirected graph obtained from K_6 by arbitrarily orienting the edges is not binet.

Lemma 6.8. *Let B be a binet matrix of size $m \times n$. If $m > 2n$, then*

(a) B has an all-zero row, or

(b) B has a row with exactly one non-zero, or

(c) B has two rows with corresponding elements having the same absolute value.

Proof: Let G be the bidirected matrix that represents B. Since B has m rows and n columns, G has m nodes and $m + n$ edges. Let $d(v)$ denote the degree of node v, i.e., the number of edge-ends incident to v. As the basis of the graph spans G, there is no node with zero degree. For the same reason, if $d(v) = 1$, then the only edge incident to v is a basic edge. The row corresponding to this edge is all-zero, as no fundamental circuit can go through the edge.

So let us assume that the degree of every node is at least 2. Let the number of half-edges in G be k. The sum of the node degrees is $\sum d(v) = 2(m + n - k) + k < 3m - k$. This implies that there must be a node (let us call it v) with degree 2 such that there is no half-edge incident to it. If a loop is incident to v, then it must be in the basis, and the corresponding row is all-zero. If two links, (say e and f) are incident to v, then at least one of them is in the basis. If one of the edges (say e) is basic and the other is non-basic, then there is only one fundamental circuit going through e, the one that corresponds to f. This means that B has a row satisfying condition (b). If both e and f are basic, then either both of them are in a basic cycle or neither of them. Furthermore, their rows can differ only in the signs of the non-zero elements as any fundamental circuit that goes through either of them goes through the other too. Thus rows e and f of B satisfy condition (c). □

Observe now that $M(K_6)$ has 15 rows and 6 columns, each row contains exactly two non-zeros, and the non-zeros are in different positions in different rows, so none of conditions (a), (b) and (c) can be satisfied for any signed version of $M(K_6)$. Therefore, $M(K_6)$ cannot be signed to a binet matrix. Note that if the edges of K_6 are oriented so that the graph is directed, then the corresponding edge-node incidence matrix is totally unimodular. Thus we have an example of a matrix which is totally unimodular, but not binet.

The counterexamples given so far were edge-node incidence matrices of bidirected graphs, i.e., they were transposes of binet matrices, hence 2-regular. A matrix that is not 2-regular cannot be binet. So the question arises: Are there matrices that are 2-regular, but neither binet, nor a transpose of a binet? The matrix $2I_2$, the 2×2 diagonal matrix with 2's in the diagonal, is such. It is clearly 2-regular, symmetric, and cannot be binet, because in any possible binet representation the basis must be connected and both basic edges are half-edges (otherwise we could not get only 2's and zeros in the matrix), which is impossible. Matrix $2I_2$ is twice a totally unimodular matrix, so it is trivially 2-regular. If our purpose is to prove the 2-regularity of a matrix, then it is enough to show that it is:

- twice a totally unimodular matrix, or
- a binet matrix, or
- the transpose of a binet matrix.

Is there any matrix which is 2-regular but does not fall into any of these categories? There is a trivial example, $[\frac{1}{2}, 2]$. No binet matrix can have a ± 2 and a $\pm\frac{1}{2}$ in the same row or column, for example because then pivoting on the $\frac{1}{2}$ would result in a ± 4. But is there any *integral* matrix that is not of the types above? The answer is yes.

Consider the following matrix.

$$A = \begin{bmatrix} 1 & 1 & 1 & 1 \\ -1 & 1 & 1 & -1 \\ 1 & 1 & -1 & -1 \end{bmatrix} \tag{6.8}$$

By careful analysis of all the cases, one can show that neither this matrix nor its transpose is binet. Alternatively, one can apply the MIP-based recognition approach described in Section 5.4. It is also easy to show that any submatrix of A is binet. The submatrices achieved by deleting rows are trivially such, as they satisfy (4.2). The 3×3 submatrices obtained by column deletions are binet matrices with a triangle basis and non-basic loops. As a consequence, matrix A is 2-regular, because any of its square submatrices is binet. Obviously, $A/2$ cannot be totally unimodular. The submatrices of A^T are also easily shown to be binet, so we get the following result.

Lemma 6.9. *The matrix A of* (6.8) *and its transpose are 2-regular, minimally non-binet matrices.*

6.3 Matrices with at most three non-zeros per row

Hochbaum [41] examines integer programs in which each constraint involves up to three variables. That is, a constraint that is not an upper bound on a variable can be of the forms:

$$a_i x_i + a_j x_j \leq b \quad \text{or} \quad a_i x_i + a_j x_j + z_{ij} \leq b$$

where a_i, a_j are rationals without any restriction on their signs. A further assumption is that variable z_{ij} appears in only one constraint. Following [41], we will call such a problem an *IP2 problem.* The generic matrix format of IP2 problems is the following:

$$\begin{aligned} &\max cx + dz \\ &\begin{bmatrix} A_1 & I \\ A_2 & 0 \end{bmatrix} \begin{pmatrix} x \\ z \end{pmatrix} \leq b \qquad \text{(IP2)} \\ &l \leq x \leq u \\ &l_z \leq z \leq u_z \end{aligned}$$

where matrices A_1 and A_2 contain at most two non-zeros in each row. It is also permitted to add further identity matrices while maintaining the results. That is, the constraint matrix can be of the form:

$$\begin{bmatrix} A_1 & I & \cdots & I \\ A_2 & 0 & \cdots & 0 \end{bmatrix}$$

A special IP2 problem, called *binarized,* is when the elements of the constraint matrix are all $0, \pm 1$. Hochbaum [41] concludes that a binarized IP2 problem always has half-integral basic solutions. In other words, she proves that the constraint matrix of (IP2) is 2-regular. Here we show that it is also the transpose of a binet matrix. So the applications given in [41] illustrates the use of binet matrices. These applications include the feasible cut problem, the complement of the maximum clique problem, the generalized independent set problem and the generalized vertex cover problem. For illustration, we describe the generalized independent set problem. Details of this and the other problems can be found in [41].

But let us first prove our claim about binetness.

Theorem 6.10. *The transpose of the constraint matrix in a binarized IP2 problem is binet.*

Proof: Matrix $A = \begin{bmatrix} A_1 \\ A_2 \end{bmatrix}$ has at most two non-zeros in each row, and these non-zeros are ± 1, so A^T is the node-edge incidence matrix of a bidirected graph. Hence, A^T is binet. According to Lemma 5.7, adding unit rows to a binet matrix maintains its binetness, so adding unit columns to A does not change the fact that it is the transpose of a binet matrix. □

The *generalized independent set problem* is the generalization of the well-known independent set problem. In the latter, the aim is to find a maximum weight node set in a graph $G(V, E)$, such that there is no edge between the selected nodes. In the generalized version, we permit edges, but at a

penalty. The IP2 formulation of the generalized independent set problem is:

$$\max \sum_{i \in V} w_i x_i - \sum_{(i,j) \in E} c_{ij} z_{ij}$$

$$\text{subject to} \quad x_i + x_j - z_{ij} \leq 1 \quad \forall (i,j) \in E$$

$$x_i, z_{ij} \in \{0,1\} \quad \forall i,j$$

Variables x_i ($i \in V$) represent nodes, $x_i = 1$ if and only if node i is selected. For each edge $e = (i,j)$, we have a variable z_{ij}, and the constraints of the model guarantee that if both end-points of e are selected, then $z_{ij} = 1$. The weight of node i is w_i, the penalty on edge (i,j) is c_{ij}.

A real-life application of the generalized independent set problem concerns the location of postal services [5]. We are given a set of potential location points and the utility value (weight) associated with each point. If two points are too close to each other, they compete for the same costumers, so their utility value is decreased. This is the penalty on the edge connecting these points. The goal is to find a set of locations that maximizes the utility value.

Chapter 7

Polyhedral results about binet matrices

In Chapter 3, we discussed the polyhedral theory related to k-regular matrices. The main result was that if A is k-regular, then the polyhedron $\{x \mid Ax \leq b,\ x \geq 0\}$ is integral for any vector b that is an integer multiple of k. We have also shown that the node-edge incidence matrix of a bidirected graph is 2-regular. In this chapter we extend this result to binet matrices by proving that they are also 2-regular. As a consequence, the polyhedron $P = \{x \mid Bx \leq b,\ x \geq 0\}$ is half-integral for any binet matrix B and integral vector b.

Unlike incidence matrices, binet matrices are not necessarily integral. We saw earlier that the integrality of a k-regular matrix has serious implications on the operations we can apply to a matrix preserving its k-regularity. For example, adding unit rows or pivoting can destroy the k-regularity of a non-integral matrix, but not if the matrix is integral. We show here that binet matrices do not have this disadvantage. In fact, from all important points of view, binet matrices behave as if they were integral 2-regular matrices. In polyhedral terms it means that we can impose upper or lower bounds on Bx and x in P without losing its half-integrality. These results about half-integral polyhedra are discussed in Section 7.1.

Section 7.2 deals with binet matrices that ensure not only half-integral, but integral polyhedra. In other words, we investigate the binet matrices that are also totally unimodular, and give a characterization of this kind of matrices.

Finally, in Section 7.3 we return to polyhedra that are not integral, and look at how the convex hull of the integer points can be achieved. It is known that if the constraint matrix is the node-edge incidence matrix of a bidirected graph, then this can be done simply by adding valid rank-1 inequalities. We show that the same holds for integral binet matrices. In other words, we prove that integral binet matrices have strong Chvátal-rank 1.

7.1 Half-integral polyhedra

In this short section we examine the polyhedron

$$P(B) = \{x \mid Bx \leq b,\ x \geq 0\} \tag{7.1}$$

in which B is a binet matrix. By introducing nonnegative variables $z = b - Bx$, the inequalities $Bx \leq b$ are converted to equalities $Bx + z = b$. Let $A = [S, R]$ be the node-edge incidence matrix of the bidirected graph representing B, with basis R and non-basic edges S. Thus, $B = R^{-1}S$. We can multiply the equality constraints $Bx + z = b$ by R, getting $Sx + Rz = Rb$. Thus, instead of $P(B)$ we can look at the following polyhedron.

$$P(A) = \{y \mid Ay = Rb,\ y \geq 0\} \tag{7.2}$$

Theorem 4.8 states that the node-edge incidence matrix of any bidirected graph is 2-regular, which implies that the extreme points of $P(A)$ are all half-integral if Rb is integral. As R is an integral matrix, this ensures the half-integrality of $P(B)$ for any integral b. By Corollary 3.11, an equivalent way of stating this fact is that B is 2-regular. We also give an alternative proof of this result, in terms of matrices.

Theorem 7.1. *Binet matrices are 2-regular.*

Proof: Let $B = R^{-1}S$ be a binet matrix, where $A = [S, R]$ is the incidence matrix of a bidirected graph. Then $[B, I] = R^{-1}A$. Let C be a non-singular square submatrix of B, and let $[B, I]$ be partitioned so that

$$[B, I] = \begin{bmatrix} D_2 & C & 0 & I_2 \\ D_3 & D_1 & I_1 & 0 \end{bmatrix} \quad \text{and let } E = \begin{bmatrix} C & 0 \\ D_1 & I_1 \end{bmatrix}$$

where I_1 and I_2 are unit matrices of appropriate size. There is a basis T of A such that $E = R^{-1}T$. A is 2-regular, so $2T^{-1}$ is integral, hence $2E^{-1} = 2T^{-1}R$ and then $2C^{-1}$ is integral too. □

In Section 2.2, it was shown that if a matrix is k-regular and integral, then adding unit rows or columns to it preserves its k-regularity, but this is not true in general for rational matrices. Binet matrices are not necessarily integral, but Lemmas 5.5 and 5.7 show that, in this sense, they behave as if they were integral, i.e., adding unit columns or rows to a binet matrix results in another binet matrix. A consequence of this fact is the following theorem.

Theorem 7.2. *Let B be a binet matrix. Then for all integral vectors l, u, a and b, the polyhedron $P = \{x \mid a \leq Bx \leq b,\ l \leq x \leq u\}$ is half-integral.*

Proof: By Lemmas 5.4 and 5.7, the matrix

$$\begin{bmatrix} B \\ -B \\ I \\ -I \end{bmatrix}$$

is binet, so it is 2-regular. Theorem 3.6 ensures the integrality of P. □

Thus a binet matrix ensures the half-integrality of a very broad set of polyhedra. Moreover, if we restrict ourselves to even vectors (i.e, b in (7.1) and l, u, a, b in P of Theorem 7.2 have even elements), then the polyhedra are integral. In the next section we take another approach and examine binet matrices that provide integrality for any vector, not only for even ones.

7.2 Integral polyhedra

The polyhedron $P(B)$ of (7.1) is integral for all integral vectors b if and only if B is 1-regular. We will give a sufficient condition for the 1-regularity of a binet matrix, but it is a more interesting question to ask (and we can give a more elegant answer) when a binet matrix is totally unimodular. First we provide a condition that is sufficient for total unimodularity.

Lemma 7.3. *If $B = R^{-1}S$ is a binet matrix which has a binet representation G, such that*
(a) all the bidirected edges in G are half-edges, or
(b) G does not contain half-edges, and all the possible bases of G are connected,
then B is totally unimodular.

Proof: We will use the notations introduced in the proof of Theorem 7.1, and show that if the conditions of this theorem are satisfied, then the inverse of each non-singular submatrix C of B is integral, so that B is 1-regular. Since B is integral by Lemma 5.12, this is equivalent to its total unimodularity.

If there are no bidirected edges in G except half-edges, then all the cycles of any basis T are half-edges, so T^{-1} is integral by Lemma 6.1. In case (b), T is connected, so a row of T^{-1} contains only $\pm\frac{1}{2}$ entries, or $0, \pm 1$ entries. There are no half-edges in $G(R)$, so the sum of absolute values in each column of R is 2. Thus $T^{-1}R$ is an integral matrix. □

This lemma proves the total unimodularity of network matrices, which clearly satisfy condition (a) (see Section 6.1.3), and that of matrices B_1 and B_2 of (6.1) and (6.2). The binet representations of B_1 and B_2 do not contain half-edges and since the binet graph is a complete graph on five points, all the possible bases must be connected. So they satisfy condition (b). For network matrices, condition (a) can be shown necessary too.

Lemma 7.4. *Binet matrix B has a representation in which all the bidirected edges are half-edges if and only if B is a network matrix.*

Proof: If B is a network matrix, then with the transformation given in Section 6.1.3 and demonstrated in Figure 6.1 we get a binet representation which has only half-edge bidirected edges.

To see the converse, simply take the inverse of this transformation. That is, insert a new node in the graph, connect all the half-edges to this node by making them directed links. The bidirected graph becomes a directed one, and the basis becomes a tree. □

Conditions (a) or (b) of Lemma 7.3 are sufficient for the total unimodularity of a binet matrix. We prove that they are also necessary. First we show that if we know that a binet matrix is totally unimodular, then a stronger version of Lemma 7.4 holds.

Lemma 7.5. *Let us suppose that a binet matrix B is totally unimodular. Then it is a network matrix if and only if it has a binet representation in which each basic cycle is a half-edge.*

Proof: The necessity clearly follows from Lemma 7.4. For sufficiency we also use that lemma, as we will show that there is a binet representation of B which contains only directed links and half-edges.

Let G be the binet representation of B in which each basic cycle is a half-edge, and the non-cycle basic edges are directed. Such a representation exists by Lemma 5.10. Let $C_1, C_2, \ldots, C_k$ be the components of G and $l_1, l_2, \ldots, l_k$ be the basic half-edges in the components. Since B is totally unimodular, it does not contain ± 2 entries, so by Algorithm 1, a non-basic edge of G is either a half-edge, or a directed link, or a bidirected link connecting two components. (See Figure 7.1.) We construct an auxiliary bidirected graph H by contracting each basic component to a node. That is, H is defined in the following way: it has k nodes labelled $C_1, C_2, \ldots, C_k$, and for each edge s of G connecting two basic components there is an edge, also labelled with s, between the corresponding nodes of H.

We show that H cannot have an odd cycle. Let us suppose the contrary, and assume that $C_{i_1}, s_{i_1}, C_{i_2}, s_{i_2}, \ldots, C_{i_t}, s_{i_t}, C_{i_1}$ is an odd cycle. Consider the submatrix B_1 of B whose rows correspond to $l_{i_1}, \ldots, l_{i_t}$ and columns correspond to $s_{i_1}, \ldots, s_{i_t}$. This matrix has exactly two non-zeros, being ± 1, in each column. By permuting rows and columns and multiplying them with -1, it can be transformed to form (4.4), so, according to Proposition 4.2, its determinant is 0 or ± 2. Since B is totally unimodular, $|det(B_1)|$ cannot be 2. On the other hand, B_1 is obtained from B by row and column deletions, so it is a binet matrix, $B_1 = R_1^{-1} S_1$, the non-basic edges of which form an odd cycle (recall from Proposition 4.3 that switching, which is applied in contractions, does not change the parity of a cycle). By Lemma 4.4, matrix S_1 is non-singular, therefore $det(B_1) \neq 0$. We arrived to a contradiction, so H cannot contain odd cycles.

According to Lemma 4.5, H can be transformed to a directed graph by switchings. (For our example graph H of Figure 7.1, the resulting directed graph H' is depicted in Figure 7.2.) This implies

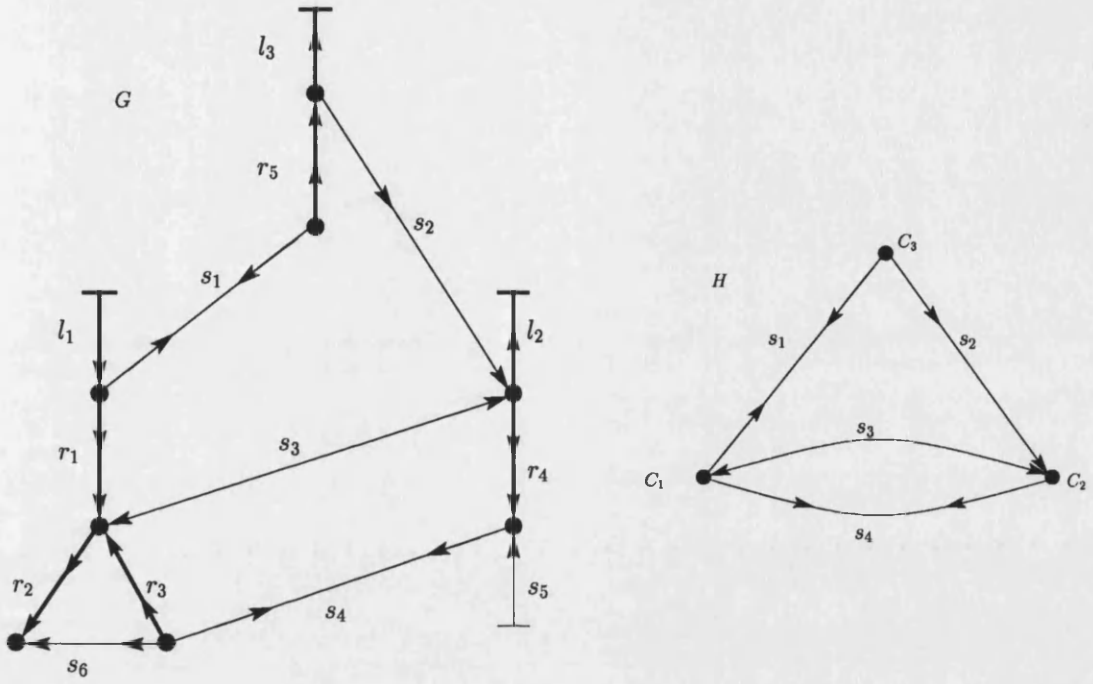


Figure 7.1: Example binet graph G and auxiliary graph H for Lemma 7.5 with basic edges in bold

a transformation of G to G': we switch at each node of component C_i in G if we switch at node C_i in H. At the end of the transformation, all the edges connecting components in G' are directed, while the edges within the component remain directed links and half-edges. Hence G' contains only directed links and half-edges, and Lemma 7.4 provides us with a network representation. □

Theorem 7.6. *A binet matrix B is totally unimodular, if and only if it has a binet representation G satisfying condition (a) or (b) of Lemma 7.3.*

Proof: The sufficiency was proved in Lemma 7.3, we prove necessity here. Let us suppose that B is totally unimodular. If it is a network matrix, then by Lemma 7.4 it falls in category (a). If B is totally unimodular but not a network matrix, then by Lemma 7.5, all of its binet representations G contain a basic cycle which is not a half-edge. The matrix is integral but this representation does not satisfy condition (a) of Lemma 5.12, so the basis must be connected and the binet graph cannot contain half-edges. By pivoting on B, we get another totally unimodular, non-network binet matrix, whose binet representation also has to have a connected basis. Any possible basis graph of G can correspond to the basis of such a matrix, hence all the possible bases are connected. We know that G cannot contain half-edges, so it satisfies condition (b). □

It is easy to decide if a bidirected graph satisfies condition (a) of Lemma 7.3 because we have only to check the bidirected edges. To check condition (b), all the subgraphs that can be a component of a basis should be taken into account. Every basic component contains an odd cycle, so odd cycles

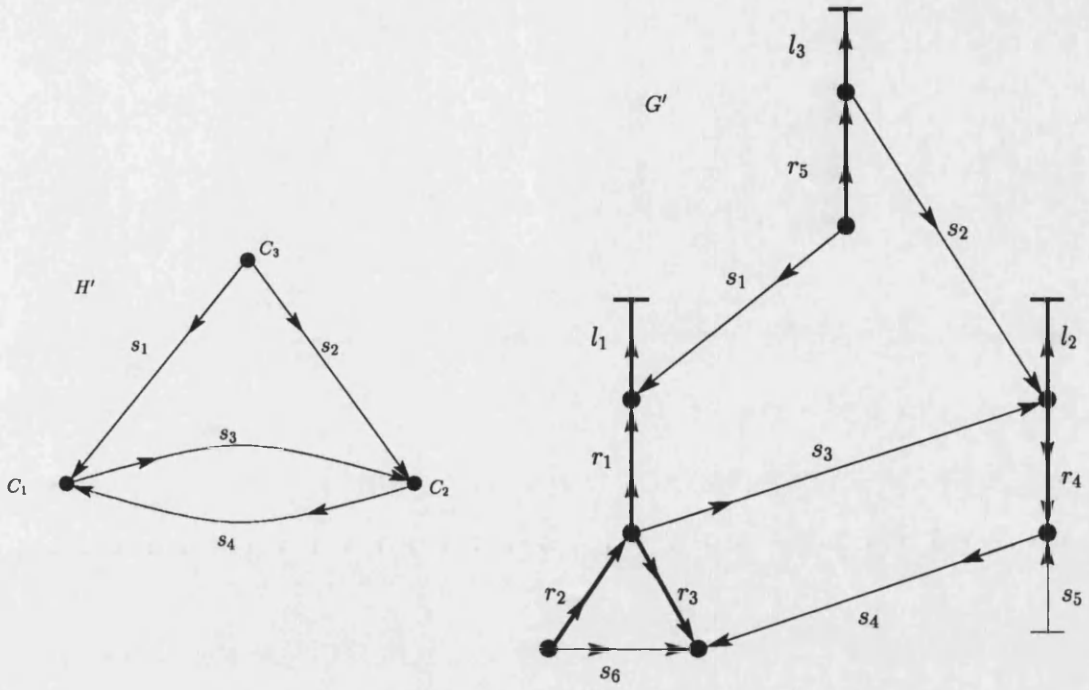


Figure 7.2: The graph H' obtained from H by a switching at node C_1, and the corresponding graph G'

are the most important subgraphs. It is obvious that a connected bidirected graph can have an unconnected basis if and only if it has two node-disjoint odd cycles. Note that we mention a matroid version of Theorem 7.6 at the end of Section 10.2.2.

Let us now give the promised sufficient condition for the 1-regularity of a binet matrix. The proof of this result does not use any new ideas as compared to that of Lemma 7.3, so we skip it. For simplifying the already complex conditions, we use T and R denoting both submatrices of the incidence matrix and the corresponding edge-sets of the graph.

Lemma 7.7. *Let $B = R^{-1}S$ be a binet matrix, and $A = [S, R]$ be the node-edge incidence matrix of a representation graph. B is 1-regular if there is no basis T of A such that*
(a) there is a basic cycle edge in $T \setminus R$ which is not a half-edge, and $R \setminus T$ contains a half-edge, or
(b) T is not connected, $R \setminus T$ connects two components of T, and at least one of these components has a basic cycle that is not a half-edge and contains an edge from $T \setminus R$.

Note that in Section 6.1.2 we showed that if R is a basis of the node-edge incidence matrix of a bidirected graph, then R^{-1} is a 1-regular but not necessarily totally unimodular matrix. We can also prove the 1-regularity of R^{-1} by Lemma 7.7. Matrix S is a unit matrix in this case, so $A = [I, R]$. It follows that for any T basis of A, $T \setminus R$ contains only half-edges, so neither of the conditions can hold.

The total unimodularity of a binet matrix B ensures integral extreme points for all integral right

hand side vectors in $P = \{x \mid Bx \leq b,\ x \geq 0\}$. For special right hand sides, however, total unimodularity is too strong a requirement. For example, for even b vectors we do not need to assume anything about a binet matrix, its 2-regularity will guarantee the integrality of P. If we know a binet representation of B, then the parity requirement on b can be relaxed.

Theorem 7.8. *Let B be a binet matrix. Polyhedron $P = \{x \mid Bx \leq b,\ x \geq 0\}$ is integral if b satisfies the following conditions*

(a) the elements of b that correspond to non-cycle basic edges are even,
(b) if the cycle of a component is not a half-edge, then the elements of b that correspond to its edges are all odd or all even.
(c) if the cycle of a component is a half-edge, then the element of b that corresponds to it is even.

Proof: For the integrality of P it is enough to prove that the polyhedron $P(A)$ of (7.2) is integral, where A is the node-edge incidence matrix of the binet graph representing B and R is the basis of A.

We show that for every vector b that satisfies the conditions of the theorem, Rb is even. Since A is the incidence matrix of a bidirected graph, it is 2-regular by Theorem 4.8, so that Rb being even is enough for the integrality of $P(A)$.

According to condition (a), the elements of b corresponding to non-cycle edges are all even, so it is enough to show that $R_1 b_1$ is even where R_1 is the submatrix of R corresponding to the basic cycles and b_1 is the part of b corresponding to the cycle edges. Consider an arbitrary component of R_1. It is either a 1×1 matrix of ± 1 or ± 2, if the cycle is a half-edge or a loop, respectively, or contains exactly two non-zeros (being ± 1) in each row. Thus, conditions (b) and (c) ensure that $R_1 b_1$ is even. □

7.3 The Chvátal rank of binet matrices

If polyhedron $P = \{x \mid Ax \leq b,\ x \geq 0\}$ is not integral, then the convex hull P_I of its integer points does not coincide with P. If this is the case, then P_I can be achieved by an iterative process: we first add inequalities of the form $cx \leq \lfloor \delta \rfloor$ where c is an integral vector and δ is a scalar such that $cx \leq \delta$ is valid for all $x \in P$. The intersection of P with the half-spaces induced by all the possible inequalities of the above form is its *rank-1 closure*, denoted P_1. (Compare this with the Chvátal-Gomory cuts discussed in Section 3.4.)

It is trivial that $cx \leq \lfloor \delta \rfloor$ is satisfied by all points of P_I (that is why, we call these inequalities *valid inequalities*, meaning that they are valid for P_I), so $P_1 \supseteq P_I$. If $P_1 \neq P_I$, then we repeat the process with P_1 instead of P, getting P_2, the rank-2 closure of P. We carry on with adding new valid inequalities until we reach the integer hull. It is known (see Chvátal [14]) that P_I is achieved after a finite number of iterations. The question is, how many of these iterations are needed. This is not only a theoretical question, since several algorithms exist that employ valid inequalities to find an

integer point of P. Obviously, polyhedra for which one iteration is enough, i.e., $P_1 = P_I$, are easier to handle than those where one has to use higher-rank closures.

Even better are those polyhedra for which this desirable property remains valid for a whole range of right hand side vectors b. If $P_1 = P_I$ for all possible integral b, then this is clearly a property of the matrix A. Matrices A for which the integer hull P_I of $P = \{x \mid Ax \leq b,\ x \geq 0\}$ is the same as the rank-1 closure P_1 for any integral b are said to have *Chvátal rank 1*. The name represents the fundamental work of Chvátal in this area.

A stronger requirement is to assume that we have lower and upper bounds on Ax and x, so the polyhedron is of the form $P = \{x \mid l \leq x \leq u,\ a \leq Ax \leq b\}$. Matrix A has *strong Chvátal rank 1*, if $P_1 = P_I$ for any integral choice for l, u, a and b (including ∞).

The theory of valid inequalities is a fundamental part of combinatorial optimization and integer programming, and presented in every textbook on these subjects. There is an extensive coverage in [51] and [55]. The latter book also contains a short discussion on Chvátal ranks. The interested reader can find a rigorous description and references in these books.

Edmonds and Johnson [25, 26] showed that if A is the node-edge incidence matrix of a bidirected graph, then it has strong Chvátal rank 1. (That is why matrices with strong Chvátal rank 1 are sometimes said to have the *Edmonds-Johnson property*.) In [33], Gerards and Schrijver treated the transpose of these matrices, the edge-node incidence matrices of bidirected graphs. They proved that the edge-node incidence matrix of a bidirected graph has strong Chvátal rank 1, if and only if the graph cannot be transformed to K_4, the complete undirected graph on four points, by a series of edge and node deletions, edge direction reversals, switchings and contractions. (See Theorems 6.6 and 6.7.)

Here we extend the set of matrices with strong Chvátal rank 1 by proving that integral binet matrices are such.

Theorem 7.9. *If B is an integral binet matrix, then it has strong Chvátal rank 1.*

Proof: Let $B = R^{-1}S$ and $A = [S, R]$ be the node-edge incidence matrix of the bidirected graph representing B. Let P be the polyhedron $\{x \mid l \leq x \leq u,\ a \leq Ax \leq b\}$ with integral l, u, a, b. Let us introduce new variables $z = -Bx$. This gives:

$$\bar{P} = \left\{ \begin{pmatrix} x \\ z \end{pmatrix} : l \leq x \leq u,\ -b \leq z \leq -a,\ Bx + z = \mathbf{0} \right\}$$

Multiplying the equality constraints by R we get

$$\bar{P} = \left\{ \begin{pmatrix} x \\ z \end{pmatrix} : l \leq x \leq u,\ -b \leq z \leq -a,\ [S, R] \begin{pmatrix} x \\ z \end{pmatrix} = \mathbf{0} \right\} \tag{7.3}$$

Obviously, $x \in P$ if and only if $\begin{pmatrix} x \\ -Bx \end{pmatrix} \in \bar{P}$.

Let $y \in P$ not be in P_I. We show that then y cannot be in P_1. This implies that $P_1 \subseteq P_I$, which suffices to prove the theorem.

It is easy to show that $\begin{pmatrix} y \\ -By \end{pmatrix} \notin \bar{P}_I$. Because A has strong Chvátal rank 1, there exists an integral vector $\bar{c} = (c_1, c_2)$ and scalar δ such that $(c_1, c_2)\begin{pmatrix} x \\ z \end{pmatrix} \leq \delta$ is valid for all $\begin{pmatrix} x \\ z \end{pmatrix} \in \bar{P}$ and $(c_1, c_2)\begin{pmatrix} y \\ -By \end{pmatrix} > \lfloor \delta \rfloor$. Hence for the integral vector $c = c_1 - c_2 B$, inequality $cx \leq \delta$ is valid for all $x \in P$ and $cy > \lfloor \delta \rfloor$. This means that y is not in the rank-1 closure of P. □

Note that Theorem 7.9 cannot be extended to rational binet matrices, as the following example shows.

$$B = R^{-1}S = \begin{bmatrix} ½ & 1 & 0 \\ 0 & 1 & 1 \\ ½ & 0 & 1 \end{bmatrix} \text{ with } R = \begin{bmatrix} 1 & 0 & -1 \\ 1 & -1 & 1 \\ 0 & 1 & 0 \end{bmatrix}, S = \begin{bmatrix} 0 & 1 & -1 \\ 1 & 0 & 0 \\ 0 & 1 & 1 \end{bmatrix}.$$

Binet matrix B does not have strong Chvátal rank 1, because the non-zero integral solutions of the polyhedron $P = \{x : 0 \leq x \leq 1,\ 0 \leq Bx \leq 1\}$ are: $(1,0,0)$, $(0,1,0)$ and $(0,0,1)$, so $x_1 + x_2 + x_3 \leq 1$ is a facet of P_I. But $(1, ½, ½) \in P$, so $\delta = 2$ is the smallest value for which $x_1 + x_2 + x_3 \leq \delta$ is valid for P.

Compare Theorem 7.9 with Theorem 3.20. The latter theorem claims that if A is a 2-regular matrix of size $m \times n$, then for polyhedron $Q = \{x \mid Ax \leq b, x \geq 0\}$ with integral b, the rank-1 closure Q_1 can be achieved by only half-integral cuts, i.e., valid inequalities of the form $\lambda Ax \leq \lfloor \lambda b \rfloor$ where $\lambda \in \{0, ½\}^m$ and λA is integral. In a compact form, this result states that $Q_1 = Q_{½}$ for 2-regular matrices and any integral right hand side vector. As binet matrices are 2-regular, we immediately get the following corollary.

Corollary 7.10. *If B is an integral binet matrix and b is an integral vector, then the integer hull of $Q = \{x \mid Bx \leq b, x \geq 0\}$ can be achieved by half-integral cuts, i.e., $Q_I = Q_{½}$.*

Proof: By Theorem 3.20, $Q_1 = Q_{½}$, and by Theorem 7.9, $Q_1 = Q_I$. □

This result has an interesting consequence in separation. The $\{0, ½\}$-*separation problem*, as defined in [11], is the following:

> Given $x \in Q = \{x \mid Ax \leq b, x \geq 0\}$, decide if x is in $Q_{½}$ or not, and in the latter case, find a half-integral cut that separates it, i.e., a $\lambda \in \{0, ½\}^m$ such that $\lambda A \in \mathbb{Z}^n$ and $\lambda Ax > \lfloor \lambda b \rfloor$.

It is well known (see e.g., [55]), that the separation problem is polynomially equivalent to the optimization problem. In the special case of $\{0, ½\}$-separation, it means that if we can optimize linear functions over $Q_{½}$ in polynomial time, then we can decide the separation question in polynomial time. We will show in Chapter 8 that the integer opimization with integral binet constraint matrices is polynomial, since it is equivalent to a matching problem, so we have the following consequence of Corollary 7.10:

Corollary 7.11. *If B is an integral binet matrix, then the $\{0, ½\}$-separation problem can be solved in polynomial time.*

If A or its tranpose is a network matrix, then the $\{0, ½\}$-separation is trivial, as $Q_{½} = Q$. This is because for totally unimodular matrices $Q_I = Q$, and for any polyhedron $Q_I \subseteq Q_{½} \subseteq Q$. Corollary 7.11 extends this result to integral binet matrices.

Chapter 8

Optimization with binet matrices

In the previous chapter we reviewed the polyhedral results about binet matrices. It turned out that a binet matrix B ensures half-integral extreme points of the polyhedron $P = \{x \mid l \leq x \leq u,\ a \leq Bx \leq b\}$ for any integral vector l, u, a and b. This statement can be rephrased in the following manner.

Theorem 8.1. *Let B be an $m \times n$ binet matrix, $l, u \in (\mathbb{Z} \cup \{\pm\infty\})^n$, $a, b \in (\mathbb{Z} \cup \{\pm\infty\})^m$, $c \in \mathbb{R}^n$. The optimization problem*

$$\max\{cx \mid l \leq x \leq u,\ a \leq Bx \leq b\} \tag{8.1}$$

has half-integral optimal solutions.

In what follows, we will call (8.1) the *binet optimization problem.* In this and the next chapter we deal with methods that are able to find the optimal solution to a binet optimization problem.

We start with the continuous case, i.e, the optimization problem where we are looking for a solution $x \in \mathbb{R}^n$; then we discuss the integer binet optimization problem, in which there are integrality conditions on the variables. Note that in a network optimization problem (that is, if B is a network matrix), the continuous and integer case coincide. This is a consequence of the total unimodularity of the network matrix, which ensures that for integral vectors l, u, a, b in (8.1) the solution vector is also integral. Furthermore, this integral solution can be found in polynomial time by the network simplex algorithm.

In the binet optimization problem, the two cases should be handled separately. We will see, however, that there are results similar to those in the network optimization problem. Notably, we can give polynomial algorithms for solving both the continuous and integer binet optimization problems. Moreover, we show that there exists a binet counterpart of the network simplex algorithm.

In Section 8.1 we treat the continuous binet optimization. We discuss two methods for this problem. The first is a general-purpose strongly polynomial algorithm that is known to work for any matrix whose elements are not too large. The second method is tailored for binet matrices, as it

makes use of the underlying combinatorial structure. We devote a whole chapter to this latter method because it exhibits very elegant parallelisms with the network simplex method. So the continuous binet optimization methods are given in two parts, in Section 8.1 of this chapter, and in Chapter 9.

Section 8.2 is about the integer binet optimization problem. We show that this problem can also be solved in polynomial time, by reducing it to another important part of combinatorial optimization, the matching problem.

8.1 Continuous binet optimization

Recall that we deal with the problem of

$$\max\{cx \mid l \leq x \leq u,\ a \leq Bx \leq b\} \tag{8.1}$$

where B is a binet matrix. Note that in this case we do not assume the integrality of vectors l, u, a and b.

The problem is a linear programming problem, so general-purpose LP solving methods can be applied to it. The obvious choice is to solve (8.1) with the simplex method. It would be unwise, however, not to capitalize on the special structure of B. This is because, knowing that the network simplex method is much more efficient than the generic simplex method for network problems, we have every hope that the bidirected graph underlying B will help us to find shortcuts in the simplex method applied to a binet optimization problem. In Chapter 9 we describe an algorithm based on this concept and show how the ideas of the network simplex algorithm can be adapted to bidirected graphs to get a binet simplex method. It is also mentioned there that the binet simplex method is very efficient in practice, though not polynomial in the worst case.

Polynomial methods for solving LP problems, like Khachiyan's or Karmarkar's algorithm, do exist. The interested reader can find a description of these methods along with references in [51] or [55]. The main drawback of these algorithms, besides that they cannot really compete with the simplex method in most of the practical applications, is that the number of iterations they execute is a function of the size of the input data. In other words, an input with larger numbers not only causes more work in the arithmetic operations, but also requires more of these operations. By arithmetic operation we mean an addition, substraction, multiplication, division or comparison. An algorithm avoiding this disadvantage is called *strongly polynomial.* More formally, in a strongly polynomial algorithm the number of arithmetic operations is a polynomial function of only the dimensions of the problem, and not the size of the input, and the operations are carried out on rationals of size polynomially bounded by the size of the input.

It is a very important result of Tardos [62] that there exists an LP solution algorithm that is almost strongly polynomial.

Theorem 8.2. (Tardos) *There exists an algorithm which solves a given rational LP problem* $max\{cx \mid Ax \leq b\}$ *in at most* $p(size(A))$ *arithmetic operations which operate on numbers of size polynomially bounded by* $size(A, b, c)$ *for some polynomial function* p.

Tardos' theorem implies a strongly polynomial algorithm for LP problems in which the elements of the constraint matrix are bounded. This is the case, for example, for network matrices, which have elements $\{0, \pm 1\}$, or binet matrices, which can contain elements $\{0, \pm ½, \pm 1, \pm 2\}$.

Corollary 8.3. (Tardos, [61]) *There is a strongly polynomial algorithm to solve a network optimization problem.*

Corollary 8.4. *There is a strongly polynomial algorithm to solve a binet optimization problem in continuous variables.*

Tardos' algorithm relies on Khachiyan's method, so it is not necessarily a practically efficient approach. In Chapter 9, we describe a method with reversed characteristics, it works very well in practice, but it is not polynomial in theory.

8.2 Integer binet optimization

Now we turn to the integer case, that is, we are to solve the following problem.

$$\max\{cx \mid l \leq x \leq u,\ a \leq Bx \leq b,\ x \text{ integral}\} \tag{8.2}$$

in which B is an integral binet matrix and l, u, a, b are integral vectors. Let us suppose that $B = R^{-1}S$ where $A = [S, R]$ is the node-edge incidence matrix of the bidirected graph representing B. By introducing a new variable vector $z = -Bx$, we transform (8.2) to

$$\max\{cx + 0z \mid l \leq x \leq u,\ -b \leq z \leq -a,\ Bx + z = 0,\ x \text{ integral}\} \tag{8.3}$$

We assumed that B is an integral matrix, so z can also be required to be integral. We have equality constraints in (8.3), so we can multiply them by R, getting

$$\max\{cx + 0z \mid l \leq x \leq u,\ -b \leq z \leq -a,\ Sx + Rz = 0,\ x, z \text{ integral}\} \tag{8.4}$$

By translations, i.e., substituting $x - l$ and $z + b$ for x and z, respectively, (8.4) can be brought to the form of

$$\begin{aligned} &\max wx \\ \text{subject to}\quad &Ax = b \\ &0 \leq x \leq c \\ &x \text{ integral} \end{aligned} \tag{8.5}$$

Note that x, b and c here denote different objects than above in (8.4). We use the same letters because (8.5) can be viewed as a *bidirected network flow problem*, where A is the node-edge incidence matrix of the bidirected graph, x represents the flow on the edges, w is the weight vector, c is the capacity vector and b represents the net supply (or demand) at the nodes. The problem (8.5) is then to find the maximum weight integer flow that satisfies the net supply and the capacity conditions. This problem was introduced by Edmonds [23]. He also showed that the bidirected network flow problem is equivalent to the matching problem (see also [47]).

A *general matching problem* is defined on an undirected graph $G(V, E)$, which can contain loops. The set of edges and loops incident to a node v is denoted by $\delta(v)$ and $\lambda(v)$, respectively. We assume that there are weights associated with the edges. Moreover, $c_e \in \mathbb{R} \cup \{\infty\}$ denotes the capacity of edge e, and $a_v, b_v \in \mathbb{R} \cup \{\infty\}$ are lower and upper degree bounds on node v. The aim is to find a minimum or maximum weight integral vector x satisfying

$$\begin{aligned} a_v \leq \sum_{e \in \delta(v)} x_e + 2 \sum_{e \in \lambda(v)} x_e \leq b_v \qquad & \forall v \in V \\ 0 \leq \qquad x_e \qquad \leq c_e \qquad & \forall e \in E \end{aligned}$$

If $a_v = b_v$ for all $v \in V$, then the matching problem is called *perfect*. If $a = 0$ and $b_v = 1$ for all nodes, then we get the classical matching problem, in which there must be at most one incident edge to every node. If $a = 0$ but b is arbitrary, then it is a *b-matching* problem.

For a comprehensive survey on matchings, see Gerards [31]. The definitions given above and the result mentioned below are also taken from this work.

Edmonds proved that a general matching problem can be reduced to a weighted perfect matching problem via network flows. Combining this result with the polynomial algorithm for weighted matchings due to Edmonds [22], and Tardos' strongly polynomial algorithm for the network problems (see Corollary 8.3), we get:

Theorem 8.5. *There exists a strongly polynomial algorithm for the general matching problem.*

Edmonds and Johnson [25] showed that this result also holds for general matching problems on bidirected graphs. Thus, as the integer binet optimization problem is essentially a bidirected network flow problem and the bidirected network flow problem is in effect a general matching problem, we have the following result.

Corollary 8.6. *There is a strongly polynomial algorithm to solve the integer binet optimization problem.*

To provide a feel for the equivalence of general matching and bidirected network flow problems, we describe, after Lawler [47], how a bidirected graph without half-edges can be transformed to an undirected graph with loops. The bidirected flow problem (8.5) is then converted by this transformation to a general matching problem.

Let G be a bidirected graph and A be its node-edge incidence matrix of size $m \times n$. We define the undirected graph G' in the following way. For each node v of G, let G' have two nodes v' and v''. For each edge e of G, there is an edge e' of G'. If v is the tail of e, then v' is an end-node of e' and if v is the head of e, then e' is incident to v''. Furthermore, there is an edge f_v in G' between any pair (v', v''). Figure 8.1 shows a small example.

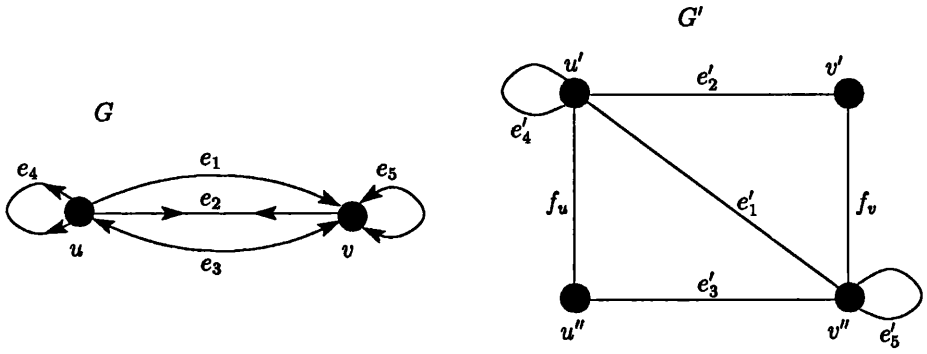


Figure 8.1: The conversion of a bidirected graph to an undirected one.

The node-edge incidence matrix A' of G' is then of size $(2m) \times (m+n)$. The incidence matrices of the graphs depicted in Figure 8.1 are the following:

	e_1	e_2	e_3	e_4	e_5
u	-1	-1	1	-2	0
v	1	-1	1	0	2

	e'_1	e'_2	e'_3	e'_4	e'_5	f_u	f_v
u'	1	1	0	2	0	1	0
v'	0	1	0	0	0	0	1
u''	0	0	1	0	0	1	0
v''	1	0	1	0	2	0	1

The weight and capacity of e' is the same as those of e, the weight of any f_v is 0, its capacity is ∞. Moreover, let b' and b'' be appropriately large vectors such that $b = b'' - b'$. They represent degree constraints associated with nodes v' and v'' for all v. Thus, (8.5) is transformed to the following capacitated perfect b-matching problem by introducing variables y for the e' edges and variables s for f_v edges.

$$\begin{aligned} &\max wx \\ \text{subject to } &A' \begin{pmatrix} y \\ s \end{pmatrix} = \begin{pmatrix} b' \\ b'' \end{pmatrix} \\ &0 \le y \le c, \quad s \ge 0 \\ &y, s \text{ integral} \end{aligned} \tag{8.6}$$

It is easily seen that solving (8.6) we get a solution to (8.5), and vice versa, so the bidirected network flow problem (8.5) and the general matching problem (8.6) are really equivalent.

Chapter 9

Generalized networks and the binet simplex method

Besides their elegant definition and important theoretical and practical advantages, network matrices are also prized for the possibility they offer to solve linear programs efficiently. In fact, if the constraint matrix of a linear programming (LP) problem is a network matrix, then the optimum can be found by using the *network simplex method*, a successful cross between the algebra of the simplex method and the combinatorics of the flow algorithms. It has been reported in the literature (e.g., in [35]) that the network simplex method can be up to 200 times faster than the general-purpose LP codes. Moreover, when considering the theoretical worst case running time, the network simplex method can be executed in polynomial time, in contrast to the exponential complexity of the simplex method.

One of the goals of this work is to investigate the extent to which the properties of network matrices can be extended to binet matrices, so it is natural to look for a bidirected version of the network simplex method. Such a method would exploit the graphical structure underlying binet matrices to solve LP problems where the constraint matrix is binet. As we will show in this chapter, a method that can achive this goal does exist, by the name of *generalized network simplex method.* This method is the adaptation of the techniques of the network simplex method to generalized networks and, in its most general form, can be applied to LP problems in which the constraint matrix has at most two non-zeros in each column. This covers the node-edge incidence matrices of bidirected graphs, so the genaralized network simplex method is indeed the bidirected version of the network simplex method. By the introduction of binet matrices, the range of constraint matrices for which the generalized network simplex method can be applied is substantially widened – the main new result of this chapter. To do so, we will show that the existing generalized network algorithm can be applied to bidirected graphs, and that an LP problem with a binet constraint matrix can be reduced

to a generalized network flow problem.

In Section 9.2 we describe the generalized networks and the generalized network simplex method. In doing so, we will use the notions introduced for the network simplex method in Section 9.1. We are fortunate enough to be able to skip the technical details of this latter method as they can be found in many good textbooks. Therefore we will focus only on the structural results that can be extended to the more general case. Section 9.2 also contains links to other parts of this dissertation.

9.1 Simplex method for network problems

The network simplex method is a specialized version of the general-purpose simplex algorithm, taking into account the underlying combinatorial structure. It can be introduced in basicly two ways. The first begins with the simplex method and shows how the special structure of the constraint matrix simplifies the calculations. This is the classical approach, initiated by Dantzig [19], who recognized the suitability of the simplex method for solving network problems. Most of the textbooks, like Dantzig [20], Nemhauser and Wolsey [51], or Murty [50] follow this approach. The second approach, which was taken, e.g., in the book of Ahuja, Magnanti and Orlin [2], develops the network simplex method in the context of network flows and applies combinatorial arguments. We will follow the first, classical route, so the layout of the section is as follows. We first give a summary of the simplex method in its general form, then highlight the points where the special stucture of the network problems can be exploited.

9.1.1 Simplex method

We provide a very concise description of the general-purpose simplex method. It is a part of the common mathematical knowledge and presented in all textbooks on linear programming (see, e.g., [20, 51, 55]), so we can be succinct in providing details and references.

The goal is to find an optimal solution of the following linear programming problem

$$\begin{aligned} &\min cx \\ \text{subject to } &Ax = b \\ &0 \leq x \leq u \end{aligned} \tag{9.1}$$

where A is a full-row rank matrix of size $m \times n$; b, c and u are vectors of appropriate size. We also assume that $u > \mathbf{0}$. Let a_i denote the i^{th} column of A.

The steps of the simplex algorithms are: (1) find a basic feasible solution, (2) check if it is optimal, and (3) if it is optimal, then stop, otherwise execute a pivot step to change the basis and start the process again.

A basic solution x^* is associated with a basis structure (B, L, U) where B is the set of basic

variables, L is the set of non-basic variables at their lower bound, U is the set of non-basic variables at their upper bound. This structure provides notations for the constraint matrix and the vectors of the problem. The submatrices of A made up of columns corresponding to variables in B, L or U are also denoted by these letters, i.e., $A = [B, L, U]$. Submatrix B is a basis of A. In the same vein, $x^* = (x_B^*, x_L^*, x_U^*)$, $c = (c_B, c_L, c_U)$ and $u = (u_B, u_L, u_U)$ are decompositions of x^*, c and u to subvectors corresponding to the appropriate variables.

Using these notations, the basic solution satisfies $Bx_B^* + Lx_L^* + Ux_U^* = b$. As $x_L^* = \mathbf{0}$ and $x_U^* = u_U$, this is equivalent to

$$Bx_B^* = b - Uu_U =: b' \tag{9.2}$$

The first phase of the two-phase simplex method is to find an initial feasible basic solution, i.e., a nonnegative x_B^*. This can be done by adding new, artifical variables.

To check whether a basic solution is optimal or not, the simplex algorithm finds the corresponding dual solution, i.e., a $\pi \in \mathbb{R}^m$ satisfying

$$\pi B = c_B \tag{9.3}$$

The basic solution associated with B is optimal if the following optimality conditions are satisfied.

$$\begin{aligned} c_i - \pi a_i &\geq 0 \quad \forall i \in L \\ c_i - \pi a_i &\leq 0 \quad \forall i \in U \end{aligned} \tag{9.4}$$

If the solution is not optimal, then there is an $i \in L \cup U$ that violates (9.4). When changing the basis, any of these indices can enter the basis. For the sake of simplicity, let us assume that the entering variable l is in L. The case of $i \in U$ can be handled similarly. To keep the basis structure, a variable must leave the basis. We can find candidates for this by expressing column a_l as a linear combination of the basic columns, i.e., calculating the vector $w = B^{-1}a_l$. At first sight, the variables i for which w_i is non-zero are all eligible to be exchanged with l, as these are the variables for which $B \setminus \{i\} \cup \{l\}$ is a basis of A. However, to keep the solution feasible, we have to be more careful. Let x' denote the solution associated with the new basis achieved by exchanging variable l with a variable in B. Then $Bx'_B + x'_l = b'$ and multiplying this by B^{-1} we get

$$x'_B + x'_l w = B^{-1}b' = x_B^*$$

and for $i \in B$

$$x'_i = x_i^* - x'_l w_i$$

To be feasible, x'_i must satisfy $0 \leq x'_i \leq u_i$. Hence $x'_l w_i \leq x_i^* \leq u_i + x'_l w_i$. This implies that by

defining

$$\delta_i = \begin{cases} x_i^*/w_i & \text{if } w_i > 0 \\ (u_i - x_i^*)/(-w_i) & \text{if } w_i < 0 \\ \infty & \text{if } w_i = 0 \end{cases} \tag{9.5}$$

x_l' can be at most δ_i for $i \in B$. We want to achieve the maximum change in the objectice function by this pivot step, so the value assigned to x_l' will be the minimum of u_l and $\delta = min\{\delta_i \mid i \in B\}$. If the minimum is u_l, then B remains unchanged, the variable exchange takes place between L and U. If the minimum is δ, then the variable leaving B will be the one at which the minimum is attained. If $\delta = 0$, then this basis exchange is *degenerate*. Degenerate pivots are undesirable from a theoretical viewpoint because they can cause *cycling*, i.e., a phenomenon when a basis structure reappears and, as a consequence, the algorithm gets into a loop and never terminates. To avoid cycling, one has to be careful with the choice of the variable leaving the basis. One of the most used techniques for this is the *lexicograpic rule.*

Even if the finiteness of the simplex algorithm is ensured, it might need exponential time to find the optimum. That is, the worst case complexity of the general simplex algorithm is exponential. In practice, however, the average case behaviour of the simplex method is much better. There are other, provably polynomial algorithms for solving LP problems, but the simplex method has outperformed these methods on most practical problems. The contradiction between the theoretical and practical complexity disappears in the special case of network problems, described in the next section.

9.1.2 Network simplex method

The network simplex method deals with LP problems in which A is the node-edge incidence matrix of a directed graph G. Without loss of generality, we can assume that G is connected. One can then regard c as a cost vector on the edges and b as a balance function on the nodes, translating (9.1) to a minimal cost flow problem. If b is a general vector, then this is a flow problem where any node can have supply or demand.

An important property of the node-edge incidence matrix A of a directed graph is that its rank is one less than the number of rows, and after deleting a row to make it full row rank there is a one-to-one relationship between the bases of A and the spanning trees of G. A consequence of this fact is that any basis B can be rearranged to be a lower triangular matrix, with unit rows corresponding to the leaf nodes of the tree. Thus, in (9.2) and (9.3) the basic solution x^* and the dual solution π can be calculated by simple substitutions. The optimality check (9.4) is also simplified, as a_i has only two non-zeros, so $c_i - \pi a_i = c_{uv} + \pi(u) - \pi(v)$ where column a_i of A correspond to edge (u, v) of G. When we arrive to a pivot operation, we first have to calculate a w satisfying $Bw = a_l$, easily done again by exploiting the lower triangular structure of B.

All these calculations can be executed on the graph. For example, calculating the basic solution

is the same as calculating a flow on the edges of the corresponding spanning tree T such that the balance $b'(u)$ at each node u is satisfied. This can be done by first assigning the appropriate flow to the leaf edges, then deleting these edges, updating the balance at their end-nodes and repeating the process with the new leaf edges. Computing the dual solution π is also very easy, we assign 0 to an arbitrarily chosen root node of T, and then calculate the π value, called *node potential*, for each node to satisfy $c_{uv} = -\pi(u) + \pi(v)$ for each edge (u, v) by fanning out along the tree edges from the root. At each step, one of $\pi(u)$ and $\pi(v)$ is known, so the cost of the edge uniquely determines the other.

There is a graphical representation for w as well. Let us suppose that column a_l correspond to edge $l = (u, v)$. Because $Bw = a_l$, the coordinates of w are flow values on T giving balance -1 at u, $+1$ at v and, zero elsewhere. This can be satisfied by sending ± 1 units on the path in the tree that connects u and v. Precisely, -1 on the edges of this path directed forward from u to v and $+1$ on edges which are backward. Thus w is a $0, \pm 1$ vector with a one-to-one connection between the sign of w_i and the direction of the corresponding edge e_i. The calculation of δ_i in (9.5) is then the following:

$$\delta_i = \begin{cases} x_i^* & \text{if } e_i \text{ is a forward edge} \\ u_i - x_i^* & \text{if } e_i \text{ is a backward edge} \end{cases}$$

The algorithm chooses the tree edge to leave the spanning tree for which δ_i is minimal and replaces it with l. Since l and the edges on which w is positive form a cycle, this step results in another tree, which differs minimally from the previous one. The calculations above, that of the basic solution or the node potentials, do not have to be restarted from scratch, it is more efficient to update the values after each basis exchange.

In the implementation of the graphical calculation several important technical questions have to be solved. For example, the representation of the trees in the code, or efficient algorithms to calculate flows and node potentials are such. We do not discuss details of these technical questions here. The books mentioned above, especially [2] and [50], contain results and references.

Theoretically, the most important question is cycling. The basic network simplex method, just as the general simplex method, may cycle. Cunningham [17] proposed a refinement that helps to avoid cycling and ensures finite termination. He introduced the *strongly feasible bases* and a pivot rule to select the leaving variable that maintains the strong feasibility of the basis. This, in combination with other techniques (see, e.g., [18]) provides a polynomial running time for the network simplex method.

So the network simplex method, at first sight, works for LP problems in which the constraint matrix is the node-edge incidence matrix of a directed graph. Where are the network matrices then? They are there, just disguised. Let N be a network matrix defined on graph G with respect to a spanning tree T. Then, if A is the node-edge incidence matrix of G, A' is a full row rank matrix obtained from A by deleting a row, and R and S are submatrices of A' corresponding to the tree and

non-tree edges, it follows that $N = R^{-1}S$. Therefore, if in the problem

$$\begin{aligned} &\min cx \\ \text{subject to } &Nx \leq b \\ &0 \leq x \leq u \end{aligned}$$

we introduce slack variables s to transform $Nx \leq b$ to $Nx + s = b$, and then multiply the now equality constraints by R, we get the equivalent

$$\begin{aligned} &\min cx \\ \text{subject to } &Sx + Rs = Rb \\ &0 \leq x \leq u \\ &0 \leq s \leq u' \end{aligned}$$

In the latter problem the constraint matrix $[S, R] = A'$ is a node-edge incidence matrix minus one dependent row, so the network simplex method can be applied to it. (Vector u' is a suitable upper bound on s. It can be shown that it is not greater than $max_i\{b_i\} + \sum u_j$ in each coordinate.)

Thus, with some preprocessing, the network simplex method works on network matrices. The range of practical applicability of the network simplex method can be extended even more, to constraint matrices that can be converted to a network matrix by row and column scalings. A heuristic algorithm for this conversion appeared in [6]. By this extension, we can find a wider set of constraint matrices for which the original network simplex method is applicable. In the next section we introduce a method that makes use of the ideas underlying the network simplex method but can be applied to instances to which the network simplex cannot.

9.2 Simplex method for generalized networks

An inherent limitation of the network flow models is that the flow is conserved on the edges, i.e., if one unit of flow leaves the tail node of the edge, then the same one unit will arrive at its head node. In some practical applications this is too strong a restriction. For example, if our network models a water pipe system, then part of the flow may disappear between nodes due to leakages. Another example is money conversion, when an edge expresses the process of exchanging a currency into another. This can be further combined with an interest rate or conversion fee, making it apparent that a unit of flow leaving the tail can change to any amount of flow by the time it arrives to the head of an edge.

Modelling these instances made it necessary to define networks in which the flow conservation rule on the edges is revoked. In this section we discuss these networks, called the *generalized networks*. We will show that they include bidirected graphs, providing a link to the main subject of this

work.

Generalized networks are, not suprisingly, more general than pure networks, but the ideas of the network simplex method, with some necessary change, can be adapted to them. In other words, there is a *generalized network simplex method* that is able to solve optimization problems on generalized networks more efficiently than the general-purpose simplex method, exploiting their combinatoral structure. In matrix terms it means that problems in which the constraint matrix is convertible to the node-edge incidence matrix of a generalized network are easier to solve. Before the invention of binet matrices, this class of constraint matrices consisted of matrices with at most two non-zeros in each column. By the definition of binet matrices, we are able to substantially widen the applicability of the generalized network simplex method.

Generalized networks are covered in many of the recent books on network problems. For instance, Glover, Klingman and Phillips [36] introduce generalized networks through a number of applications. A more formal approach is followed in [2] and [50], which also describe the generalized network simplex method.

In the remainder of the section we first define the generalized networks, give some examples and show its relationship to bidirected graphs and gain graphs. Then we present the generalized network simplex method.

9.2.1 Generalized networks

A generalized network is defined on a connected directed graph $G(N, E)$. There is a real non-zero multiplier p_e associated with each edge $e = (i, j)$ of G. We assume that if a unit flow leaves the tail i of e, then p_e units arrive at j. G can also contain loops, i.e., edges whose tail and head nodes coincide. They usually represent slacks, or surplus capacity, as in the example presented shortly. We assume that the multiplier of a loop cannot be $+1$, as it would mean that the same flow leaves and enters the node on such a loop, making the loop redundant. If all the multipliers equal 1, then we have the well-known pure network. Furthermore, at any node we can assume a supply or demand, so the flow balance at a node $i \in N$ can take any value $b(i)$. An example of generalized networks is the model of *machine loading*, taken from [2].

Imagine that we would like to schedule the production of r products $P_1, \ldots, P_r$ on s machines $M_1, \ldots, M_s$. Machine M_j is available for α_j hours, and we must produce β_i units of product P_i. Producing one unit of product P_i on machine M_j takes a_{ij} hours and costs c_{ij}. We wish to find a feasible production plan at the least possible cost. The generalized network formulation is shown in Figure 9.1. There are r nodes representing the products and s nodes representing the machines. There is an edge (i, j) with a multiplier a_{ij} between any product node P_i and machine node M_j. This edge represents the quantity of product P_i produced on machine M_j. The cost of the edge is c_{ij}, i.e., any unit of flow leaving P_i costs c_{ij}. Node P_i has a supply β_i ($i = 1, \ldots, r$), node M_j has

a demand α_j ($j = 1 \ldots, s$). This means that to keep the balance at P_i the outgoing edges from this node must altogether carry β_i flow, in other words, the total production of product P_i should be β_i. Furthermore, there is a loop with multiplier 2 and cost 0 at each machine node. They ensure that the total of the production hours at a machine, i.e., the incoming flow from the product nodes, is less than or equal to the capacity of the machine.

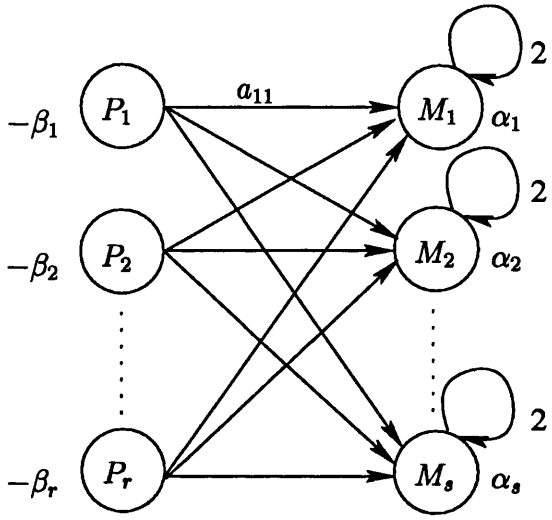


Figure 9.1: The generalized network model of the machine loading problem

The aim of this example is to find the minimum cost flow on the generalized network. Just as in pure networks, this minimum cost flow problem, can be formulated as an LP problem. For an edge e, define $T(e)$ and $H(e)$ as the set of tail and head nodes of e. So edge e is a loop if and only if $T(e) = H(e)$. The LP formulation of the minimal cost flow problem is the following:

$$\begin{aligned} \min \quad & \sum_{e \in E} c_e x_e \\ \text{subject to} \quad & \sum_{e:i \in T(e)} -x_e + \sum_{e:i \in H(e)} p_e x_e = b(i) \quad \forall i \in N \\ & 0 \leq x_e \leq u_e \quad \forall e \in E \end{aligned} \tag{9.6}$$

where x_e is the flow on edge e, $b(i)$ is the balance at node i, c_e is the cost of one unit of flow leaving the tail of e, and $u_e > 0$ is the capacity of edge e. Note that if there is a supply β at a node i, then the balance value $b(i)$ is $-\beta$, while if node j has a demand α, then $b(j) = \alpha$.

A compact form of this formulation can be given by introducing the node-edge incidence matrix of a generalized network. The column of this matrix (denote it by A) that corresponds to a non-loop edge $e = (i, j)$ has -1 in row i and p_e in row j, zeros elsewhere. If e is a loop at node i, then its

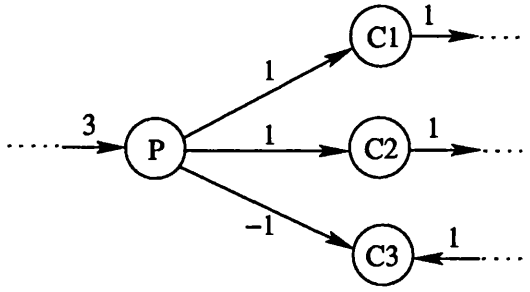
column has only one non-zero, $(p_e - 1)$ in row i. So (9.6) takes the form

$$\begin{aligned} &\min cx \\ \text{subject to} \quad &Ax = b \\ &0 \leq x \leq u \end{aligned} \tag{9.7}$$

with vectors x, b, c and u made up from the flows, balance values, costs and capacities, respectively.

Note that the multipliers in the machine loading problem are all positive. This is the case in most of the applications, and the theory of general networks with positive multipliers is more developed than that of the general case, when p_e can also take negative values. This is mainly because some problems are easier to resolve when one deals with only positive multipliers, as discussed later. Most of the results, however, hold for both cases, so if we do not mention the opposite explicitly, we assume that p_e can be negative. Among the references mentioned, [2] presents only the positive case, while [50] shortly discusses the effect of negative multipliers. There is an extensive coverage of general network models with negative multipliers in [36]. The following example is taken from this latter book.

We model a *class scheduling* problem. Suppose that there are some students and some classes and our task is to assign the students to the classes satisfying some conditions. A possible settings can be when the students are pre-assigned, but they can change this assignment according to their priorities. For instance, one individual would like to remove his name from Class 3 and put it on Class 1 and Class 2. The following part of the general network model accomplishes this task.



Node P represent the student's priority, nodes C1, C2 and C3 are the classes. There is a supply of 3 (or an incoming flow, if this is a part of a bigger model) at P, and classes C1 and C2 have a demand, or outgoing flow of 1, C3 has a supply of 1. These represent the changes in the number of students assigned to the classes. The multipliers on the edges from P to C1 and C2 are 1, whereas the multiplier of edge $(P, \text{C3})$ is -1. So the network must carry one unit of flow on all three edges, increasing the number of people assigned to Class 1 and 2 by 1, and taking one student off Class 3.

As it is apparent from their incidence matrices, there is a strong connection between bidirected graphs and generalized networks. The node-edge incidence matrix of a bidirected graph can have columns of the following forms:

(a) columns with one non-zero, being $+1$ or ± 2,

(b) columns with one -1 and another non-zero, being ± 1,

(c) columns with one non-zero, being -1

(d) columns with two non-zeros, both being 1.

Columns of types (a) and (b) can be columns of the node-edge incidence matrix of a generalized network: columns of type (a) are loops, columns of type (b) are ordinary non-loop edges. Columns with a single non-zero -1 cause problem, because representing them with $(p_e - 1)$ would require $p_e = 0$. But this problem can be easily overcome by multiplying such columns by 2, obtaining a column of type (a). The multiplication of a column by 2 is equivalent to dividing the corresponding variable by 2. Columns of the last type are also problematic, as an edge corresponding to such a column cannot be a generalized network edge as defined in the current literature, as this edge would have no tail. We provide some simple transformations to deal with columns of type (d). As a result, LP problems on bidirected graphs can fit into the context of generalized networks.

Let us suppose that we have an LP problem

$$\begin{aligned} \min\ & c_1x_1 + c_2x_2 \\ \text{subject to}\ & A_1x_1 + A_2x_2 = b \\ & 0 \le x_1 \le u_1 \\ & 0 \le x_2 \le u_2 \end{aligned} \tag{9.8}$$

where $A = [A_1, A_2]$ is the node-edge incidence matrix of a bidirected graph with A_1 containing the columns of type (d). Then with the new variable $x_1' = -x_1 + u_1$ we get the equivalent problem

$$\begin{aligned} \min\ & -c_1x_1' + c_1u_1 + c_2x_2 \\ \text{subject to}\ & -A_1x_1' + A_2x_2 = b - A_1u_1 \\ & 0 \le x_1' \le u_1 \\ & 0 \le x_2 \le u_2 \end{aligned} \tag{9.9}$$

in which all columns are of types (a), (b) or (c), so the constraint matrix can be considered the node-edge incidence matrix of a generalized network.

However, there is no need to transform type (d) columns. The generalized network simplex method, which we describe in the next section, can be easily adapted to the case where the network contains edges with two heads and no tails. So, this method works on bidirected graphs too, justifying our earlier claim that the generalized network simplex method is the bidirected version of the network simplex algorithm.

9.2.2 Generalized network simplex method

To apply the simplex method to generalized networks, we have to investigate the basis structures, i.e., subnetworks that correspond to bases of the node-edge incidence matrix. It turns out that a basis subnetwork is not necessarily a spanning tree. We need some definitions first.

Let C be a cycle in the underlying graph G. By C we will also denote the edge set of the cycle. Choose arbitrarily a node of the cycle, and walk along the edges of C from this point in either of the directions. Edges that are forward in this walk are members of $\overline{C}$, the backwards edges are in $\underline{C}$. Obviously, $C = \overline{C} \cup \underline{C}$. The *multiplier of the cycle*, is defined as

$$p(C) = \frac{\prod_{e \in \overline{C}} p_e}{\prod_{e \in \underline{C}} p_e}$$

It is clear from the definition that the multiplier of a cycle depends on the direction the cycle is walked along, unless $p(C) = 1$. Cycles with multipliers 1 are called *balanced*, other cycles are *unbalanced*. (Compare this definition with the balancedness in gain graphs, defined in Section 10.1.2.)

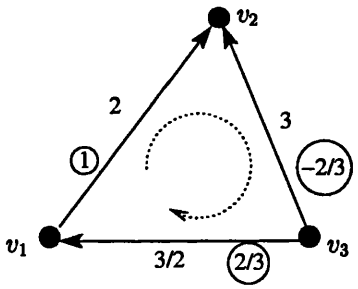


Figure 9.2: An example of a balanced cycle. The plain numbers on the edges are their multipliers, the circled ones are the flow values.

To explain the term 'balanced', consider now the example depicted in Figure 9.2, a cycle with three edges and multipliers $2, 3$ and $3/2$. This is a balanced cycle, as the product of the multipliers on forward edges equals the multiplier of the only backward edge: $2 \cdot \frac{3}{2} = 3$. Imagine that we send a unit flow on edge (v_1, v_2) and we would like to keep zero balance at each node. Two units of flow arrive at v_2 from v_1, so to balance this out -2 should arrive from v_3. That is, edge (v_3, v_2) should carry a flow of $-2/3$. The flow conservation at v_3 then forces a flow of $2/3$ on edge (v_3, v_2). The multiplier of this edge is $3/2$, so 1 unit of flow arrives at v_1 from v_3, exactly what is needed to offset the flow on (v_1, v_2). So in a balanced cycle, zero balance at each node can be achieved. A different but equivalent phrasing of this fact is the following. If a cycle is balanced, then one can find weights d_v assigned to the nodes of the cycle such that for each edge $(u, v) \in C$:

$$\frac{d_u p_{uv}}{d_v} = 1 \tag{9.10}$$

It is easy to see that the same is true for cycle-less subnetworks, i.e. forests, but not for unbalanced cycles. This implies the following result.

Lemma 9.1. *The following statements are equivalent for a generalized network $\mathcal{G}$ on a connected directed graph G.*

(a) $\mathcal{G}$ can be transformed to a pure network.

(b) There are weights d_v on the nodes of G satisfying (9.10) for all edges (u, v) of G.

(c) There are no unbalanced cycles or loops in $\mathcal{G}$.

(d) The rank of the node-edge incidence matrix A of $\mathcal{G}$ is $n - 1$, where n is the number of nodes in G, i.e., the number of rows in A.

Therefore, if there is an unbalanced cycle in $\mathcal{G}$, then A is of full row rank. In this case the basis structures are easy to identify.

Lemma 9.2. *There is a one-to-one connection between the bases of a full row rank node-edge incidence matrix of a generalized network and the spanning subnetworks satisfying the following conditions:*

(a) the subnetwork may consist of more than one connected components,

(b) each component consists of a tree plus one additional edge, i.e., it is a 1-tree,

(c) the additional edge is either a loop or forms an unbalanced cycle with some tree edges.

Let us call a basis structure of a generalized network simply a *basis of the generalized network*, and its components the *unbalanced 1-trees*.

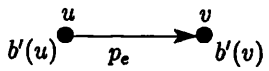
Lemma 9.2 has nice consequences for the layout of a basis B of the node-edge incidence matrix. It can be easily shown that any component of B can be rearranged to an *almost lower triangular* matrix, that is a matrix which has at most one non-zero element above its main diagonal. This property of the bases can be exploited in the simplex method, when calculating e.g., the basic or the dual solution. Basically, one has to assign a parametric value to a variable and then find the values of the other variables by simple substitutions. We do not describe this algebraic procedure because we will present a graphical representation of the calculations.

We give the steps of the generalized network simplex algorithm, applied to the LP problem (9.7) where A is the node-edge incidence matrix of the generalized network $\mathcal{G}$. We will use the notations introduced for the simplex method in Section 9.1.1.

(1) Find an initial basis structure (B, L, U). This can be an artificial one in which $L = A$ and $B = I$, the unit matrix, by adding artifical variables, effectively executing the first phase of the two-phase simplex method. The columns of the artificial basis I have graphical representation, they

correspond to loops with multiplier 2. A subnetwork made up of such loops at each node satisfies the conditions of Lemma 9.2.

(2) Find the basic solution associated with the basis structure, i.e., an x satisfying $Bx = b'$ (see (9.2)). This amounts to finding a flow on the basis that has a balance $b'(v)$ at any node v of G. We now describe an algorithm for determining such a flow on an unbalanced 1-tree T. The algorithm can be repeated for each component of the basis. Let a non-loop edge of T that is connected to only one other edge be called a *leaf edge*. We first determine the flow on the leaf edges. One end-node of any leaf edge is connected to exactly one 1-tree edge, the leaf edge itself. Let us call these nodes the *terminal nodes*. The balance value of a terminal node determines the flow on the connected leaf edge. This flow then, in turn, effects the balance to be satisfied at the other end-node of the leaf edge. For example, if $e = (u, v)$ is the leaf edge with multiplier p_e and v is a terminal node, then the flow on e is $x_e = b'(v)/p_e$ and the balance at u is increased by x_e.



If u is the terminal end-node of e, then $x_e = -b'(u)$ and the balance at v is decreased by $p_e x_e$.

When the flow on all leaf edges is set and the balance values at their non-terminal edges are updated, we can delete the leaf edges. In the resulting graph there are either new leaf edges, which can be dealt with in the same way, or there are no leaf edges, which means that the graph is a loop or a cycle. If it is a loop then we can easily set a flow on it to satisfy the residual balance at its end-node. If we have a cycle, then we try to find the flow values by allowing a flow of θ on an arbitrary edge, say (u, v) of the cycle. This flow and the balance values at the cycle nodes consecutively determine the flow on all other cycle edges, in much the same way as we did for the cycle in Figure 9.2. The last step of this procedure is to find the flow on edge h, the cycle edge incident to the starting edge (u, v) through node u. The flow on this edge is a linear function f of θ. If u is the head of h, then the balance at node u requires θ to satisfy $p_h f(\theta) - \theta = b'(u)$. If u is the tail of h, then the equation θ should satisfy is $-f(\theta) - \theta = b'(u)$. Note that the cycle in question must be unbalanced, and it is such in a basis structure, otherwise there would be multiple θ solutions of the equations. The interested reader can find a pseudo-code of the algorithm just described, and of the following ones too, in [50].

(3) Compute the node potentials. Again, we describe the algorithm only for one connected component of the basis, an unbalanced 1-tree, T. We want to determine $\pi(u)$ values for the nodes of T satisfying the edge condition $c(e) = -\pi(u) + p_e\pi(v)$ for each edge $e = (u, v)$ of the 1-tree. Let us choose an arbitrary node r of T. A 1-tree consists of a tree and an additional edge. We assign node potential θ to r and determine the node potential for the other nodes by going along the tree, just as in the pure network case. At the end of this procedure, we have a parametric node potential $\pi_\theta(u)$ at each node, satisfying all edge conditions except the one on the additional edge. Solving the edge

condition equality for this additional edge provides the value for θ, which we substitute in $\pi_\theta(u)$ to get the node potentials for the nodes.

(4) Check the optimality conditions. These are $c_{uv} + \pi(u) - p_{uv}\pi(v) \geq 0$ for nonbasic edges (u, v) that carry zero flow, with their corresponding variables in L; and $c_{uv} + \pi(u) - p_{uv}\pi(v) \leq 0$ for nonbasic edges (u, v) the flow of which reached the capacity of the edge, i.e., their variables are in U.

(5) Find an edge l for which the optimality condition does not hold, and calculate $w = B^{-1}a_l$. This task boils down to finding a flow on the basis that has a balance -1 at the tail node of l, p_l at the head node of l, and 0 at all other nodes. We can use the algorithm descibed in step (2). Note that the end-nodes of l can belong to different components of the basis, in which case the algorithm has to be executed twice, once for either 1-tree components.

(6) Determine the δ_i values, select the leaving variable, and update the basis.

In the last two steps it is helpful to understand what a flow that defines w looks like, that is, what kind of a subnetwork is formed by the edges with non-zero flow. In the pure network case, this subnetwork was a path on the basic tree connecting the end-nodes of l. In other words, l and the subnetwork formed a cycle. With a case by case analysis it is not difficult to prove that in a generalized network the counterpart of this cycle can be (see Figure 9.3 for examples):

(a) a balanced cycle, or

(b) the union of two unbalanced cycles which meet at one point, or

(c) the union of two node-disjoint unbalanced cycles and a path that connects them but has no common point with the cycles other than its end-points, or

(d) the union of three internally node-disjoint paths that connect the same pair of nodes and none of the three cycles is balanced.

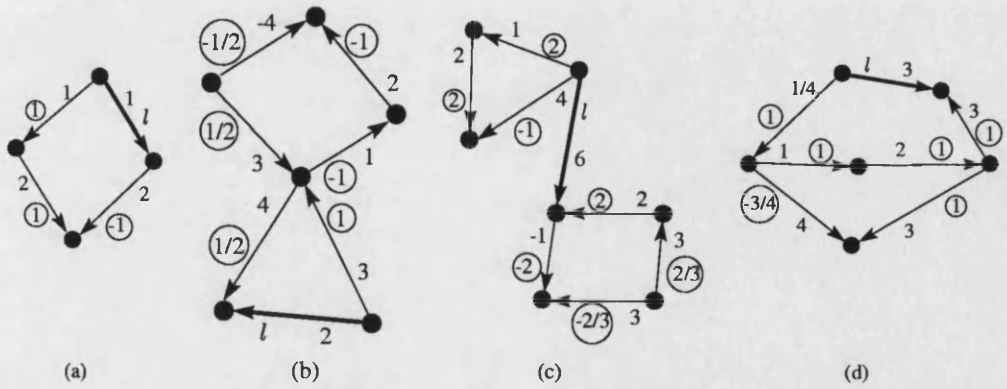


Figure 9.3: Examples of flows to calculate w. The plain numbers on the edges are their multipliers, the circled ones are the flow values. Bold edges represent incoming edge l.

Observe that these subgraphs are the same as the subgraphs composing a circuit of a bias matroid, listed in Section 10.2.2. The equivalence, of course, is not coincidental. A generalized network is in effect a gain graph (see Section 10.1.2), and it can be easily shown following the arguments in Section 10.2.2 that the bias matroid of a gain graph is the linear matroid represented by the node-edge incidence matrix of the generalized network.

When implementing the steps presented above, a good representation of the basis structure is needed. The books [2] and [50], besides describing the algoritmh in much greater detail, deal with the problem of tree representation as well.

The generalized network simplex algorithm, as described above, can cycle. To avoid this, Elam, Glover and Klingman [27] devised a version of the method that uses only special bases, called *strongly feasible bases*. However, their method works only for positive multipliers. Arantes, Birge and Murty [4] resolved the problem for negative multipliers. Both methods are lexicographic, though this is not apparent at first sight.

Applying these methods, the finite termination of the generalized network simplex method can be ensured. The polynomal complexity, however, is not. Although the time needed for one iteration, i.e., the calculations on one basis, is only a low order polynomial function of the input size, the number of iterations can be more than polynomial in the worst case. In practice, however, the algorithm is usually very fast, only two or three times slower than the network simplex method, as experiments show (see the reference notes in [2]).

It is time now to fulfill our promise and demonstrate that the generalized network simplex method is the bidirected version of the network simplex algorithm, and it can be applied to LP problems in which the constraint matrix is binet. Recall from the previous section that bidirected graphs can be considered almost entirely as generalized networks, the only problem is with the edges with two head nodes. However, the generalized network simplex algorithm can be easily adjusted to accomodate such edges. When defining balanced cycles, one can view the two-headed edges as a normal edge with multiplier -1. It does not matter if they are forward or backward, as their contribution to the multiplier of the cycle is -1, so they can equally be in the numerator or the denominator. Taking all this into account, it is easy to show that unbalanced cycles are exactly the odd cycles. Lemma 9.2 remains valid, so the basis structures are the odd 1-tree forests. In the calculations there are some necessary refinements, caused by the exceptional edges, but they can be very easily realized. The biggest difference is that type (d) flow of Figure 9.3 cannot occur, for the same reason why there are no theta graph fundamental circuits, see the proof of Lemma 5.2.

The extension of the applicability of the generalized network simplex method to binet matrices goes in the same way as the extension of network simplex method to network matrices. Having an LP problem with constraint matrix B, where $B = R^{-1}S$, a binet matrix obtained from the node-edge incidence matrix $A = [S, R]$ of a bidirected graph, then by converting the possible inequalities to equalities and multiplying the constraints with R we get a problem of a form for which the gener-

alized network simplex method, with the above mentioned adjustments, can be applied. A further extension could be for matrices that can be scaled to binet matrices, as it is done for network matrices in [6].

To sum up this chapter, we showed that there is a combinatorial algorithm, namely the generalized network simplex method that can efficiently solve linear programming problems in which the constraint matrix is binet.

Chapter 10

General graphs and their matroids

A substantial part of this dissertation deals with bidirected graphs, which we introduced as a convenient generalization of directed and undirected graphs. There are, however, even more general graphs that can be discussed. As we will show in this chapter, by further steps of generalizations, we can obtain wider and more abstract sets namely, signed graphs, gain graphs and biased graphs.

At each generalizing step a special characteristic of the graphs is chosen and its definition is relaxed to get to the next level. Thus, signed graphs can be achieved from bidirected graphs by ignoring the signs at the ends of the edges and focusing on only whether an edge is bidirected or directed. Gain graphs, extending the range of alternatives, allow more possible options for an edge. In fact, edges can take any value from a given group, called the gain group. The value of a cycle is then defined as the product of the values assigned to the edges of the cycle. Cycles that take the value of the identity element of the gain group are special, and this is the characteristic generalized in biased graphs. A biased graph is a graph in which a special class of cycles is earmarked. The class is such that it generalizes the class of unitary valued cycles in gain graphs.

Knowing about all these generalizations of bidirected graphs, a naturally arising question needs to be answered. Why did we stop at the level of bidirected graphs, why do we not deal in this work with more general graphs? The answer is: because our focus is on matrices related to graphs, basically on the node-edge incidence matrix and its derivatives, and it is the class of bidirected graphs where the node-edge incidence matrix is naturally defined and has properties that provide a link to total unimodularity. For signed graphs, one has to orient the edges first to get an unambiguous incidence matrix, and the result of this operation is a bidirected graph. For gain graphs, if the group contains real numbers, the incidence matrix can be defined, but it will not, in general, be 2-regular. In biased graphs, the speciality of the cycles cannot be represented by an incidence matrix.

There is another combinatorial structure, however, that has a meaningful definition for all these general graphs, namely matroids. In the second part of this chapter we give a brief summary of the most basic matroidal results concerning biased and signed graphs. Naturally, these results have a

strong connection with what is described in other chapters. For example, we will show that binet matrices are representative matrices of signed graphic matroids, therefore results about these matroids have consequences in terms of binet graphs.

The most important reference about biased graphs is Thomas Zaslavsky's work. His research into signed and biased graphs is the basis of this chapter. The term and definition of biased graphs first appeared in [72]. Signed graphs were introduced by Harary [38]. Zaslavsky's annotated bibliography of signed and gain graphs [74], which contains hundreds of references, is an essential tool for anyone interested in the subject. Our terminology and notations mainly follow the Glossary of Signed and Gain Graphs [75], also by Zaslavsky.

The chapter is organised as follows. First we give the formal definitions of the generalized graphs and list some basic results. Then, in Section 10.2 we deal with the matroids associated with these graphs. The second section also contains an overview of the necessary theory about matroids.

10.1 Generalized graphs

This section introduces three generalized graphs, namely signed graphs, gain graphs and biased graphs. Signed graphs are in intimate connection with bidirected graphs, as a bidirected graph is an oriented signed graph. Based on this relationship, every result we give about bidirected graphs in this dissertation could be rephrased to signed graphs. We chose to use the term bidirected graphs, as opposed to oriented signed graphs, because the orientation of the edges play a cruical role in, for example, the determination of the elements of a binet matrix. One can imagine the relationship of bidirected and signed graphs as that of undirected and directed graphs.

Gain graphs are more abstract structures, having less connection to other parts of this work. However, we discuss them here for two reasons. First, they provide a natural step towards biased graphs, and secondly, gain graphs arise in the context of generalized network flows, handled in Chapter 9.

Biased graphs are the most general graphs we give. Their importance for us lies in their matroids. As we will see in Section 10.2, matroids of biased graphs provide the general setting in which most of the relevant results can be stated.

The generalized graphs in this section are based on an underlying undirected graph $\Gamma(N, E)$ with set N of nodes and E of edges. Any edge $e \in E$ can have at most two end-nodes in N. An edge with no end-nodes is called a *loose edge*; edges with one end-node are the *half-edges*; if an edge has two end-nodes and they coincide, then it is a *loop*; if the two end-nodes are distinct, then the edge is called a *link*. Other definitions are familiar from basic graph theory, or from Section 4.1. A *path* is a sequence of links $e_1, \ldots, e_k$ where e_i and e_{i+1} $(i = 1, \ldots, k-1)$ have a common end-node, but none of these nodes is repeated. If e_1 and e_k also have a common end-node, then the path is *closed*. A closed path, a loop or a half-edge is called a *cycle*. A *tree* is a connected subgraph that does not

contain a cycle. A *1-tree* is a connected subgraph that contain exactly one cycle. Sometimes we need to differentiate half-edges from the other types of edges.

Definition 10.1. Let E_* be the subset of E containing all the links and loops, but not the half-edges or loose edges. Edges in E_* are called *ordinary edges*. A cycle which is not a half-edge, i.e. contains only ordinary edges, is an *ordinary cycle*.

Two links forming a cycle are called *parallel*. Recall the definition of *theta graphs* from Section 1.2.1, illustrated by Figure 1.1.

10.1.1 Signed graphs

In a signed graph, the edges can fall into one of two classes. There are two equally used, and equivalent, notations of the classes. One employs signs + or −, the other works with parity. Both notations have advantages, so we give and use both of them.

Definition 10.2. In a *signed graph* $\Sigma = (\Gamma, \sigma)$, the edges of Γ are labelled by + or −, that is, there is a mapping $\sigma : E \to \{+, -\}$ on the edges.

Usually it is also required that the label of a loose edge is + and that of a half-edge is −. We also assume that there are only negative loops. The alternative definition of signed graphs is the following.

Definition 10.3. A *signed graph* is a pair (Γ, Λ) where $\Lambda \subseteq E$. Edges that are in Λ are called *odd*, otherwise they are *even*.

Odd (respectively, even) edges usually correspond to edges with label − (+, resp.). In respect to what was mentioned above, loops and half-edges are in Λ, loose edges are not. The parity of the edges can be extended to the parity of subgraphs. An edge-set is called odd (even), if it contains odd (even) number of odd edges.

Clearly, a bidirected graph is a signed graph, bidirected edges are negative (odd), directed edges are positive (even). Conversely, the edges of a signed graph Σ can be *oriented* to get a bidirected graph, denoted $\vec{\Sigma}$. To do so, we allocate arbitrary signs to the ends of every edge so that positive edges become directed and negative edges become bidirected. More formally, if u and v are (possibly coinciding) end-nodes of an ordinary edge e and the sign of e at u is $s(e, u)$, then the sign of e at v is $s(e, v) = -\sigma(e)s(e, u)$. For convenience, Table 10.1.1 lists the possible notations for the edges of a signed graph Σ, and the corresponding bidirected graph $\vec{\Sigma}$.

Now we derive the node-edge incidence matrix of a signed graph, it is the node-edge incidence matrix of the bidirected graph obtained by an orientation. Different orientations yield different node-edge incidence matrices. That is, the incidence matrix of a signed graph is not unambiguously defined, but any incidence matrix conveys all the information about the signed graph. In fact, two

	links	loops	half-edges	loose edges
σ	+ or -	-	-	+
Λ	even or odd	odd	odd	even
$\vec{\Sigma}$	directed or bidirected	bidirected	bidirected	
E_*	ordinary		not ordinary	

Table 10.1: Different notations for a a signed graph Σ.

bidirected graphs can be the oriented versions of the same signed graph if and only if one can be obtained from the other by switchings (for the definition of switching in a bidirected graph, see Section 4.1).

Basic operations on signed graphs, such as deletion, contraction and switching are defined in the same way as for bidirected graphs, by replacing the term 'bidirected' with 'odd' and directed with 'even'. A subgraph of a signed graph achieved by deletions and contractions of edges is sometimes called the *minor* of the graph. Two signed graphs that can be obtained from each other by switchings are called *switching equivalent.*

10.1.2 Gain graphs

The original definition of gain graphs comes from the generalized network flows (see Chapter 9). Allowing more general possible gains on the edges, we achieve the abstract construction of gain graphs.

Definition 10.4. A *gain graph* (Γ, ϕ) is a graph Γ with a mapping $\phi : E_* \to \mathcal{G}$ from the ordinary edges of Γ to a *gain group* $\mathcal{G}$.

If $\mathcal{G}$ is a group of two elements, then a gain graph is a signed graph with only one difference in terminology. The orientation of a gain graph means making Γ a directed graph, instead of bidirected as in the case of signed graphs. It is understood that $\phi(e^{-1}) = \phi(e)^{-1}$ where e^{-1} denotes the edge e with direction reversed.

Generalized network flows are based on gain graphs with $\mathcal{G} = \mathbb{R} \setminus \{0\}$ or $\mathbb{R}_+$, the multiplicative group of the real or positive real numbers. In this case the node-edge incidence matrix of the oriented gain graph is straightforward. There is a non-zero element in the intersection of a column and a row if the node of the row is an end-node of the edge that corresponds to the column. The columns of half-edges contain a unique $+1$ or -1, a column of a link e has a -1 in the out-node of the edge and $\phi(e)$ in the row of its in-node. If e is a loop, then its column has only one non-zero, $\phi(e) - 1$.

Let us suppose that $C = \{e_1, e_2, \ldots, e_k\}$ is a cycle with oriented edges. Change the direction of the edges, together with their gains, so that all of them are oriented in the same direction, i.e., C becomes a directed cycle $C' = \{e'_1, e'_2, \ldots, e'_k\}$. The gain of C is the product of the direction-adjusted gains of cycle-edges, i.e., $\phi(C) = \phi(e'_1)\phi(e'_2)\cdots\phi(e'_k)$. The gain of a cycle obviously

depends on the chosen starting point and direction unless the gain is 1, the identity element of the gain group. An ordinary cycle with gain 1 is called *balanced.*

Balanced cycles form a linear class in the following sense. If the union of two balanced cycles C_1 and C_2 is a theta graph, then the symmetric difference $C_1 \triangle C_2$ (i.e., the third cycle in the theta graph) is also a balanced cycle. This property is generalized in biased graphs.

10.1.3 Biased graphs

Definition 10.5. A *biased graph* $\Omega = (\Gamma, \mathcal{B})$ consists of the underlying graph Γ and a linear subclass $\mathcal{B}$ of the ordinary cycles in Γ such that if $C_1, C_2 \in \mathcal{B}$ and $C_1 \cup C_2$ is a theta graph, then $C_1 \triangle C_2 \in \mathcal{B}$. Ordinary cycles in $\mathcal{B}$ are called the *balanced cycles.* Any cycle (ordinary or not) which is not in $\mathcal{B}$ is *unbalanced.*

A subgraph of Ω is biased in the obvious way, a cycle of a subgraph is balanced if and only if it is balanced in the whole graph. A subgraph or edge-set of Γ is *balanced* if every cycle in it is balanced. It is *contrabalanced* if it has no balanced cycles and no loose edges. Recall that a 1-tree contains exactly one cycle. If that cycle is balanced or unbalanced, then we also call the 1-tree balanced or unbalanced, respectively.

The deletion of $S \subseteq E$ results in a biased subgraph on edge-set $E \backslash S$. The biased graph Ω/e achieved by the contraction of the edge e is $(\Gamma/e, \mathcal{B}/e)$, that is, its underlying graph is obtained from Γ by contracting e, and the balanced cycles are those which can be extended to a balanced cycle in Ω. Any biased graph is a minor of Ω if achieved by deletions and contractions from Ω.

A subclass of the ordinary cycles in Γ is called *additive* if in any theta subgraph an odd number of cycles belong to the subclass. In signed graphs, the balanced cycles are exactly the even ordinary cycles, and they form an additive class. Zaslavsky proved that the converse is also true.

Theorem 10.6. ([69], Th. 6.) *A biased graph* $(\Gamma, \mathcal{B})$ *is a signed graph, i.e., there is a signed graph* (Γ, σ) *such that* $\mathcal{B} = \{$*even ordinary cycles in* $\Gamma\}$*, if and only if* $\mathcal{B}$ *is additive.*

10.2 Matroids

In this section we give an overview of the matroidal results related to biased graphs. Contrary to the previous section, where we moved from the specific to the more general, here we start with the most general objects, namely the matroids based on biased graphs, then show how it is specialized to signed graphs. First of all, however, let us outline the necessary theory about matroids. For a comprehensive coverage, we refer the reader to Oxley [53] and Truemper [64].

10.2.1 Relevant matroid terminology

Definition 10.7. $M=(H,\mathcal{I})$ is a matroid on ground set H if $\mathcal{I}$ is a set of subsets of H satisfying:

(i) $\emptyset \in \mathcal{I}$,

(ii) if $I_1 \in \mathcal{I}$ and $I_2 \subseteq I_1$, then $I_2 \in \mathcal{I}$,

(iii) for any subset F of H the maximal subsets of F that are in $\mathcal{I}$ have the same cardinality, called the *rank* of F.

Subsets of H in $\mathcal{I}$ are called the *independent* sets, a maximal independent subset in H is the *basis* of M. The rank of H, which is the cardinality of a basis, is called the rank of the matriod, denoted as $r(M)$. A *circuit* is a minimal dependent subset of H. If R is a basis of M and $s \in H\backslash R$, then there is a unique circuit in $R \cup s$, called the *fundamental circuit of s with respect to R.* There are several different but equivalent ways to define a matroid. For example, one can define a matroid on a given ground set through its bases, rank-function, circuits, or fundamental circuits.

A standard example of matroids is when H is a finite set of vectors from a vectorspace over a field $\mathcal{F}$ and $\mathcal{I}$ contains the linearly independent subsets of H. This kind of matroid is called the *linear matroid.* If for a matroid M there exists a field $\mathcal{F}$ such that M is a linear matroid over $\mathcal{F}$, then M is *representable* over $\mathcal{F}$. Matroids representable over GF(2), the field with two elements, are called *binary*, and matroids representable over GF(3), the field with three elements, are called *ternary.* A matroid representable over every field is *regular.* For a matroid to be regular, it must obviously be binary and ternary. It turns out that this condition is also sufficient.

Theorem 10.8. (Tutte [65, 66]) *A matroid is regular if and only if it is GF(2)- and GF(3)-representable.*

Linear matroids can be represented by matrices. Let M be a linear matroid on a finite set of vectors over $\mathcal{F}$. The matrix A made up of these vectors is a *standard representation matrix* of M. There is a one-to-one correspondence between linearly independent columns of A and independent sets in M, so the linear matroid $M = M(A)$ can be fully given by its representation matrix A. Obviously, deleting linearly dependent (over $\mathcal{F}$) rows from A does not have any effect on the structure of independent columns. Therefore, we can assume that the rows of A are linearly independent.

If A is the node-edge incidence matrix of a (connected) undirected graph G, then the corresponding binary matroid $M(A)$ is called *graphic.* A graphic matroid can be viewed as defined on the edges of the graph, the independent sets are the (edge-sets of) forests, a circuit is a cycle, a basis is a spanning tree.

There is another, more compact representation matrix of a linear matroid $M=M(A)$. To get it, first delete linearly dependent (over $\mathcal{F}$) rows from A, if there are any, then choose a basis R of M. It corresponds to a basis of A, also denoted by R. By pivoting (executed in $\mathcal{F}$) on elements of R, it can be converted to an identity matrix. It is clear that the independence is not affected by pivots, so

the transformed matrix also represents M. This remains true if the full-row-size identity submatrix is deleted. The remaining matrix, denoted as B, is a *compact representation matrix* of M. The rows (columns, respectively) of B correspond to the vectors in (out of, respectively) the basis R. Note that a matroid might have several different compact representation matrices, depending on the selection of basis R. A column s of B gives us the fundamental circuit of s with respect to the chosen basis R. In fact, the basic vectors whose rows contain non-zeros in column s are the basic elements of the fundamental circuit. As an example, take the following matrix over GF(2).

$$B = \begin{array}{c|cccc} & h_4 & h_5 & h_6 & h_7 \\ \hline h_1 & 0 & 1 & 1 & 1 \\ h_2 & 1 & 0 & 1 & 1 \\ h_3 & 1 & 1 & 0 & 1 \end{array}$$

The binary matroid represented by B is called the *Fano-matroid*, denoted as F_7. Its ground set has seven vectors $\{h_1, \ldots, h_7\}$. Subset R=$\{h_1, h_2, h_3\}$ is a basis, so $r(F_7) = 3$. The fundamental circuit of, say, h_4 is $\{h_2, h_3, h_4\}$.

Graphic matroids provide another example. Take a connected undirected graph G and its node-edge incidence matrix A, which is the standard representation matrix of the graphic matroid based on G. First we delete a row to make A a full row rank matrix A'. Selecting a basis R of A', which corresponds to a spanning tree of G, and pivoting on its elements is equivalent to premultiplying A' by the inverse of R (all the operations are done modulo 2). As a result, we get a matrix B the columns of which give the fundamental cycles of G with respect to R, i.e., the unique cycles that contains exactly one non-tree edge. In other words, B is an unsigned (i,.e., modulo 2) network matrix.

Lemma 10.9. *Any compact representation matrix of a graphic matroid can be signed with* $\{+,-\}$ *to obtain a network matrix. Conversely, the binary support of any network matrix is the compact representation matrix of a graphic matroid.*

Basic operations on matroids are dualization, deletion and contraction. If M=$(H, \mathcal{I})$ is a matroid, then M^*=$(H, \{H \backslash I \mid I \in \mathcal{I}\})$ is also a matroid, called the *dual* of M. If M is linear and B is its compact representation matrix, then B^T is the compact representation matrix of M^*. A consequence of this fact is that if M is representable over a field $\mathcal{F}$, then so is the dual of M. In matroid terminology, dualization is usually expressed by the 'co' prefix. Thus, if M^* is graphic, then M is *cographic*.

The *deletion* of $U \subseteq H$ from M results in a matroid (denoted as $M \backslash U$) on $H \backslash U$ the independent sets of which are in $\{I \subseteq H \backslash U \mid I \in \mathcal{I}\}$. The matroid resulting from the *contraction* of a set $T \subseteq H$ in M is defined as M/T=$(M^* \backslash T)^*$. Deletion and contraction in graphic matroids are naturally expressed by deletion and contraction of edges. A matroid achieved by contractions and deletions

in M is called a *minor* of M. For any minor N of a linear matroid M one can find a compact representation matrix B of M such that N is represented by a submatrix of B. As a corollary, representability over a field is maintained under minor-taking.

Another important class of matroids is that of the *uniform matroids*. If $|H| = m$, and $\mathcal{I}$ contains all the subsets with at most r elements ($r \leq m$), then $(H, \mathcal{I})$ is the uniform matroid of rank r on m elements, denoted as U_m^r. It is a classical result that binary matroid can be characterized by forbidden uniform minors.

Theorem 10.10. (Tutte [65]) *A matroid is binary if and only if it does not have U_4^2 minors.*

This theorem initiated a number of similar results on representability. One of them is about GF(3) representability, due to Reid, who never published his proof. The first published proofs were given independently by Bixby [8] and Seymour [57].

Theorem 10.11. *A matroid is ternary if and only if it has no F_7, F_7^*, U_5^2 or U_5^3 minors.*

Another kind of characterization is when matroids are decomposed to simpler matroids in special ways. The most striking of this kind of results is the decomposition of regular matroid, due to Seymour [58]. It claims that the building blocks of a regular matroid are graphic matroids, cographic matroids, or matroids represented by the following compact representation matrix (Although it is a bit imprecise, R_{10} wil denote all matroids of this latter kind).

$$A(R_{10}) = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 1 \end{bmatrix}$$

The special way regular matroids are built up are through 1-sums, 2-sums and 3-sums, which we do not define here. The interested reader can find the definitions in e.g., [64]. The decomposition theorem of regular matroids goes as follows.

Theorem 10.12. (Seymour [58]) *Every regular matroid can be produced from graphic and cographic matroids and R_{10} by consecutive 1-, 2-, and 3-sums. Conversely, every matroid produced this way is regular.*

The importance of regular matroids is very much (from our point of view, almost exclusively) connected to the following theorem.

Theorem 10.13. (Tutte [65, 66]) *A matroid is regular if and only if it has a binary compact representation matrix the 1s of which can be replaced by ± 1 so that the resulting real matrix is totally unimodular.*

For the translation of the decompostion theorem of regular matroids to matrix language, we first have to find the matrix version of the matroids and operations used. We know that graphic and cographic matroids correspond to unsigned network matrices and their transposes. There are two possible totally unimodular representation matrix of R_{10}:

$$B_1 = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 1 \end{bmatrix} \quad \text{and} \quad B_2 = \begin{bmatrix} 1 & 0 & 0 & 1 & -1 \\ -1 & 1 & 0 & 0 & 1 \\ 1 & -1 & 1 & 0 & 0 \\ 0 & 1 & -1 & 1 & 0 \\ 0 & 0 & 1 & -1 & 1 \end{bmatrix} \tag{10.1}$$

Note that as R_{10} is not graphic or cographic, these matrices are not network matrices neither the transposes of a network matrix. 1-, 2-, and 3-sums of matrices are defined as

1-sum: $A \oplus_1 B := \begin{bmatrix} A & 0 \\ 0 & B \end{bmatrix}$

2-sum: $\begin{bmatrix} A & a \end{bmatrix} \oplus_2 \begin{bmatrix} b \\ B \end{bmatrix} := \begin{bmatrix} A & ab \\ 0 & B \end{bmatrix}$

3-sum: $\begin{bmatrix} A & a & a \\ c & 0 & 1 \end{bmatrix} \oplus_3 \begin{bmatrix} 1 & 0 & b \\ d & d & B \end{bmatrix} = \begin{bmatrix} A & ab \\ dc & B \end{bmatrix}$

where A, B are matrices, a, d and b, c are column and row vectors of appropriate sizes. Furthermore, for technical reasons, we assume that A and B are large enough, i.e., the number of rows plus columns is at least 4.

Hence, the matrix version of Theorem 10.12 is the following theorem, also due to Seymour [59], appearing with proof in [64] and without proof in [51] and [55].

Theorem 10.14. *Up to row and column permutations and scaling by ± 1 factors, any totally unimodular matrix is a network matrix, or the transpose of a network matrix, or matrix B_1 or B_2 of* (10.1)*, or can be constructed recursively from these matrices by 1-, 2-, and 3-sums.*

Now we are ready to define the matroids of biased graphs.

10.2.2 Bias matroid

In Lemma 10.9 we stated the well-known result that unsigned network matrices are compact representation matrices of graphic matroids. In graphical terms this means that with any direction of the edges of G, the undirected graph underlying the graphical matroid, leads to a network matrix defined on the now directed graph G. This construction can be copied for binet matrices. Take a bidirected

graph, remove the orientation of the edges to get a signed graph Σ. Any binet matrix based on the bidirected graph is the compact representation matrix of the linear matroid of the node-edge incidence matrix of Σ.

In this section we investigate this matroid, and its corresponding, more general matroid based on biased graphs. We do this because several matroidal results about signed graphs can be naturally extended to biased graphs, and the latter provide a useful insight into the specific, signed graphic case.

So let $\Omega=(\Gamma, \mathcal{B})$ be a biased graph. The definition of its bias matroid is the following.

Definition 10.15. (Zaslavsky [73]) The ground set of the *bias matroid* $G(\Omega)$ of the biased graph Ω is the edge set of Γ. A basis of $G(\Omega)$ consists of a spanning tree in each balanced component of Ω, and an unbalanced 1-tree in any other component.

Equivalently, one can specify the possible circuits of $G(\Omega)$. A circuit can be a balanced cycle; the union of two unbalanced cycles which meet at one point (a *tight handcuff*); the union of two node-disjoint unbalanced cycles and a path that connects them but has no common point with the cycles other than its end-points (a *loose handcuff*); or a contrabalanced theta graph.

In signed graphs, viewed as biased graphs, there are no contrabalanced theta graphs (cf. Theorem 10.6), so the circuits can be only even cycles or handcuffs. To express this difference, we will call the bias matroid of a signed graph, a *signed graphic matroid* (the name was given by Zaslavsky in [70]). The set of circuits in a signed graphic matroid is exactly the same set as the fundamental circuits defined in Section 5.1. This equivalence is not coincidental. The bias matroid of a signed graph is the linear matroid represented by the node-edge incidence matrix of the graph. It follows that a binet matrix is a compact representation matrix (over $\mathbb{R}$) of a signed graphic matroid as it is obtained from the node-edge incidence matrix by $\mathbb{R}$-pivots.

Theorem 10.16. *If $G(\Sigma)$ is a signed graphic matroid based on signed graph Σ, and B is a compact representation matrix of $G(\Sigma)$ over $\mathbb{R}$, then B is a binet matrix. Conversely, any binet matrix is the compact representation matrix of a signed graphic matroid.*

Proof: Let A be the node-edge incidence matrix of Σ. As defined in Section 10.1.1, A is the incidence matrix of the bidirected graph $\vec{\Sigma}$ achieved by orienting the edges of Σ. In Corollary 4.7, we stated that columns of A are linearly independent if and only if the subgraph determined by them has only isolated node, tree, and odd 1-tree components. It follows that the bases of the linear matroid $M(A)$ of A are exactly those listed in Definition 10.15. Hence $M(A) = G(\Sigma)$. As described above, the derivation of a compact representation matrix B of $M(A)$ includes making A a full row rank matrix A', selecting a basis R of A', and premultiplying A' by R^{-1}. This is exactly the derivation of binet matrices from incidence matrices, thus B is a binet matrix. This proves both parts of the theorem. □

We know from Lemma 5.3 that switchings in a bidirected graph do not alter its binet matrices. This phenomenon can be expressed in matroidal terms too. Notably, if two signed graphs are switching equivalent, then their signed graphic matroids are the same.

To be even more specific, the bias matroid of an undirected graph, which is a signed graph where all the edges are odd, is the even-cycle matroid, employed by Doob [21]. At the other end of the scale, if Ω is balanced, that is, all its cycles are balanced, then $G(\Omega)$ is the graphic matroid of Γ.

It is shown in [73] that minors of the bias matroid of Ω correspond to the minors of Ω. This is equivalent to saying that deletions and contractions of edges in a biased graph correspond to deletions and contractions in its bias matroid.

Theorem 10.17. (Zaslavsky [70, 73]) *The class of biased matroids is closed under minor-taking. The same is true for signed graphic matroids.*

It is a standard technique of matroid theory to find *minimal violators* for a given property, i.e., matroids that do not have this property but all their minors do. Zaslavsky [71, 73] gave some minimal violators of bias matroids. He showed that the Fano-matroid F_7, its dual F_7^*, U_7^3 are not bias matroids but all their minors are such. For illustration, we prove that U_4^2 is signed graphic, but U_5^2 is not.

Lemma 10.18. (Zaslavsky [70]) *U_4^2 is a signed graphic matroids. U_5^2 is not a signed graphic matroid.*

Proof: We have to show that there cannot be a signed graph whose bias matroid is U_5^2, but U_4^2 is a signed graphic matroid. First we prove the latter. In the signed graph of Figure 10.1, any two edges form an independent set, but none of the subgraphs with 3 edges do. Thus, its signed graphic matroid is U_4^2.

Figure 10.1: A signed graph whose signed graphic matroid is U_4^2. The heavy edges are odd.

To show that U_5^2 is not signed graphic, we eliminate all the cases. First, if a signed graph Σ has the signed graphic matroid U_5^2, then it must have 5 edges and any two of them form a basis. In other words, any subgraph with at least three edges is not independent, but all subgraphs with two edges are such. This rules out signed graphs on a single node, as they do not have two independent edges. Let us assume that there are no isolated nodes in Σ. If a signed graph has more than one component, then the union of independent subgraphs of the components is also independent. It follows that Σ cannot have more than two components. If it has two components, then at least one of them has at least two edges. Taking two edges forming a basis from this component and one edge from the other component would result in an independent set with three edges, a contradiction.

Thus, Σ is connected. There cannot be three parallel edges in Σ because they would form a contrabalanced theta graph, which is impossible in signed graphs, according to Theorem 10.6. If Σ

has more than three nodes, then there would be a tree with three edges in it, i.e., an independent set with more than two edges. For a similar reason, there cannot be a half-edge or a loop in Σ if it has three nodes. If Σ has three nodes, the only possible structure left is a triangle in which two edges are repeated so that the graph contains two 2-cycles. The 2-cycles must be odd to from a basis, so adding an extra edge to either of them would form an independent set with more than two edges.

We are left with the case when Σ has exactly two nodes. It cannot have more than two of half-edges and loops because then two of them would be incident to the same node and they would not be independent. But then Σ must contain three parallel links, which we already ruled out. We examined all the possible cases and they all led to a contradiction, so U_5^2 is not a signed graphic matroid. □

With similar case analysis, it can also be shown that U_5^3 is not signed graphic, which leads to the following result.

Lemma 10.19. (Zaslavsky [70]) *The signed graphic matroid is ternary.*

Proof: By Theorem 10.11 and Theorem 10.17, we have only to show that F_7, F_7^*, U_5^2 and U_5^3 cannot be a signed graphic matroid. As mentioned above, F_7 and F_7^* are not bias matroids, U_5^2 and U_5^3 are not signed graphic. □

A very recent result due to Daniel Slilaty and independently by Hogxun Qin and Thomas A. Dowling [60] gives a technique to find minimal forbidden minors for signed graphic matroids. They proved that a connected cographic matroid (i.e., a matroid that is not a 1-sum of two smaller matroids, and its dual is a graphic matroid based on an undirected graph G) is signed graphic, if and only if G cannot be embedded into the projective plane. Thus connected cographic matroids of minimally non-embeddable graphs are all minimal forbidden minors for signed graphic matroids.

Lemma 10.19 is not all that we can say about the representability of signed graphic matroids. In fact, they are representable over not only $GF(3)$ but over any field where 2 is not a zero-divisor, which in technical terms is refered to as a field the characteristic of which is not equal to 2.

Theorem 10.20. (Zaslavsky, [70]) *A signed graphic matroid is representable over any field of characteristic not equal to 2.*

Proof: Let Σ be a signed graph and $\vec{\Sigma}$ is the bidirected graph achieved by an arbitrary orientation of its edges. The signed graphic matroid of Σ is represented by the node-edge incidence matrix A of $\vec{\Sigma}$. By Theorem 4.8 and Theorem 2.4, matrix A is totally 2-modular, so any non-singular square submatrix has determinant $\pm 2^r$ ($r \geq 0$). It follows that if a set of column vectors in A are linearly independent over $\mathbb{R}$, then they are linearly independent in any field whose characteristic is not 2. □

The proof only exploits the total 2-modularity of the incidence matrix, so the following theorem is straightforward.

Theorem 10.21. *Let $k = p^r$ where p is a prime and $r \in \mathbb{N}$. The linear matroid $M(A)$ represented by a totally k-modular matrix A is representable over any field of characteristic not equal p.*

The next corollary is by Theorem 2.5, which claims that an integral k-regular matrix is totally p-modular for prime-power $k = p^r$.

Corollary 10.22. *Let $k = p^r$ where p is a prime, $r \in \mathbb{N}$, and A is an integral k-regular matrix. Then the linear matroid represented by A is representable over any field of characteristic not equal p.*

Let us now turn to GF(2)-representability. Zaslavsky showed in [71, Cor. 3.2] that binary bias matroids are actually signed graphic matroids.

Theorem 10.23. *If $G(\Omega)$ is a binary bias matroid, then there is a signed graph Σ such that $G(\Omega) = G(\Sigma)$.*

Combining this result with Lemma 10.19 and Theorem 10.8 we get

Theorem 10.24. (Zaslavsky [71]) *If a biased matroid is binary, then it is regular.*

We know from Theorem 10.20 that signed graphic matroids are representable over any field whose characteristic is not 2, so it is interesting to characterize signed graphs whose bias matroids are representable over the remaining fields. Zaslavsky [71] and Pagano [54] accomplished this task. The former author gave a characterization of signed graphs that give rise to binary bias matroids, the latter simplified the characterization and showed with the help of Whittle's [68] results that a signed graphic matroid can be one of the following three types:

(i) regular,

(ii) representable over any field except $GF(2)$, or

(iii) representable over any field of characteristic other than 2.

These characterizations imply an alternative proof of Theorem 7.6. The exact details would need too deep a coverage of signed and biased graphs, therefore we give only the main ideas. Let Σ be a signed graph, and $\vec{\Sigma}$ is the bidirected graph obtained by orienting Σ. A binet matrix B based on $\vec{\Sigma}$ is the compact representation matrix of the signed graphic matroid $G(\Sigma)$. B is totally unimodular if and only if $G(\Sigma)$ is regular, which, in turn, occurs if and only if $G(\Sigma)$ is binary. The characterizations mentioned above essentially claim that this can happen if and only if either Σ has nodes deleting which results in a graph with no odd edges, or Σ is connected and has no two node-disjoint odd cycles. These cases correspond to the conditions given in Theorem 7.6. Lovász and Schrijver dealt with graphs that satisfy the second condition, i.e., have no two node-disjoint odd cycles. Unfortunately, this result has not been published, though it is refered to in several papers as [32]. One of the results of this work appeared, without proof, in Truemper [64, (11.5.20) Theorem].

10.2.3 Lift matroids

Lifting a matroid M is a general matroidal operation, the precise definition of which can be found in Oxley [53]. It essentially means finding a matroid N such that M can be obtained from N by contracting an element e_0, and then deleting e_0 from N to get a lift of M. Zaslavsky defined two lifted versions of the bias matroid in [73]. The first, the extended lift matroid, is the matroid we achive if we execute lifting but stop before deleting e_0, the second is the precise lifted matroid. The extended lift matroid is the more important for us as it provides a key to characterize signed graphic matroids whose edge-node incidence matrices have strong Chvátal rank 1. These kind of graphs appeared in Section 7.3.

Definition 10.25. Let Ω be a biased graph on the underlying graph $\Gamma(N, E)$. The *extended lift matroid* $L_0(\Omega)$ on the ground set $E \cup e_0$ is given by either of the following definitions:

(a) A basis of $L_0(\Omega)$ consists of a maximal forest F of Γ together with one more element which is either e_0 or an edge $e \in E$ forming an unbalanced cycle in $F \cup e$.

(b) A circuit of $L_0(\Omega)$ is either a balanced cycle, a contrabalanced theta graph, the union of two unbalanced cycles having at most one common node, or the union of e_0 and an unbalanced cycle.

Definition 10.26. The *lift matroid* $L(\Omega)$ of Ω is $L_0(\Omega)\backslash e_0$.

There is a natural binary representation matrix of the extended lift matroid of a signed graph which has no half-edges or loops. Let $\Sigma = (\Gamma, \Lambda)$ be such a signed graph and build the representation matrix A_Σ as follows. We start with the node-edge incidence matrix of the undirected Γ (having 1's at the end-nodes of the edges, zeros elsewhere), then adjoin an additional row recording if an edge is odd or even. That is, this row has 1 in a column whose edge is in Λ and 0 in columns whose edge is not in Λ. Then we add a new column that corresponds to e_0 and has 1 in the additional row, zeros in other rows.

Lemma 10.27. *A_Σ is a standard representation matrix of $L_0(\Sigma)$ over GF(2).*

Proof: We show that the GF(2)-independent columns of A_Σ correspond to a subgraphs of Σ with componets that are trees or odd 1-trees, and vice versa. Let T denote a column submatrix of A_Σ. The columns of T are GF(2)-dependent if and only if there is a subset of them, T' such that all rows of T' have even number of 1s. Hence, the columns corresponding to a tree are independent (as all subgraphs of a tree has a leaf node, whose row has only one 1) and so are the columns of an odd 1-tree (any of its subgraphs has a leaf node or is an odd-cycle, the last row of which has odd number of 1s). Furthermore, if a connected subgraph is not a tree or an odd 1-tree, then it either contains an even cycle or two odd cycles, and the rows of the corresponding submatrices have even number of 1s. □

This representation shows that if Ω is a signed graph, then its extended lift matroid is binary. In [71], Zaslavsky proved the converse.

Theorem 10.28. *$L_0(\Omega)$ is binary if and only if Ω is a signed graph.*

The proof relies on the fact that if $L_0(\Omega) = U_4^2$, then Ω contains a contrabalanced theta graph. Full characterizations of bias graphs for which $L(\Omega)$ is binary is also given in [71].

An important application of the extended lift matroids is the construction of signed graphs whose edge-node incidence matrix has strong Chvátal rank 1. As it is also mentioned elsewhere in this dissertation (see Theorem 6.6), Gerards and Schrijver [33] gave a characterization of bidirected graphs with this property. Their result extends to signed graphs in a natural way. The key graph in the characterization is K_4 the complete graph on four nodes. A signed K_4 is called *odd-K_4*, if it has three odd edges forming a triangle, or it is obtained from such a graph by switchings.

Theorem 10.29. (Gerards and Schrijver [33]) *Let A be the edge-node incidence matrix of a signed graph Σ. It has strong Chvátal rank 1, if and only if Σ has no odd-K_4 minors.*

Before we show how it is related to matroids, we need one more definition. A graph is called *2-connected*, if it has no node the deletion of which would increase the number of unconnected components with at least one edge. In the following theorem, also due to Gerards and Schrijver, we assume that the graphs are 2-connected, a technical assumption.

Theorem 10.30. *Let Σ be a 2-connected signed graph. Then Σ has no odd-K_4 minors if and only if $L_0(\Sigma)$ has no F_7^* minor that contain e_0.*

Using this result, Truemper [64] gave a construction of signed graphs without odd-K_4 minors.

Chapter 11

Connection to the row spaces of matrices

The approach we followed in this dissertation to generalize totally unimodular matrices focuses on the matrices themselves. This is, however, not the only possibility because total unimodularity is a very prevalent notion and it reappears in several, seemingly distant parts of combinatorial optimization. For instance, in Section 10.2.1 we mentioned that totally unimodular matrices and regular matroids are two faces of the same structure (see Theorem 10.13). Another area where totally unimodular matrices play an important role is Tutte's theory of chain groups, see [66].

A *chain group* C is a subgroup of the additive group $\mathbb{Z}^n$, and it is called *regular*, if for any non-zero element x of C with minimal support (defined formally later), the non-zero entries of x all have the same absolute value. Now Tutte proved the following.

Theorem 11.1. (Tutte) *Let A be an integral matrix and let $M = [I, A]$. Then the chain group $\{x \mid Mx = 0,\ x$ integral$\}$ is regular if and only if A is totally unimodular.*

This theorem shows that the connection of totally unimodular matrices and regular chain groups is established through a special vectorspace, namely the null space of M. This observation led to the definiton of regular vectorspaces, and consequently a generalization of total unimodularity through vectorspaces, due to Lee [48].

As Lee explained in [48, p 22], he opted for generalizing subspaces instead of matrices because "much of the practical and theoretical significance of the totally unimodular matrices lies in the structure of the associated spaces". As we will show in this chapter, this statement is largely true, and several of our results about k-regular matrices can be proved in this setting. However, we will also point out instances where the k-regularity of a matrix reveals more than what was proved in [48]. This is because Lee employes determinants, that is, he essentially follows the k-modular approach.

As we argued earlier in the thesis, total k-modularity cannot grasp the essence of total unimodularity, and it is k-regularity that generalizes total unimodularity the best.

We start the chapter with the necessary definitions, in a form that is slightly less general form than in [48], but broad enough for our purposes. Then we exhibit the relationship between vectorspaces and matrices in Section 11.2. This section also contains some relevant results from [48] and shows how they can be translated to our terminology. Finally, in Section 11.3 we mention some matroidal results connected to vectorspaces, due to Lee [49], and show their connection to k-regular matrices.

11.1 Definitions

Let V be a linear vectorspace over $\mathbb{Q}$. In other words, V is a subspace of $\mathbb{Q}^E$ for a finite non-empty set E. That is why we use, following Lee [48], the term 'subspace' for V. For any $v \in V$, the *support* of v is the set of coordinates for which v is non-zero, i.e., $supp(v) = \{i \in E \mid v_i \neq 0\}$. A vector x of V is called *elementary*, if it is not the all-zero vector, and it has minimal support in $V \setminus \mathbf{0}$, i.e.,

$$x \in V \setminus \mathbf{0} \text{ and } \nexists y \in V \setminus \mathbf{0}, \text{ such that } supp(y) \subsetneq supp(x).$$

For example, if $V = \mathbb{Q}^E$, then the elementary vectors are exactly the vectors with only one non-zero entry.

For a subset S of $\mathbb{Q}$, V is called *S-regular* if every elementary vector of V can be scaled by a non-zero rational so that all of its non-zero elements are in S, i.e.,

$$\text{if } x \in V \text{ is elementary, then } \exists \lambda \in \mathbb{Q} \setminus 0 \text{ such that } \lambda x_j \in S \; \forall j \in supp(x).$$

This definition generalizes the notion of regular subspaces. Recall from above that a chain group is called regular, if the non-zero elements with minimal support have non-zero entries with the same absolute value. By adapting this definition to vectorspaces, we call a subspace V *regular* if each elementary vector of V is a rational multiple of a $0, \pm 1$ vector. Clearly, these are the S-regular subspaces for $S = \{+1, -1\}$.

If T is a finite subset of $\mathbb{Q}_+$ and $\omega \in \mathbb{N}$, then any V that is S-regular for

$$S = \left\{ \pm \prod_{t \in T} t^{p_t} \mid p_t \in \mathbb{N} \, (t \in T), \sum_{t \in T} p_t \leq \omega \right\}$$

is called *T-adic of order ω*. For example, if $T = \{1, 2, 3\}$ and V is T-adic of order 2, then any elementary vector of V can be scaled to a non-zero vector with elements of $\{0, \pm 1, \pm 2, \pm 3, \pm 4, \pm 6, \pm 9\}$. Obviously, a T-adic subspace of order 1 is $\pm(T \cup \{1\})$-regular. For $T = \{k\}$, when k is a positive integer, we will use the term *k-adic*. The word 2-adic is replaced by *dyadic*. Lee defines *k-regular*

subspaces as the $\{1,2,\ldots,k\}$-adic subspaces of order 1. A subspace that is dyadic of order 1 is also 2-regular.

Note that this k-regularity is not exactly the same as the k-regularity defined in Definition 2.2 and used throughout the dissertation. What we have here are k-regular vectorspaces, while Definition 2.2 is about matrices. We will discuss the relationship of the two definitions later, in Section 11.2.

As Theorem 11.1 implies, the main interest is in linear vectorspaces that arise from matrices. So let M be a full row rank rational matrix of order $m \times n$, and $rs(M)$ denote its *row space*, i.e., $rs(M) = \{v \in \mathbb{Q}^n \mid \exists u \in \mathbb{Q}^m, v = uM\}$. Trivially, if B is a basis of M, then $rs(B^{-1}M) = rs(M)$, so one can assume that M contains an identity matrix of size m. The *null space* of M is defined as $ns(M) = \{u \in \mathbb{Q}^n \mid Mu = 0\}$. It is easy to see that if $M = [I, A]$, then $ns(M) = rs([-A^T, I])$.

A central result about row spaces is the following.

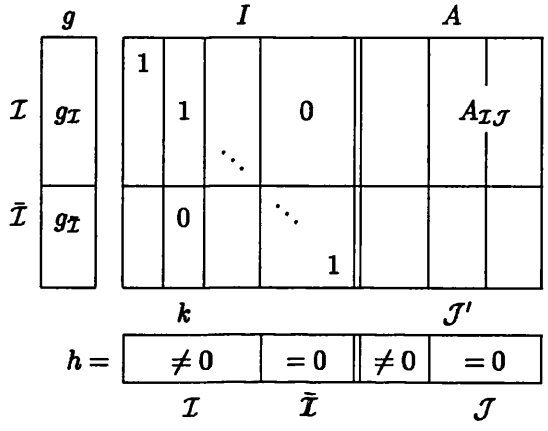
Lemma 11.2. *Let $M = [I, A]$ be an $m \times n$ rational matrix. Then it is true that:*

(a) For every basis B of M, the rows of $B^{-1}M$ are elementary vectors of $rs(M)$.

(b) For every elementary vector h of $rs(M)$ there exists a basis B of M, such that up to non-zero scalar multiplication, h is a row of $B^{-1}M$.

Proof: (a) Let us permute the rows and columns of $B^{-1}M$ so that it has the form $[I, C]$ for a suitable matrix C. Without loss of generality, we will only show that the first row of this matrix, let us call it h, is elementary. For any vector $f \in rs(M)$ satisfying $supp(f) \subseteq supp(h)$ it holds that $f_2 = \cdots = f_m = 0$. As $rs(M) = rs(B^{-1}M)$, f is a linear combination of the rows of $[I, C]$. This implies that $f = \lambda h$ for a $\lambda \in \mathbb{Q} \setminus 0$, which means that h is elementary.

(b) The elementary vector h is in $rs(M)$, so there exists a $g \in \mathbb{Q}^m$ such that $h = gM = (g, gA)$. Let $\mathcal{I} = \{i \mid g_i \neq 0\}$, $\bar{\mathcal{I}} = \{i \mid g_i = 0\}$, $\mathcal{J} = \{j \mid (gA)_j = 0\}$ and $A_{\mathcal{I}\mathcal{J}}$ be the submatrix of A defined by the rows and columns whose indices are in $\mathcal{I}$ and $\mathcal{J}$, respectively. Furthermore, $g_{\mathcal{I}}$ and $g_{\bar{\mathcal{I}}}$ are subvectors of g corresponding to indices in $\mathcal{I}$ and $\bar{\mathcal{I}}$. The following diagram illustrates these notations.



The rows of $A_{\mathcal{I}\mathcal{J}}$ are linearly dependent as $g_{\mathcal{I}}A_{\mathcal{I}\mathcal{J}} = 0$, but any matrix obtained from $A_{\mathcal{I}\mathcal{J}}$ by deleting a row is a full row rank matrix. If not, then there exists an $f_{\mathcal{I}} \in \mathbb{Q}^{|\mathcal{I}|} \setminus \mathbf{0}$ such that $f_{\mathcal{I}}A_{\mathcal{I}\mathcal{J}} = 0$ and at least one element of $f_{\mathcal{I}}$ is 0. This means that if we extend $f_{\mathcal{I}}$ with zeros for indices in $\bar{\mathcal{I}}$, we get a non-zero vector fM from $rs(M)$ whose support is a real subset of $supp(h)$, contradicting the elementary of h. So the rank of $A_{\mathcal{I}\mathcal{J}}$ is $|\mathcal{I}| - 1$, therefore there is a subset $\mathcal{J}'$ of $\mathcal{J}$ of cardinality $|\mathcal{I}| - 1$ such that the columns of A determined by these indices are linearly independent. One can now find an index $k \in \mathcal{I}$, such that the matrix B made up of the columns of M indexed by k, $\bar{\mathcal{I}}$ and $\mathcal{J}'$ (in this order) is non-singular.

This B serves as the basis claimed in statement (b) of the lemma. In fact, the first row h' of $B^{-1}M$ has a unit vector part corresponding to the columns of B, so $h'_i = 0$ if $i \in \bar{\mathcal{I}} \cup \mathcal{J}'$. Let now $a_{\cdot j}$ be a column of A for $j \in \mathcal{J} \setminus \mathcal{J}'$. From the structure of B it is clear that to get h'_j, a_{ij} is multiplied by 0 for $i \in \bar{\mathcal{I}}$. Furthermore, by the choice of $\mathcal{J}'$, $a_{\mathcal{I}j}$ (i.e., the part of $a_{\cdot j}$ that lies in $A_{\mathcal{I}\mathcal{J}}$) is a linear combination of the columns of $A_{\mathcal{I}\mathcal{J}}$ indexed by $\mathcal{J}'$, ensuring $h'_j = 0$. We showed that $supp(h') \subseteq supp(h)$ and h is elementary, hence $h' = \lambda h$ for a $\lambda \in \mathbb{Q} \setminus 0$. □

This lemma is the key to the relationship between k-adic subspaces and k-regular matrices, discussed in the next section.

11.2 From k-adic vectorspaces to k-regular matrices

If we wish to see how results about k-adic vectorspaces can be exploited in the theory of k-regular matrices, we have to relate the two notions. In this section we accomplish this task. The main result is that for integral matrices and prime k, a matrix is k-regular if and only if $rs([I, A])$ is k-adic of order 1. This covers the 2-regularity of integral matrices, one of the most important cases for us. On the other hand, it fails to tell something about rational matrices. In this sense it is similar to total k-modularity, which also worked only for integral matrices, as shown in Chapter 2.

However, one direction of the equivalence between matrices and row spaces does not need integrality or assumptions on k.

Theorem 11.3. *For any A and any $k \in \mathbb{N}$, if $rs([I, A])$ is k-adic of order 1, then A is k-regular.*

Proof: Any non-singular square submatrix R of A can be extended to a basis B of $[I, A] = M$. Lemma 11.2 states that for any row h of $B^{-1}M$ there is a $\lambda \in \mathbb{Q} \setminus 0$ such that $\lambda h \in \{0, \pm 1, \pm k\}$. The unit matrix is part of $B^{-1}M$, so 1 is an element of h. It follows then that $|\lambda| = 1$ or k and the non-zero elements of h are from $\{\pm\frac{1}{k}, \pm 1, \pm k\}$. Thus kB^{-1} and then kR^{-1} are integral. □

Note that this result cannot be extended to higher orders. For example, for $A = [4]$ the row space of $[I, A]$ is dyadic of order 2, but A is not 2-regular.

In the reverse direction integrality becomes crucial. Take, for example, the rational matrix $A = \begin{bmatrix} 2/3 \end{bmatrix}$. A is 2-regular, but $rs([I, A])$ is not dyadic of any order. For integral matrices, however, one can give a partial converse of Theorem 11.3. As we will see, the value of k determines the strength of the result. We start with prime-powers.

Theorem 11.4. *If A is an integral k-regular matrix and $k = p^\omega$ where p is a prime or 1, then the row space of $M = [I, A]$ is p-adic of order ω.*

Proof: In the view of Lemma 11.2, we have to prove that for any basis B of M the rows of $B^{-1}M$ can be multiplied by a $\lambda \in \mathbb{Q} \setminus 0$ so that they are vectors of elements in $\{0, \pm p^r, r = 0, 1, \ldots, \omega\}$. The k-regular matrix A is integral, so by Lemma 2.12, $B^{-1}M$ is also k-regular. Therefore for any element $\mu \neq 0$ of $B^{-1}M$, $k\mu^{-1} \in \mathbb{Z}$. Moreover, the k-regularity of A implies that kB^{-1} is integral, so $k\mu \in \mathbb{Z}$ too. Let $\mu = \frac{a}{b}$ where a and b are relative prime integers. It easily follows that a and b are both divisors of k, and then $|\mu| = p^{\omega'}$ for $\omega' \in \mathbb{Z}$, $-\omega \leq \omega' \leq \omega$.

This would be enough to prove that $rs(M)$ is k-adic of order 2ω, because multiplying any row h of $B^{-1}M$ by k yields a vector of elements in $\{0, \pm p^r, r = 0, 1, \ldots, 2\omega\}$. To push the order down to ω, we show that if h is a row, say the first, of $B^{-1}M$ having non-zero elements $h_i = \pm p^{\lambda_i}$ and $\underline{\lambda} = \min\{\lambda_i\} < 0$, $\bar{\lambda} = \max\{\lambda_i\}$, then $\bar{\lambda} - \underline{\lambda} \leq \omega$. This implies that $p^{-\underline{\lambda}}h$ has elements in $\{0, \pm p^r, r = 0, 1, \ldots, \omega\}$. If $\underline{\lambda} \geq 0$, this holds without multiplying h by $p^{-\underline{\lambda}}$.

One can identify three parts of $B^{-1}M$. It contains a unit matrix I, the not necessarily disjoint B^{-1}, and disjoint $B^{-1}C$ where C is the submatrix of A consisting of columns not in B. Because C is integral, the element $h_j = \pm p^{\underline{\lambda}}$ of h with the minimal exponent is not in $B^{-1}C$. By Cramer's rule, $|h_j| = det(B')/det(B)$ where B' is a submatrix of B obtained by deleting the first column and the j^{th} row. If $h_k = \pm p^{\bar{\lambda}}$ is the element with the maximal exponent, then let s be a column of M such that $h_k = (B^{-1}s)_1$. By Cramer's rule again, $|h_k| = det(B'')/det(B)$ where B'' is the matrix obtained from B by replacing the first column with s. Taking the inverse of B'', we get

$$|(B'')^{-1}_{1j}| = \frac{det(B')}{det(B'')} = p^{\underline{\lambda}} \frac{1}{p^{\bar{\lambda}}} = p^{\underline{\lambda} - \bar{\lambda}}$$

In the first part of the proof we showed that for any basis of M the non-zero elements in its inverse are of the form $\pm p^{\omega'}$ where $\omega' \geq -\omega$. Hence $\bar{\lambda} - \underline{\lambda} \leq \omega$. □

Applying this theorem to prime k, we get the following result.

Theorem 11.5. *If A is an integral k-regular matrix for a k that is prime or 1, then the row space of $M = [I, A]$ is k-adic of order 1.*

Note that the order in Theorem 11.5 cannot be strengthened if k is not the first power of p, as the following example shows. Let $A = \begin{bmatrix} 2 & 1 \\ 0 & 2 \end{bmatrix}$, a 4-regular matrix. However, the row space of $[I, A]$ is dyadic of order 2, but not of order 1.

Theorems 11.3 and 11.5 together provide the following result.

Theorem 11.6. *For any integral matrix A and prime number k, A is k-regular if and only if $rs([I, A])$ is k-adic of order 1.*

Recall that in the terminology used by Lee in [48], 2-regular subspaces are equivalent to subspaces which are dyadic of order 1.

Corollary 11.7. *An integral matrix A is 2-regular if and only if $rs([I, A])$ is 2-regular.*

This fits well with our terminology, at least for the most important case of $k = 2$, as the integral 2-regular matrices are in par with 2-regular row spaces. Unfortunately, this does not extend to 3-regularity.

If k is not the power of a prime, then the results of Theorem 11.4 and Theorem 11.5 do not hold. Take, for example, $A = \begin{bmatrix} 2 & 1 \\ 0 & 3 \end{bmatrix}$, a 6-regular matrix. Choosing $B = A$, $B^{-1}[I, A]$ has the row $(\frac{1}{2}, -\frac{1}{6}, 0, 1)$ showing that $rs([I, A])$ is not 6-adic of any order. This example suggests, however, a theorem for general k.

Theorem 11.8. *If A is an integral k-regular matrix and the prime factorization of k is $p_1^{\omega_1} \cdots p_l^{\omega_l}$, then the row space of $M = [I, A]$ is $\{\pm 1, \pm p_1, \ldots, \pm p_l\}$-adic of order $\omega_1 + \cdots + \omega_l$.*

Proof: Goes along the same lines as the proof of Theorem 11.4. □

In the rest of this section, we list some results from [48], and show how they are related to our findings. In what follows, T is a finite set of positive rationals and ω and η are nonnegative integers. We start with results which deal with determinants.

Theorem 11.9. ([48], Proposition 5.1.) *For any rational $m \times n$ matrix A, $rs([I, A])$ is T-adic of order ω only if every non-singular square submatrix of order r $(1 \leq r \leq m)$ has its determinant in the set*

$$\left\{ \pm \prod_{t \in T} t^{p_t} \mid p_t \in \mathbb{Z} \ \ (t \in T), \sum_{p_t > 0} p_t \leq r\omega \sum_{p_t < 0} (-p_t) \leq r\omega \right\}$$

Specifically, for an integral A that is p-adic of order ω where p is 1 or the power of a prime, we get Theorem 2.5. The following partial converse of Theorem 11.9 is a consequence of Cramer's rule.

Theorem 11.10. ([48], Proposition 5.3.) *If all the bases of a full row rank matrix M have determinants in*

$$\left\{ \pm \prod_{t \in T} t^{p_t} \mid p_t \in \mathbb{Z} \ \ (t \in T), \sum_{p_t > 0} p_t \leq \omega \sum_{p_t < 0} (-p_t) \leq \eta \right\}$$

then $rs(M)$ and $ns(M)$ are T-adic of order $\omega + \eta$. If M is integral and T is a set of integers, then the spaces are T-adic of order ω.

In the special case of $M = [I, A]$, $T = \{k\}$, and $\eta = 0$, this theorem gives Theorem 2.6.

One of the most important results in this dissertation is Theorem 3.8, which deals with the integrality of special polyhedra. One way to state the theorem is that A is k-regular if and only if for any integral vector b and any basic solution x of $max\{cx \mid Ax \leq b,\ x \geq 0\}$, kx is an integral vector. The easy part is the sufficiency of k-regularity, which can be extended to T-adic subspaces. The following theorem claims that if the subspaces related to M are T-adic, then the basic solutions of $max\{cx \mid Mx = b,\ x \geq 0\}$ can be scaled to integers for any integer right hand side vectors, or equivalently, if the right hand side vector is scaled, then the basic solutions are integer.

Theorem 11.11. ([48], Proposition 6.1.) *Let T be a set of positive integers. If $rs(m)$ is T-adic of order ω and $ns(M)$ is T-adic of order η, then for each basis B of M there exists $\delta_i \in \{\pm\prod_{t\in T} t^{p_t} \mid p_t \in \mathbb{N}\ (t \in T),\ \sum_{t\in T} p_t \leq \omega\}$ and $\theta_i \in \{\pm\prod_{t\in T} t^{p_t} \mid p_t \in \mathbb{N}\ (t \in T),\ \sum_{t\in T} p_t \leq \eta\}$ $(i = 1, \ldots, m)$ such that for any integral right hand side vector b and for all basic indices i, $\delta_i(B^{-1}b)_i \in \mathbb{Z}$; and if $b_i = \theta_i z_i$ for $z_i \in \mathbb{Z}$ $(i = 1, \ldots, m)$, then $(B^{-1}b) \in \mathbb{Z}^m$.*

If $M = [I, A]$, then the maximization takes the form $max\{cx \mid Ax \leq b,\ x \geq 0\}$ and by setting $\omega = 1$, $\eta = 1$ we get back the easy direction of Theorem 3.8. Lee also gives a result which is very similar to the substantial part of Theorem 3.8.

Theorem 11.12. ([48], Proposition 6.2.) *Let k be a prime and M be an $m \times n$ full row rank matrix having the property that for each row $M_{i\cdot}$ of M there exists a $q_i \in \mathbb{N}$ $(i = 1, \ldots, m)$ such that $k^{q_i} M_{i\cdot}$ is integral and $\sum_1^m q_i \leq \eta$. If for each basis B of M and for all integral right hand side vectors b there exist $p_i \in \mathbb{N}$ with $\sum_1^m p_i \leq \omega$, such that the basic solution corresponding to B satisfies $k^{p_i}(B^{-1}b)_i \in \mathbb{Z}$ for all basic indices i, then $rs(M)$ and $ns(M)$ are k-adic of order $\omega + \eta$.*

Let us suppose that $M = [I, A]$ is integral. Then with $\eta = 0$ this theorem claims that if for any integral right hand side vector b and for any basis, the basic solution x satisfies $kx \in \mathbb{Z}^m$, then $rs(M)$ is k-adic of order m. On the other hand, for this case Theorem 3.8 ensures that A is k-regular, that is, $rs(M)$ is k-adic of order 1, a stronger statement. The difference is because Lee proves his theorem through Theorem 11.10, while we get around determinants and use basis inverses directly.

To sum up, T-adic subspaces provide a powerful generalization of total unimodularity. They have a substantial connection to k-regular matrices, but in some sense k-regularity is the stronger. That is to say, exploiting the k-regularity of a matrix, we can prove stronger results than with the corresponding property of vectorspaces.

11.3 Matroids of subspaces

In [49], Lee also examined the matroidal structure of T-adic subspaces. He examined the representability of a matroid by special subspaces. By the relationship between subspaces and matrices,

described in the previous section, we can translate some of his results to representability by matrices. The representation of matroids by matrices is presented in Section 10.2.1.

A matroid M on ground set E is represented by a linear subspace $V \subseteq \mathbb{Q}^E$ if the circuits of M are the sets of E with characteristic vector in $\{supp(x) \mid x \text{ is an elementary vector of } V\}$.

Lemma 11.13. *A matroid M is represented by subspace $rs([I, A])$, if and only if $-A^T$ is a compact representation matrix of M.*

Proof: One can easily show that $rs([I, A]) = ns([-A^T, I])$. This implies that the elementary vectors of $rs([I, A])$ correspond to minimally dependent columns of $[-A^T, I]$. The circuits of the matroid represented by the compact representation matrix $-A^T$ are exactly the minimal dependent columns of $[-A^T, I]$. □

Combining this lemma with Theorems 11.3 and 11.5 yields the following result.

Theorem 11.14. *(a) If a matroid M is representable by a subspace that is k-adic of order 1 for a prime k, then M has a compact representation matrix that is k-regular.*

(b) If M is a linear matroid and it has a compact representation matrix that is integral and k-regular, then M is represented by a subspace that is k-adic of order 1.

Proof: Let $V = rs([I, A]$. By Theorem 11.3, V is k-adic of order 1 only if A, and then $-A^T$, is k-regular.

In part (b), if the compact representation matrix A of M is integral and k-regular, then $-A^T$ is also such, and then Lemma 11.13 ensures that M is represented by $rs([I, A])$, a k-adic subspace of order 1 by Theorem 11.5. □

Part (b) of Theorem 11.14 has an interesting consequence for signed graphic matroids (see Section 10.2.2):

Corollary 11.15. *If M is a signed graphic matroid, then it can be represented by a 2-regular subspace.*

Proof: By Theorem 10.16, M has a compact representation matrix B such that B is binet. By pivoting on B, it can be converted to an integral binet matrix B', as Lemma 5.14 claims. The matroid represented by B' is still M, so it satisfies the conditions of Theorem 11.14(b) for $k = 2$. A subspace that is dyadic of order 1 is also 2-regular, which completes the proof. □

This theorem implies that if a matroid is not represenatable by a 2-regular subspace, then it cannot be signed graphic. For example, Lee gave a matroid that is not representable by a 2-regular subspace, and minimal with respect to this property. In other words, he proved that this matroid

is minimal violator for representability by 2-regular subspaces. His matroid is the linear matroid represented by the compact representation matrix C:

$$C = \begin{bmatrix} -2 & 1 & 1 & 1 & 1 \\ -1 & 0 & 1 & 1 & 1 \\ -1 & 1 & 0 & 1 & 1 \\ -1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & -1 & 1 \end{bmatrix}$$

Chapter 12

Conclusions

In this last chapter we point out the main results obtained and suggest some directions for further research.

The main goal of this work was to examine how totally unimodular matrices can be generalized. To achieve this, we introduced two definitions, both extending a specific property of totally unimodular matrices. Total k-modularity allows more possible values for the subdeterminants, whereas k-regularity permits rational values in the inverses of submatrices. We showed that k-regularity is the proper generalization, as it has similar consequences on the integrality of polyhedra as total unimodularity has. Namely, k-regular matrices are exactly the rational constraint matrices for which a polyhedron with a right hand side vector that is an integer multiple of k has integral vertices. This extends the connection between totally unimodular matrices and integral polyhedra, expressed in Hoffman and Kruskal's theorem (Theorem 1.8), in two senses. First, it covers rational matrices, as opposed to integral ones in Theorem 1.8, and secondly it gives necessary and sufficient conditions on the integrality of polyhedra with special right hand sides.

We analysed the relation of totally k-modular and k-regular matrices, and proved that k-regularity implies total k-modularity if k is prime and the matrix is integral. On the other hand, if a k-regular matrix is not integral, then several basic operations on it, for example pivoting, do not preserve its k-regularity.

We gave two polyhedral characterizations of k-regular matrices. The first (Theorem 3.8), which holds for any rational matrix, is the extension of Hoffman and Kruskal's theorem for right hand sides that are integer multiples of k. The second (Theorem 3.21), true only for integral matrices, states that the rank-1 Chvátal-Gomory cuts can be replaced by mod-k cuts if and only if the the constraint matrix is k-regular.

Having introduced k-regularity for a general integer k, we continued with focusing on the most important case of $k = 2$. For 2-regularity we completed the generalization of total unimodularity by generalizing the most important class of totally unimodular matrices, i.e., the network matrices.

To this end, we defined binet matrices on bidirected graphs. We showed that binet matrices can be calculated by a graphical method, very similar to the one used to define network matrices. We investigated the matrix operations that maintain binetness, and proved that binet matrices, though not necessarily integral, have all the advantages of integral 2-regular matrices. For example, in Theorem 7.2 we stated that the polyhedron $\{x \mid a \leq Bx \leq b,\ l \leq x \leq u\}$ has half-integral vertices for any integral vectors l, u, a and b if B is a binet matrix. We also gave characterisations of binet matrices that are totally unimodular or network matrices. Moreover, we proved that integral binet matrices have strong Chvátal rank 1, extending by this the set of matrices known to have this property.

We showed that optimization with a binet constraint matrix can be done efficiently. For continuous optimization (which in case of integral right hand sides yields half-integral solutions, due to the 2-regularity of binet matrices), one can use the binet simplex method. We demonstrated that the existing and well-researched generalized network simplex method can be adapted to binet optimization problems. The integer optimization over binet matrices is equivalent to the matching problem, therefore it can also be dealt with very efficiently.

We pointed out the connection of our generalizations to other approaches. The main difference between our and these other methods to generalize total unimodularity is that we focus on matrices, not matroids or linear vectorspaces. Concerning matroids, we showed that binet matrices represent signed graphic matroids, in the same vein as network matrices represent graphic matroids. On the other hand, we related totally k-modular and k-regular matroids to k-adic vectorspaces, a different way to generalize total unimodularity. We argued that in some cases our approach can lead to stronger results.

We would like to mention two areas for further research. One is the recognition of binet or 2-regular matrices. We described some ideas towards this, but we believe that a combinatorial recognition algorithm is not impossible. Probably it would be similar to the recognition algorithm for network and totally unimodular matrices, originating from the decomposition theory of regular matroids, due to Seymour [58]. Just to mention some points which would need to be settled: Does the 2-sum or 3-sum of binet matrices result in another binet or 2-regular matrix? Is there any other well-defined class of 2-regular matrices that are not binet or transpose of binets? We gave an example of such a matrix in Section 6.2, but is it only a sporadic example or a member of a class? Is there any necessary and sufficient condition for binet or 2-regular matrices? The necessary conditions we gave in Section 5.3 for binet matrices are not sufficient. Is there any way to prove that a matrix is 2-regular other than to check the inverses of all submatrices or to show that it is binet or the transpose of a binet matrix?

The second possibility for further research is connected to half-integrality. We know that if the constraint matrix in an optimization problem is 2-regular and the right hand side vector is integral, then half-integral optimal solutions are guaranteed. There are problems in combinatorial optimization where half-integrality appears. Examples, besides those mentioned in Section 6.3, include the

network synthesis problem with unit cost (see Ford and Fulkerson [28]), the 2-commodity flow problem (see Hu [45]), the multiflow problem (Karzanov [46]), the multiway undirected cuts (Garg, Vazirani and Yannakakis [30]), the ring routing problem (Okamura and Seymour [52]), or the stable matching problem (Abeledo and Rothblum [1]). In some of these instances only the existence of a half-integral solution is claimed, not the half-integrality of all optimal solutions. Connected to this is the fact that the constraint matrix is not necessarily 2-regular in some of these problems. The question arising here is the following: Is the constraint matrix 2-regular in the examples where it has not been refuted yet? If yes, is it binet? If not, is there any underlying reason why these problems have half-integral solution? For example, the constraint matrix is not 2-regular, but the right hand side vector is such that it ensures half-integrality. This is connected to total dual half-integrality introduced in Section 3.3, so can we prove that the inequality systems in these or other problems are TDHI?

Bibliography

[1] H.G. Abeledo and U.G. Rothblum. Stable matchings and linear inequalities. *Discrete Applied Mathematics*, 54:1–27, 1994.

[2] R.K. Ahuja, Th.K. Magnanti, and J.B. Orlin. *Network Flows.* Prentice Hall, New Jersey, 1993.

[3] G. Appa. k-integrality, an extension of total unimodularity. *Operations Research Letters*, 13:159–163, 1993.

[4] J.C. Arantes, J.R. Birge, and K.G. Murty. Studies of lexicography in the generalized network simplex method. *Annals of Operations Research*, 47:237–248, 1993.

[5] M. Ball. Locating competitive new facilities in the presence of existing facilities. In *Proceedings of the 5th United States Postal Service Advanced Technology Conference*, pages 1169–1177, 1992.

[6] V.J.D. Baston, M.K. Rahmouni, and Williams H.P. A practical conversion of linear programmes to network flow models. *European Journal of Operational Research*, 50:325–334, 1991.

[7] C. Berge. Sur certains hypergraphes généralisant les graphes bipartis. In P. Erdős, A. Rényi, and V. T. Sós, editors, *Combinatorial Theory and Its Applications I*, Colloquia Mathematica Societatis János Bólyai 4, pages 119–133. North-Holland, Amsterdam, 1970.

[8] R.E. Bixby. On Reid's characterization of the ternary matroids. *Journal of Combinatorial Theory (B)*, 26:174–204, 1979.

[9] R.E. Bixby and W.H. Cunningham. Converting linear programs to network problems. *Mathematics of Operations Research*, 5:321–357, 1980.

[10] P. Camion. Characterization of totally unimodular matrices. *Proceedings of the American Mathematical Society*, 16:1068–1073, 1965.

[11] A. Caprara and M. Fischetti. $\{0, \frac{1}{2}\}$-Chvátal-Gomory cuts. *Mathematical Programming*, 74:221–235, 1996.

[12] A. Caprara, M. Fischetti, and A.N. Letchford. On the separation of maximally violated mod-k cuts. *Mathematical Programming (A)*, 87:37–56, 2000.

[13] R. Chandrasekaran. Total unimodularity of matrices. *SIAM Journal on Applied Mathematics*, 17:1032–1034, 1969.

[14] V. Chvátal. Edmonds polytopes and a hierarchy of combinatorial problems. *Discrete Mathematics*, 4:305–337, 1973.

[15] M. Conforti and G. Cornuéjols. Balanced $0, \pm 1$ matrices, bicoloring and total dual integrality. *Mathematical Programming*, 71:249–258, 1995.

[16] G. Cornuéjols. *Combinatorial Optimization: packing and covering*. Number 74 in CBMS-NSF regional conference series in applied mathematics. SIAM, 2001.

[17] W.H. Cunningham. A network simplex method. *Mathematical Programming*, 11:105–116, 1976.

[18] W.H. Cunningham. Theoretical properties of the network simplex method. *Mathematics of Operations Research*, 4:196–208, 1979.

[19] G.B. Dantzig. Application of the simplex method to a transportation problem. In Tj.C. Koopmans, editor, *Activity Analysis of Production and Allocation*, pages 339–347. Wiley, New York, 1951.

[20] G.B. Dantzig. *Linear programming and extensions*. Princeton University Press, Princeton, 1963.

[21] M. Doob. An interrelation between line graphs, eigenvalues, and matroids. *Journal of Combinatorial Theory (B)*, 15:40–50, 1973.

[22] J. Edmonds. Maximum matching and a polyhedron with 0,1-vertices. *Journal of Research of the National Bureau of Standards (B)*, 69:125–130, 1965.

[23] J. Edmonds. An introduction to matching. Mimeographed notes, Engineering Summer Conference, The University of Michigan, 1967.

[24] J. Edmonds and R. Giles. A min-max relation for submodular functions on graphs. *Annals of Discrete Mathematics*, 1:185–204, 1977.

[25] J. Edmonds and E.L. Johnson. Matching: a well-solved class of integer linear programs. In R.K. Guy et al., editors, *Combinatorial Structures and Their Applications*. Gordon and Breach, New York, 1970.

[26] J. Edmonds and E.L. Johnson. Matching: Euler tours and the Chinese postman. *Mathematical Programming*, 5:88–124, 1973.

[27] J. Elam, F. Glover, and D. Klingman. A strongly convergent primal simplex algorithm for generalized networks. *Mathematics of Operations Research*, 4(1):39–59, 1979.

[28] L. R. Ford, Jr. and D.R. Fulkerson. *Flows in networks*. Princeton University Press, Princeton, N.J., 1962.

[29] D.R. Fulkerson and O.A. Gross. Incidence matrices and interval graphs. *Pacific Journal of Mathematics*, 15:835–855, 1965.

[30] N. Garg, V.V. Vazirani, and Yannakakis M. Multiway cuts in directed and node weighted graphs. In *Proceedings of the 21th International Colloquium on Automata Languages and Programming*, pages 487–498, 1994.

[31] A.M.H. Gerards. Matching. In M.O. Ball et al., editors, *Handbooks in Operations Research and Management Science – Network Models*, volume 7. Elsevier Science, 1995.

[32] A.M.H. Gerards, L. Lovász, A. Schrijver, P.D. Seymour, and K. Truemper. Regular matroids from graphs. Unpublished manuscript, 1991.

[33] A.M.H. Gerards and A. Schrijver. Matrices with the Edmonds-Johnson property. *Combinatorica*, 6:365–379, 1986.

[34] A. Ghouila-Houri. Caractérisation des matrices totalement unimodulaires. *Comptes Rendus Hebdomadaires des Séances de l'Académie des Sciences (Paris)*, 254:1192–1194, 1962.

[35] F. Glover, D. Karney, and D. Klingman. Implementation and computational comparisons of primal, dual and primal-dual computer codes for minimum cost netrwok flow problem. *Networks*, 4:191–212, 1974.

[36] F. Glover, D. Klingman, and N.V. Phillips. *Network models in optimization and their applications in practice*. Wiley, New York, 1992.

[37] R.E. Gomory. Outline of an algorithm for integer solutions to linear programs. *Bulletin of the American Mathematical Society*, 64:275–278, 1958.

[38] F. Harary. On the notion of balance of a signed graph. *Michigan Mathematical Journal*, 2:143–146, 1953-1954.

[39] M. Held and R. M. Karp. The traveling-salesman problem and minimum spanning trees. *Operations Research*, 18:1138–1162, 1970.

[40] I. Heller and C.B. Tompkins. An extension of a theorem of Dantzig. In K. Kuhn and A. Tucker, editors, *Linear Inequalities and Related Systems*, pages 247–254. Princeton University Press, Princeton, NJ, 1956.

[41] B.S. Hochbaum. Solving integer programs over monotone inequalities in three variables: A framework for half integrality and good approximations. *European Journal of Operational Research*, 140(2):291–321, 2002.

[42] B.S. Hochbaum, N. Megiddo, J. Naor, and A. Tamir. Tight bounds and 2-approximation algorithms for integer programs with two variables per inequality. *Mathematical Programming*, 62:69–83, 1993.

[43] A. Hoffman. A generalization of max flow-min cut. *Mathematical Programming*, 6:352–359, 1974.

[44] A. Hoffman and J. Kruskal. Integral boundary points of convex polyhedra. In K. Kuhn and A. Tucker, editors, *Linear Inequalities and Related Systems*, pages 223–246. Princeton University Press, Princeton, NJ, 1956.

[45] T.C. Hu. *Integer programming and network flows.* Addison-Wesley, Reading, Mass., 1969.

[46] A.V. Karzanov. Multiflows and disjoint paths of minimum total cost. *Mathematical programming*, 78:219–242, 1997.

[47] E.L. Lawler. *Combinatorial Optimization: Networks and Matroids.* Holt, Rinehart & Winston, New York, 1976.

[48] J. Lee. Subspaces with well-scaled frames. *Linear Algebra and its Applications*, 114/115:21–56, 1989.

[49] J. Lee. The incidence structure of subspaces with well-scaled frames. *Journal of Combinatorial Theory (B)*, 50:265–287, 1990.

[50] K.G. Murty. *Network Programming.* Prentice Hall, New Jersey, 1992.

[51] G.L. Nemhauser and L.A. Wolsey. *Integer and Combinatorial Optimization.* Wiley, New York, 1988.

[52] H. Okamura and P. D. Seymour. Multicommodity flows in planar graphs. *Journal of Combinatorial Theory (B)*, 31:75–81, 1981.

[53] J.G. Oxley. *Matroid Theory.* Oxford University Press, Oxford, 1992.

[54] S.R. Pagano. *Separability and Representability of Bias Matroids of Signed Graphs.* PhD thesis, Binghamton University, 1998.

[55] A. Schrijver. *Theory of Linear and Integer Programming*. Wiley, Chichester, 1986.

[56] A. Schrijver. Disjoint homotopic paths and trees in a planar graph. *Discrete and Computational Geometry*, 6:527–574, 1991.

[57] P.D. Seymour. Matroid representation over $GF(3)$. *Journal of Combinatorial Theory (B)*, 26:159–173, 1979.

[58] P.D. Seymour. Decomposition of regular matroids. *Journal of Combinatorial Theory (B)*, 28:305–359, 1980.

[59] P.D. Seymour. Applications of the regular matroid theory decomposition. In L. Lovász and A. Recski, editors, *Matroid Theory (Colloquia Mathematica Societatis János Bolyai, 40)*, pages 345–357. North Holland, Amsterdam, 1985.

[60] D. Slilaty. Private communication, 2002.

[61] É. Tardos. A strongly polynomial minimum cost circulation algorithm. *Combinatorica*, 5:247–255, 1985.

[62] É. Tardos. A strongly polynomial algorithm to solve combinatorial linear programs. *Operations Research*, 34:250–256, 1986.

[63] K. Truemper. Alfa-balanced graphs and matrices and GF(3)-representability of matroids. *Journal of Combinatorial Theory (B)*, 32:112–139, 1982.

[64] K. Truemper. *Matroid decomposition*. Academic Press, San Diego, CA, 1992.

[65] W.T. Tutte. A homotopy theorem for matroids I, II. *Transactions of the American Mathematical Society*, 88:144–160, 161–174, 1958.

[66] W.T. Tutte. Lectures on matroids. *Journal of Research of the National Bureau of Standards (B)*, 69:1–47, 1965.

[67] A. Veinott and G. Dantzig. Integral extreme points. *SIAM Review*, 10:371–372, 1968.

[68] G. Whittle. A characterization of the matroids representable over $GF(3)$ and the rationals. *Journal of Combinatorial Theory (B)*, 65:222–261, 1995.

[69] Th Zaslavsky. Characterizations of signed graphs. *Journal of Graph Theory*, 5:401–406, 1981.

[70] Th. Zaslavsky. Signed graphs. *Discrete Applied Mathematics*, 4:47–74, 1982.

[71] Th. Zaslavsky. Biased graphs whose matroids are binary. *Journal of Combinatorial Theory (B)*, 42:337–347, 1987.

[72] Th. Zaslavsky. Biased graphs. I. Bias, balance, and gains. *Journal of Combinatorial Theory (B)*, 47:32–52, 1989.

[73] Th. Zaslavsky. Biased graphs. II. The three matroids. *Journal of Combinatorial Theory (B)*, 51:46–72, 1991.

[74] Th. Zaslavsky. A mathematical bibliography of signed and gain graphs and allied areas. *Electronic Journal of Combinatorics*, 1998. Dynamic Surveys in Combinatorics #DS8.

[75] Th. Zaslavsky. Glossary of signed and gain graphs and allied areas. *Electronic Journal of Combinatorics*, 1999. Dynamic Surveys in Combinatorics #DS9.