On Integer Programming and the Branch-Width of the Constraint Matrix

William H. Cunningham and Jim Geelen

Department of Combinatorics and Optimization
University of Waterloo
Waterloo, Canada N2L 3G1
{whcunnin,jfgeelen}@uwaterloo.ca
http://www.math.uwaterloo.ca/C_andO_Dept/index.shtml

Abstract. Consider an integer program $\max(c^t x : Ax = b,\ x \geq 0,\ x \in \mathbf{Z}^n)$ where $A \in \mathbf{Z}^{m\times n}$, $b \in \mathbf{Z}^m$, and $c \in \mathbf{Z}^n$. We show that the integer program can be solved in pseudo-polynomial time when A is non-negative and the column-matroid of A has constant branch-width.

1 Introduction

For positive integers m and n, let $A \in \mathbf{Z}^{m\times n}$, $b \in \mathbf{Z}^m$, and $c \in \mathbf{Z}^n$. Consider the following integer programming problems:

(IPF) Find $x \in \mathbf{Z}^n$ satisfying $(Ax = b,\ x \geq 0)$.
(IP) Find $x \in \mathbf{Z}^n$ maximizing $c^t x$ subject to $(Ax = b,\ x \geq 0)$.

Let $M(A)$ denote the column-matroid of A. We are interested in properties of $M(A)$ which lead to polynomial-time solvability for (IPF) and (IP). Note that, even when A (or, equivalently, $M(A)$) has rank one, the problems (IPF) and (IP) are NP-hard. Papadimitriou [9] considered these problems for instances where A has constant rank.

Theorem 1 (Papadimitriou). *There is a pseudopolynomial-time algorithm for solving (IP) on instances where the rank of A is constant.*

Robertson and Seymour [10] introduced the parameter "branch-width" for graphs and also, implicitly, for matroids. We postpone the definition until Section 2. Our main theorem is the following; a more precise result is given in Theorem 6.

Theorem 2. *There is a pseudopolynomial-time algorithm for solving (IP) on instances where A is non-negative and the branch-width of $M(A)$ is constant.*

The branch-width of a matroid M is at most $r(M)+1$. Theorem 2 does not imply Papadimitriou's theorem, since we require that A is non-negative. In Section 6 we show that the non-negativity can be dropped when we have bounds on the variables. However, the following result shows that we cannot just relax the non-negativity.

M. Fischetti and D.P. Williamson (Eds.): IPCO 2007, LNCS 4513, pp. 158–166, 2007.

Theorem 3. *(IPF) is NP-hard even for instances where $M(A)$ has branch-width ≤ 3 and the entries of A are in $\{0, \pm 1\}$.*

We also prove the following negative result.

Theorem 4. *(IPF) is NP-hard even for instances where the entries of A and b are in $\{0, \pm 1\}$ and $M(A)$ is the cycle matroid of a graph.*

We find Theorem 4 somewhat surprising considering the fact that graphic matroids are regular. Note that, if A is a $(0, \pm 1)$-matrix and $M([I, A])$ is regular, then A is a totally unimodular matix and, hence, we can solve (IP) efficiently. It seems artificial to append the identity to the constraint matrix here, but for inequality systems it is more natural.

Recall that $M(A)$ is regular if and only if it has no $U_{2,4}$-minor (see Tutte [13] or Oxley [8], Section 6.6). Moreover, Seymour [12] found a structural characterization of the class of regular matroids. We suspect that the class of $\mathbf{R}$-representable matroids with no $U_{2,l}$- or $U_{2,l}^*$-minor is also "highly structured" for all $l \geq 0$ (by which we mean that there is likely to be a reasonable analogue to the graph minors structure theorem; see [11]). Should such results ever be proved, one could imagine using the structure to solve the following problem.

Problem 1. Given a non-negative integer $l \geq 0$, is there a polynomial-time algorithm for solving $\max(c^t x \,:\, Ax \leq b,\, x \geq 0,\, x \in \mathbf{Z}^n)$ on instances where A is a $(0, \pm 1)$-matrix and $M([I, A])$ has no $U_{2,l}$- or $U_{2,l}^*$-minor?

2 Branch-Width

For a matroid M and $X \subseteq E(M)$, we let $\lambda_M(X) = r_M(X) + r_M(E(M) - X) - r(M) + 1$; we call λ_M the *connectivity function* of M. Note that the connectivity function is *symmetric* (that is, $\lambda_M(X) = \lambda_M(E(M) - X)$ for all $X \subseteq E(M)$) and *submodular* (that is, $\lambda_M(X) + \lambda_M(Y) \geq \lambda_M(X \cap Y) + \lambda_M(X \cup Y)$ for all $X, Y \subseteq E(M)$).

Let $A \in \mathbf{R}^{m \times n}$ and let $E = \{1, \ldots, n\}$. For $X \subseteq E$, we let

$$S(A, X) := \mathrm{span}(A|X) \cap \mathrm{span}(A|(E - X)),$$

where span(A) denotes the subspace of $\mathbf{R}^m$ spanned by the columns of A and $A|X$ denotes the restriction of A to the columns indexed by X. By the modularity of subspaces,

$$\dim S(A, X) = \lambda_{M(A)}(X) - 1.$$

A tree is *cubic* if its internal vertices all have degree 3. A *branch-decomposition* of M is a cubic tree T whose leaves are labelled by elements of $E(M)$ such that each element in $E(M)$ labels some leaf of T and each leaf of T receives at most one label from $E(M)$. The *width* of an edge e of T is defined to be $\lambda_M(X)$ where $X \subseteq E(M)$ is the set of labels of one of the components of $T - \{e\}$. (Since λ_M is symmetric, it does not matter which component we choose.) The *width* of T

is the maximum among the widths of its edges. The *branch-width* of M is the minimum among the widths of all branch-decompositions of M.

Branch-width can be defined more generally for any real-valued symmetric set-function. For graphs, the branch-width is defined using the function $\lambda_G(X)$; here, for each $X \subseteq E(G)$, $\lambda_G(X)$ denotes the number of vertices incident with both an edge in X and an edge in $E(G) - X$. The branch-width of a graph is within a constant factor of its tree-width. Tree-width is widely studied in theoretical computer science, since many NP-hard problems on graphs can be efficiently solved on graphs of constant tree-width (or, equivalently, branch-width). The most striking results in this direction were obtained by Courcelle [1]. These results have been extended to matroids representable over a finite field by Hliněný [4]. They do not extend to all matroids or even to matroids represented over the reals.

Finding Near-Optimal Branch-Decompositions

For any integer constant k, Oum and Seymour [7] can test, in polynomial time, whether or not a matroid M has branch-width k (assuming that the matroid is given by its rank-oracle). Moreover their algorithm *finds* an optimal branch-decomposition in the case that the branch-width is at most k. The algorithm is not practical; the complexity is $O(n^{8k+13})$. Fortunately, there is a more practical algorithm for finding a near-optimal branch-decomposition. For an integer constant k, Oum and Seymour [6] provide an $O(n^{3.5})$ algorithm that, for a matroid M with branch-width at most k, finds a branch-decomposition of width at most $3k - 1$. The branch decomposition is obtained by solving $O(n)$ matroid intersection problems. When M is represented by a matrix $A \in \mathbf{Z}^{m \times n}$, each of these matroid intersection problems can be solved in $O(m^2 n \log m)$ time; see [2]. Hence we can find a near-optimal branch-decomposition for $M(A)$ in $O(m^2 n^2 \log m)$ time.

3 Linear Algebra and Branch-Width

In this section we discuss how to use branch decompositions to perform certain matrix operations more efficiently. This is of relatively minor significance, but it does improve the efficiency of our algorithms.

Let $A \in \mathbf{Z}^{m \times n}$ and let $E = \{1, \ldots, n\}$. Recall that, for $X \subseteq E$, $S(A, X) = \text{span}(A|X) \cap \text{span}(A|(E - X))$ and that $\dim S(A, X) = \lambda_{M(A)}(X) - 1$. Now let T be a branch-decomposition of $M(A)$ of width k, let e be an edge of T, and let X be the label-set of one of the two components of $T - e$. We let $S_e(A) := S(A, X)$. The aim of this section is to find bases for each of the subspaces $(S_e(A) : e \in E(T))$ in $O(km^2 n)$ time.

Converting to Standard Form

Let $B \subseteq E$ be a basis of $M(A)$. Now let $A_B = A|B$ and $A' = (A_B)^{-1} A$. Therefore $M(A) = M(A')$ and $S_e(A) = \{A_B v : v \in S_e(A')\}$. Note that we can find B

and A' in $O(m^2n)$ time. Given a basis for $S_e(A')$, we can determine a basis for $S_e(A)$ in $O(km^2)$ time. Since T has $O(n)$ edges, if we are given bases for each of $(S_e(A') : e \in E(T))$ we can find bases for each of $(S_e(A) : e \in E(T))$ in $O(km^2n)$ time.

Matrices in Standard Form

Henceforth we suppose that A is already in standard form; that is $A|B = I$ for some basis B of $M(A)$. We will now show the stronger result that we can find a basis for each of the subspaces $(S_e(A) : e \in E(T))$ in $O(k^2mn)$ time (note that $k \leq m+1$).

We label the columns of A by the elements of B so that the identity $A|B$ is labelled symmetrically. For $X \subseteq B$ and $Y \subseteq E$, we let $A[X, Y]$ denote the submatrix of A with rows indexed by X and columns indexed by Y.

Claim. *For any partition (X, Y) of E,*

$$\lambda_{M(A)}(X) = rank\ A[X \cap B, X - B] + rank\ A[Y \cap B, Y - B] + 1.$$

Moreover $S(A, X)$ is the column-span of the matrix

$$\begin{array}{c} \\ X \cap B \\ Y \cap B \end{array} \begin{array}{c} \begin{array}{cc} X - B & \qquad\qquad Y - B \end{array} \\ \begin{pmatrix} A[X \cap B, X - B] & 0 \\ 0 & A[Y \cap B, Y - B] \end{pmatrix}. \end{array}$$

Proof. The formula for $\lambda_{M(A)}(X)$ is straightforward and well known. It follows that $S(A, X)$ has the same dimension as the column-space of the given matrix. Finally, it is straightforward to check that each column of the given matrix is spanned by both $A|X$ and $A|(E - X)$.

Let (X, Y) be a partition of E. Note that $B \cap X$ can be extended to a maximal independent subset B_X of X and $B \cap Y$ can be extended to a maximal independent subset B_Y of Y. Now $S(A, X) = S(A|(B_X \cup B_y), B_X)$. Then, by the claim above, given B_X and B_Y we can trivially find a basis for $S(A, X)$.

Finding Bases

A set $X \subseteq E$ is called T-*branched* if there exists an edge e of T such that X is the label-set for one of the components of $T - e$. For each T-branched set X we want to find a maximal independent subset $B(X)$ of X containing $X \cap B$. The number of T-branched sets is $O(n)$, and we will consider them in order of non-decreasing size. If $|X| = 1$, then we can find $B(X)$ in $O(m)$ time. Suppose then that $|X| \geq 2$. Then there is a partition (X_1, X_2) of X into two smaller T-branched sets. We have already found $B(X_1)$ and $B(X_2)$. Note that X is spanned by $B(X_1) \cup B(X_2)$. Moreover, for any T-branched set Y, we have $r_{M(A)}(Y) - |Y \cap B| \leq r_{M(A)}(Y) + r_{M(A)}(E - Y) - r(M(A)) = \lambda_{M(A)}(Y) - 1$. Therefore $|(B(X_1) \cup B(X_2)) - (B \cap X)| \leq 2(k-1)$. Recall that $A|B = I$. Then in $O(k^2m)$ time ($O(k)$ pivots on an $m \times k$-matrix) we can extend $B \cap X$ to a basis $B(X) \subseteq B(X_1) \cup B(X_2)$. Thus we can find all of the required bases in $O(k^2mn)$ time.

4 The Main Result

In this section we prove Theorem 2. We begin by considering the feasibility version.

IPF(k).
INSTANCE: Positive integers m and n, a non-negative matrix $A \in \mathbf{Z}^{m\times n}$, a non-negative vector $b \in \mathbf{Z}^m$, and a branch-decomposition T of $M(A)$ of width k.
PROBLEM: Does there exist $x \in \mathbf{Z}^n$ satisfying $(Ax = b,\ x \geq 0)$?

Theorem 5. *IPF(k) can be solved in* $O((d+1)^{2k}mn + m^2n)$ *time, where* $d = \max(b_1, \ldots, b_m)$.

Note that for many combinatorial problems (like the set partition problem), we have $d = 1$. For such problems the algorithm requires only $O(m^2n)$ time (considering k as a constant). Recall that $S(A, X)$ denotes the subspace $\text{span}(A|X) \cap \text{span}(A|(E - X))$, where E is the set of column-indices of A.

The following lemma is the key.

Lemma 1. *Let* $A \in \{0, \ldots, d\}^{m\times n}$ *and let* $X \subseteq \{1, \ldots, n\}$ *such that* $\lambda_{M(A)}(X) = k$*. Then there are at most* $(d+1)^{k-1}$ *vectors in* $S(A, X) \cap \{0, \ldots, d\}^m$.

Proof. Since $\lambda_{M(A)}(X) \leq k$, $S(A, X)$ has dimension $k-1$; let $a_1, \ldots, a_{k-1} \in \mathbf{R}^m$ span $S(A, X)$. There is a $(k-1)$-element set $Z \subseteq \{1, \ldots, n\}$ such that the matrix $(a_1|Z, \ldots, a_{k-1}|Z)$ is non-singular. Now any vector $x \in \mathbf{R}$ that is spanned by $(a_1, \ldots, a_{k-1})$ is uniquely determined by $x|Z$. So there are at most $(d+1)^{k-1}$ vectors in $\{0, \ldots, d\}^m$ that are spanned by $(a_1, \ldots, a_{k-1})$.

Proof (Proof of Theorem 5.). Let $A' = [A, b]$, $E = \{1, \ldots, n\}$, and $E' = \{1, \ldots, n+1\}$. Now, let T be a branch-decomposition of $M(A)$ of width k and let T' be a branch-decomposition of $M(A')$ obtained from T by subdividing an edge and adding a new leaf-vertex, labelled by $n+1$, adjacent to the degree 2 node. Note that T' has width $\leq k+1$. Recall that a set $X \subseteq E$ is T*-branched* if there is an edge e of T such that X is the label-set of one of the components of $T - e$. By the results in the previous section, in $O(m^2n)$ time we can find bases for each subspace $S(A', X)$ where $X \subseteq E$ is T'-branched.

For $X \subseteq E$, we let $\mathcal{B}(X)$ denote the set of all vectors $b' \in \mathbf{Z}^m$ such that

(1) $0 \leq b' \leq b$,
(2) there exists $z \in \mathbf{Z}^X$ with $z \geq 0$ such that $(A|X)z = b'$, and
(3) $b' \in \text{span}(A'|(E' - X))$.

Note that, if $b' \in \mathcal{B}(X)$, then, by (2) and (3), $b' \in S(A', X)$. If $\lambda_{M(A')}(X) \leq k+1$, then, by Lemma 1, $|\mathcal{B}(X)| \leq (d+1)^k$. Moreover, we have a solution to the problem (IPF) if and only $b \in \mathcal{B}(E)$.

We will compute $\mathcal{B}(X)$ for all T'-branched sets $X \subseteq E$ using dynamic programming. The number of T'-branched subsets of E is $O(n)$, and we will consider them in order of non-decreasing size. If $|X| = 1$, then we can easily find $\mathcal{B}(X)$ in $O(dm)$ time. Suppose then that $|X| \geq 2$. Then there is a partition (X_1, X_2) of X into two smaller T'-branched sets. We have already found $\mathcal{B}(X_1)$ and $\mathcal{B}(X_2)$. Note that $b' \in \mathcal{B}(X)$ if and only if

(a) there exist $b'_1 \in \mathcal{B}(X_1)$ and $b'_2 \in \mathcal{B}(X_2)$ such that $b' = b'_1 + b'_2$,
(b) $b' \leq b$, and
(c) $b' \in S(A', X)$.

The number of choices for b' generated by (a) is $O((d+1)^{2k})$. For each such b' we need to check that $b' \leq b$ and $b' \in S(A', X)$. Since we have a basis for $S(A', X)$ and since $S(A', X)$ has dimension $\leq k$, we can check whether or not $b' \in S(A', X)$ in $O(m)$ time (considering k as a constant). Therefore we can find $\mathcal{B}(E)$ in $O((d+1)^{2k}mn + m^2n)$ time.

We now return to the optimization version.

IP(k).
INSTANCE: Positive integers m and n, a non-negative matrix $A \in \mathbf{Z}^{m\times n}$, a non-negative vector $b \in \mathbf{Z}^m$, a vector $c \in \mathbf{Z}^n$, and a branch-decomposition T of $M(A)$ of width k.
PROBLEM: Find $x \in \mathbf{Z}^n$ maximizing $c^t x$ subject to $(Ax = b,\ x \geq 0)$.

Theorem 6. *IP(k) can be solved in $O((d+1)^{2k}mn + m^2n)$ time, where $d = \max(b_1, \ldots, b_m)$.*

Proof. The proof is essentially the same as the proof of Theorem 5, except that for each $b' \in \mathcal{B}(X)$ we keep a vector $x \in \mathbf{Z}^X$ maximizing $\sum(c_i x_i : i \in X)$ subject to $((A|X_e)x = b',\ x \geq 0)$. The details are easy and left to the reader.

Theorem 6 implies Theorem 2.

5 Hardness Results

In this section we prove Theorems 3 and 4. We begin with Theorem 3. The reduction is from the following problem, which is known to be NP-hard; see Lueker [5].

Single Constraint Integer Programming Feasibility (SCIPF).
INSTANCE: A non-negative vector $a \in \mathbf{Z}^n$ and an integer b.
PROBLEM: Does there exist $x \in \mathbf{Z}^n$ satisfying $(a^t x = b,\ x \geq 0)$?

Proof (Proof of Theorem 3.). Consider an instance (a, b) of (SCIPF). Choose an integer k as small as possible subject to $2^{k+1} > \max(a_1, \ldots, a_n)$. For each $i \in \{1, \ldots, n\}$, let $(\alpha_{i,k}, \alpha_{i,k-1}, \ldots, \alpha_{i,0})$ be the binary expansion of a_i. Now consider the following system of equations and inequalities:

(1) $$\sum_{i=1}^{n}\sum_{j=0}^{k} \alpha_{ij} y_{ij} = b.$$

(2) $y_{ij} - x_i - \sum_{l=0}^{i-1} y_{i,l} = 0$, for $i \in \{1, \ldots, n\}$ and $j \in \{0, \ldots, k\}$.

(3) $x_i \geq 0$ for each $i \in \{1, \ldots, n\}$.

If $(y_{ij} \ :\in \{1,\ldots,n\},\ j \in \{0,\ldots,k\})$ and $(x_1,\ldots,x_n)$ satisfy (2), then $y_{ij} = 2^j x_i$, and (1) simplifies to $\sum(a_i x_i \ : \ i \in \{1,\ldots,n\}) = b$. Therefore there is an integer solution to (1), (2), and (3) if and only if there is an integer solution to $(a^t x = b,\ x \geq 0)$.

The constraint matrix B for system (2) is block diagonal, where each block is a copy of the matrix:

$$C = \begin{array}{c} \\ 1 \\ 2 \\ \vdots \\ k+1 \end{array} \begin{array}{c} \begin{array}{cccccc} 1 & 2 & 3 & \ldots & k+1 & k+2 \end{array} \\ \begin{pmatrix} 1 & -1 & -1 & \cdots & -1 & -1 \\ 0 & 1 & -1 & & -1 & -1 \\ & & \ddots & \ddots & & \\ 0 & 0 & 0 & \cdots & 1 & -1 \end{pmatrix} \end{array}.$$

It is straightforward to verify that $M(C)$ is a circuit and, hence, $M(C)$ has branch-width 2. Now $M(B)$ is the direct sum of copies of $M(C)$ and, hence, $M(B)$ has branch-width 2. Appending a single row to B can increase the branch-width by at most one.

Now we turn to Theorem 4. Our proof is by a reduction from 3D Matching which is known to be NP-complete; see Garey and Johnson [3], pp. 46.

3D Matching.
INSTANCE: Three disjoint sets X, Y, and Z with $|X| = |Y| = |Z|$ and a collection $\mathcal{F}$ of triples $\{x, y, z\}$ where $x \in X$, $y \in Y$, and $z \in Z$.
PROBLEM: Does there exist a partition of $X \cup Y \cup Z$ into triples, each of which is contained in $\mathcal{F}$?

Proof (Proof of Theorem 4.). Consider an instance $(X, Y, Z, \mathcal{F})$ of 3D Matching. For each triple $t \in \mathcal{F}$ we define elements u_t and v_t. Now construct a graph $G = (V, E)$ with

$$V = X \cup Y \cup Z \cup \{u_t \ : \ t \in \mathcal{F}\} \cup \{v_t \ : \ t \in \mathcal{F}\}, \text{ and}$$

$$E = \bigcup_{t=\{x,y,z\}\in\mathcal{F}} \{(u_t, x),\ (u_t, y),\ (u_t, v_t),\ (v_t, z)\}.$$

Note that G is bipartite with bipartition $(X \cup Y \cup \{v_t \ : \ t \in \mathcal{F}\},\ Z \cup \{u_t \ : \ t \in \mathcal{F}\})$.

Now we define $b \in \mathbf{Z}^V$ such that $b_{u_t} = 2$ for each $t \in \mathcal{F}$ and $b_w = 1$ for all other vertices w of G. Finally, we define a matrix $A = (a_{ve}) \in \mathbf{Z}^{V \times E}$ such that $a_{ve} = 0$ whenever v is not incident with e, $a_{ve} = 2$ whenever $v = u_t$ and $e = (u_t, v_t)$ for some $t \in \mathcal{F}$, and $a_{ve} = 1$ otherwise; see Figure 1.

It is straightforward to verify that $(X, Y, Z, \mathcal{F})$ is a YES-instance of the 3D Matching problem if and only if there exists $x \in \mathbf{Z}^E$ satisfying $(Ax = b,\ x \geq 0)$. Now A and b are not $(0, \pm 1)$-valued, but if, for each $t \in \mathcal{F}$, we subtract the v_t-row from the u_t-row, then the entries in the resulting system $A'x = b'$ are in $\{0, \pm 1\}$.

It remains to verify that $M(A)$ is graphic. It is straightforward to verify that A is equivalent, up to row and column scaling, to a $\{0, 1\}$-matrix A''. Since G

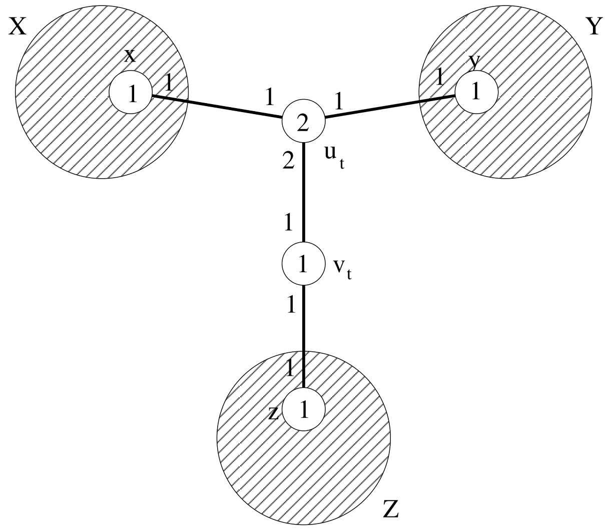

Fig. 1. The reduction

is bipartite, we can scale some of the rows of A'' by -1 to obtain a matrix B with a 1 and a -1 in each column. Now $M(B) = M(A)$ is the cycle-matroid of G and, hence, $M(A)$ is graphic.

6 Bounded Variables

In this section we consider integer programs with bounds on the variables.

Integer Programming with Variable Bounds (BIP)
INSTANCE: Positive integers m and n, a matrix $A \in \mathbf{Z}^{m\times n}$, a vector $b \in \mathbf{Z}^m$, and vectors $c, d \in \mathbf{Z}^n$.
PROBLEM: Find $x \in \mathbf{Z}^n$ maximizing $c^t x$ subject to ($Ax = b$, $0 \le x \le d$).

We can rewrite the problem as: *Find* $y \in \mathbf{Z}^{2n}$ *maximizing* $\hat{c}^t y$ *subject to* ($\hat{A}y = \hat{b}$, $y \ge 0$), where

$$\hat{A} = \begin{bmatrix} A & 0 \\ I & I \end{bmatrix},\ \hat{b} = \begin{bmatrix} b \\ d \end{bmatrix},\ \text{and } \hat{c} = \begin{bmatrix} c \\ 0 \end{bmatrix}.$$

Note that, for $i \in \{1, \ldots, n\}$, the elements i and $i+n$ are in series in $M(\hat{A})$, and, hence, $M(\hat{A})$ is obtained from $M(A)$ by a sequence of series-coextensions. Then it is easy to see that, if the branch-width of $M(A)$ is k, then the branch-width of $M(\hat{A})$ is at most $\max(k, 2)$.

Now note that the all-ones vector is in the row-space of $\hat{A}$. Therefore, by taking appropriate combinations of the equations $\hat{A}y = \hat{b}$, we can make an equivalent system $\tilde{A}y = \tilde{b}$ where $\tilde{A}$ is non-negative. Therefore, we obtain the following corollary to Theorem 2.

Corollary 1. *There is a pseudopolynomial-time algorithm for solving (BIP) on instances where the branch-width of $M(A)$ is constant.*

Acknowledgements

We thank Bert Gerards and Geoff Whittle for helpful discussions regarding the formulation of Problem 1 and the proof of Theorem 4. This research was partially sponsored by grants from the Natural Science and Engineering Research Council of Canada.

References

1. B. Courcelle, "Graph rewriting: An algebraic and logical approach", in: *Handbook of Theoretical Computer Science, vol. B*, J. van Leeuwnen, ed., North Holland (1990), Chapter 5.
2. W.H. Cunningham, *Improved bounds for matroid partition and intersection algorithms*, SIAM J. Comput. **15** (1986), 948-957.
3. M.R. Garey and D.S. Johnson, *Computers and Intractability. A guide to the theory of NP-completeness*, A series of books in the mathematical sciences, W.H. Freeman and Co., San Francisco, California, 1979.
4. P. Hliněný, Branch-width, parse trees and monadic second-order logic for matroids, manuscript, 2002.
5. G.S. Lueker, *Two NP-complete problems in non-negative integer programming*, Report No. 178, Department of Computer Science, Princeton University, Princeton, N.J., (1975).
6. S. Oum and P. D. Seymour, *Approximating clique-width and branch-width*, J. Combin. Theory, Ser. B **96** (2006), 514-528.
7. S. Oum and P. D. Seymour, *Testing branch-width*, to appear in J. Combin. Theory, Ser. B.
8. J. G. Oxley, *Matroid Theory*, Oxford University Press, New York, 1992.
9. C.H. Papadimitriou, *On the complexity of integer programming*, J. Assoc. Comput. Mach. **28** (1981), 765-768.
10. N. Robertson and P. D. Seymour, *Graph Minors. X. Obstructions to tree-decomposition*, J. Combin. Theory, Ser. B **52** (1991), 153–190.
11. N. Robertson and P. D. Seymour, *Graph Minors. XVI. Excluding a non-planar graph*, J. Combin. Theory, Ser. B **89** (2003), 43-76.
12. P. D. Seymour, *Decomposition of regular matroids,* J. Combin. Theory, Ser. B **28** (1980), 305–359.
13. W. T. Tutte, *A homotopy theorem for matroids, I, II,* Trans. Amer. Math. Soc. **88** (1958), 144–174.