A Primal Analogue of Cutting Plane Algorithms

Robert T. Firla* Bianca Spille* Robert Weismantel*

Institute of Mathematical Optimization
Department of Mathematics
Otto-von-Guericke-University Magdeburg
Universitätsplatz 2
D-39106 Magdeburg
Germany
[firla,spille,weismantel]@imo.math.uni-magdeburg.de

September 15, 1999

Abstract

This paper deals with algorithmic issues related to the design of an augmentation algorithm for general integer programs. It is shown that every phase of a primal method has a natural analogue in cutting plane algorithms. In particular, the role that the Chvátal-Gomory cuts play in cutting plane algorithms is taken on by so-called Gomory-Young augmentation vectors. The latter family of vectors can naturally be derived from a simplex tableau.

There is also a primal counterpart of families of combinatorial cutting planes. To demonstrate this we introduce an augmentation network for the intersection of any two integer programs. Paths and cycles in this network correspond to candidates for improving feasible solutions. This network gives rise to an algorithmic characterization of the weighted bipartite $\mathfrak{b}$-matching problem.

Key words: primal algorithm, local search, augmentation network, bipartite $\mathfrak{b}$-matching, Gomory-Young vectors

*The work of the three authors is supported by a "Gerhard-Hess-Forschungsförderpreis" (WE 1462/2-2) of the German Science Foundation (DFG) awarded to R. Weismantel.

The third author is also supported by the "Kultusministerium Sachsen-Anhalt" and an EC DONET Project.

1 A Primal versus a Dual Approach

This paper deals with algorithmic approaches to solve integer programs of the following form

$$\max c^T x : x \in \mathcal{F} := \{Ax \leq b, x \in \mathbb{Z}^n_+\}, \tag{1}$$

where $A \in \mathbb{Z}^{m \times n}$, $b \in \mathbb{Z}^m$, and $c \in \mathbb{Z}^n$.

One common approach to solve such an optimization problem (OPT) is the *dual approach* that is based on a separation strategy. From the point of view of computational complexity this approach relies on the polynomial time equivalence of the optimization problem (OPT) and the separation problem (SEP), see [GLS88].

> **Separation Problem** (SEP)
> Given a convex set K and a vector y, decide whether $y \in K$ and if not, return a hyperplane separating y from K.

Adapted to an integer program in the form (1), a dual algorithm works in principle via the following scheme.

Let $y \in \mathbb{R}^n$ be an optimal solution of the LP-relaxation

$$\max c^T x : Ax \leq b,\ x \in \mathbb{R}^n_+.$$

If y is integral, then y is the integer optimum of (1) and we are done. Otherwise, a cut $\{x \in \mathbb{R}^n : d^T x \leq \delta\}$ must be determined that is valid for all $x \in \mathcal{F}$ and separates the current point y. It is one of the nice features of a dual algorithm that there is a small certificate for proving that a point is optimal.

Proposition 1 (Dual Optimality Test)
Given $k \leq n$ valid inequalities $\{x : f^i x \leq g_i\}$ with $(1 \leq i \leq k)$ of $P := \operatorname{conv} \mathcal{F}$. If $x^* \in \{x \in P : f^i x = g_i, 1 \leq i \leq k\}$ and $c \in \operatorname{cone}\{f^1, \ldots, f^k\}$, then x^* is c-optimal.

One of the prospective advantages of a primal algorithm is the possibility of stopping the calculation at any time and still being able to deliver a feasible solution. From an algorithmic point of view the *primal approach* to solve problem (1) relies on a study of the augmentation problem.

> **Augmentation Problem** (AUG)
> Given vectors $c \in \mathbb{Z}^n$ and $x \in \mathcal{F}$, return an *augmentation vector* $t \in \mathbb{Z}^n$ such that $c^T t > 0$ and $x + t \in \mathcal{F}$ or assert that no such vector t exists.

It was shown in [GL95] and [SWZ95] that 0/1-linear optimization problems can be solved by a polynomial number of calls of the oracle (AUG). Indeed, for 0/1-problems (OPT) and (AUG) are polynomial time equivalent. For general integer programs, one needs access to a directed augmentation problem (DIR-AUG) in order to solve the optimization problem (OPT) within a polynomial number of steps, see [SW99].

Thus, in order to solve the optimization problem (1) with a primal approach, one starts with a feasible solution and then proceeds through a sequence of augmentations to obtain the IP-optimum. Like in the case of dual algorithms there is also for the primal algorithm a small certificate for proving that a point is optimal.

Proposition 2 (Primal Optimality Test)
*Given l vectors $h^1, \ldots, h^l$ such that $\mathcal{F} - x^* \subseteq \operatorname{cone}\{h^1, \ldots, h^l\}$.
If $c^T h^i \leq 0$ for $1 \leq i \leq l$, then x^* is c-optimal.*

Proof. For any $x \in \mathcal{F}$, we have that $x - x^* = \sum_{i=1}^{l} \lambda_i h^i$ with $\lambda_1, \ldots, \lambda_l \geq 0$. Therefore, $c^T(x - x^*) = \sum_{i=1}^{l} \lambda_i (c^T h^i) \leq 0$, which implies $c^T x \leq c^T x^*$. This verifies the optimality of x^*. □

We have seen so far that there is an analogy between primal and dual algorithms w.r.t. oracles, complexity issues and optimality certificates. This analogy becomes even more striking when we analyze augmentation and separation strategies in further detail. Namely from a high level prospective there are two important viewpoints to attack the separation problem, one is the algebraic one and the second one a combinatorial one. We will show in this paper that both viewpoints lead to a natural analogue on the primal side.

More precisely, the analogue of families of combinatorial cutting planes are combinatorial augmentation vectors. The design of such a family is based on the combinatorial structure of the problem under investigation. We define combinatorial augmentation vectors in Section 2 by introducing an augmentation network. Furthermore, we show that this network supplies an algorithmic characterization of the bipartite $\mathfrak{b}$-matching problem.

The role that the Chvátal-Gomory cuts play in the dual approach is taken on by the family of so-called Gomory-Young augmentation vectors. Their definition and properties are discussed in Section 3.

In Section 4 we explain how combinatorial augmentation vectors can be added to a given system of equations in a way such that in the subsequent step we again have the choice of performing an augmentation step either with a Gomory-Young vector or with a combinatorial augmentation vector.

2 An Augmentation Network and the Bipartite $\mathfrak{b}$-Matching Problem

Effective cutting plane algorithms for combinatorial optimization problems rely on a study of the polyhedron defined as the convex hull of all feasible solutions to the problem. If this problem is NP-hard, then one cannot expect to have knowledge about all inequalities needed to describe the associated polyhedron. This motivates the search for a partial description.

It is the objective of such an effort to find a family of inequalities of exponential size that define deep cuts for the polyhedron and to simultaneously design a polynomial time algorithm for solving the separation problem restricted to this family. Examples of this kind are the odd cycle-inequalities for the stable set problem [P73], [GLS88] or the 2-matching constraints for the TSP [PR82]. Such a design of a family of cuts and a corresponding local separation algorithm is usually problem dependent and exploits the combinatorial structure of the polyhedron under investigation.

In this section we study the primal analogue of this question. It will be shown that for the intersection of two 0/1-integer programs one can design a family of combinatorial augmentation vectors of exponential size and an algorithm that augments over this family in polynomial time. The vectors that we investigate are combinatorial in the sense that they can be interpreted as paths and cycles in a network. Similar ideas have been used earlier for designing augmentation algorithms for the matroid intersection problem (see [E70], [E79], [L76], [CCPS98], or [FSW99]).

To be more precise, let us consider the combinatorial optimization problem of intersecting two 0/1-integer programs,

$$\max c^T x : Ax \le b,\ Cx \le d,\ x \in \{0,1\}^n, \tag{IIP}$$

where $A \in \mathbb{Z}_+^{m_1 \times n}$, $C \in \mathbb{Z}_+^{m_2 \times n}$, $b \in \mathbb{Z}_+^{m_1}$, $d \in \mathbb{Z}_+^{m_2}$, and $c \in \mathbb{Z}^n$.

Depending on a feasible solution x of the problem we define an augmentation network $\mathcal{N}(x)$ as follows:
Let $J := \operatorname{supp}(x) := \{i \in \{1,\dots,n\} : x_i \ne 0\}$. The network $\mathcal{N}(x)$ (in the following also denoted by $\mathcal{N}(J)$) has vertex-set $\{1,\dots,n\} \cup \{r,s\}$ and two types of arcs that reflect conditions on the matrices A and C. The first type of arcs is called *A-arcs* and is defined as follows.

- (r,i) for every $i \notin J$ with $A(x+e^i) \le b$;
- (j,i) for every $j \in J$, $i \notin J$ with $A(x+e^i) \not\le b$ and $Ae^i \le Ae^j$;
- (j,s) for every $j \in J$;

The conditions below characterize the set of all *C-arcs* of the network.

- (r,j) for every $j \in J$;
- (i,j) for every $j \in J$, $i \notin J$ with $C(x+e^i) \not\le d$ and $Ce^i \le Ce^j$;
- (i,s) for every $i \notin J$ with $C(x+e^i) \le d$,

A-arcs are illustrated in Figure 1 by solid arcs and C-arcs by dashed arcs.

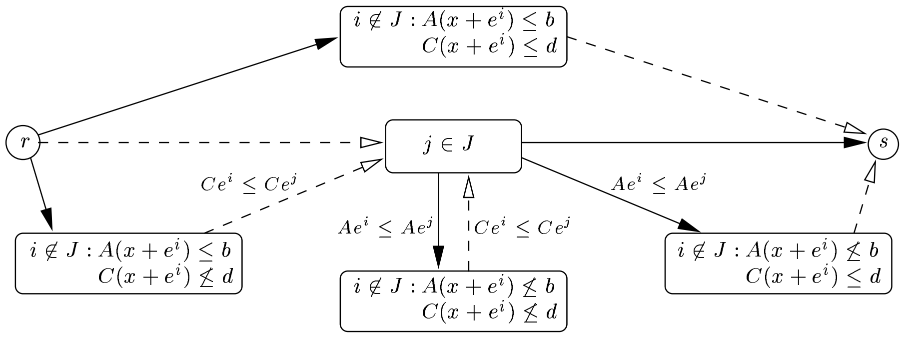

Figure 1: The augmentation network $\mathcal{N}(x)$ for the intersection of two 0/1-programs with $J = \mathrm{supp}(x)$.

Finally we assign weights to the vertices as follows:
$w(j) := c(j)$ for $j \in J$, $w(i) := -c(i)$ for $i \notin J$ and $w(r) := w(s) := 0$.

Note that we associate a network $\mathcal{N}(x)$ with every feasible solution x. Augmentation networks for the matroid intersection problem are defined for particular feasible solutions only, see [CCPS98], [L76]. Secondly, the arcs in the network $\mathcal{N}(x)$, except the arcs (r, j) and (j, s) with $j \in J$, are in general a strict subset of the arcs in the networks used for the matroid intersection algorithm.

For an easier notation we denote in the following an r-s-path (r, P, s) by P, i.e., P consists of the inner vertices of such a sequence.

It is not too difficult to see that every r-s-path or cycle in $\mathcal{N}(x)$ corresponds to a vector t such that $x + t$ is feasible. If the weight of this r-s-path or cycle is negative (the weight of a path or cycle is given by the weights of its vertices), then this yields an improving direction (see Lemma 3). One may apply a shortest path algorithm to detect a path or cycle of negative weight. Therefore, there is a polynomial time algorithm for checking whether a given solution x can be improved by an augmentation vector corresponding to an r-s-path or cycle in the augmentation network $\mathcal{N}(x)$.

Lemma 3 *Every negative r-s-path or cycle T in the network $\mathcal{N}(x)$ corresponds to an augmentation vector for x, i.e.,*

$$y := \chi^{(J \triangle T)} = x + \Big(\sum_{i \in T \setminus J} e^i - \sum_{j \in T \cap J} e^j \Big)$$

is feasible for (IIP) *and has better objective value than x.*

Proof. Let T be an r-s-path or cycle of the form $T = (i_0, j_1, i_1, \ldots, j_k, i_k)$, where i_0 and i_k are optionally depending on the structure of the path or cycle.

Let $y := \chi^{(J \triangle T)}$, then we have

$$Ay = \underbrace{A(x + e^{i_0})}_{\leq b} + \sum_{l=1}^{k-1} \underbrace{(Ae^{i_l} - Ae^{j_l})}_{\leq 0} + \underbrace{(Ae^{i_k} - Ae^{j_k})}_{\leq 0} \leq b.$$

Analogously we get $Cy = (Ce^{i_0} - Ce^{j_1}) + \sum_{l=2}^{k}(Ce^{i_{l-1}} - Ce^{j_l}) + C(x + e^{i_k}) \leq d$ and therefore, y is feasible. If T is negative then $c^T y = c^T x - w(T) > c^T x$. $\square$

We illustrate the network on an example.

Example 4 Consider the intersection of two 0/1-programs,

$$\begin{array}{rcl}
\max \begin{pmatrix} 2 & 1 & 9 & 8 & 4 & 5 & 4 & 6 \end{pmatrix} x & & \\
\begin{pmatrix} 1 & 1 & & & & & & \\ & & 1 & 1 & 1 & 1 & & \\ & & & & & & 1 & 1 \end{pmatrix} x & \leq & \begin{pmatrix} 1 \\ 1 \\ 2 \end{pmatrix} \\
\begin{pmatrix} 3 & 1 & 4 & 2 & 5 & 4 & 4 & 5 \end{pmatrix} x & \leq & 13 \\
x & \in & \{0,1\}^8.
\end{array}$$

The first 3×8-matrix represents the system $Ax \leq b$. The system $Cx \leq d$ corresponds in our example to the second 1×8-matrix. We start with the feasible solution $x := e^1 + e^3 + e^8$ that attains an objective function value of 17. It may be checked that x cannot be improved by a two exchange, i.e., a vector of the form $(e^i - e^j)$ for $i \in \{2, 4, 5, 6, 7\}$ and $j \in \{1, 3, 8\}$.

The augmentation network $\mathcal{N}(x)$ is depicted in the figure below (the w-weights are denoted beside the vertices).

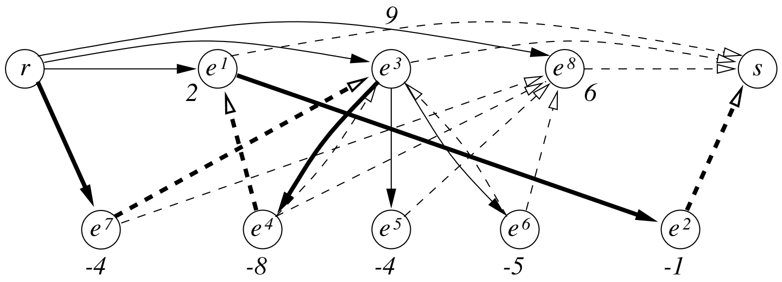

Figure 2: The network $\mathcal{N}(x)$ for the starting solution $x = e^1 + e^3 + e^8$. The thick lines represent the r-s-path of minimal weight.

There is no negative cycle in the network $\mathcal{N}(x)$. The r-s-path of minimal weight $w(P) = -2$ is $P := (e^7, e^3, e^4, e^1, e^2)$ whose arcs are marked in the figure by thick lines. Let $y := \chi^{(J \triangle P)} = e^2 + e^4 + e^7 + e^8$. By Lemma 3, y is feasible for the above system and attains an objective value of 19.

For arbitrary matrices A and C one may not expect that every feasible non-optimal solution x can be improved by applying only vectors that correspond to cycles and paths in the network $\mathcal{N}(x)$. It seems natural to characterize the subset of problems (IIP) for which the network is exact, where the precise meaning of the latter is as follows.

Definition 5 *We say that the family of networks $\mathcal{N}(\cdot)$ is* exact *for a problem of the form* (IIP) *if x is optimal for* (IIP) *if and only if there is no negative r-s-path or cycle in $\mathcal{N}(x)$.*

If a family of networks $\mathcal{N}(\cdot)$ is exact, then on account of Lemma 3 and the polynomial time equivalence of augmentation and optimization, an optimal solution of (IIP) can be computed via shortest path techniques in polynomial time.

We begin with the bipartite $\mathfrak{b}$-matching problem and show that for this problem the network is exact (see Lemma 8).

Definition 6 (The weighted bipartite $\mathfrak{b}$-matching problem).
Given a bipartite graph $G = (V, E)$ with bipartition $V = V_1 \times V_2$, $c \in \mathbb{Z}^E$, and $\mathfrak{b} \in \mathbb{Z}_+^V$. Find a c-maximal $\mathfrak{b}$-matching M in G, where a set $M \subseteq E$ is called a $\mathfrak{b}$-matching *in G if $|M \cap \delta(v)| \leq \mathfrak{b}_v$ for all $v \in V$.*

Set $\mathcal{I}_i := \{M \subseteq E : |M \cap \delta(v)| \leq \mathfrak{b}_v \text{ for all } v \in V_i\}$ for $i = 1, 2$. Then $\mathcal{I}_1$ and $\mathcal{I}_2$ are partition matroids and hence, the bipartite $\mathfrak{b}$-matching problem is a special case of the matroid intersection problem. Let A and C be the incidence matrices of (V_1, E) and (V_2, E), respectively, and $b := (\mathfrak{b}_v)_{v \in V_1}$ and $d := (\mathfrak{b}_v)_{v \in V_2}$. Then the feasible solutions of the system $Ax \leq b,\ Cx \leq d,\ x \in \{0,1\}^E$ are precisely the characteristic vectors of the $\mathfrak{b}$-matchings in G. In order to characterize the symmetric difference of two $\mathfrak{b}$-matchings in G, we resort to the notion of M-exposed nodes and M-alternating paths and cycles.

Definition 7 (M-exposed, M-alternating) [B57]. *Let M be a $\mathfrak{b}$-matching in G. A vertex $v \in V$ is called* M-exposed *if $|M \cap \delta(v)| < \mathfrak{b}_v$.*
A path P (or cycle C) in G is called M-alternating *if*
(i) the edges of P (or C) are alternately in and not in M and
(ii) if the first (or last) edge of the path P is not in M, then the corresponding first (or last) node of P is M-exposed.

Note that this definition implies that the symmetric difference of the $\mathfrak{b}$-matching M with an M-alternating path (or cycle) is again a $\mathfrak{b}$-matching in the graph G. We obtain

Lemma 8 *Let M be a $\mathfrak{b}$-matching in the bipartite graph $G = (V, E)$. Then M is a c-maximal $\mathfrak{b}$-matching in G if and only if there exists no negative r-s-path or cycle in the augmentation network $\mathcal{N}(M)$.*

Proof. It is a slight extension of Berge's augmenting path theorem for matchings ([B57]) that the symmetric difference $M \triangle N$ of two $\mathfrak{b}$-matchings M and N in the bipartite graph G is the edge-disjoint union of M-alternating paths and cycles in G. Since every M-alternating path or cycle in G splits into the vertex-disjoint union of r-s-paths and cycles in $\mathcal{N}(M)$, the claim follows. $\square$

So far we have shown that the family of networks $\mathcal{N}(\cdot)$ is exact for the bipartite $\mathfrak{b}$-matching problem. Therefore, shortest path algorithms lead to a polynomial-time exact augmentation and finally optimization algorithm.

We will now address the converse question. If we know that the family of networks $\mathcal{N}(\cdot)$ is exact for a problem of the form (IIP), then what can we say about the problem under investigation? It will follow from Theorem 9 that the problem we started with is a bipartite $\mathfrak{b}$-matching problem. This together with Lemma 8 yields an algorithmic characterization of the bipartite $\mathfrak{b}$-matching problem.

Theorem 9 *Let $P = \operatorname{conv}\{x \in \{0,1\}^n : Ax \leq b,\ Cx \leq d\}$ and assume that each inequality defines a $0/1$-facet of the polytope P.*
If the network $\mathcal{N}(\cdot)$ is exact then $P \cap \{0,1\}^n$ is the set of all characteristic vectors of $\mathfrak{b}$-matchings in a bipartite graph.

Proof. Let $\mathcal{I} := \{M \subseteq \{1,\dots,n\} : \chi^M \in P\}$. Since $\mathcal{I}$ is an independence system every inequality of $Ax \leq b$, $Cx \leq d$ has the form $x(F) \leq r(F)$ with $F \subseteq \{1,\dots,n\}$ and $r(F)$ is the rank of F in $\mathcal{I}$.
We claim that each column of A and C contains at most one nonzero entry, respectively. Assume that there are inequalities $x(F) \leq r(F)$ and $x(F') \leq r(F')$ of $Ax \leq b$ with $F \neq F'$ and $F \cap F' \neq \emptyset$. Then

$$S := P \cap \{x : x(F) = r(F)\} \text{ and } S' := P \cap \{x : x(F \cap F') = r(F \cap F')\}.$$

define faces of P with $S \not\subseteq S'$, since S is even a facet, $F \neq F'$ and $S' \neq P$. Therefore, there exists a point $y \in (S \setminus S') \cap \{0,1\}^n$. Letting $J := \operatorname{supp}(y) \cap F$, we have $J \in \mathcal{I}$, $J \subseteq F$, $|J| = r(F)$, and $|J \cap F'| < r(F \cap F')$.
Let J^* be a maximal independent set in $F \cap F'$ and $c := \chi^{(J \cup J^*) \cap (F \cap F')}$. Then

$$c(J) = |J \cap F'| < r(F \cap F') = |J^*| = c(J^*).$$

Since J^* is a better solution than J and the network $\mathcal{N}(\cdot)$ is exact there exists a negative r-s-path or cycle T in $\mathcal{N}(J)$, i.e., we have

$$|\,T \cap \big(J \cap (F \cap F')\big)\,| < |\,T \cap \big((J^* \setminus J) \cap (F \cap F')\big)\,|.$$

Since J is maximally independent in F it follows that

$$J^* \setminus J \subseteq \{i \notin J : A(y + e^i) \not\leq b\}.$$

Thus there exists an A-arc (j, i) in T with $j \in J$, $i \notin J$ such that $c(j) = 0$ and $c(i) = 1$. This implies that $j \in J \setminus (F \cap F')$ and $i \in J^* \setminus J \subseteq F \cap F'$. We obtain $j \notin F'$ and $i \in F'$, i.e., $\chi_i^{F'} = 1$ and $\chi_j^{F'} = 0$. This proves the claim, since this means that $A_{.i} \not\leq A_{.j}$ contradicting the condition of an A-arc.

Extend the systems $Ax \leq b$ and $Cx \leq d$ by adding the upper bound inequalities $x_i \leq 1$ such that any of the new systems $A'x \leq b'$ and $C'x \leq d'$ contain exactly one 1-entry in each column. Now these systems represent the incidence matrix of a bipartite graph $G = (V_1 \times V_2, E)$, where V_1 and V_2 correspond to the rows of A' and C', respectively. With $\mathfrak{b} := (b', d')$ the elements of $\mathcal{I}$ are exactly the $\mathfrak{b}$-matchings in G. $\square$

3 Gomory-Young Augmentation Vectors

Branch and cut algorithms of today make use of "general" cutting planes for separation issues. Such cuts are generated algebraically, i.e., they ignore the combinatorial structure of the problem. Among general cutting planes are cuts based on disjunctive programming [Ba98] and Gomory's integer and mixed integer cutting planes. Of major relevance in the following will be the second family, often called Chvátal-Gomory cuts. Such cuts have – in our context – the form $\{x \in \mathbb{R}^n : \lfloor u^T A \rfloor x \leq \lfloor u^T b \rfloor\}$ for a vector $u \in \mathbb{R}^m_+$. It is important to note that adding iteratively such cuts to the system yields a finite algorithm to detect the optimal solution, see [Go69].

Gomory-Young augmentation vectors can be seen as a primal analogue of general cutting planes especially of Chvátal-Gomory cuts. Their derivation is purely algebraic and requires no problem specific information. Let us describe informally the derivation of Gomory-Young augmentation vectors.

Starting with an integral basic feasible solution x^* that is not optimal, we try to detect an augmenting direction so as to obtain a new basic feasible solution x^{**} with $c^T x^{**} \geq c^T x^*$.

We resort to the simplex method for performing this task. A major first problem one faces with this method is that after executing an ordinary pivot step the new solution will in general not be integral.

The essential modification consists of generating a Chvátal-Gomory cut from one of the legitimate pivot rows in the corresponding simplex-tableau in a way that ensures that the cut itself will become a legitimate pivot row with pivot coefficient 1. This will guarantee the integrality of the new solution. Geometrically, we determine at the point x^* an improving direction from the simplex tableau. In order to reach a new integral point we add to the system a cut that prevents us from becoming fractional. This approach is discussed in [H69] and goes back to [G68] and [Y68]. It may be viewed as a modification of Gomory's all-integer algorithm [Go63] in the sense that the latter algorithm resorts to the dual simplex method, whereas the approach of [Y68] uses the primal simplex algorithm in order to maintain at any time in the process a primal feasible integral solution.

To provide a notational context for describing the procedure, we shall consider integer programs of the following type:

$$\max c^T x : x \in \mathcal{F} := \{x \in \mathbb{Z}_+^n : Ax = b\} \tag{2}$$

with $A = (I, A_N) \in \mathbb{Z}^{m\times n}$, $b \in \mathbb{Z}_+^m$, and $c \in \mathbb{Z}^n$. Denoting $x = (x_B, x_N)$ and $c = (c_B, c_N)$ this problem can also be formulated as follows

$$\begin{array}{rcl} \max\ c_B^T x_B + c_N^T x_N & & \\ x_B + A_N x_N & = & b \\ x & \geq & 0,\ \text{integer}. \end{array}$$

Let $x^* = (x_B^*, x_N^*) = (b, 0) \in \mathcal{F}$ be the corresponding basic feasible solution and $c_0 := c^T x^* = c_B^T b$ its c-weight.

With $\bar{c}^T := c_N^T - c_B^T A_N \in \mathbb{Z}^{n-m}$, we may write $c^T x = c_0 + \bar{c}^T x_N$ for $x \in \mathcal{F}$. The integer program can be written as an all-integer simplex-tableau as follows:

$$\begin{array}{c|cc} & & x_N \\ & c_0 & -\bar{c} \\ \hline x_B & b & A_N \end{array}$$

If we assume that the current solution x^* is not c-optimal, then there exists a column s with $\bar{c}_s > 0$, the pivot column. Choose a pivot row r such that $b_r/A_{rs} \leq b_i/A_{is}$ for all i with $A_{is} > 0$. Now, before carrying out the Gaussian elimination of the simplex method, we consider the following valid cutting plane that one may generate from the pivot row,

$$\sum_{j\in N} \left\lfloor \frac{A_{rj}}{A_{rs}} \right\rfloor x_j \leq \left\lfloor \frac{b_r}{A_{rs}} \right\rfloor.$$

This inequality defines a Chvátal-Gomory cut and is valid for all $x \in \mathcal{F}$. Adjoining this cut to the simplex-tableau with slack variable s_1 yields a tableau

$$\begin{array}{c|cc} & & x_N \\ & c_0 & -\bar{c} \\ \hline x_B & b & A_N \\ s_1 & \lfloor \frac{b_r}{A_{rs}} \rfloor & \lfloor \frac{A_{r\cdot}}{A_{rs}} \rfloor \end{array}$$

Obviously, the vector $(x^*, \lfloor \frac{b_r}{A_{rs}} \rfloor)$ is the new basic solution for the extended system. Since

$$\frac{\lfloor b_r/A_{rs} \rfloor}{\lfloor A_{rs}/A_{rs} \rfloor} = \left\lfloor \frac{b_r}{A_{rs}} \right\rfloor \leq \frac{b_r}{A_{rs}}$$

we can choose this new s_1-row as the pivot row. This row has the property that the new pivot element is equal to 1. Therefore, the resulting simplex-tableau will again be all-integer.

Let $(x^{**}, 0)$ be the new basic integer solution after applying the pivot. The difference vector $x^{**} - x^*$ is called a *Gomory-Young augmentation vector*. Below the procedure is illustrated on an example to demonstrate that the optimal integral solution can be found by an iteration of this procedure.

Unfortunately, the above algorithm does not automatically support a proof of finiteness, because the right hand side of the cut $\lfloor \frac{b_r}{A_{rs}} \rfloor$ may be zero. In this case we do not move away from the old basic feasible solution x^* but represent this point by a new basis only. This problem of degeneracy may be overcome, but it requires some effort and care, in particular in the selection of the pivot column and row. For further details including a complete description of the mentioned modifications and a proof of finiteness we refer the reader to [H69].

We illustrate the procedure on a small example.

Example 10 Consider the integer program

$$\begin{array}{rrcrcrl} & \max \quad x_1 & & & & & \\ \text{subject to} & -\,3x_1 & + & 5x_2 & \leq & 5, & \text{(a)} \\ & x_1 & - & x_2 & \leq & 1, & \text{(b)} \\ & 10x_1 & - & 4x_2 & \leq & 15, & \text{(c)} \\ & & & x_1, x_2 & \in & \mathbb{Z}_+. & \end{array}$$

The feasible domain of this program and the cuts that this primal algorithm produces are illustrated in the figure below.

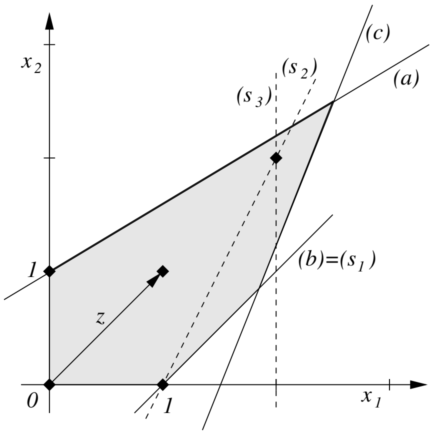

Figure 3: Illustration of the feasible domain and the cuts generated.

We add the slack variables (x_3, x_4, x_5) to the system and start with the feasible solution $x^0 = (0, 0, 5, 1, 15)$, i.e., $B = \{3, 4, 5\}$ and $N = \{1, 2\}$. The resulting problem is expressed in tableau (1a) (ignore the s_1-row for a moment).

(1a)

		x_1	x_2
	0	-1	0
x_3	5	-3	5
x_4	1	1^*	-1
x_5	15	10	-4
s_1	1	$\boxed{1}$	-1

We choose x_1 as the pivot column, x_4 as the pivot row (the resulting pivot element is marked by *) and generate from this row the cut. We adjoin the cut in the last row, add the slack variable s_1 to the basis and perform the pivot operation. This leads to the new tableau (1b) and solution $x^1 = (1, 0, 8, 0, 5)$.

The process is repeated accordingly. The results are described in tableau (1c) and (1d), respectively.

(1b)

		s_1	x_2
	1	1	-1
x_1	1	1	-1
x_3	8	3	2
x_4	0	-1	0
x_5	5	-10	6^*
s_2	0	-2	$\boxed{1}$

(1c)

		s_1	s_2
	1	-1	1
x_1	1	-1	1
x_2	0	-2	1
x_3	8	7^*	-2
x_4	0	-1	0
x_5	5	2	-6
s_3	1	$\boxed{1}$	-1

(1d)

		s_3	s_2
	2	1	0
x_1	2	1	0
x_2	2	2	-1
x_3	1	-7	5
x_4	1	1	-1
x_5	3	-2	-4

The final tableau (1d) is optimal and corresponds to the new basic solution $x^* = (2, 2, 1, 1, 3)$. The cuts which are generated during this procedure are (s_1) $x_1 - x_2 \leq 1$, (s_2) $2x_1 - x_2 \leq 2$, and (s_3) $x_1 \leq 2$.

With $h^1 := (0, 1, -5, 1, 4)$ (which is the negative of the s_2-column in the last tableau) and $h^2 := (-1, -2, 7, -1, 2)$ (negative of the s_3-column) we have $\mathcal{F} - x^* \subseteq \operatorname{cone}\{h^1, h^2\}$, $c^T h^1 = 0 \leq 0$, and $c^T h^2 = -1 \leq 0$, i.e., the primal optimality certificate for x^*.

4 Combinatorial vs. General Augmentation Vectors

In the previous two sections we described the primal analogue of general and combinatorial cutting planes. It is clear from the way branch and cut systems are designed today that a combination of the two approaches is particularly fruitful. This transformed to the primal side leads to the question of studying the interaction between general augmentation vectors and combinatorial augmentation vectors.

To be more formal, let $\mathcal{T}$ be a family of (combinatorial) augmentation vectors and $t \in \mathcal{T}$ such that $x^* + t \in \mathcal{F}$ and $c^T t > 0$. Our objective is to augment x^* by t (or a positive integer multiple of t) in such a way, that in the subsequent step we are still left with the possibility of generating a Gomory-Young augmentation vector or using again a combinatorial augmentation vector from the family $\mathcal{T}$ for improving. Note however that in order to produce a Gomory-Young vector a basic feasible solution is needed. Since we can not expect that $x^* + \lambda t$ is itself a basic feasible solution of the original problem (2) for any $\lambda \in \mathbb{Z}$, $\lambda > 0$, we transform the problem by lifting it into higher dimension.

To do this let $\gamma := \bar{c}^T t_N = c^T t > 0$ and $a := A_N t_N = -t_B \in \mathbb{Z}^m$. We now extend the given system by an additional column, i.e., define $y^* := (x^*, 0)$ and

$$\mathcal{F}(t) := \{y \in \mathbb{Z}_+^{n+1} : (I, A_N, a)y = b\} \quad \text{and} \quad c(t) := (c, c_N^T t_N).$$

There is a one to many correspondence between the points in $\mathcal{F}$ and the points in $\mathcal{F}(t)$ that is summarized below.

Proposition 11 *$x \in \mathcal{F}$ implies that $(x, 0) \in \mathcal{F}(t)$.*
Conversely, $y = (z, z_{n+1}) \in \mathcal{F}(t)$ corresponds to $x := (z_B, z_N + z_{n+1} t_N) \in \mathcal{F}$ such that $c(t)^T y = c^T z + c_N^T t_N \cdot z_{n+1} = c^T x$.

On account of this proposition we may consider instead of the original integer program (2) the integer program

$$\max\ c(t)^T y : y \in \mathcal{F}(t) = \{y \in \mathbb{Z}_+^{n+1} : (I, A_N, a)y = b\}. \tag{3}$$

Since $c(t)^T y = c_0 + \bar{c}^T y_N + \gamma y_{n+1}$ the simplex-tableau extended to the higher dimensional space reads

$$\begin{array}{c|ccc} & & y_N & y_{n+1} \\ & c_0 & -\bar{c} & -\gamma \\ \hline y_B & b & A_N & a \end{array}$$

At this point we choose as the pivot column the y_{n+1}-column (note that $\gamma > 0$) and as a pivot row any row r with $b_r/a_r \leq b_i/a_i$ for all i with $a_i > 0$. Now we perform the same operations as described in Section 3 , i.e., we extend the system by the Chvátal-Gomory cut

$$\sum_{j \in N} \left\lfloor \frac{A_{rj}}{a_r} \right\rfloor y_j + y_{n+1} \leq \left\lfloor \frac{b_r}{a_r} \right\rfloor$$

with slack variable s_1. This results in the simplex-tableau

$$\begin{array}{c|ccc} & & y_N & y_{n+1} \\ & c_0 & -\bar{c} & -\gamma \\ \hline y_B & b & A_N & a \\ & & & \\ s_1 & \lfloor \frac{b_r}{a_r} \rfloor & \lfloor \frac{A_{r\cdot}}{a_r} \rfloor & 1 \end{array}$$

Since $\lfloor \frac{b_r}{a_r} \rfloor \leq \frac{b_r}{a_r}$, the s_1-row is a legitimate pivot row with pivot element 1. Let $y^{**} \in \mathcal{F}(t)$ be the new basic solution (ignoring the s_1-component) after executing the pivot step. Then

$$y^{**} = (b - \left\lfloor \frac{b_r}{a_r} \right\rfloor a\,,\ 0\,,\ \left\lfloor \frac{b_r}{a_r} \right\rfloor)$$

and this corresponds to the vector

$$x^{**} = \left(b - \left\lfloor \frac{b_r}{a_r} \right\rfloor a\,,\ 0 + \left\lfloor \frac{b_r}{a_r} \right\rfloor t_N\right) = (b,\ 0) + \left\lfloor \frac{b_r}{a_r} \right\rfloor (t_B, t_N) = x^* + \left\lfloor \frac{b_r}{a_r} \right\rfloor t.$$

Note that $1 \leq \lfloor \frac{b_r}{a_r} \rfloor = \max\{\lambda \in \mathbb{Z}_+ : x^* + \lambda\, t \in \mathcal{F}\}$, which means that x^* is augmented by t as often as we remain feasible. Moreover, the resulting vector x^{**} corresponds to a basic feasible solution of (3).

To illustrate this procedure we continue with Example 10.

Example 12 (Example 10 continued)

We start again with the equation representation of the problem and the feasible solution $x^0 = (0,0,5,1,15)$. Suppose that we can apply the vector $t = (1,1,-2,0,-6)$ for augmenting x^0, which corresponds to the vector $z = (1,1)$ in Figure 3. Then we extend our system by the column $a := A_N t_N = (2,0,6)$ and obtain the tableau (2a) (ignore the s_1-row in the meantime).

(2a)

		y_1	y_2	y_6
	0	-1	0	-1
y_3	5	-3	5	2^*
y_4	1	1	-1	0
y_5	15	10	-4	6
s_1	2	-2	2	$\boxed{1}$

(2b)

		y_1	y_2	s_1
	2	-3	2	1
y_3	1	1	1	-2
y_4	1	1	-1	0
y_5	3	22	-16	-6
y_6	2	-2	2	1

We select the new column corresponding to a as the pivot column and perform the steps outlined before. This leads to a new cut which we adjoin to the system. Performing the pivot operation results in tableau (2b).

Associated with this tableau the vector $y = (0,0,1,1,3,2) \in \mathcal{F}(t)$ is the basic feasible integral solution. Its corresponding feasible solution in $\mathcal{F}$ is the vector $x = (0,0,1,1,3) + 2 \cdot (1,1,0,0,0) = (2,2,1,1,3)$, which is equal to $x^0 + 2t$.

Acknowledgments: We wish to thank Giovanni Rinaldi for many valuable discussions on the subject of this paper.

Thanks are also due to Laurence Wolsey for providing us the literature of Gomory's work.

References

[Ba98] E. Balas, *Disjunctive Programming: Properties of the Convex Hull of Feasible Points,* Discrete Applied Mathematics **89**, 3–44, 1998

[B57] C. Berge, *Two Theorems in Graph Theory,* Proceedings of the National Academy of Sciences (U.S.A.) **43**, 842–844, 1957

[CCPS98] W. J. Cook, W. H. Cunningham, W. R. Pulleyblank and A. Schrijver, *Combinatorial Optimization,* Wiley-Interscience, New York, 1998

[E70] J. Edmonds, *Submodular Functions, Matroids and Certain Polyhedra,* in: Comb. Structures and their Applications (R. K. Guy, H. Hanai, N. Sauer and J. Schönheim, eds.), Gordon and Brach, New York, 69–87, 1970

[E79] J. EDMONDS, *Matroid Intersection,* Annals of Discrete Mathematics **4**, 39–49, 1979

[FSW99] R. T. FIRLA, B. SPILLE AND R. WEISMANTEL, *An Algorithmic Characterization of Matroid Intersection,* in preparation

[G68] F. GLOVER, *A New Foundation for a Simplified Primal Integer Programming Algorithm,* Operations Reserach **16** (4), 727–740, 1968

[Go63] R. E. GOMORY, *An All-Integer Integer Programming Algorithm,* in: Industrial Scheduling (J. F. Muth and G. L. Thompson, eds.), Prentice-Hall, Englewood Cliffs, N.J., 193–206, 1963

[Go69] R. E. GOMORY, *An Algorithm for Integer Solutions to Linear Programming,* in: Recent Advances in Mathematical Programming (R. L. Graves and P. Wolfe, eds.), McGraw-Hill, New York, 269–302, 1969

[GL95] M. GRÖTSCHEL AND L. LOVÁSZ, *Combinatorial Optimization,* in: Handbook of Combinatorics, Volume II (R. Graham, M. Grötschel, and L. Lovász, eds.), North–Holland, 1541–1597, 1995

[GLS88] M. GRÖTSCHEL, L. LOVÁSZ AND A. SCHRIJVER, *Geometric Algorithms and Combinatorial Optimization,* Springer, Berlin, 1988

[H69] T. C. HU, *Integer Programming and Network Flows,* Addison-Wesley, Reading, Mass., 1969

[L76] E. L. LAWLER, *Combinatorial Optimization: Networks and Matroids,* Holt, Rinehart and Winston, New York, 1976

[P73] M. PADBERG, *On the Facial Structure of Set Packing Polyhedra,* Mathematical Programming **5**, 199–215, 1973

[PR82] M. PADBERG, M. R. RAO, *Odd Minimum Cut Sets and b-Matchings,* Mathematics of Operations Research **7**, 67–80, 1982

[SW99] A. S. SCHULZ AND R. WEISMANTEL, *A Polynomial Time Augmentation Algorithm for Integer Programming,* in: Proceedings of the 10th ACM-SIAM Symposium on Discrete Algorithms, Baltimore, 967–968, 1999

[SWZ95] A. S. SCHULZ, R. WEISMANTEL AND G. M. ZIEGLER, 0/1 *Integer Programming: Optimization and Augmentation are Equivalent,* in: Lecture Notes in Computer Science 979, (P. Spirakis, ed.), Springer, Berlin, 473–483, 1995

[Y68] R. D. YOUNG, *A Simplified Primal (All-Integer) Integer Programming Algorithm,* Operations Research **16** (4), 750–782, 1968