Convex Programming Tools for Disjunctive Programs

João Soares,
Departamento de Matemática,
Universidade de Coimbra,
Portugal

Abstract

A Disjunctive Program (DP) is a mathematical program whose feasible region is the convex hull of the union of convex sets. The objective function is also convex. Disjunctive Programming models are very frequent as a modeling framework for setup costs or constraints, and models with constraints that are better expressed as disjunctions. Some Process Sinthesis Design models arising from Chemical Engineering are mixed integer convex programming models which are particular instances of a Disjunctive Program. In this talk we will address questions that are raised when conceptualizing a Branch-and-cut algorithm for mixed-integer convex programming.

1 Introduction

The process synthesis network problem problem in Chemical Engineering is the problem of simultaneously determining the optimal structure and operating parameters for a chemical synthesis problem. This problem can be modeled as a mixed 0-1 convex program where the continuous variables represent process parameters such as flowrates and the 0-1 variables represent the potential existence of a process unit. The nonlinear elements come from the intrinsic nonlinear input-output performance equations of some process units, see [3] where this model is proposed and [8, 4] for related models.

Other models of Network Design in communication and transportation networks are discrete by nature. The 0-1 variables represent the potential existence of multiplexers, concentrators, or interface message processors in computer communication networks, junctions in pipeline networks, interchanges in highway networks, and so on. Discrete variables may also represent the discrete quantity of physical arc units of certain characteristics between two junctions of the network. In a simple model, described in [5], the nonlinear element comes from modelling *delay* at some link (i, j) as proportional to the fraction of the rate of messages crossing the link (i, j) to the available capacity of the same link.

We propose a cutting-plane algorithm for solving the following mathematical program that we will refer to as a mixed zero-one convex program,

$$\begin{array}{ll} \min & f(x) \\ \text{s.t.} & G(x) \le 0 \\ & x_i \in \{0,1\}, i = 1, \ldots, p, \end{array} \tag{1}$$

where $f\colon \mathbb{R}^n \to \mathbb{R}$ is a closed convex function and $G\colon \mathbb{R}^n \to \mathbb{R}^m$ is a vector function of closed convex functions. The variables x_i, for $i = 1, \ldots, p$, are zero-one constrained and the variables x_i, for $i = p+1, \ldots, n$, are simply nonnegative. We will further assume that both f and G are continuous in an open set containing the continuous relaxation, i.e., when $\{0,1\}$ is replaced by $[0,1]$.

Our work extends to the nonlinear setting the lift-and-project approach of Balas, Ceria and Cornuéjols [1, 2], which is seen as one of the most important practical contributions to the solution of mixed zero-one linear programs by general-purpose cutting-plane based algorithms since the work of Gomory in the sixties. As proposed by Stubbs and Mehrotra [7], we solve the cut generation problem in its dual form. Some of the distinctive features of our algorithm are the following: our algorithm guarantees the existence of a cut whenever an optimal solution was not yet found; we solve the cut generation problem using standard nonlinear programming algorithms; and, we fully extend the lifting procedure to the nonlinear setting.

The article is structured in the following way. In Section 2 we describe the basic cutting-plane algorithm specialized to solve Program (1). In Section 3 we explain how the cut generation problem can be solved in a smaller-dimension space, taking advantage of the fact that some variables are already integral. In the talk, the algorithm will be illustrated on a small example. A practical implementation of this method is part of an ongoing research project.

2 The basic cutting plane algorithm

Our approach requires that we use the following equivalent formulation of program (1),

$$\begin{array}{ll} \min & x_{n+1} \\ \text{s.t.} & f(x) \le x_{n+1} \\ & G(x) \le 0 \\ & x_i \in \{0,1\}, i = 1, \ldots, p. \end{array} \tag{2}$$

Since f is convex then this formulation is still a mixed zero-one convex program. Moreover, the feasible region K can be replaced by $P = \text{conv}\ (K)$ without loss of generality, where we note that P is closed. As a matter of notation, we will still use the same $f(x)$ and $G(x)$ eventhough we are refering to these functions as functions of the first n components of the vector x that now lies in $\mathbb{R}^{n+1}$.

A specialization of the basic cutting-plane algorithm is presented in Figure 1. The algorithm requires performing three basic steps in each iteration. In the first step, the *relaxation step*, we seek an optimal solution $\bar{x}$ of the following convex program

$$\begin{array}{ll} \min & x_{n+1} \\ \text{s.t.} & x \in \bar{P}, \end{array} \tag{3}$$

whose feasible region $\bar{P}$ is defined by

$$\bar{P} \quad \equiv \quad \left\{ x \in \mathbb{R}^{n+1} \colon \begin{array}{ll} f(x) \le x_{n+1}, & a^i x \le b_i, i = 1, \ldots, m_1, \\ G(x) \le 0, & x_i \in [0,1], i = 1, \ldots, p, \end{array} \right\}, \tag{4}$$

where m_1 is the number of cuts generated so far. In the second step, the *optimality check step*, we try to reduce as much as possible the number of fractional components of $\bar{x}$ while keeping the same value of the component $\bar{x}_{n+1}$. In the third step, the *separation step*, we use the last index j tried in the second step to define the following disjunctive programming relaxation $\bar{P}_j$ of P,

$$\bar{P}_j \equiv \text{conv}\ \left(\left(\bar{P} \cap \{x\colon x_j = 0\}\right) \cup \left(\bar{P} \cap \{x\colon x_j = 1\}\right)\right). \tag{5}$$

The following proposition shows that a nonoptimal $\bar{x} \notin \bar{P}_j$, from where we are able to guarantee the existence of a separating hyperplane.

Proposition 1 *In each iteration of the algorithm BCP4MINLP, Step 2 is performed at most p times. Moreover, if j is the last index tried in Step 2 then either $\bar{x}$ is optimal or $\bar{x} \notin \bar{P}_j$.*

Proof: We recall that in Step 2 of the algorithm BCP4MINLP, the integer-constrained variables are sequentially fixed at one of their bounds, zero or one, until an index j is found such that

$$\min_{i=0,1} \left(\begin{array}{ll} \min & x_{n+1} \\ \text{s.t.} & x \in \bar{P}, \\ & x_{F'} = \bar{x}_{F'}, \quad x_j = i \end{array} \right) > \bar{x}_{n+1}, \tag{6}$$

where F' identifies the variables that are fixed in the process. Since F' can have at most p elements then Step 2 is performed at most p times until either (6) holds or all the integer constrained variables are fixed in which case we would have found an optimal solution.

Now, we prove the second part of this proposition. Let j be the last index tried in Step 2 so that (6) holds. Assume, by contradiction, that $\bar{x} \in \bar{P}_j$. Then, the point $\bar{x}$ can be represented by one of the following three possible ways:

a. $\bar{x} = \delta z + (1-\delta)y$, where $\delta \in (0,1)$, $z \in \bar{P} \cap \{x\colon x_j = 0\}$ and $y \in \bar{P} \cap \{x\colon x_j = 1\}$;

b. $\bar{x} = z + dy$, where $z \in \bar{P} \cap \{x\colon x_j = 0\}$ and dy is a direction of the set $\bar{P} \cap \{x\colon x_j = 1\}$ if this set is nonempty or the zero vector otherwise.

c. $\bar{x} = dz + y$, where $y \in \bar{P} \cap \{x\colon x_j = 1\}$ and dz is a direction of the set $\bar{P} \cap \{x\colon x_j = 0\}$ if this set is nonempty or the zero vector otherwise.

If $\bar{x}$ can be decomposed as in a. then, since $\bar{x}_k \in \{0,1\}$, for every $k \in F'$, we must have $z_k = y_k = \bar{x}_k$, for every $k \in F'$. Thus,

$$\bar{x} \in \text{conv}\ \left(\left(\bar{P} \cap \{x\colon x_{F'} = \bar{x}_{F'}, x_j = 0\}\right) \cup \left(\bar{P} \cap \{x\colon x_{F'} = \bar{x}_{F'}, x_j = 1\}\right)\right)$$

which contradicts (6). If $\bar{x}$ can be decomposed as in b. then, since $dy_i = 0$, for every $i \in \{1,\ldots,p\}$, $z_k = \bar{x}_k$, for every $k \in F' \cup \{j\}$. Thus, $\bar{x} \in \bar{P} \cap \{x\colon x_{F'} = \bar{x}_{F'}, x_j = 0\}$ which contradicts (6) once again. If $\bar{x}$ can be decomposed as in c. an analogous argument as in b. applies. $\square$

3 The cut generation problem

We explain how the cut generation solution procedure should be implemented to take advantage of the fact that many variables have been fixed during the second step of the algorithm BCP4MINLP. Our cut generation problem uses the following duality result

$$\begin{array}{lll} \sup & \alpha\bar{x} - \beta \\ s.t. & (\alpha,\beta) \in \text{polar}\ (\bar{P}_j), \\ & \|\alpha_F\|_* \leq 1 \end{array} \quad = \quad \begin{array}{ll} \inf & \|x - \bar{x}\| \\ s.t. & x \in \bar{P}_j, \\ & x_{F'} = \bar{x}_{F'}. \end{array} \tag{7}$$

```
**Data:**            Functions $f$ and $G$. The scalars $n$ and $p$.
**Initialization:**  Set $k = 0$ and define $P^0$ as
                     $P^0 \equiv \left\{ x \in \mathbb{R}^{n+1} \colon \begin{array}{ll} f(x) \leq x_{n+1}, & \\ G(x) \leq 0, & x_i \in [0,1], i = 1, \ldots, p, \end{array} \right\},$
**Iteration-$k$:**

    *Step 1:*   (Relaxation) Let $\bar{x}$ be the optimal solution of
                $\begin{array}{ll} \min & x_{n+1} \\ \text{s.t.} & x \in P^k \end{array},$

    *Step 2:*   (Optimality check) Define $F \equiv \{j \in \{1, \ldots, p\} \colon 0 < \bar{x}_j < 1\}$ and
                $F' = \{1, \ldots, p\} \setminus F$. If $F$ is empty then stop: $\bar{x}$ is an optimal solution
                and $\bar{x}_{n+1}$ is the optimal value. Otherwise, let $j \in F$.

        *Step 2.1:*   Find an optimal solution $\hat{x}$ of
                      $\min \left( \begin{array}{ll} \min & x_{n+1} \\ \text{s.t.} & x \in P^k \\ & x_{F'} = \bar{x}_{F'} \\ & x_j = 0 \end{array}, \begin{array}{ll} \min & x_{n+1} \\ \text{s.t.} & x \in P^k \\ & x_{F'} = \bar{x}_{F'} \\ & x_j = 1 \end{array} \right)$

        *Step 2.2:*   If $\hat{x}_{n+1} = \bar{x}_{n+1}$ then let $\bar{x} = \hat{x}$ and restart Step 2;
                      Otherwise set $x^k = \bar{x}$ and continue to Step 3.

    *Step 3:*   (Separation) Let $j$ be the last index tried in Step 2. Find a
                separating hyperplane "$a^{k+1}x \leq b_{k+1}$" between $P_j^k$ and $x^k$.
                Define $P^{k+1} = P^k \cap \left\{x \colon a^{k+1}x \leq b_{k+1}\right\}$ and set $k := k + 1$.
```

Figure 1: The basic cutting-plane algorithm for mixed zero-one convex programming (BCP4MINLP)

Let us assume without loss of generality that the relaxation $\bar{P}$ is defined by

$$\bar{P} \equiv \{x \in \mathbb{R}^n\colon \quad G(x) \leq 0, \quad x \geq 0, \quad x_i \leq 1, i = 1, \ldots p\}. \tag{8}$$

and $\bar{P}_j$ is defined by (5). Let F be an index set that may or may not be related to the Step 2 of the cutting-plane algorithm, and $F' = \{1, \ldots, n\} \backslash F$ be its complement. If $\hat{x} = (\hat{x}_F, \bar{x}_{F'}) \in \bar{P}_j$ is a known optimal primal solution in (7) then the subgradient $\hat{\xi} = (\hat{\xi}_F, \hat{\xi}_{F'})$ of the function

$$f(x) = \begin{cases} \|x_F - \bar{x}_F\| & \text{if } x_F = \bar{x}_F, \\ +\infty & \text{otherwise,} \end{cases} \tag{9}$$

at the point $\hat{x}$ that satisfies $\hat{\xi}(x - \hat{x}) \geq 0$, for every $x \in \bar{P}_j$, defines an optimal dual solution $(\hat{\alpha}, \hat{\beta}) \in \text{polar } (\bar{P}_j)$.

However, the main purpose of (7) is to define the cut generation problem using a smaller number of variables. This means that after solving the primal problem

$$\begin{array}{ll} \min & \|x_F - \bar{x}_F\| \\ \text{s.t.} & x_F \in \left\{x_F\colon (x_F, \bar{x}_{F'}) \in \bar{P}_j\right\} \end{array} \tag{10}$$

we have at hand an optimal primal solution $\hat{x}_F$ and a subgradient $\hat{\xi}_F$ of the function $\|\cdot - \bar{x}_F\|$ at $\hat{x}_F$, such that $\hat{\xi}_F(x_F - \hat{x}_F) \geq 0$, for every $x_F \in \left\{x_F\colon (x_F, \bar{x}_{F'}) \in \bar{P}_j\right\}$. Thus, a natural question is whether we can extend $\hat{\xi}_F$ so that $\hat{\xi} = (\hat{\xi}_F, \hat{\xi}_{F'})$ is a subgradient of the function f defined by (9) at $\hat{x} = (\hat{x}_F, \bar{x}_{F'})$ that satisfies $\hat{\xi}(x - \hat{x}) \geq 0$, for every $x \in \bar{P}_j$. Another natural question is whether we can apply a similar mechanism even when $\hat{x}_F$ is not optimal.

Our answers to these questions require that $\bar{x}_{F'} = 0$. This can be done without loss of generality because when $\bar{x}_k$, for some $k \in F'$, is nonzero then as long as it coincides with one of its bounds on Program (2) a variable transformation allows for the requirement to hold. In this setting, x_F is feasible for Program (10) if and only if x_F belongs to

$$\text{conv } \left(\left\{x_F\colon (x_F, 0) \in \bar{P}, x_j = 0\right\} \cup \left\{x_F\colon (x_F, 0) \in \bar{P}, x_j = 1\right\}\right). \tag{11}$$

Note that the two individual sets that define this convex hull are the feasible regions in (6) and consequently at the end of Step 2 we already know whether those sets are empty or nonempty. This feature is important because it determines which is the best solution procedure to use on Program (10). If both sets are nonempty then the program can be handled using the solution procedures described on Sections 5.4 and 5.5 of [6]. If one of them is empty then Program (10) is a standard convex program, and may therefore be solved by a standard nonlinear programming algorithm. If the two sets are empty then there is no feasible solution x to Program (2) such that $x_{F'} = \bar{x}_{F'}$. In this case the following inequality

$$\sum_{k \in F' : \bar{x}_k = 0} x_k + \sum_{k \in F' : \bar{x}_k = 1} (1 - x_k) \geq 1,$$

separates $\bar{x}$ from the convex hull of the feasible region of Program (2).

Now, we explain how the lifting procedure works under two distinct situations, depending on the fact that one or none of the sets in (11) is empty. We start by assuming that none of them is empty. Let $\hat{x}_F$ be a feasible solution for Program (10) and $\hat{\xi}_F$ be a subgradient of the function $\|\cdot - \bar{x}_F\|$ at $\hat{x}_F$ such that for a given scalar β satisfying $\hat{\xi}_F \bar{x}_F < \beta$ the following holds

$$\min_{i=0,1} \left(\begin{array}{ll} \min & \hat{\xi}_F z_F \\ \text{s.t.} & (z_F, 0) \in \bar{P}, \\ & z_j = i \end{array} \right) \geq \beta. \tag{12}$$

We remark that $\hat{x}_F$ need not be optimal for Program (10), though if it were optimal then the existence of a subgradient and a scalar satisfying (12) would be guaranteed. Under a constraint qualification, the optimal solution $\hat{z}_F^i$ of each one of the problems in (12) also solves a linear program defined by a suitable matrix $A^i \in \partial G(\hat{z}_F^i, 0)$ so that

$$\min_{i=0,1} \left(\begin{array}{ll} \min & \hat{\xi}_F z_F \\ s.t. & \left\{ \begin{array}{l} G(\hat{z}_F^i, 0) + A^i(z_F - \hat{z}_F^i, 0) \leq 0, \\ z_k \geq 0, k \in F, \qquad z_j = i, \\ z_k \leq 1, k \in F \cap \{1, \ldots, p\}, \end{array} \right. \end{array} \right) \geq \beta \tag{13}$$

The feasible region of each one of these linear programs defines an outer-approximation of each one of the sets in (11). Our lifting procedure applies to these linear programs, so that by the outer-approximation argument it also applies to our original nonlinear sets. Proposition 2 below describes the lifting mechanism in generic terms.

Proposition 2 *Let F be an index set and $F' = \{1, \ldots, n\} \setminus F$. For a given arbitrary vector α_F, let $\hat{z}_F$ be an optimal solution of the following linear program:*

$$\begin{array}{ll} \min & \alpha_F z_F \\ \text{s.t.} & A_F z_F \leq b, \\ & l_F \leq z_F \leq u_F, \end{array} \tag{14}$$

where l_F and u_F are the, possibly infinite, lower and upper bounds, $A_F \in \mathbb{R}^{m \times |F|}$ and $b \in \mathbb{R}^m$. Then, for any extended matrix $A = [A_F, A_{F'}] \in \mathbb{R}^{m \times n}$ there is a closed-form extended vector $\alpha = (\alpha_F, \alpha_{F'})$ such that the vector $\hat{z} = (\hat{z}_F, 0)$ is an optimal solution of the following linear program:

$$\begin{array}{ll} \min & \alpha z \\ \text{s.t.} & Az \leq b, \\ & l_F \leq z_F \leq u_F, \\ & z_{F'} \geq 0. \end{array} \tag{15}$$

Proof: Let $\hat{v} \leq 0$ be the optimal dual multipliers associated with the matrix constraints in Program (14) and define

$$\alpha_k \equiv \max \left(0, \sum_{l=1}^{m} \hat{v}_l a_{lk} \right),$$

for every $k \in F'$. Now, consider Program (15) and use the same dual variables to price the new primal variables z_k, for every $k \in F'$. Since the reduced costs are $\rho_k = \alpha_k - \sum_{l=1}^m \hat{v}_l a_{lk} \geq 0$, for every $k \in F'$, we conclude that $\hat{z} = (\hat{z}_F, 0)$ is optimal for Program (15). $\square$

This proposition shows by construction how to define extended vectors $\hat{\xi}^i = (\hat{\xi}_F, \hat{\xi}_{F'}^i)$ such that $\hat{z}^i = (\hat{z}_F^i, 0)$, for $i = 0, 1$ are still optimal in the following linear programs

$$\min_{i=0,1} \left(\begin{array}{ll} \min & \hat{\xi}^i z \\ s.t. & \left\{ \begin{array}{l} G(\hat{z}^i) + A^i(z - \hat{z}^i) \leq 0, \\ z \geq 0, \qquad z_j = i, \\ z_k \leq 1, k \in F \cap \{1, \ldots, p\}, \end{array} \right. \end{array} \right) \geq \beta \tag{16}$$

whose feasible regions are larger than the set $\bar{P} \cap \{x : x_j = i\}$, respectively. Since $z_{F'} \geq 0$, for every $z \in \bar{P}_j$, then

$$\hat{\xi} = (\hat{\xi}_F, \max_{i=0,1}(\hat{\xi}_{F'}^i))$$

is a subgradient of f at $\hat{x}$ such that $\hat{\xi}x \geq \beta$, for every $x \in \bar{P}_j$. Moreover, since $\hat{\xi}\bar{x} < \beta$ then we have found a separating hyperplane.

Now, we assume that one of the sets in (11) is empty. Thus, Program (10) is solved as a standard convex program because its feasible region is defined by the nonempty set only. However, the fact that one of the sets in (11) is empty does not imply that the same has to occur in (5), when the variables $x_{F'}$ are no longer fixed. Proposition 3 below describes in generic terms how to define the extended vector $\hat{\xi}^i = (\hat{\xi}_F, \hat{\xi}^i_{F'})$ so that $\hat{\xi}^i z \geq \beta$, for every $z \in \bar{P} \cup \{x: x_j = i\}$, when the set $\bar{P} \cap \{x: x_{F'} = 0, x_j = i\}$ is empty.

Proposition 3 *Let F be an index set and $F' = \{1, \ldots, n\} \setminus F$. For a given arbitrary vector α_F, let $\hat{z}_F$ be the optimal value of the following linear program:*

$$\begin{array}{ll} \min & \alpha_F z_F \\ \text{s.t.} & A_F z_F \leq b + \hat{t}e, \\ & l_F \leq z_F \leq u_F, \end{array} \tag{17}$$

where l_F and u_F are the, possibly infinite, lower and upper bounds, $A_F \in \mathbb{R}^{m \times |F|}$, $b, e \in \mathbb{R}^m$ where e is a vector of "all-ones", and $\hat{t} \equiv \min\{t: A_F z_F \leq b + te, l_F \leq z_F \leq u_F\} > 0$. Then, for any extended matrix $A = [A_F, A_{F'}] \in \mathbb{R}^{m \times n}$ and scalar β there is a closed-form extended vector $\alpha = (\alpha_F, \alpha_{F'})$ such that $\alpha z \geq \beta$, for every z such that $Az \leq b, l_F \leq z_F \leq u_F, z_{F'} \geq 0$.

Proof: First, consider the linear program that defines $\hat{t}$. Let $(\hat{t}, \tilde{z})$ be an optimal solution and $\hat{w}$ be the optimal dual multipliers associated with the matrix constraints. Then,

$$\hat{t} = \hat{w}b + \hat{\gamma}_F \tilde{z}_F, \tag{18}$$

where $\hat{\gamma}_k = 0 - \sum_{l=1}^m \hat{w}_l a_{lk}$ is the reduced cost associated with the variable z_k, for each $k \in F$.

Now, consider Program (17) and let $\hat{v}$ be the optimal dual multipliers associated with the matrix constraints. Then,

$$\begin{array}{rl} & \alpha_F \hat{z}_F = \hat{v}\left(b + \hat{t}e\right) + \hat{\rho}_F \hat{z}_F \\ \iff & \alpha_F \hat{z}_F - \hat{t}\hat{v}e = \hat{v}b + \hat{\rho}_F \hat{z}_F, \end{array} \tag{19}$$

where $\hat{\rho}_k = \alpha_k - \sum_{l=1}^m \hat{v}_l a_{lk}$ is the reduced cost associated with the variable z_k, for each $k \in F$.

If $\beta \leq \alpha_F \hat{z}_F - \hat{t}\hat{v}e$ then define $\alpha_k = \max(0, \sum_{l=1}^m \hat{v}_l a_{lk})$, for every $k \in F'$. For every z such that $Az \leq b, l_F \leq z_F \leq u_F, z_{F'} \geq 0$ we have that

$$\begin{array}{rll} \alpha z & = & \alpha_F z_F + \alpha_{F'} z_{F'} \\ & \geq & \sum_{k \in F} \left(\hat{\rho}_k + \sum_{l=1}^m \hat{v}_l a_{lk}\right) z_k + \sum_{k \in F'} \left(\sum_{l=1}^m \hat{v}_l a_{lk}\right) z_k \qquad (20) \\ & = & \hat{\rho}_F z_F + \hat{v}Az \\ & \geq & \hat{\rho}_F z_F + \hat{v}b \qquad (21) \\ & \geq & \hat{\rho}_F \hat{z}_F + \hat{v}b \qquad (22) \\ & = & \alpha_F \hat{z}_F - \hat{t}\hat{v}e \qquad (23) \\ & \geq & \beta, \qquad (24) \end{array}$$

where the inequality (20) follows from the definition of $\hat{\rho}_F$, the definition of $\alpha_{F'}$ and the fact that $z_{F'} \geq 0$; the inequality (21) follows from the fact that $\hat{v} \leq 0$ and $Az \leq b$; the inequality

(22) follows the fact that $\hat{\rho}_k(z_k - \hat{z}_k) \geq 0$, for every $k \in F$, which is consequence of the values of the reduced costs at optimality; the inequality (23) follows from (19); and finally the inequality (24) holds by hypothesis.

If $\beta > \alpha_F \hat{z}_F - \hat{t}\hat{v}e$ then a similar formula works but we need to increase $\hat{v}$ by a suitable positive multiplier of $\hat{w}$. Observe that $\hat{z}_F$ is feasible for the linear program that defines $\hat{t}$ and so, from (18), we have that $\hat{t} \leq \hat{w}b + \hat{\gamma}_F \hat{z}_F$, or equivalently,

$$\beta - \left(\alpha_F \hat{z}_F - \hat{t}\hat{v}e\right) \leq \delta\hat{w}b + \delta\hat{\gamma}_F \hat{z}_F, \tag{25}$$

where $\delta = \left(\beta - \left(\alpha_F \hat{z}_F - \hat{t}\hat{v}e\right)\right)/\hat{t} > 0$. Now, define $\alpha_k = \max\left(0, \sum_{l=1}^m (\hat{v} + \delta\hat{w})_l \, a_{lk}\right)$, for every $k \in F'$. For every z such that $Az \leq b, l_F \leq z_F \leq u_F, z_{F'} \geq 0$ we have that

$$\begin{aligned}
\alpha z &= \alpha_F z_F + \alpha_{F'} z_{F'} \\
&\geq \sum_{k \in F} \left(\hat{\rho}_k + \sum_{l=1}^m \hat{v}_l a_{lk}\right) z_k + \sum_{k \in F'} \left(\sum_{l=1}^m (\hat{v} + \delta\hat{w})_l \, a_{lk}\right) z_k && (26) \\
&= \hat{\rho}_F z_F + (\hat{v} + \delta\hat{w}) Az + \delta\hat{\gamma}_F z_F \\
&\geq \hat{\rho}_F z_F + \hat{v}b + \delta\hat{w}b + \delta\hat{\gamma}_F z_F && (27) \\
&\geq \hat{\rho}_F \hat{z}_F + \hat{v}b + \delta\hat{w}b + \delta\hat{\gamma}_F \hat{z}_F && (28) \\
&\geq \hat{\rho}_F \hat{z}_F + \hat{v}b + \left(\beta - \left(\alpha_F \hat{z}_F - \hat{t}\hat{v}e\right)\right) && (29) \\
&= \beta && (30)
\end{aligned}$$

where the inequality (26) follows from the definition of $\hat{\rho}_F$, the definition of $\alpha_{F'}$ and the fact that $z_{F'} \geq 0$; the inequality (27) follows from the fact that $\hat{v} + \delta\hat{w} \leq 0$ and $Az \leq b$; the inequality (28) follows the fact that $\hat{\rho}_k(z_k - \hat{z}_k) \geq 0$ and $\hat{\gamma}_k(z_k - \hat{z}_k) \geq 0$, for every $k \in F$, which is consequence of the values of the reduced costs at optimality; the inequality (29) follows from (25); and finally the equality (30) is a consequence of (19). □

This result can be easily generalized to a situation in which a distinct t variable occurs for each constraint. This is in fact the usual procedure with most phase-one implementations of the Simplex algorithm for linear programs.

When solving a nonlinear program whose feasible region $\bar{P} \cap \{z \colon z_{F'} = 0, z_j = i\}$ is empty, most standard nonlinear programming algorithms are not ready to provide some point $(\hat{t}, \tilde{z}_F^i)$ that solves the following program

$$\begin{array}{ll}
\min & t \\
\text{s.t.} & \left\{ \begin{array}{l} G(z_F, 0) \leq te, \\ z_k \geq 0, k \in F \qquad z_j = i, \\ z_k \leq 1, k \in F \cap \{1, \ldots, p\}, \end{array} \right.
\end{array} \tag{31}$$

and in this way proving infeasibility. In fact, it may occur that what seems to be an infeasible problem is just a numerical difficulty of meeting the constraints to a desired accuracy. The solution of the program (31) provides a verification of infeasibility and, as saw in the proof of Proposition 3, the dual variables that may be required for the lifting procedure. Since the Slater condition holds, the optimal solution $(\hat{t}, \hat{z}_F^i)$ of Program (31) also solves the following linear program defined by a suitable matrix $A^i \in \partial G(\hat{z}_F^i, 0)$,

$$\begin{array}{ll}
\min & t \\
\text{s.t.} & \left\{ \begin{array}{l} G(\hat{z}_F^i, 0) + A^i(z_F - \hat{z}_F^i, 0) \leq te, \\ z_k \geq 0, k \in F, \qquad z_j = i, \\ z_k \leq 1, k \in F \cap \{1, \ldots, p\}, \end{array} \right.
\end{array} \tag{32}$$

Then, to complete the lifting procedure we just have to solve

$$\begin{array}{ll} \min & \hat{\xi}_F z_F \\ s.t. & \left\{ \begin{array}{l} G(\hat{z}^i_F, 0) + A^i(z_F - \hat{z}^i_F, 0) \leq \hat{t}e, \\ z_k \geq 0, k \in F, \qquad z_j = i, \\ z_k \leq 1, k \in F \cap \{1, \ldots, p\}, \end{array} \right. \end{array} \tag{33}$$

and, as explained in the proof of Proposition 3 we are now able to define $\hat{\xi}^i = (\hat{\xi}_F, \hat{\xi}^i_{F'}) \in \partial f(\hat{x})$ such that $\hat{\xi}^i z \geq \beta$, for every $z \in \bar{P} \cap \{z\colon z_j = i\}$. Since $z_{F'} \geq 0$, for every $z \in \bar{P}_j$, then again

$$\hat{\xi} = (\hat{\xi}_F, \max_{i=0,1}(\hat{\xi}^i_{F'}))$$

is a subgradient of the function f at $\hat{x}$ such that $\hat{\xi} x \geq \beta$, for every $x \in \bar{P}_j$. Moreover, since $\hat{\xi}\bar{x} < \beta$ then we have found a separating hyperplane.

References

[1] Egon Balas, Sebastián Ceria, and Gérard Cornuéjols. A lift-and-project cutting plane algorithm for mixed 0-1 programs. *Math. Programming*, 58(3, Ser. A):295–324, 1993.

[2] E. Balas, S. Ceria, and G. Cornuejols. Mixed 0-1 programming by lift-and-project in a branch-and-cut framework. *Management Science*, 42(9):1229–1246, Sep 1996.

[3] M. Duran and I. Grossmann. An outer-approximation algorithm for a class of mixed-integer nonlinear programs. *Mathematical Programming*, 36:307–339, 1986.

[4] C. Floudas. *Nonlinear and Mixed-integer Optimization.* Oxford University Press, 1995.

[5] B. Gavish. Topological design of computer communication networks - the overall design problem. *European Journal of Operational Research*, 58:149–172, 1992.

[6] J. Soares. *Disjunctive Convex Optimization.* PhD thesis, Graduate School of Business, Columbia Univeristy, Jun 1998.

[7] Robert A. Stubbs and Sanjay Mehrotra. A branch-and-cut method for 0-1 mixed convex programming. *Math. Program.*, 86(3, Ser. A):515–532, 1999.

[8] M. Turkay and I. Grossmann. Disjunctive programming techniques for the optimization of process systems with discontinuous investment costs - multiple size regions. *Industrial & Engineering Chemistry Research*, 35:2611–2623, 1996.