

2014

Cutting planes algorithm for convex Generalized Disjunctive Programs

Francisco Trespacios
Carnegie Mellon University

Ignacio E. Grossmann
Carnegie Mellon University, grossmann@cmu.edu

Follow this and additional works at: <http://repository.cmu.edu/cheme>

 Part of the [Chemical Engineering Commons](#)

This Working Paper is brought to you for free and open access by the Carnegie Institute of Technology at Research Showcase @ CMU. It has been accepted for inclusion in Department of Chemical Engineering by an authorized administrator of Research Showcase @ CMU. For more information, please contact research-showcase@andrew.cmu.edu.

Authors are encouraged to submit new papers to INFORMS journals by means of a style file template, which includes the journal title. However, use of a template does not certify that the paper has been accepted for publication in the named journal. INFORMS journal templates are for the exclusive purpose of submitting to an INFORMS journal and should not be used to distribute the papers in print or online or to submit the papers to another publication.

Cutting planes algorithm for convex Generalized Disjunctive Programs

Francisco Trespalcios

Carnegie Mellon University, ftrespal@andrew.cmu.edu,

Ignacio E. Grossmann

Carnegie Mellon University, grossmann@cmu.edu,

In this work, we propose a cutting plane algorithm to improve optimization models that are originally formulated as convex Generalized Disjunctive Programs (GDP). GDPs are traditionally reformulated as MINLPs using either the Big-M (BM) or the hull-reformulation (HR). The former yields a smaller MILP/MINLP while the later a tighter one. The (HR) reformulation can be further strengthened, by using the concept of basic step from disjunctive programming. The proposed algorithm uses the strengthened formulation to derive cuts for the Big-M formulation, generating a stronger formulation with small growth in problem size. We test the algorithm with several instances. The results show that the algorithm improves GDP convex models, in the sense of providing formulations with stronger continuous relaxations than the (BM) with few additional constraints. In general, the algorithm also leads to a reduction in the solution time of the problems.

Key words: generalize disjunctive programming; mixed-integer nonlinear programming

1. Introduction

The modeling of many real-world application requires use of mixed-integer nonlinear programming. In particular, convex MINLP problems seek to minimize a convex objective function over a feasible region that is convex when the integrality variables are relaxed as continuous. Process design(Duran and Grossmann 1986, Floudas 1995), layout problems(Sawaya 2006), financial modeling and electrical power management(Stubbs and Mehrotra 1999) are a few of the areas where convex MINLP has been successfully applied. There are, however, different ways of formulating MINLP models. The performance of the MINLP methods strongly depends on the size of the problem formulation and tightness of its continuous relaxation.

Methods to solve convex MINLP problems include branch and bound and branch and cut(Dakin 1965, Gupta and Ravindran 1985, Stubbs and Mehrotra 1999), outer-approximation(Duran and Grossmann 1986, Fletcher and Leyffer 1994, Abhishek et al. 2010), generalized Benders decomposition(Geoffrion 1972), LP/NLP based branch and bound(Quesada and Grossmann 1992) and extended cutting planes(Westerlund and Petersson 1995). For a comprehensive review of MINLP we refer the reader to the work by Belotti et al.(Belotti et al. 2013), and to the review article by Bonami et al.(Bonami et al. 2012) for convex MINLP.

An alternative way to represent MINLP problems is Generalized Disjunctive Programming (GDP)(Raman and Grossmann 1994). GDP involves not only algebraic equations, but also disjunctions and logic propositions. This higher level representation, which is of special importance in engineering(Trespalacios and Grossmann 2014b), allows us to exploit the logic structure of the problem to obtain better formulations.

Generalized Disjunctive Programming (GDP) is an alternative higher-level representation of these problems(Raman and Grossmann 1994) that involves not only algebraic equations, but also disjunctions and logic propositions in terms of Boolean and continuous variables. The modeling of problems through GDP allows a systematic approach for the formulation process(Grossmann and Trespalacios 2013). Special techniques to solve GDP problems include the disjunctive branch and bound(Lee and Grossmann 2005) and the logic based outer-approximation(Turkay and Grossmann 1996). However, GDPs are normally reformulated as MILP/MINLP(Nemhauser and Wolsey 1988, Lee and Grossmann 2000) to exploit the developments in these solvers(Grossmann 2002, Trespalacios and Grossmann 2014c). Big-M (BM) and Hull-Reformulation (HR) are the traditional MINLP reformulations of GDP. The former generates a smaller MILP/MINLP, while the latter generates a tighter one(Grossmann and Lee 2003, Vecchietti et al. 2003).

In addition to the BM and HR reformulations, it is possible to obtain formulations that are even stronger than the traditional HR. These stronger formulations are obtained by using the HR reformulation after applying a logic operation called basic step(Balas 1985). Basic steps can be applied sequentially to generate a hierarchy of relaxations, where the limiting case is given by the convex hull of the original GDP problem. The application of basic steps has proven to improve the tightness in linear GDP formulations(Sawaya and Grossmann 2012), and in general convex GDP formulations(Ruiz and Grossmann 2012).

The drawback in the application of this operation is the exponential growth of continuous variables and number of constraints. Recent work proposed a hybrid MINLP reformulation to exploit both the strength of the relaxation of the HR after basic steps, and the small problem size of the BM (Trespalcios and Grossmann 2014a).

In this work, we follow a different approach to exploit the advantages of having small but weak formulations (BM), and strong but larger formulations (HR after the application of some basic steps). In order to do this, we propose a cutting plane algorithm for convex GDP problems. The algorithm iteratively derives valid inequalities (or cutting planes) for the BM reformulations. These inequalities cut-off sections of the feasible region of the continuous relaxation of the BM, but they do not cut-off any valid region of the stronger formulation. Once the cuts stop having a relevant impact in the improvement of the relaxation, the final MINLP is generated by using the BM of the GDP and including all the generated cuts. This MINLP is then solved using traditional methods. In the proposed algorithm, the cutting plane methodology is used before a branch and bound method is applied. Therefore, it can be considered as a pre-processing of the problem in the GDP space. Figure 1 illustrates the proposed framework. It is important to note that, within the context of disjunctive branch and bound, these cuts could be derived at any node. The addition of this cuts would in fact yield a disjunctive branch and cut algorithm. However, the disjunctive branch and cut algorithm is out of the scope of this paper, and it is subject of future research.

This paper is organized as follows. Section 2 provides a high-level description of relevant concepts in GDP. Section 3 presents the proposed algorithm, which is illustrated in detailed with an example in Section 4. Section 5 presents convex GDP examples that were solved to test the algorithm. Section 6 presents the statistics, results and performance of different test examples.

2. Background

In this section we present the general form of GDP, and its (BM) and (HR) reformulations. We then present an overview on the logic operation called basic step. For a more comprehensive description of the topics we refer the reader to previous work that reviews generalized disjunctive programming (Grossmann and Trespalcios 2013).

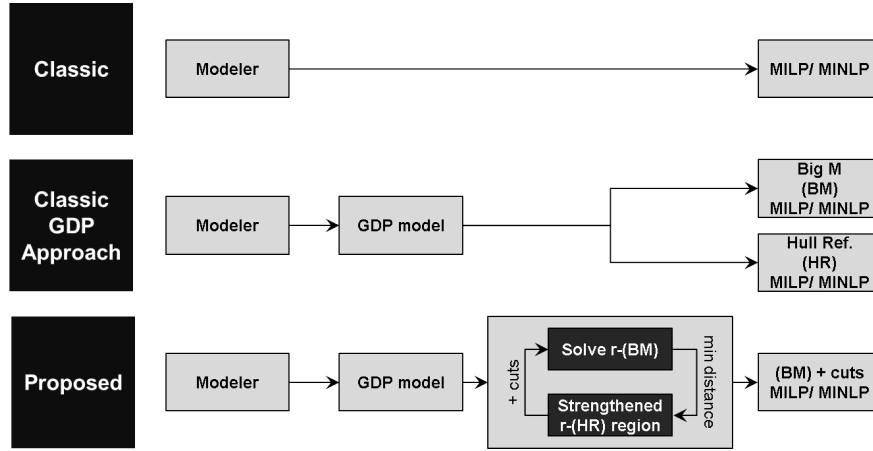


Figure 1 Different modeling approaches

2.1. Generalized disjunctive programming

The general GDP formulation is represented as follows:

$$\begin{aligned}
 & \min z = f(x) \\
 \text{s.t.} \quad & g(x) \leq 0 \\
 & \bigvee_{i \in D_k} \left[\begin{array}{c} Y_{ki} \\ r_{ki}(x) \leq 0 \end{array} \right] \quad k \in K \\
 & \bigvee_{i \in D_k} Y_{ki} \quad k \in K \quad (\text{GDP}) \\
 & \Omega(Y) = \text{True} \\
 & x^{lo} \leq x \leq x^{up} \\
 & x \in \mathbb{R}^n \\
 & Y_{ki} \in \{\text{True}, \text{False}\} \quad k \in K, i \in D_k
 \end{aligned}$$

The objective function $f(x)$ in (GDP) depends only on the continuous variables x . The global constraints $g(x)$ must hold true. (GDP) contains $k \in K$ disjunctions. Each disjunction, in turn, contains $i \in D_k$ disjunctive terms. Each of these terms is associated with a Boolean variable Y_{ki} and a set of constraints $r_{ki}(x) \leq 0$. The disjunctive terms in a disjunction are linked together by an OR operator (\vee). In each disjunction, only one disjunctive term can be selected $\bigvee_{i \in D_k} Y_{ki}$. The constraints associated to a selected (or active) term ($Y_{ki} = \text{True}$) are enforced. The constraints associated to a disjunctive term that is not active ($Y_{ki} = \text{False}$) are ignored. The logic relations among the disjunctive terms is represented by $\Omega(Y) = \text{True}$. In this work, we consider the particular case in which $f(x)$, $g(x)$ and $r_{ki}(x)$ are convex. In such a case, the problem is called a convex GDP.

GDP problems are normally reformulated as MILP/MINLP by using either the Big-M(Nemhauser and Wolsey 1988) (BM) or Hull Reformulation(Lee and Grossmann 2000) (HR). The (BM) reformulation is as follows:

$$\begin{aligned}
& \min z = f(x) \\
s.t. \quad & g(x) \leq 0 \\
& r_{ki}(x) \leq M^{ki}(1 - y_{ki}) \quad k \in K, i \in D_k \\
& \sum_{i \in D_k} y_{ki} = 1 \quad k \in K \\
& Hy \geq h \\
& x^{lo} \leq x \leq x^{up} \\
& x \in \mathbb{R}^n \\
& y_{ki} \in \{0, 1\} \quad k \in K, i \in D_k
\end{aligned} \tag{BM}$$

The (HR) formulation is given as follows:

$$\begin{aligned}
& \min z = f(x) \\
s.t. \quad & g(x) \leq 0 \\
& x = \sum_{i \in D_k} \nu^{ki} \quad k \in K \\
& y_{ki} r_{ki}(\nu^{ki}/y_{ki}) \leq 0 \quad k \in K, i \in D_k \\
& \sum_{i \in D_k} y_{ki} = 1 \quad k \in K \\
& Hy \geq h \\
& x^{lo} y_{ki} \leq \nu^{ki} \leq x^{up} y_{ki} \quad k \in K, i \in D_k \\
& x \in \mathbb{R}^n \\
& y_{ki} \in \{0, 1\} \quad k \in K, i \in D_k
\end{aligned} \tag{HR}$$

In both reformulations, the Boolean variables Y_{ki} are transformed into binary variables y_{ki} with a one-to-one correspondence ($Y_{ki} = True$ is equivalent to $y_{ki} = 1$, while $Y_{ki} = False$ is equivalent to $y_{ki} = 0$). The logic constraints ($\Omega(Y) = True$) can be transformed to linear constraints ($Hy \geq h$) using well-known methods(Clocksink and Mellish 1981, Williams 1985, Biegler et al. 1997). As previously described, only one disjunctive term in each disjunction can be selected ($\sum_{i \in D_k} y_{ki} = 1$).

The (BM) and (HR) differ in the representation of the set of equations inside the disjunctive terms. For the (BM), the constraints associated with a selected term ($y_{ki} = 1$)

are enforced ($r_{ki}(x) \leq 0$). For a term not selected ($y_{ki} = 0$) and a large enough M^{ki} , the corresponding constraints become redundant ($r_{ki}(x) \leq M^{ki}$).

In (HR), the continuous variables x are disaggregated. One additional continuous variable is created for each disjunctive term in each disjunction $\nu^{ki}, i \in D_k$. When a disjunctive term is selected ($y_{ki} = 1$), the variable corresponding to that term has to lie within the bounds of the original variable ($x^{lo} \leq \nu^{ki} \leq x^{up}$). When it is not selected, the disaggregated variable takes a value of 0. The original continuous variables have the same value as the disaggregated variables that correspond to a selected term ($x = \sum_{i \in D_k} \nu^{ki}$) in each of the disjunctions $k \in K$. The constraints associated to a disjunctive term are represented by the perspective function $y_{ki}r_{ki}(\nu^{ki}/y_{ki})$. When a term is selected, these constraints are enforced on the disaggregated variable ($r_{ki}(\nu^{ki}) \leq 0$). The constraints for a non-selected disjunctive term are trivially satisfied ($0 \leq 0$). Note that when the constraints corresponding to a disjunctive term are linear ($A^{ki}x \leq a^{ki}$), then $y_{ki}r_{ki}(\nu^{ki}/y_{ki}) \leq 0$ becomes $A^{ki}\nu^{ki} \leq a^{ki}y_{ki}$ (Balas 1985). In order to avoid singularities in the nonlinear case, the following approximation can be used for the perspective function (Sawaya 2006):

$$y_{ki}r_{ki}(\nu^{ki}/y_{ki}) \approx ((1 - \epsilon)y_{ki} + \epsilon)r_{ki} \left(\frac{\nu^{ki}}{(1 - \epsilon)y_{ki} + \epsilon} \right) - \epsilon r_{ki}(0)(1 - y_{ki}) \quad (\text{APP})$$

where ϵ is a small finite number (e.g. 10^{-4}). (APP) is exact at $y_{ki} = 0$ and $y_{ki} = 1$, and it is convex if r_{ki} is convex.

It is important to note that (HR) involves more variables and constraints than (BM). However, (HR) provides a stronger formulation (Grossmann and Lee 2003, Vecchietti et al. 2003). The (HR) reformulation represents the intersection of the convex hulls of each disjunction. This representation of convex hulls of disjunctions, using the perspective function, has been previously presented for convex MINLP (Ceria and Soares 1999, Stubbs and Mehrotra 1999).

2.2. Basic steps

Consider the following definitions from disjunctive convex programming:

Convex inequality: $C = \{x \in \mathbb{R}^n | \Phi(x) \leq 0\}$, where $\Phi(x) : \mathbb{R}^n \rightarrow \mathbb{R}^1$ is a convex function.

Convex set: $P = \bigcap_{m \in M} C_m$

Elementary disjunctive set: $H = \bigcup_{m \in M} C_m$

Disjunction: $S_k = \bigcup_{i \in D_k} P_i = \bigcup_{i \in D_k} \bigcap_{m \in M_i} C_m$

A disjunction such that $S_k = P_i$ for some $i \in D_K$ is called improper disjunction, otherwise it is called a proper disjunction. Note that if D_k is a singleton then S_k is improper.

There are alternative forms to represent disjunctive convex sets. In particular:

$$\text{Regular form: } F = \bigcap_{k \in K} S_k$$

$$\text{Disjunctive normal form (DNF): } F = S = \bigcup_{i \in D} P_i$$

A basic step is the intersection between two disjunctions to form a new disjunction. Basic steps bring a disjunctive set in regular form closer to its DNF. The definition of basic step is as follows (Balas 1985, Ruiz and Grossmann 2012):

THEOREM 2.1. Let F be a disjunctive set in regular form. Then F can be brought to DNF by $|K| - 1$ recursive applications of the following basic step which preserves regularity. For some $k, l \in K$, bring $S_k \cap S_l$ to DNF by replacing it with: $S_{kl} = S_k \cap S_l = \bigcup_{i \in D_k, j \in D_l} (P_i \cap P_j)$

Any convex GDP is equivalent to a disjunctive convex program (Ruiz and Grossmann 2012). Therefore, it is possible to use the concept of basic step to strengthen GDP formulations. In particular, there are two main consequences of the basic steps for GDP problems (Sawaya and Grossmann 2012, Ruiz and Grossmann 2012): a) The continuous relaxation of the (HR) of a disjunctive set after a basic step is at least as tight as the one before the basic step; and b) The (HR) of the DNF describes the convex hull of the problem

The results indicate that we can improve the strength of the continuous relaxation of the (HR) by applying basic steps. Furthermore, in the extreme case in which we intersect all of the disjunctions and all of the global constraints into a single disjunction, we obtain the convex hull of the problem. The drawback in the application of basic steps is the growth of the problem size, which is exponential when applying proper basic steps.

The application of proper basic steps not only increases the problem size, but also results in an exponential growth of disjunctive terms. As described in Section 2.1, each disjunctive term is associated with a binary variable in any GDP reformulation. Therefore, the growth of disjunctive terms implies an exponential increase in the number of binary variables. However, it is possible to avoid the exponential growth in binary variables by using the following theorem (Balas 1985):

THEOREM 2.2. Consider MILP/MINLP representation of two disjunctions $k, l \in K$, whose disjunctive terms are represented by the 0-1 variables $y_{ki}, y_{lj}, i \in D_k, j \in D_l$. If a basic step is applied between disjunction k and disjunction l , the variables representing the

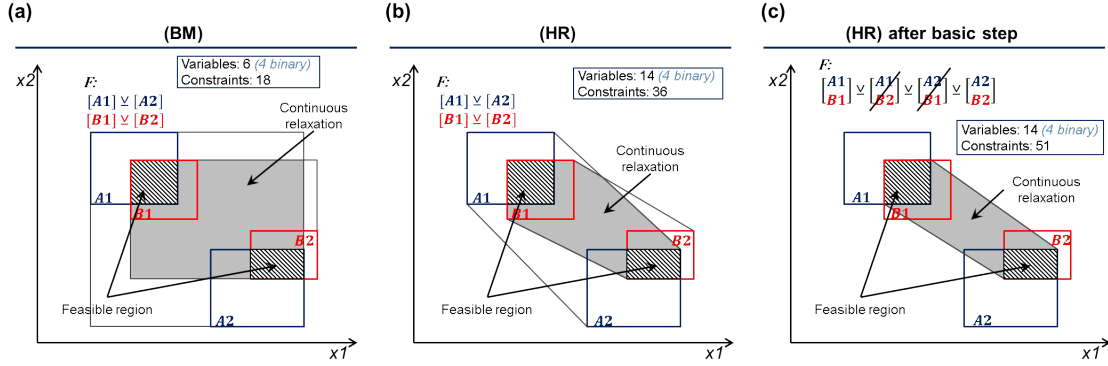


Figure 2 (a) Illustration of (BM), (b) (HR), and (c) (HR) after the application of a basic step.

disjunctive terms of the resulting disjunction $\hat{y}_{ij} \in \{0, 1\}$ can be equivalently represented by:

$$\begin{aligned}
 y_{ki} &= \sum_{j \in D_l} \hat{y}_{ij} & i \in D_k \\
 y_{lj} &= \sum_{i \in D_k} \hat{y}_{ij} & j \in D_l \\
 \sum_{i \in D_k, j \in D_l} \hat{y}_{ij} &= 1 \\
 \sum_{i \in D_k} y_{ki} &= 1 \\
 \sum_{j \in D_l} y_{lj} &= 1 \\
 0 \leq \hat{y}_{ij} &\leq 1 \\
 y_{ki}, y_{lj} &\in \{0, 1\} & i \in D_k, j \in D_l
 \end{aligned}$$

Proof. The proof follows from Theorem 4.4 of Balas (Balas 1985) ■.

Theorem 2.2 relates the new terms after a basic step to those before the basic step. Only the variables associated to the original disjunctive terms are required to be binary, while the ones related to the new terms can be continuous between 0 and 1.

Figure 2 illustrates tightness of relaxation of the (BM), (HR), and (HR) after the application of a basic step. Figure 2 presents the feasible region described by two disjunctions with two disjunctive terms each, that is $([A_1] \vee [A_2]) \wedge ([B_1] \vee [B_2])$. Figure 2.a shows the continuous relaxation of the (BM) reformulation projected into x_1 and x_2 . Figure 2.b shows the continuous relaxation of the (HR) reformulation projected into x_1 and x_2 . It is easy to see from the figure that the continuous relaxation of the (HR) has a tighter feasible region. Figure 2.c shows that the feasible region is even tighter after the application of a basic step.

The key element in the application of basic steps is the trade-off between the problem size and the tightness improvement. There has been some work that identifies heuristics

to decide when to apply basic steps (Balas 1985, Sawaya and Grossmann 2012, Ruiz and Grossmann 2012). Some recent work tries to use a hybrid (BM)-(HR) reformulation to avoid the exponential growth in problem size (Trespacios and Grossmann 2014a). This work presents an alternative method, in which the strength of the (HR) after basic steps is captured through cutting planes. Section 3 describes how this procedure is performed.

3. Cutting plane algorithm to improve GDP formulations

A method for using the strong extended reformulation, obtained through the application of basic steps, is to derive cutting planes for the (BM) reformulation (Stubbs and Mehrotra 1999, Vecchietti et al. 2003, Sawaya and Grossmann 2005). The main idea of the cutting planes method is to solve the continuous relaxation of the (BM) of the convex GDP, and use the strong formulation of the (HR) after basic steps to derive cutting planes. The cutting planes are determined by solving a separation problem (NLP), in which the feasible solution corresponds to the continuous relaxation of the (HR) formulation after a sequence of basic steps.

Without loss of generality, any convex GDP can be formulated as follows:

$$\begin{aligned}
& \min x_n \\
s.t. \quad & g(x) \leq 0 \\
& \bigvee_{i \in D_k} \left[\begin{array}{c} Y_{ki} \\ r_{ki}(x) \leq 0 \end{array} \right] & k \in K \\
& \bigvee_{i \in D_k} Y_{ki} & k \in K \\
& \Omega(Y) = True \\
& x^{lo} \leq x \leq x^{up} \\
& x \in \mathbb{R}^n \\
& Y_{ki} \in \{True, False\} \quad k \in K, i \in D_k
\end{aligned} \tag{1}$$

where, $g(x)$ and $r_{ki}(x)$ are convex functions.

Let (F-HR) be the feasible region of the continuous relaxation of the (HR) reformulation, and (P-HR) the projection of (F-HR) to the original space. Let (R-BM) be the continuous relaxation of the (BM) reformulation of (1). Let (F-BM) be the feasible region and $z^{BM} = (x^{BM}, y^{BM})$ the optimal solution of (R-BM). Also, it is possible to define the feasible region of the continuous relaxation (HR) of the GDP after application of basic steps with the following constraints:

$$g_e(x) \leq 0 \quad E \setminus \tilde{E} \quad (2a)$$

$$x = \sum_{i \in D_k} \nu^{ki} \quad k \in K \setminus \tilde{K} \quad (2b)$$

$$y_{ki} r_{ki} (\nu^{ki} / y_{ki}) \leq 0 \quad k \in K \setminus \tilde{K}, i \in D_k \quad (2c)$$

$$\sum_{i \in D_k} y_{ki} = 1 \quad k \in K \quad (2d)$$

$$Hy \geq h \quad (2e)$$

$$x^{lo} y_{ki} \leq \nu^{ki} \leq x^{up} y_{ki} \quad k \in K \setminus \tilde{K}, i \in D_k \quad (2f)$$

$$0 \leq y_{ki} \leq 1 \quad k \in K, i \in D_k \quad (2g)$$

$$x = \sum_{\hat{i} \in D_{\hat{k}}} \nu^{\hat{k}\hat{i}} \quad \hat{k} \in \hat{K} \quad (2h)$$

$$\hat{y}_{\hat{k}\hat{i}} g_e (\nu^{\hat{k}\hat{i}} / \hat{y}_{\hat{k}\hat{i}}) \leq 0 \quad \hat{k} \in \hat{K}, \hat{i} \in D_{\hat{k}}, e \in \hat{E}_{\hat{k}} \quad (2i)$$

$$\hat{y}_{\hat{k}\hat{i}} r_{ki} (\nu^{\hat{k}\hat{i}} / \hat{y}_{\hat{k}\hat{i}}) \leq 0 \quad \hat{k} \in \hat{K}, \hat{i} \in D_{\hat{k}}, ki \in \hat{K}I_{\hat{k}\hat{i}} \quad (2j)$$

$$\sum_{\hat{i} \in D_{\hat{k}}} \hat{y}_{\hat{k}\hat{i}} = 1 \quad \hat{k} \in \hat{K} \quad (2k)$$

$$x^{lo} \hat{y}_{\hat{k}\hat{i}} \leq \nu^{\hat{k}\hat{i}} \leq x^{up} \hat{y}_{\hat{k}\hat{i}} \quad \hat{k} \in \hat{K}, \hat{i} \in D_{\hat{k}} \quad (2l)$$

$$0 \leq \hat{y}_{\hat{k}\hat{i}} \leq 1 \quad \hat{k} \in \hat{K}, \hat{i} \in D_{\hat{k}} \quad (2m)$$

$$y_{ki} = \sum_{\substack{\hat{k} \in \hat{K}, \hat{i} \in D_{\hat{k}} \\ ki \in \hat{K}I_{\hat{k}\hat{i}}}} \hat{y}_{\hat{k}\hat{i}} \quad k \in \tilde{K}, i \in D_k \quad (2n)$$

The feasible region (SEP) is defined by constraints (2a) - (2n). (2a) - (2g) is the relaxed (HR) of the global constraints and disjunctions to which no basic steps were applied ($K \setminus \tilde{K}$ and $E \setminus \tilde{E}$). (2h) - (2m) is the relaxed (HR) of the global constraints and disjunctions in which basic steps were applied. In these constraints $\hat{k} \in \hat{K}$ are the resulting disjunctions after applying basic steps (that we denote “key disjunctions”), and $\hat{i} \in D_{\hat{k}}$ their corresponding disjunctive terms. The set $\hat{E}_{\hat{k}}$ maps the intersection of global constraints $e \in E$ with the new disjunctions $\hat{k} \in \hat{K}$. Note that a global constraint, which corresponds to an improper disjunction, can be intersected with more than one disjunction. Intersecting a global constraint with multiple disjunctions might provide further tightening of the relaxation in some cases. The set $\hat{K}I_{\hat{k}\hat{i}}$ maps the original disjunctive terms $ki, k \in K, i \in D_k$ to the resulting disjunctive terms after the application of basic steps $\hat{k} \in \hat{K}, \hat{i} \in D_{\hat{k}}$. Finally, constraint (2n) relates the original binary variables y_{ki} to the resulting ones $\hat{y}_{\hat{k}\hat{i}}$ as described in Theorem 2.2.

Let (P-SEP) be the projection of (SEP) over the original space: (P-SEP) = $Proj_x(\text{SEP})$. The following propositions allows us to derive valid cutting planes for the (BM) reformulation:

PROPOSITION 3.1. (P-SEP) \subseteq (F-BM)

Proof. For GDP (P-HR) \subseteq (F-BM) (Grossmann and Lee 2003). Also, after applying basic steps (P-SEP) \subseteq (P-HR) (Balas 1985, Ruiz and Grossmann 2012) ■.

PROPOSITION 3.2. (SEP) and (P-SEP) are convex regions.

Proof. 1) The original functions $g(x)$ and $r_{ki}(x)$ are convex. (SEP) contains either the original functions or the perspective function of the original functions. The perspective function is an operation that preserves convexity, therefore, (SEP) is convex. 2) (P-SEP) is convex, since it is the projection of a convex region, and projection preserves convexity ■.

Let $z = (x, y)$. In order to derive a separating hyperplane that cuts off a point z^{BM} , consider the two following separation problems:

$$\begin{aligned} \min \phi(z) &= \|z - z^{BM}\| \\ \text{s.t.} \quad & (z, \nu, \hat{y}) \in (SEP) \end{aligned} \tag{3}$$

and,

$$\begin{aligned} \min \phi(z) &= \|z - z^{BM}\| \\ \text{s.t.} \quad & z \in (P-SEP) \end{aligned} \tag{4}$$

(SEP) is convex, and (P-SEP) \subseteq (F-BM). Therefore, following propositions hold true (Sawaya 2006):

PROPOSITION 3.3. Let $(z^{sep}, \nu^{sep}, \hat{y}^{sep})$ be an optimal solution of (3). Then z^{sep} is an optimal solution of (4).

PROPOSITION 3.4. Let z^{BM} be the optimal solution of the continuous relaxation of the (BM) reformulation of (1), and z^{sep} an optimal solution of (3). If $z^{BM} \notin (P-SEP)$, then $\exists \xi$ such that $\xi^T(z - z^{sep}) \geq 0$ is a valid linear inequality in z^{sep} that cuts off z^{BM} , and such ξ is a subgradient of $\phi(x)$ at z^{sep} .

PROPOSITION 3.5. Let (SEP) \subset S, where S is a convex set. If $\phi : S \rightarrow \mathbb{R}$ is differentiable over its entire domain, then the collection of subgradients of ϕ at z^{sep} is the singleton set $\partial\phi \equiv \{\xi^{sep} | \xi^{sep} = \nabla\phi(z^{sep})\}$.

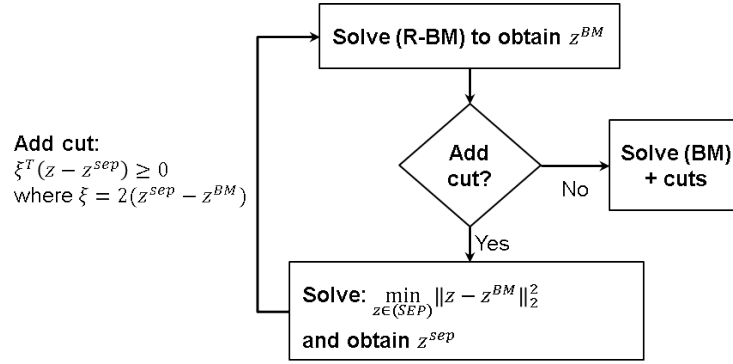


Figure 3 Outline of Algorithm 2

PROPOSITION 3.6. Let $(SEP) \subset S$, where S is a convex set. If $\phi : S \rightarrow \mathbb{R}$ is defined as $\phi(z) = \|z - z^{BM}\|_2^2$, then the collection of subgradients of ϕ at z^{sep} is the singleton set $\partial\phi \equiv \{\xi^{sep} | \xi^{sep} = 2(z^{sep} - z^{BM})\}$.

With these propositions, it is possible to derive cuts for (BM) using the (HR) reformulation after basic steps. The outline of the algorithm is shown in Figure 3. Figure 4 illustrates the algorithm in a simple example with three disjunctions, each one with two terms. There are several decisions and heuristics in the algorithm that have to be considered:

a) Number of cuts: A predetermined number of cuts can be established. The optimal value of the objective function in (3) can also be used as an indicator of the performance of the cuts.

b) The number of new resulting disjunctions, or “key disjunctions” has to be decided at each iteration.

c) The number of basic steps to apply in each “key disjunction” at each iteration.

d) Heuristics to select which disjunctions to intersect in each “key disjunction” at each iteration.

e) Selection of which global constraints to intersect with each disjunction. Includes generation of “redundant” constraints (i.e. intersecting a global constraint with more than one disjunction) at each iteration.

f) Selection of the norm in (3). In particular, the norm-2 squared is convenient for nonlinear convex GDP, but norm-1 or the infinity-norm might be computationally more convenient for linear GDP, since (3) then becomes linear.

In particular for b) - e), only a few heuristics have been developed to select intersection of disjunctions (Balas 1985, Sawaya and Grossmann 2012, Ruiz and Grossmann 2012, Trespacios and Grossmann 2014a). Important improvements in the algorithm could be

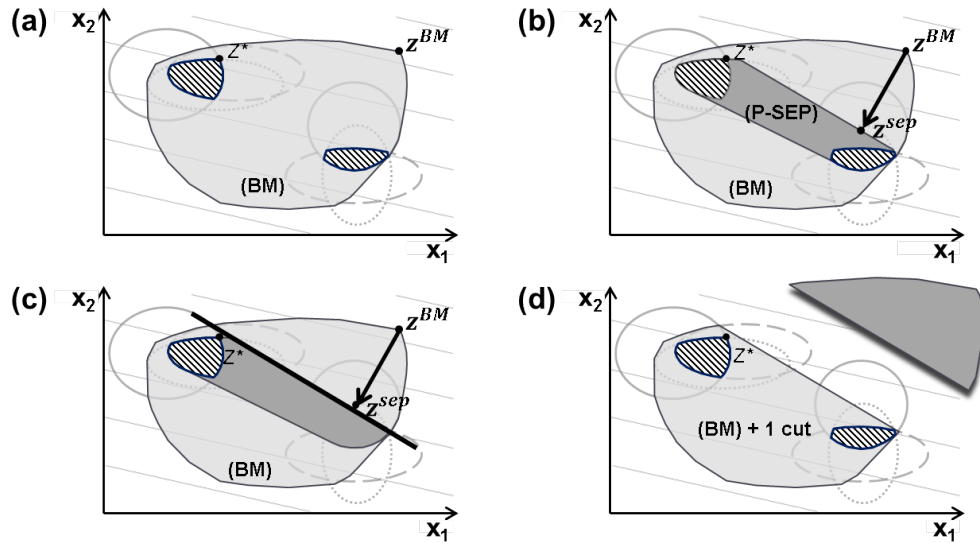


Figure 4 (a) Solution of the relaxation of the (BM) formulation. (b) Solution of (3). (c) Cutting plane $\xi(z - z^{sep}) \geq 0$. (d) Addition of cutting plane to (BM) formulation

achieved by using better heuristics. The development of these heuristics are out of the scope of this work.

It is important to note that in the proposed algorithm, the derived cutting planes are stronger than the ones proposed by Vecchietti et al (Vecchietti et al. 2003). These authors use the (HR) formulation to derive cuts for the (BM). In this work, the cuts are generated using the (HR) after the application of basic steps, so the separation problem (SEP) has a tighter feasible region.

4. Illustration of algorithm

To illustrate the algorithm, consider the simple convex GDP analytical example (5):

$$\begin{aligned}
& \min l \\
s.t. \quad & l \geq x_1 \\
& l \geq x_2 \\
& l \geq x_3 \\
& l \geq x_4 \\
& \left[\begin{array}{c} Y_{11} \\ x_1^2/50 - x_2 + 2 \leq 0 \end{array} \right] \vee \left[\begin{array}{c} Y_{12} \\ -x_1 + x_2^2/80 + 4 \leq 0 \end{array} \right] \\
& \left[\begin{array}{c} Y_{21} \\ x_1^2/60 - x_3 \leq 0 \end{array} \right] \vee \left[\begin{array}{c} Y_{22} \\ -x_1 + x_3^2/60 + 5 \leq 0 \end{array} \right] \\
& \left[\begin{array}{c} Y_{31} \\ x_1^2/60 - x_4 \leq 0 \end{array} \right] \vee \left[\begin{array}{c} Y_{32} \\ -x_1 + x_4^2/70 + 6 \leq 0 \end{array} \right] \\
& \left[\begin{array}{c} Y_{41} \\ x_2^2/60 - x_3 \leq 0 \end{array} \right] \vee \left[\begin{array}{c} Y_{42} \\ -x_2 + x_3^2/90 + 4 \leq 0 \end{array} \right] \\
& \left[\begin{array}{c} Y_{51} \\ x_2^2/70 - x_4 + 9 \leq 0 \end{array} \right] \vee \left[\begin{array}{c} Y_{52} \\ -x_2 + x_4^2/50 + 7 \leq 0 \end{array} \right] \\
& \left[\begin{array}{c} Y_{61} \\ x_3^2/90 - x_4 + 6 \leq 0 \end{array} \right] \vee \left[\begin{array}{c} Y_{62} \\ -x_3 + x_4^2/80 + 3 \leq 0 \end{array} \right] \\
& Y_{i1} \vee Y_{i2} \quad i = 1, \dots, 6 \\
& 3 \leq x_1 \leq 100; 0 \leq x_2 \leq 100; 3 \leq x_3 \leq 100; 0 \leq x_4 \leq 100 \\
& Y_{i1}, Y_{i2} \in \{True, False\} \quad i = 1, \dots, 6 \\
& x_1, x_2, x_3, x_4 \in \mathbb{R}
\end{aligned} \tag{5}$$

The optimal solution of the continuous relaxation of the (BM) of this problem is **3**, and of the (HR) is **3.94**. The optimal solution of this problem is **7**. Note that if the (BM) of this problem is solved with SBB from GAMS(Brooke et al. 1998) it takes 18 nodes to find and prove the optimal solution. Consider the feasible region described in (6):

$$\begin{aligned}
& \left[\begin{array}{c} Y_{11} \\ l \geq x_1 \\ x_1^2/50 - x_2 + 2 \leq 0 \end{array} \right] \vee \left[\begin{array}{c} Y_{12} \\ l \geq x_1 \\ -x_1 + x_2^2/80 + 4 \leq 0 \end{array} \right] \\
& \left[\begin{array}{c} Y_{21} \\ x_1^2/60 - x_3 \leq 0 \end{array} \right] \vee \left[\begin{array}{c} Y_{22} \\ -x_1 + x_3^2/60 + 5 \leq 0 \end{array} \right] \\
& \left[\begin{array}{c} Y_{31} \\ x_1^2/60 - x_4 \leq 0 \end{array} \right] \vee \left[\begin{array}{c} Y_{32} \\ -x_1 + x_4^2/70 + 6 \leq 0 \end{array} \right] \\
& \left[\begin{array}{c} Y_{41} \\ x_2^2/60 - x_3 \leq 0 \end{array} \right] \vee \left[\begin{array}{c} Y_{42} \\ -x_2 + x_3^2/90 + 4 \leq 0 \end{array} \right] \\
& \left[\begin{array}{c} \hat{Y}_{11} \\ l \geq x_2 \\ l \geq x_3 \\ l \geq x_4 \\ x_2^2/70 - x_4 + 9 \leq 0 \\ x_3^2/90 - x_4 + 6 \leq 0 \end{array} \right] \vee \left[\begin{array}{c} \hat{Y}_{12} \\ l \geq x_2 \\ l \geq x_3 \\ l \geq x_4 \\ x_2^2/70 - x_4 + 9 \leq 0 \\ -x_3 + x_4^2/80 + 3 \leq 0 \end{array} \right] \vee \left[\begin{array}{c} \hat{Y}_{13} \\ l \geq x_2 \\ l \geq x_3 \\ l \geq x_4 \\ -x_2 + x_4^2/50 + 7 \leq 0 \\ x_3^2/90 - x_4 + 6 \leq 0 \end{array} \right] \vee \left[\begin{array}{c} \hat{Y}_{14} \\ l \geq x_2 \\ l \geq x_3 \\ l \geq x_4 \\ -x_2 + x_4^2/50 + 7 \leq 0 \\ -x_3 + x_4^2/80 + 3 \leq 0 \end{array} \right] \\
& Y_{i1} \vee Y_{i2} \quad i = 1, \dots, 4 \\
& \hat{Y}_{12} \vee \hat{Y}_{13} \vee \hat{Y}_{14} \\
& 3 \leq x_1 \leq 100; 0 \leq x_2 \leq 100; 3 \leq x_3 \leq 100; 0 \leq x_4 \leq 100 \\
& Y_{i1}, Y_{i2} \in \{True, False\} \quad i = 1, \dots, 4 \\
& \hat{Y}_{12}, \hat{Y}_{13}, \hat{Y}_{14} \in \{True, False\} \\
& x_1, x_2, x_3, x_4 \in \mathbb{R}
\end{aligned} \tag{6}$$

Note that (6) represents the feasible region of (5) after the following basic steps: the first global constraint is intersected with the first disjunction, a basic step is performed with disjunctions 5 and 6; the remaining global constraints are intersected with the disjunction that resulted from the basic step between disjunction 5 and 6.

Also, the constraints that relate the original variables y to the new variables \hat{y} , described in Theorem 2.2, are presented in (7).

$$\begin{aligned}
y_{51} &= \hat{y}_{11} + \hat{y}_{12} \\
y_{52} &= \hat{y}_{13} + \hat{y}_{14} \\
y_{61} &= \hat{y}_{11} + \hat{y}_{13} \\
y_{62} &= \hat{y}_{12} + \hat{y}_{14}
\end{aligned} \tag{7}$$

Let $z = [l, x_1, x_2, x_3, x_3, y_{11}, y_{12}, y_{21}, y_{22}, y_{31}, y_{32}, y_{41}, y_{42}, y_{51}, y_{52}, y_{61}, y_{62}]^T$

Step 1.

The relaxation of the (BM) reformulation of (5) is solved. The following solution is obtained:

$$z^{BM} = [3, 3, 0, 3, 0, 0.001, 0.999, 0.002, 0.998, 0.003, 0.997, 0.004, 0.996, 0.007, 0.993, 0, 1]^T$$

Step 2.

The separation problem is solved by minimizing $\Phi(z) = \|z - z^{BM}\|_2^2$:

$$\begin{aligned}
\Phi(z) &= (l - 3)^2 + (x_1 - 3)^2 + (x_2 - 0)^2 + (x_3 - 3)^2 + (x_4 - 0)^2 + (y_{11} - 0.001)^2 + (y_{21} - \\
&0.999)^2 + (y_{21} - 0.002)^2 + (y_{22} - 0.998)^2 + (y_{31} - 0.003)^2 + (y_{32} - 0.997)^2 + (y_{41} - 0.004)^2 + \\
&(y_{42} - 0.996)^2 + (y_{51} - 0.007)^2 + (y_{52} - 0.993)^2 + (y_{61} - 0)^2 + (y_{62} - 1)^2
\end{aligned}$$

Subject to the continuous relaxation of the (HR) of (6) and to (7). The solution of this separation problem is: $z^{SEP} = [7.5, 3, 5.2, 3, 2.3, 1, 0, 1, 0, 1, 0, 1, 0, 0.3, 0.7, 0.3, 0.7]^T$.

Cut generation.

Since $\Phi(z) = \|z - z^{BM}\|_2^2$, then $\xi^{sep} = 2(z^{sep} - z^{BM})$:

$$\xi^{sep} = [9, 0, 10.4, 0, 4.7, 2, -2, 2, -2, 2, -2, 2, -2, 0.5, -0.5, 0.5, -0.5].$$

The following cut is then added to (BM):

$$\begin{aligned}
9(l - 7.5) + 0(x_1 - 3) + 10.4(x_2 - 5.2) + 0(x_3 - 3) + 4.7(x_4 - 2.3) + 2(y_{11} - 1) - 2y_{21} + \\
2(y_{21} - 1) - 2y_{22} + 2(y_{31} - 1) - 2y_{32} + 2(y_{41} - 1) - 2y_{42} + 0.5(y_{51} - 0.3) - 0.5(y_{52} - 0.7) + \\
0.5(y_{61} - 0.3) - 0.5(y_{62} - 0.7) \geq 0
\end{aligned}$$

The continuous relaxation of the (BM) with this additional cut is **6.01**.

Iteration.

This procedure can be repeated with the solution of the (BM) relaxation after adding the cut. With a second cut the continuous relaxation becomes **6.79**, and with a third iteration it becomes **6.96**. Solving the MINLP by doing a (BM) reformulation and adding the three cuts just described, takes SBB **5** nodes (in contrast to solving the (BM) reformulation without the cuts, which requires **18** nodes).

5. Convex GDP examples

The algorithm was tested with 19 instances of four problems: constrained layout, farm layout, and two types of process flowsheets.

5.1. Constrained Layout (C-Lay)

In the constrained layout problem there is a set of rectangles that have to be packed inside a set of fixed circles. The objective function is to minimize the distance in x and y axis, with a cost associated to every pair of rectangles. Figure 5 illustrates the constrained layout problem. It can be formulated as the following convex GDP(Sawaya 2006):

$$\begin{aligned}
\min Z &= \sum_i \sum_j c_{ij} (delx_{ij} + dely_{ij}) \\
s.t. \quad delx_{ij} &\geq x_i - x_j && i, j \in N, i < j \\
delx_{ij} &\geq x_j - x_i && i, j \in N, i < j \\
dely_{ij} &\geq y_i - y_j && i, j \in N, i < j \\
dely_{ij} &\geq y_j - y_i && i, j \in N, i < j \\
\left[\begin{array}{c} Y_{ij}^1 \\ x_i + L_i/2 \leq x_j - L_j/2 \end{array} \right] \vee & \left[\begin{array}{c} Y_{ij}^2 \\ x_j + L_j/2 \leq x_i - L_i/2 \end{array} \right] \\
\vee \left[\begin{array}{c} Y_{ij}^3 \\ y_i + H_i/2 \leq y_j - H_j/2 \end{array} \right] \vee & \left[\begin{array}{c} Y_{ij}^4 \\ y_j + H_j/2 \leq y_i - H_i/2 \end{array} \right] && i, j \in N, i < j \\
\bigvee_{t \in T} \left[\begin{array}{c} W_{it} \\ (x_i + L_i/2 - xc_t)^2 + (y_i + H_i/2 - yc_t)^2 \leq r_t^2 \\ (x_i + L_i/2 - xc_t)^2 + (y_i - H_i/2 - yc_t)^2 \leq r_t^2 \\ (x_i - L_i/2 - xc_t)^2 + (y_i + H_i/2 - yc_t)^2 \leq r_t^2 \\ (x_i - L_i/2 - xc_t)^2 + (y_i - H_i/2 - yc_t)^2 \leq r_t^2 \end{array} \right] && i \in N \\
Y_{ij}^1 \vee Y_{ij}^2 \vee Y_{ij}^3 \vee Y_{ij}^4 && i, j \in N, i < j \\
\bigvee_{t \in T} W_{it} && i \in N \\
0 \leq x_i \leq x_i^{up} && i \in N \\
0 \leq y_i \leq y_i^{up} && i \in N \\
Y_{ij}^1, Y_{ij}^2, Y_{ij}^3, Y_{ij}^4 \in \{True, False\} && i, j \in N, i < j \\
W_{it} \in \{True, False\} && i \in N, t \in T
\end{aligned} \tag{8}$$

In formulation (8) x_i and y_i represent the coordinates of the centre of the rectangles $i \in N$. $delx_{ij}$ and $dely_{ij}$ represent the distance between two rectangles $i, j \in N, i < j$, and

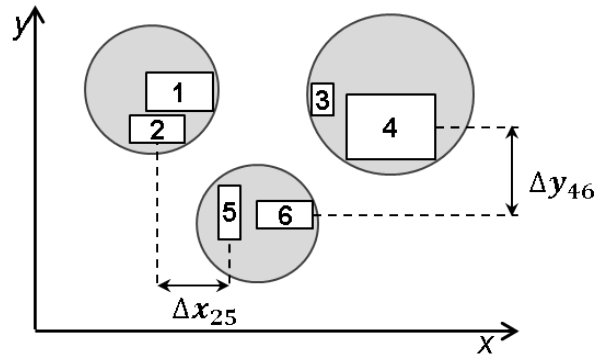


Figure 5 Illustration of constrained layout problem

c_{ij} is the cost associated with these. The first disjunctions ensures that there is no overlap by expressing the possible relative position between rectangles i and j . The second set of disjunctions ensure that every rectangle i is inside one of the $t \in T$ circles. For a circle t , its coordinates (x_{c_t}, y_{c_t}) and its radius r_t are given.

5.2. Farm Layout (F-Lay)

In the farm layout problem, the objective is to determine the width and length of a number of rectangles with fixed area in order to minimize the total perimeter. Figure 6 illustrates this problem, which can be formulated as the following convex GDP (Sawaya 2006):

$$\begin{aligned}
 & \min Z = 2(\text{Length} + \text{Width}) \\
 \text{s.t.} \quad & \text{Length} \geq x_i + L_i && i \in N \\
 & \text{Width} \geq y_i + W_i && i \in N \\
 & A_i/W_i - L_i \leq 0 && i \in N \\
 & \left[\begin{array}{c} Y_{ij}^1 \\ x_i + L_i \leq x_j \end{array} \right] \vee \left[\begin{array}{c} Y_{ij}^2 \\ x_j + L_j \leq x_i \end{array} \right] \vee \left[\begin{array}{c} Y_{ij}^3 \\ y_i + W_i \leq y_j \end{array} \right] \vee \left[\begin{array}{c} Y_{ij}^4 \\ y_j + W_j \leq y_i \end{array} \right] && i, j \in N, i < j \\
 & Y_{ij}^1 \vee Y_{ij}^2 \vee Y_{ij}^3 \vee Y_{ij}^4 && i, j \in N, i < j \\
 & 0 \leq \text{Length} \leq \text{Length}^{up}; \quad 0 \leq \text{Width} \leq \text{Width}^{up} \\
 & L_i^{lo} \leq L_i \leq L_i^{up}; \quad W_i^{lo} \leq W_i \leq L_i^{up} && i \in N \\
 & 0 \leq x_i \leq \text{Length}^{up} - L_i^{lo}; \quad 0 \leq y_i \leq \text{Width}^{up} - L_i^{lo} && i \in N \\
 & Y_{ij}^1, Y_{ij}^2, Y_{ij}^3, Y_{ij}^4 \in \{\text{True}, \text{False}\} && i, j \in N, i < j \\
 & && (9)
 \end{aligned}$$

In formulation (9) the variables x_i and y_i represent the coordinates of lower-left corner of each rectangle $i \in N$, while L_i and W_i represent their corresponding length and width.

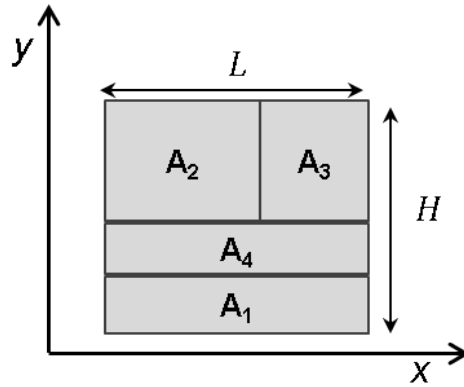


Figure 6 Illustration of farm layout problem

Length and *Width* represent the length and width of the total area. A_i is the given area for each rectangle. There is one disjunction for each pair of rectangles. Each term in the disjunction represents the possible relative position between the two rectangles: rectangle i is either to the left, or to the right, or below, or above rectangle j , respectively.

5.3. Process Network type 1 (Proc-1)

This problem seeks to maximize the profit of selling a set of products, taking into account the cost of raw materials and equipment. In this example the model that describes the performance is simplified to single input-output relations that give rise to a convex GDP (Ruiz and Grossmann 2012). The GDP problem formulation is as follows:

$$\begin{aligned}
 \min Z &= \sum_{i \in I} c_i + \sum_{j \in J} p_j x_j + \alpha \\
 \text{s.t.} \quad & \sum_{j \in J} r_{jn} x_j \leq 0 \quad \forall n \in N \\
 & \bigvee_{i \in D_k} \left[\begin{array}{c} Y_{ki} \\ \sum_{j \in J^i} d_{kij} (e^{x_j/t_{kij}} - 1) - \sum_{j \in J^i} s_{kij} x_j \leq 0 \\ c_i = \gamma_{ki} \end{array} \right] \quad k \in K \\
 & \Omega(Y) = True \\
 & c_i, x_j \geq 0 \\
 & Y_i \in \{True, False\}
 \end{aligned} \tag{10}$$

In (10) there are $k \in K$ processes that need to be installed. For each process, only one of the alternative units $i \in D_k$ must be selected. c_i is the cost associated to each process. x_j represents each of the flows $j \in J$, and p_j the profit or cost associated to each one.

The global constraints represent the mass balance in each of the $n \in N$ nodes, where r_{jn} is the coefficient of the mass balance for flow j . If a unit is selected ($Y_{ki} = True$) then the corresponding mass balance has to be satisfied, and the cost of the unit c_i takes the value associated to that equipment γ_i . Finally $\Omega(Y) = True$ represents the topology of the superstructure. Note that for a some processes one of the alternatives might be to select no units. In such cases one of the terms $i \in D_k$ will have $c_i = 0$, and $x_j = 0 \forall j \in J^i$ constraints associated with it.

5.4. Process Network type 2 (Proc-2)

This problem is similar to (Proc-1). The main difference is that in this case more than one alternative can be selected for each process. The formulation is presented in (11).

$$\begin{aligned}
\min Z &= \sum_{i \in I} c_i + \sum_{j \in J} p_j x_j + \alpha \\
s.t. \quad & \sum_{j \in J} r_{jn} x_j \leq 0 && \forall n \in N \\
& \left[\begin{array}{c} Y_i \\ \sum_{j \in J^i} d_{ij} (e^{x_j/t_{ij}} - 1) - \sum_{j \in J^i} s_{ij} x_j \leq 0 \\ c_i = \gamma_i \end{array} \right] \vee \left[\begin{array}{c} \neg Y_i \\ x_j = 0 \forall j \in J^i \\ c_i = 0 \end{array} \right] && i \in I \quad (11) \\
& \Omega(Y) = True \\
& c_i, x_j \geq 0 \\
& Y_i \in \{True, False\}
\end{aligned}$$

6. Results

In this section we present the computational results of applying the algorithm described in section 3 to different problems. The algorithm applies basic steps in a “key disjunction” following the heuristics presented by Trespalacios and Grossmann (Trespalacios and Grossmann 2014a), as long as the “key disjunction” contains less than 10 disjunctive terms (before the basic step). Four different strategies were tested for the number of “key disjunctions”. In strategy *K0* no proper basic steps were applied. In *K1* there is only one “key disjunction”. In *K5* five “key disjunctions” were generated. In *KK* there are as many key disjunctions as the instance allows. Two strategies were tested for the basic steps with the global constraints. Strategy *I1*, where all global constraints are intersected once with a disjunction (as long as they share variables in common, “key disjunctions” where

preferred over regular ones). In strategy *I2* all global constraints are intersected with all of the disjunctions with which they share variables in common. For this work, the selection of disjunctions and basic steps does not change after each iteration, but in general it is possible to do so. Finally, the algorithm was tested by generating 1, 2, 3, 5, 7 and 10 cuts.

Nineteen nonlinear convex instances were solved for the problems presented in section 5. The instances were generated by defining problem size and structure, and randomly generating the parameters of the problems. The ϵ has a value of 10^{-4} , and the Big M parameter was determined using by estimating a basic feasible solution of the problem. The problem size and solution of this instances is presented in Table 1. The instances are solved using SBB. The algorithm and models were implemented in GAMS 24.2(Brooke et al. 1998) and solved in an Intel(R) Core(TM) i7 CPU 2.93 GHz and 4 GB of RAM.

This section first presents three plots on the general performance of the algorithm. It then provides in-depth tables with the statistics and computational results of the different problems and strategies.

Figure 7 shows the percentage of problems solved vs. time for the (BM), (HR) and the algorithm, using the strategy *K5 – I1*, and using 3 cuts. The time for the algorithm includes the time to generate the cuts and time to solve the MINLP. The figure shows that the algorithm, using strategy *K5 – I1* and 3 cuts, performs considerably better than the direct (BM) and (HR) reformulations.

Figure 8 shows the relaxation gap for the (BM), (HR) and the algorithm for the different strategies and different numbers of cuts. It is important to note that after one single cut, the formulation presents a stronger relaxation than the (HR). This is an important result, considering that the new formulation has the same number of variables than the (BM), and just an additional constraint. It can also be seen that after the first cut, there is a small improvement in the relaxation. In terms of strategies, intersecting global constraints with every disjunction with which they share variables provides the best continuous relaxation. Note that this strategy involves generating redundant constraints. With this strategy *I2*, and using the proposed heuristics, the number and strategy for proper basic steps has small impact in the continuous relaxation(strategies *K0 – I2, K1 – I2, K5 – I2, KK – I2*). However, when the global constraints are intersected only once, the use of proper basic steps helps to improve the continuous relaxation (e.g. strategy *K0 – I1* has a weaker continuous

Table 1 Number of constraints and variables for the test problems

Instance	Solution	Binary	(BM)		(HR)	
			Cont. vars.	# constraints	Cont. vars.	# constraints
C-Lay-4-2	8,469	32	22	93	134	349
C-Lay-4-3	9,746	36	22	109	142	381
C-Lay-4-4	7,923	40	22	125	150	413
C-Lay-4-5	8,781	0	66	141	202	445
C-Lay-5-2	11,472	50	32	138	212	548
C-Lay-5-3	20,799	55	32	158	222	569
C-Lay-5-4	10,876	60	32	178	232	608
C-Lay-5-5	9,223	65	32	198	242	668
F-Lay-04	20	24	20	45	212	381
F-Lay-05-a	68	40	24	68	349	633
F-Lay-05-b	63	40	24	68	349	633
F-Lay-05-c	57.5	40	24	68	349	633
Proc-1-21	17.2	21	21	56	84	137
Proc-1-31	12.2	41	36	102	159	255
Proc-1-36	12.1	46	36	112	174	280
Proc-1-48	12.1	61	45	149	228	371
Proc-2-21	17	42	27	125	169	341
Proc-2-31	12.3	82	46	235	336	675
Proc-2-36	12.1	92	46	260	376	760

relaxation than strategy $K1 - I1$). It is important to note that different heuristics in the application of basic steps will impact this behavior.

Figure 9 shows the accumulated solution time to solve all instances, using different strategies, with different number of cuts. The maximum time allowed was 7,200 seconds. The figure shows that the solution time decreases with the first 3 cuts or so, but it starts increasing after that. This behaviour is expected, since the first cuts reduce the solution time of the MINLP considerably. However, as the number of cuts increases, the solution time of the MINLP does not improve much, and the time to generate the cuts becomes relatively more expensive. It is interesting to note that, even though the relaxation after

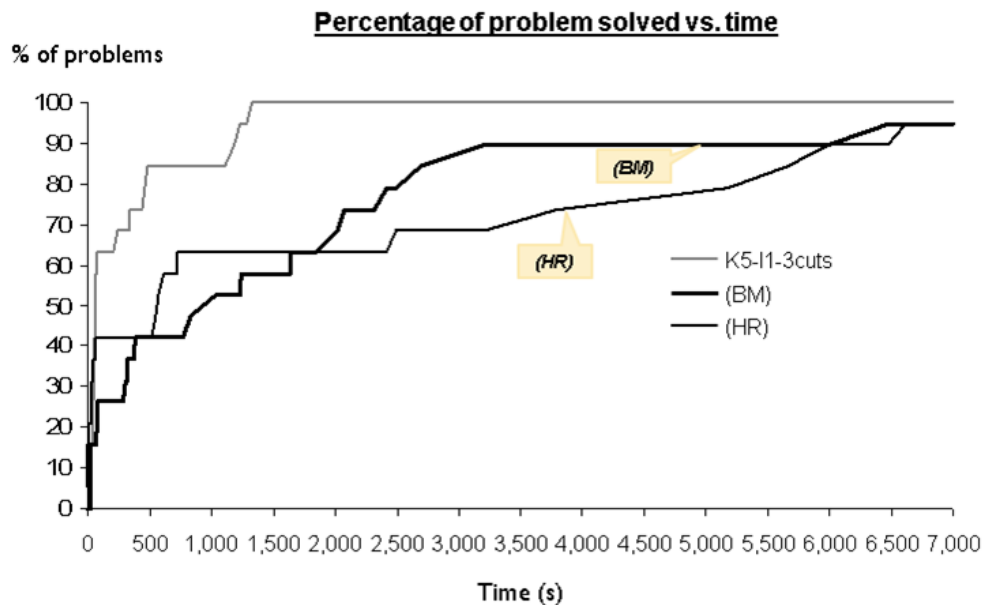


Figure 7 Percentage of problems solved vs. time

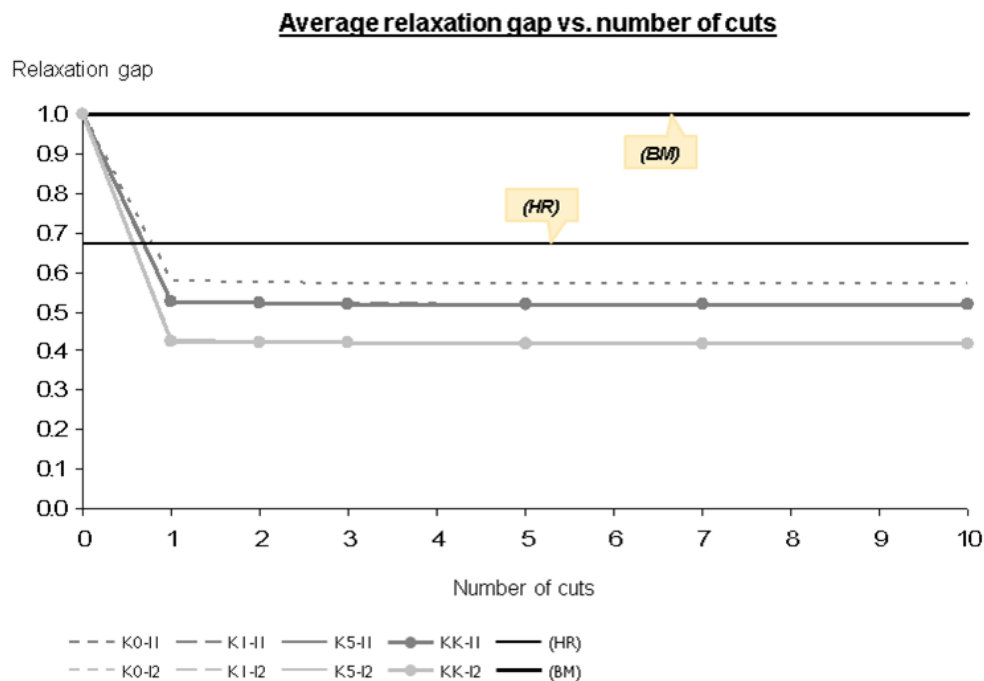


Figure 8 Average relaxation gap vs. number of cuts for different strategies in the algorithm

the first cut does not change much, generating around 3 cuts seems to be the best strategy for the tested problems. As expected, in the strategies that generate large problem sizes ($K5 - I2$ and $KK - I2$) the time to generate cuts is expensive, so the solution times after 3 cuts increases considerable.

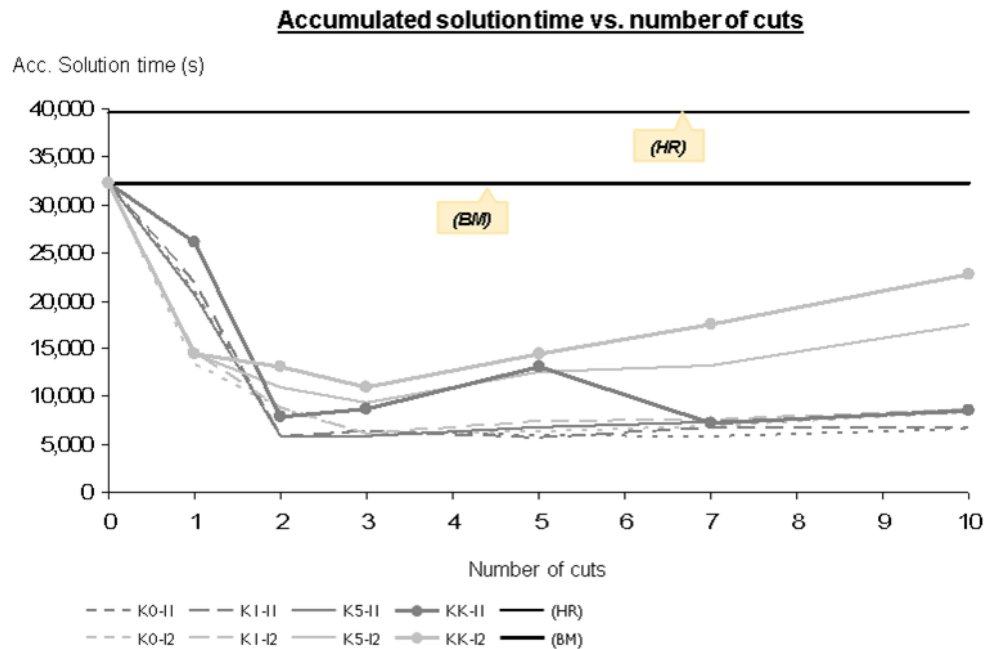


Figure 9 Accumulated solution time vs. number of cuts for different strategies in the algorithm

Table 2 summarizes the performance of the (BM), (HR), and of the algorithm for strategy $K5-I1$ with three cuts. It is easy to see that the relaxation of the MINLP after adding the cutting planes is stronger than the (HR) in most cases. The algorithm has the strongest continuous relaxation in 13 of the 19 instances, while the (HR) is the strongest in 2. Note that in the 2 instances in which (HR) is stronger, the value relaxation of the problem after the cutting planes is very close to that of the (HR). On the C-lay problems, the algorithm provides a value of the continuous relaxation much stronger than either the (BM) or the (HR). In terms of solution times, the algorithm is the fastest in 14 of the 19 instances, the (HR) in 4, and the (BM) in 1. Except in Proc-1-21, in all other instances in which the algorithm is not the fastest it is the second fastest. It is also important to notice that the algorithm solves all of the problems in less than 1,325 seconds, while the (BM) and (HR) cannot solve all of the problems within the two-hour limit.

Table 3 presents the separation problem size for the different strategies, compared to the (BM) reformulation. It is easy to see that, in general, strategy $I2$ generates a much larger problems than strategy $I1$. The difference in size lies in the generation of the redundant global constraints that are intersected with every single disjunction. In most cases, $K1$ is twice the size of $K0$, while $k5$ is thrice its size. It is important to note that in some

Table 2 Performance of the algorithm for strategy $K5 - I1$ and 3 cuts

		Continuous relaxation			Solution time (s)				
Instance	Solution	(BM)	(HR)	Algo.	(BM)	(HR)	Algorithm		
							cut-gen	MINLP	Total
C-Lay-4-2	8,469	0	0	2,877	24	34	8.3	9.6	18
C-Lay-4-3	9,746	0	0	2,774	74	31	16.8	7.2	24
C-Lay-4-4	7,923	0	0	2,788	312	548	25.6	36.7	62
C-Lay-4-5	8,781	0	0	3,384	318	727	35.6	39.9	76
C-Lay-5-2	11,472	0	0	5,187	383	574	20.8	48.3	69
C-Lay-5-3	20,799	0	0	4,826	2,696	2,498	27.4	27	54
C-Lay-5-4	10,876	0	0	4,417	2,075	5,175	34.2	306	340
C-Lay-5-5	9,223	0	0	3,680	6,477	> 7,200	59.5	429	488
F-Lay-04	20	0	11.9	15.2	27	48	22.6	21.8	44
F-Lay-05-a	68	0	39.8	51.7	2,019	6,617	70.1	1,255	1,325
F-Lay-05-b	63	0	35.3	45.3	1,247	5,660	53.7	1,138	1,192
F-Lay-05-c	57.5	0	31.6	42.5	1,642	6,000	73.4	1,156	1,229
Proc-1-21	17.2	0	15.7	15.6	24	4.7	20.9	20.3	41
Proc-1-31	12.2	0	12.2	12.2	840	1	31.6	1.3	33
Proc-1-36	12.1	0	12.1	12.1	2,414	5	62.2	1.1	63
Proc-1-48	12.1	0	12.1	12	> 7,200	10	44.9	13.4	58
Proc-2-21	17	0	0.3	0.3	76	71	3.9	48.3	52
Proc-2-31	12.3	0	0.2	0.2	1,030	616	5.4	240.4	246
Proc-2-36	12.1	0	0.1	0.2	2,770	3,791	5.2	369.1	374

instances strategy $K5$ and KK provide the same separation problem. This happens when strategy $K5$ applies all of the possible basic steps.

7. Conclusion

In this paper, we have proposed a cutting plane algorithm that improves the relaxation of the (BM) reformulation of convex GDP formulation. The cutting planes for the (BM) are derived through a separation problem. The separation problem minimizes the distance between the optimal solution of the continuous relaxation of the (BM), and a point that

Table 3 Ratios of problem size of (SEP) compared to (BM)

Instance	Ratio of number of constraints								Ratio of number of variables							
	Strategy I1				Strategy I2				Strategy I1				Strategy I2			
	<i>K0</i>	<i>K1</i>	<i>K5</i>	<i>KK</i>	<i>K0</i>	<i>K1</i>	<i>K5</i>	<i>KK</i>	<i>K0</i>	<i>K1</i>	<i>K5</i>	<i>KK</i>	<i>K0</i>	<i>K1</i>	<i>K5</i>	<i>KK</i>
C-Lay-4-2	6.1	12	18	18	18	23	34	34	12	23	33	33	29	37	55	55
C-Lay-4-3	5.5	11	17	17	16	21	34	34	13	23	35	35	32	40	64	64
C-Lay-4-4	5.1	9.9	16	16	16	19	36	36	14	24	39	39	35	43	76	76
C-Lay-4-5	4.8	9	17	17	15	18	39	39	4.7	8.2	14	14	12	15	31	31
C-Lay-5-2	6.7	13	18	19	25	30	46	49	14	24	32	34	40	49	73	78
C-Lay-5-3	6	12	16	18	23	28	44	51	14	25	35	38	43	52	83	95
C-Lay-5-4	5.6	11	15	17	22	26	44	53	15	25	37	42	46	55	92	110
C-Lay-5-5	5.4	9.9	15	17	21	25	44	55	16	26	39	46	49	58	101	125
F-Lay-04	12	21	33	33	17	26	43	43	15	25	40	40	18	27	46	46
F-Lay-05-a	12	18	33	33	18	24	48	48	20	28	52	52	25	32	63	63
F-Lay-05-b	12	18	33	33	18	24	48	48	20	28	52	52	25	32	63	63
F-Lay-05-c	12	18	33	33	18	24	48	48	20	28	52	52	25	32	63	63
Proc-1-21	5.4	10	13	13	12	15	19	19	11	20	25	25	22	29	37	37
Proc-1-31	5.7	9.6	17	17	20	23	39	39	12	22	35	35	42	49	88	88
Proc-1-36	5.6	9.1	19	19	20	22	48	48	13	23	43	43	47	54	119	119
Proc-1-48	5.6	9	13	18	25	27	36	51	14	23	33	42	62	69	94	135
Proc-2-21	4.3	4.6	10	11	13	13	19	19	12	14	27	29	35	35	53	55
Proc-2-31	3.8	4.1	6.5	10	20	20	23	28	13	14	21	30	66	66	81	103
Proc-2-36	3.8	4.1	6.2	9.4	21	21	23	27	14	15	22	31	74	74	89	112

lies within a tighter continuous region. The tighter continuous region is still valid for the original GDP. We have proposed the use of basic steps in order to obtain the tighter region of the separation problem. This region is obtained by performing basic steps on the original GDP, and then applying the (HR) reformulation. The continuous relaxation of this region is as tight, and generally much tighter, than the continuous relaxation of the (BM). We have presented the results of applying this algorithm to several test problems, using different suggested strategies. The algorithm improves the relaxation of the (BM)

in all cases. Also, the algorithm solves the test problems faster than the (HR) and (BM) direct reformulations in most cases.

Acknowledgments

The authors would like to acknowledge financial support from the Center for Advanced Process Decision-making (CAPD).

References

- Abhishek, K., S. Leyffer, J. Linderoth. 2010. Filmint: An outer approximation-based solver for convex mixed-integer nonlinear programs. *INFORMS Journal on Computing* **22** 555–567.
- Balas, E. 1985. Disjunctive programming and a hierarchy of relaxations for discrete continuous optimization problems. *SIAM. Journal on Algebraic and Discrete Methods* **6** 466–486.
- Belotti, P., C. Kirches, S. Leyffer, J. Linderoth, J. Luedtke, A. Mahajan. 2013. Mixed-integer nonlinear optimization. *Acta Numerica* **22** 1–131.
- Biegler, L. T., I. E. Grossmann, A. W. Westerberg. 1997. *Systematic methods of chemical process design*. Prentice-Hall international series in the physical and chemical engineering sciences, Prentice Hall PTR. URL <http://books.google.com/books?id=59NTAAAAMAAJ>.
- Bonami, P., M. Kilinc, J. T. Linderoth. 2012. Algorithms and software for solving convex mixed integer nonlinear programs. *Chapter in Jon Lee et Sven Leyffer editors, Mixed Integer Nonlinear Programming, IMA Volumes in Mathematics and its Applications* **154** 1–39.
- Brooke, A., D. Kendrick, A. Meeraus, Raman. 1998. Gams, a users guide. *The Scientific Press* .
- Ceria, S., J. Soares. 1999. Convex programming for disjunctive convex optimization. *Mathematical Programming* **86** 595–614.
- Clocksinn, W. F., C. S. Mellish. 1981. *Programming in Prolog*. Springer.
- Dakin, R. J. 1965. A tree-search algorithm for mixed programming problems. *The Computer Journal* **8** 250–255.
- Duran, M., I. E. Grossmann. 1986. An outer-approximation algorithm for a class of mixed-integer nonlinear programs. *Mathematical Programming* **36** 307–339.
- Fletcher, Roger, Sven Leyffer. 1994. Solving mixed integer nonlinear programs by outer approximation. *Mathematical programming* **66** 327–349.
- Floudas, C. A. 1995. *Nonlinear and mixed integer optimization: Fundamentals and applications*. Oxford University Press.
- Geoffrion, A. M. 1972. Generalized benders decomposition. *Journal of Optimization Theory and Applications* **10** 237–260.

- Grossmann, I. E. 2002. Review of nonlinear mixed-integer and disjunctive programming techniques. *Optimization and Engineering* **3** 227–252.
- Grossmann, I. E., S. Lee. 2003. Generalized convex disjunctive programming: nonlinear convex hull relaxation. *Computational Optimization and Applications* **26** 83–100.
- Grossmann, I. E., F. Trespalacios. 2013. Systematic modeling of discrete-continuous optimization models through generalized disjunctive programming. *AIChE Journal* **59** 3276–3295. doi:10.1002/aic.14088. URL <http://dx.doi.org/10.1002/aic.14088>.
- Gupta, O. K., A. Ravindran. 1985. Branch and bound experiments in convex nonlinear integer programming. *Management Science* **31** 1533–1546.
- Lee, S., I. E. Grossmann. 2000. New algorithms for nonlinear generalized disjunctive programming. *Computers and Chemical Engineering* **24** 2125–2141.
- Lee, S., I. E. Grossmann. 2005. Logic-based modeling and solution of nonlinear discrete/continuous optimization problems. *Annals of Operations Research* **139** 267–288.
- Nemhauser, G. L., L. A. Wolsey. 1988. *Integer and Combinatorial Optimization*, Wiley-Interscience. Wiley.
- Quesada, I., I. E. Grossmann. 1992. An lp/nlp based branch and bound algorithm for convex minlp optimization problems. *Computers and Chemical Engineering* **16** 937–947.
- Raman, R., I. E. Grossmann. 1994. Modeling and computational techniques for logic-based integer programming. *Computers and Chemical Engineering* **18** 563–578.
- Ruiz, J. P., I. E. Grossmann. 2012. A hierarchy of relaxations for nonlinear convex generalized disjunctive programming. *European Journal of Operational Research* **218** 38–47.
- Sawaya, N. 2006. Reformulations, relaxations and cutting planes for generalized disjunctive programming. Ph.D. thesis, Carnegie Mellon University.
- Sawaya, N., I. E. Grossmann. 2012. A hierarchy of relaxations for linear generalized disjunctive programming. *European Journal of Operational Research* **216** 70–82.
- Sawaya, Nicolas W, Ignacio E Grossmann. 2005. A cutting plane method for solving linear generalized disjunctive programming problems. *Computers and chemical engineering* **29** 1891–1913.
- Stubbs, R., S. Mehrotra. 1999. A branch-and-cut method for 0-1 mixed convex programming. *Mathematical Programming* **86** 515–532.
- Trespalacios, F., I. E. Grossmann. 2014a. Algorithmic approach for improved mixed-integer reformulations of convex generalized disjunctive programs. *INFORMS Journal on Computing* Approved for publication.
- Trespalacios, F., I. E. Grossmann. 2014b. Review of mixed-integer nonlinear and generalized disjunctive programming applications in process systems engineering. *SIAM* submitted for publication.
- Trespalacios, F., I. E. Grossmann. 2014c. Review of mixed-integer nonlinear and generalized disjunctive programming methods. *Chemie Ingenieur Technik* Approved for publication.

- Turkay, M., I. E. Grossmann. 1996. A logic-based outer-approximation algorithm for minlp optimization of process flowsheets. *Computers and Chemical Engineering* **20** 959–978.
- Vecchietti, A., S. Lee, I. E. Grossmann. 2003. Modeling of discrete/continuous optimization problems: characterization and formulation of disjunctions and their relaxations. *Computers and Chemical Engineering* **27** 433–448.
- Westerlund, T., F. Pettersson. 1995. An extended cutting plane method for solving convex minlp problems. *Computers and Chemical Engineering* **19** 131–136.
- Williams, H. P. 1985. *Model Building in Mathematical Programming*. Wiley.