A computationally useful algebraic representation of nonlinear disjunctive convex sets using the perspective function

Kevin C. Furman[1,2], Nicolas W. Sawaya[3], Ignacio E. Grossmann[4]

Abstract

Nonlinear disjunctive convex sets arise naturally in the formulation or solution methods of many discrete-continuous optimization problems. Often, a tight algebraic representation of the disjunctive convex set is sought, with the tightest such representation involving the characterization of the convex hull of the disjunctive convex set. In the most general case, this can be explicitly expressed through the use of the perspective function in higher dimensional space – the so-called extended formulation of the convex hull of a disjunctive convex set. However, there are a number of challenges in using this characterization in computation which prevents its wide-spread use, including issues that arise because of the functional form of the perspective function. In this paper, we propose an explicit algebraic representation of a fairly large class of nonlinear disjunctive convex sets using the perspective function that addresses this latter computational challenge. This explicit representation can be used to generate (tighter) algebraic reformulations for a variety of different problems containing disjunctive convex sets, and we report illustrative computational results using this representation for several nonlinear disjunctive problems.

Keywords: disjunctive convex sets; perspective function; epsilon; MINLP

[1] Corresponding author: kevin.c.furman@exxonmobil.com
[2] ExxonMobil Upstream Research Company, Spring, TX 77389
[3] ExxonMobil Gas and Power Marketing Company, Spring, TX 77389
[4] Department of Chemical Engineering, Carnegie Mellon University, Pittsburgh, PA 15213

1. Introduction

Disjunctive convex sets arise naturally in the formulation or solution methods of many discrete-continuous optimization problems, in areas such as process synthesis (heat exchanger networks and reactor networks) [13,24,37,48], engineering design (truss structures and feed location in distillation columns) [34], retrofit planning [23,33,44], optimal positioning of products [13], scheduling of batch and continuous multiproduct batch plants [11,37,40,43,50], facility location problems [26], network design [5,9], machine scheduling [1], unit commitment for power generation [51] and stochastic service system design [14].

Often, a tight algebraic representation of the disjunctive convex set is sought in order to improve the computational performance of the solution method used to solve these aforementioned problems. The tightest such representation involves the characterization of the convex hull of the disjunctive convex set, which in the most general case, can be explicitly expressed through the use of the perspective function in higher dimensional space – the so-called extended formulation of the convex hull of a disjunctive convex set. However, there are a number of challenges in using this characterization in computation for nonlinear problems which prevents its wide-spread use, including issues that arise because of the functional form of the perspective function. In this paper, we propose an explicit algebraic representation of a fairly large class of disjunctive convex sets using the perspective function that addresses this latter computational challenge. This explicit representation can be used to generate (tighter) algebraic reformulations for a variety of different problems containing disjunctive convex sets, and we report illustrative computational results using this representation for the Synthesis, Retrofit-Synthesis and Constrained Layout problems (see Appendix for problem descriptions).

We also note that this explicit representation of nonlinear disjunctive convex sets can be used in the generation of cutting planes for problems that contain disjunctive convex sets as part of their formulation (see [44] and [47] for some preliminary work in that direction) or from disjunctions that are generated from a Mixed-Integer Non-Linear Programming (MINLP) formulation (e.g. split disjunctions) in a manner similar to that in

[46]; however, the details of cutting plane approaches that exploit our formulation are beyond the scope of this paper.

In order to appreciate the computational challenge alluded to above, let us examine the perspective of a given function. For a proper closed convex function $g(v):\mathbb{R}^n \to \mathbb{R}\cup\{\infty\}$, the perspective function $\tilde{g}(\lambda,v):\mathbb{R}^{n+1} \to \mathbb{R}\cup\{\infty\}$ is defined as [32, Section B 2.2]:

$$\tilde{g}(\lambda,v) = \begin{cases} \lambda g\left(\frac{v}{\lambda}\right) & \text{if } \lambda > 0 \\ +\infty & \text{if } \lambda \le 0. \end{cases}$$

It is well known that the perspective function of a closed convex function is convex [32, Section B 2.2.1], although it need not be closed. As such, it is typical to discuss the closure of the perspective function $(cl\ \tilde{g})\ (\lambda,v)$, which is defined as:

$$(cl\ \tilde{g})\ (\lambda,v) = \begin{cases} \lambda g\left(\frac{v}{\lambda}\right) & \text{if } \lambda > 0 \\ g'_\infty(v) & \text{if } \lambda = 0 \\ +\infty & \text{if } \lambda < 0, \end{cases} \tag{1}$$

and where $g'_\infty(v)$ is the recession function of $g(v)$ [32, Section B Proposition 2.2.2 and Example 3.2.3]. We note that in general, $g'_\infty(v)$ does not have a closed-form expression.

Let us now examine the following disjunctive set

$$F = \underset{j\in J}{\cup} C_j, \tag{2}$$

where J is finite, $C_j \equiv \left\{x\in\mathbb{R}^n \,\middle|\, G_j(x)\le 0\right\}$ and $G_j:\mathbb{R}^n \to \left(\mathbb{R}\cup\{\infty\}\right)^{m_j}$ are vector mappings whose components $g_{ij}, i=1\ldots m_j$ are proper closed convex functions.

Ceria and Soares [12] characterize the closure of the convex hull of F, denoted as $cl\ conv(F)$, using the closure of the perspective function. Their main result is as follows:

Theorem 1 [12]. Let $J' \equiv \left\{ j\in J \,\middle|\, C_j \ne \varnothing\right\}$. Then $x \in cl\ conv(F)$ if and only if the following system is feasible:

$$x = \sum_{j \in J'} \nu_j$$

$$(cl\ \tilde{G}_j)\ (\lambda_j, \nu_j) \le 0, \quad j \in J' \tag{3}$$

$$\sum_{j \in J'} \lambda_j = 1$$

$$\lambda_j \ge 0, \quad j \in J'.$$

We note that if the set C_j is compact, then $\lambda_j = 0$ implies $\nu_j = 0$ in (3) since $(cl\ \tilde{G}_j)\ (0, \nu_j) \le 0$ if and only if $G'_{\infty j}(\nu) \le 0$ from (1), and the recession cone of the set C_j, denoted as $C_{\infty j}$, is $C_{\infty j} = \left\{ \nu \in \mathbb{R}^n \middle| G'_{\infty j}(\nu) \le 0 \right\}$ [32, Section B, Proposition 3.2.4], with $C_{\infty j} = \{0\}$ when C_j is compact [32, Section A Proposition 2.2.3].

The above formulation in higher dimensional space extends the result of Balas [2,3,4] where sets $C_j, \forall j \in J$ are polyhedral. Stubbs and Mehrotra [46] also derive a similar extended formulation for convex programming problems within the context of generating disjunctive cutting planes. See also Jeroslow [35, Example 4.1], who describes a mixed-integer convex set that is equivalent to the set F when sets $C_j, \forall j \in J$ contain bounds on x by setting $\lambda_j \in \{0,1\}, j \in J$ in the formulation above.

One of the major computational challenges that arises when using the extended formulation in (3) is the algebraic representation of the perspective function $\tilde{g}$ when $\lambda = 0$. Indeed, $g'_\infty(\nu)$ does not have a closed-form expression in general, and $\lambda g\left(\frac{\nu}{\lambda}\right)$ is not defined, and therefore not differentiable, at $\lambda = 0$. Ceria and Soares comment on this problem and propose a log-barrier approach to address this issue. However, their method requires the solution of many convex programs, and the termination criteria to guarantee equivalence with the original problem is not straight-forward. Moreover, no readily implemented version of the algorithm is available for the general case. Stubbs and Mehrotra report numerical convergence issues when trying to solve a program based on the set described in (3) using an algorithm for continuously differentiable optimization within the context of generating disjunctive cutting planes. Finally, Jeroslow does not discuss the implementation of his formulation in practice.

In order to address the aforementioned computational challenge, different classes of approaches have been devised. In section 2, we identify and compare these different methods, and highlight some of the drawbacks of each of them. In section 3, we present our proposed method. In section 4, we describe some applications where disjunctive convex sets arise, and present illustrative computational results using our method. Finally, we conclude this paper in section 5 and discuss future work and next steps.

2. Literature Review

Several approaches in the literature address the computational issue in different ways. The first approach involves generating the explicit representation of $conv(F)$, either exactly or approximately, in a way that avoids the problem at $\lambda = 0$. If the sets C_j have a particular special structure, one may be able to generate an exact algebraic representation of the convex hull that does just that. Gunluk and Linderoth succeed in doing so within the context of indicator-induced {0,1}-MINLPs for a disjunctive set F representing the union of a fixed point or a ray and a convex set (with bounds on x), and provide the convex hull in the original space of variables when the functions g_{ij} are polynomial [26,27,28,29]; see also the closely related work of Akturk, Atamturk and Gurel [1]. Furthermore, when the functions g_{ij} are "SOCP-representable", Gunluk and Linderoth show that efficient computational methods exist to solve these reformulated (and perspective-strengthened) problems. However, for more general structures of the sets C_j, although one can theoretically generate an explicit algebraic representation of the convex hull, in practice, a small value ε must be added to the appropriate constraints in order to avoid division by 0. We note that in this case, the "ε-approximate" algebraic representation can be used directly in the derivation of tighter reformulations in the original space [7,31] or in the extended space [25,37,38,39,45] (depending on the application), or indirectly in the generation of cutting planes that are added to the original formulation [44,47]. Indeed, Hijazi et al [31] examine the case where the disjunctive set F represents the union of a box and a convex set (described by one nonlinear constraint and bounds on x). They provide the convex hull in the original space of variables under

the condition that the nonlinear function g is isotone, though with an exponential number of constraints. In practice, however, they show that, at least for their application of choice (the delay-constrained routing problem), the addition of one particular constraint yields a relaxation nearly as tight as that of the convex hull; they also note that a small value of ε is needed in their formulation to avoid division by 0, although their formulation is exact when λ_j belongs to {0,1}. Bonami et al [7] examine the more general case of two complementary disjunctions, with each disjunction as in [31], and where the "activation" of the convex set of the first disjunction via its indicator variable "deactivates" the convex set in the other disjunction. They show that the convex hull of such complementary disjunctions can be described in the original space of variables as long as the functions g are isotone and a certain technical condition holds on the set of indices over which the functions g are independently increasing or decreasing. For more general convex sets (but that still contain bounds on x), Lee and Grossmann [37, 38, 39] and Grossmann and Lee [25], within the context of Generalized Disjunctive Programming (GDP), propose to replace the perspective constraints in (3) by $(\lambda_j + \varepsilon)\, G_j\left(\frac{\nu_j}{\lambda_j + \varepsilon}\right) \le 0$ and $\nu_j^{LB}\lambda_j \le \nu_j \le \nu_j^{UB}\lambda_j$, while restricting λ_j to belong to {0,1} in order to represent the disjunctive set F – see [42] for an introduction on GDP and [7,22] for correspondence between GDP and indicator-induced {0,1}-MINLPs. While this approximation is exact for the limiting case when ε tends to zero, given that $\varepsilon > 0$ in practice, this approximation fails to represent the set F for cases where $G_j(0) > 0$ since $(\lambda_j + \varepsilon)\, G_j\left(\frac{\nu_j}{(\lambda_j + \varepsilon)}\right) > 0$ when $\lambda_j = 0$. In order to circumvent this problem, one could attempt to reduce ε to a value small enough to numerically satisfy the constraint within the solver tolerance, but this can lead to numerical difficulties, if not failure of the solver since it is not uncommon to require values of ε to be of the order of 10^{-15} in order to maintain feasibility. Sawaya and Grossmann note this in [45], and propose to replace the perspective approximation by $(\lambda_j + \varepsilon) G_j\left(\frac{\nu_j}{\lambda_j + \varepsilon}\right) - \max_{\nu_j, \lambda_j} \varepsilon G_j\left(\frac{\nu_j}{\lambda_j + \varepsilon}\right) \le 0$ or

$(\lambda_j + \varepsilon)\left(G_j\left(\frac{v_j}{\lambda_j + \varepsilon}\right) + G_j(0)(\lambda_j - 1)\right) \leq 0$. Although these ε-approximations solve the issue previously highlighted with the Lee and Grossmann formulation when $\lambda_j = 0$, the second approximation is convex only when $G_j(0) \geq 0$, and both of them remain inexact representations of the perspective function when $\lambda_j = 1$ if $\varepsilon > 0$ (as required in practice); as such, their use will result in an optimal solution to the ε-approximated problem that does not always correspond to the optimal solution of the original disjunctive problem (and under certain conditions, can be "very different" versus the true optimal solution).

The second approach involves the implicit approximation of $conv(F)$, typically through the derivation of valid cutting planes that are added to the original formulation of the problem, and which circumvents the issue at $\lambda = 0$ since the explicit convex hull is never described. This is the approach taken by Frangioni and Gentile [16,17,18,19] in their development of perspective cuts, which Gunluk and Linderoth [27] show to be equivalent to outer-approximations of the $conv(F)$ when the disjunctive set F represents the special case of the union of a fixed point (namely the "0" point) and a convex set. For the more general, but still special case of a disjunctive set F representing a split disjunction (within the context of generating cutting planes that are added to the original MINLP problem), Zhu and Kuno [53] suggest replacing $conv(F)$ by a linear approximation taken about the solution of the MINLP relaxation. Their preliminary results suggest that their method is effective on small problems, although their generated cuts may be weak [52]. Kilinc, Linderoth and Luetdke [36] attempt to address this issue by extending Zhu and Kuno's framework within the context of an iterative scheme that updates their polyhedral outer-approximations in an appropriate manner that at the limit, results in a cut that is as strong as that generated from Stubbs and Mehrotra's nonlinear cut-generating program [46]. Finally, Bonami [6] develops a two-phase cutting plane method for split disjunctions where a nonlinear program (NLP) in the original space of variables (but with twice the constraints) is first solved and its solution checked to see whether it belongs to $conv(F)$; if so, an outer-approximation is generated in the second phase such that a cut can be derived using linear programming (although this cut is not guaranteed to be as strong as the Stubbs and Mehrotra cut). We note that recently, a

hybrid approach of the Kilinc et al and Bonami methods has been implemented in CPLEX 12.6.2 [8].

All of these approaches, however, suffer from certain drawbacks from the perspective of failing to meet at least one of the following useful criteria: (1) ability to guarantee that the approach used will result in an optimal solution that is *equivalent* to that of the original problem; (2) applicability of the approach to a very *general* class of nonlinear disjunctive convex sets; (3) *robustness* of the approach in the sense of avoiding numerical difficulties related to precision; (4) *ease of implementation* of the approach in the sense of requiring only algebraic modeling software that directly calls modern off-the-shelf solvers (this is particularly important for practitioners); (5) and *tight* relaxations resulting from the approach. In contrast, our proposed method, as described in section 3, meets all of these useful criteria. We should mention, however, that our convex hull formulation is given in extended space such that additional variables are needed (adding at most $(n+1)\times|J|$ variables), which could potentially increase the computational burden of the approach relative to other alternatives. Although the larger size of the problem could be more than mitigated by the other applicable criteria (e.g. tightness of the relaxation), and this trade-off could be well-worth it for certain classes of problems, typically, a problem with additional variables is more computationally expensive to solve than one without (all else being equal). As such, an additional criterion reflecting whether the approach stays in the (6) *original space* of variables should be added to our list.

We now qualitatively compare the various methods in Table 1 according to the six aforementioned criteria. We also include in the table the traditional Big-M approach used to convert disjunctions into algebraic form as a point of reference for comparison.

Table 1: Comparison of Existing Approaches in the Literature

	Equivalent	General	Robust	Easy to Implement	Tight	Original Space
Ceria and Soares Formulation + Log-Barrier Method [12]	X	X			X	
Exact Algebraic Representation of Union of Point and Convex Set + SOCP Solver [26,27,28,29]	X		X	X	X	X
$\mathcal{E}$ -Approximate Algebraic Representation of Union of Box and Convex Set [7,31]	X		X	X	X[1]	X
$\mathcal{E}$ -Approximate Algebraic Representation of General Convex Sets [25,37,38,39,45]		X		X	X	
MINLP + Cutting Planes from Union of Point and Convex Set (perspective cuts) [16,17,18,19]	X		X		X	X
MINLP + Cutting Planes using Zhu and Kuno method [52,53]	X	X	X			X
MINLP + Cutting Planes using Stubbs and Mehrotra method [46]	X	X			X	
MINLP + Cutting Planes using Bonami method [6]	X	[2]	X		X[2]	X
MINLP + Cutting Planes using Kilinc et al method [36]	X	X	X		X	X
Big-M Method	X	X	X	X		X
Proposed Method[3]	X	X	X	X	X	

[1] Hijazi et al show that an exponential number of constraints are needed to describe the convex hull for the union of a box and a convex set in the original space. Still, at least for the delay-constrained routing problem, they show that the addition of only one specific constraint amongst that exponential number yields a formulation nearly as tight as the convex hull. We also note that all constraints can be added to the cut pool and used only when needed.

[2] Bonami's method only works for split disjunctions (thus is not "general"). Furthermore, it doesn't generate cuts that are guaranteed (at the limit) to be as strong as the Stubbs and Mehrotra cuts; however, we consider the resulting cuts "strong enough" (based on empirical evidence) to yield "tight" relaxations

[3] Our proposed method could be used either directly as a reformulation of disjunctive convex sets or in the generation of cutting planes that are added to the original formulation of the problem via a (numerically robust) nonlinear cut-generating program in extended space

3. New ε-approximate Formulation of the Convex Hull of Disjunctive Convex Sets

The proposed formulation in this section is based on a personal communication by Furman [20] that was further modified and first appeared in Sawaya's Ph.D. thesis [44]. The formulation has been presented at conferences [21,22], and has been mentioned by Gunluk and Linderoth in their literature review of the perspective function and its applications [29]. It has also been applied by Trespalacios and Grossmann within the context of preliminary work on generating cutting planes for nonlinear GDP [47] and has been implemented in experimental mathematical programming modeling software such as GAMS's Extended Mathematical Programming extension [15] via the LOGMIP solver [49] as well as the new Python-based Optimization platform PYOMO [30]. Although qualitative inspection suggests the reformulation technique to be an exact reformulation of the disjunctive set *F*, to date, the necessary theoretical underpinnings of the formulation have not been clearly established, which is important to ensure that the reformulation is rigorously grounded and correctly used under the right assumptions; as such, the purpose of this section is to remedy the situation. Using this new ε-approximation to the perspective function allows us to derive an explicit algebraic representation of the convex hull of the disjunctive convex set described in (2) that can be used in the practical solution of mixed integer convex programming problems. This new ε-approximation avoids the issue at $\lambda = 0$ in the original perspective function, is applicable to general disjunctive convex sets, results in a tight convex relaxation that approximates the convex hull closely, is easy to implement in any general purpose algebraic software and importantly, is equivalent to the perspective function at $\lambda = 0$ and 1 for *any* value of ε.

Let us now consider the disjunctive convex set in (2) and assume that the sets $C_j \equiv \left\{ x \in \mathbb{R}^n \middle| G_j(x) \le 0 \right\} \ \forall j \in J$ are compact, though not necessarily non-empty. Furthermore, we assume that $C_j, \forall j \in J$ are such that $G_j(0)$ is defined and the following condition holds

$$\left\{ x \in \mathbb{R}^n \middle| G_j(x) - G_j(0) \le 0 \right\} = \{0\}, \quad \forall j \in J . \tag{4}$$

It is worth nothing that although not all disjunctions satisfy the condition in (4) (for example, the disjunction $[(x-1)^2-2\le 0]\cup[3\le x\le 4]$ does not satisfy it), this condition is not very restrictive. Indeed, simply having a bounded range on $x\in C_j, \forall j\in J$ is sufficient to satisfy this condition (however, this is not necessary; for example, the disjunction $[x^2-1\le 0]\cup[3\le x\le 4]$ satisfies (4)).

Let us now define $eps-rel\ F(\varepsilon)$ to be the set of all those $(x,v,\lambda)\in\mathbb{R}^{n+n|J|+|J|}$ that satisfy the following set of constraints for some $0<\varepsilon<1$:

$$x=\sum_{j\in J} v_j \tag{5}$$

$$((1-\varepsilon)\lambda_j+\varepsilon)g_{ij}\left(\frac{v_j}{(1-\varepsilon)\lambda_j+\varepsilon}\right)-\varepsilon g_{ij}(0)(1-\lambda_j)\le 0,\quad i=1\ldots m_j,\ j\in J \tag{6}$$

$$\sum_{j\in J}\lambda_j=1 \tag{7}$$

$$\lambda_j\ge 0,\quad j\in J\ . \tag{8}$$

If we define $eps-MIP\ F(\varepsilon)\equiv\left\{(x,v,\lambda)\in eps-rel\ F(\varepsilon)\,\middle|\,\lambda_j\in\{0,1\},\ j\in J\right\}$, then the projection of $eps-MIP\ F(\varepsilon)$ onto the x space is $proj_{(x)}(eps-MIP\ F(\varepsilon))\equiv\left\{x\in\mathbb{R}^n\,\middle|\,(x,v,\lambda)\in eps-MIP\ F(\varepsilon)\right\}$.

In Proposition 1, we show that $proj_{(x)}(eps-MIP\ F(\varepsilon))$ is equivalent to F for any $0<\varepsilon<1$ as long as the condition in (4) holds, which is needed in order to ensure that $v_{j'}=0$ when $\lambda_j=1$ for all $j'\in J\setminus\{j\}$ in our reformulation. This is a very useful equivalence as it allows us to replace any disjunctive convex set satisfying (4) with $eps-MIP\ F(\varepsilon)$, where ε can be *any value* between (0,1), and whose relaxation $eps-rel\ F(\varepsilon)$ is a compact convex set that at the limit is equivalent to the closure of the convex hull of *F* (as we will prove in subsequent propositions).

Proposition 1 For any $0<\varepsilon<1$, $proj_{(x)}(eps-MIP\ F(\varepsilon))=F$.

Proof: Assume that $F \neq \varnothing$. We begin by proving $proj_{(x)}(eps - MIP\ F(\varepsilon)) \supseteq F$. Let $x' \in F$. There there exists some $j' \in J$ such that $g_{ij'}(x') \leq 0, i = 1 \ldots m_{j'}$. Now let $0 < \varepsilon' < 1$, and $(v_j, \lambda_j, \lambda_{j'}) = (0,0,1),\ j \in J \setminus \{j'\}$. Then constraints (5) to (8) reduce to:

$$g_{ij'}(x) \leq 0, \quad i = 1 \ldots m_{j'}. \tag{9}$$

Clearly, $x' \in \left\{ x \mid g_{ij'}(x) \leq 0, \quad i = 1 \ldots m_{j'} \right\}$, and therefore, $x' \in proj_{(x)}(eps - MIP\ F(\varepsilon'))$. It remains to be proven that $proj_{(x)}(eps - MIP\ F(\varepsilon)) \subseteq F$.

Let $\bar{x} \in proj_{(x)}(eps - MIP\ F(\bar{\varepsilon}))$ for some $0 < \bar{\varepsilon} < 1$. Then there exists some vector $(\bar{v}, \bar{\lambda})$ such that $(\bar{x}, \bar{v}, \bar{\lambda}) \in eps - MIP\ F(\bar{\varepsilon})$. Specifically, for some $\bar{j} \in J$,

$$\bar{\lambda}_{\bar{j}} = 1, \tag{10}$$

and from (7),

$$\bar{\lambda}_j = 0,\ j \in J \setminus \{\bar{j}\}. \tag{11}$$

From (6), (10) and (11)

$$g_{\bar{i}\bar{j}}(\bar{v}_{\bar{j}}) \leq 0, i = 1 \ldots m_{\bar{j}} \tag{12}$$

$$g_{ij}\left(\frac{\bar{v}_j}{\bar{\varepsilon}}\right) - g_j(0) \leq 0, \quad i = 1 \ldots m_j,\ j \in J \setminus \{\bar{j}\}, \tag{13}$$

From (4),

$$g_{ij}\left(\frac{\bar{v}_j}{\bar{\varepsilon}}\right) - g_j(0) \leq 0 \Rightarrow \frac{\bar{v}_j}{\bar{\varepsilon}} = 0, \quad i = 1 \ldots m_j,\ j \in J \setminus \{\bar{j}\}. \tag{14}$$

Therefore,

$$\bar{v}_j = 0,\ j \in J \setminus \{\bar{j}\} \tag{15}$$

and from (5) and (15),

$$\bar{x} = \bar{v}_{\bar{j}}. \tag{16}$$

Finally, from (12), (15) and (16), we have

$$g_{\bar{i}\bar{j}}(\bar{x}) \leq 0, i = 1 \ldots m_{\bar{j}}. \tag{17}$$

Now assume that $F = \varnothing$. We claim that $proj_{(x)}(eps - MIP\ F(\bar{\varepsilon})) = \varnothing$. For if we assume this to not be the case, then there exists some $\bar{x} \in proj_{(x)}(eps - MIP\ F(\bar{\varepsilon}))$ for some $0 < \bar{\varepsilon} < 1$. As was shown above, (5) – (8) reduces to $g_{i\bar{j}}(\bar{x}) \le 0, i = 1 \ldots m_{\bar{j}}$ for some $\bar{j} \in J$. But since $F = \varnothing$, then $C_j = \varnothing, \forall j \in J$. Therefore, $\left\{ x \middle| g_{i\bar{j}}(\bar{x}) \le 0, i = 1 \ldots m_{\bar{j}} \right\} = \varnothing$, and thus, $proj_{(x)}(eps - MIP\ F(\varepsilon)) = \varnothing$.

■

Next, we prove that the new "ε-approximate" perspective function is convex, and use that to show that $eps - rel\ F(\varepsilon)$ is a compact convex set.

Lemma 1 For any $0 < \varepsilon < 1,\ i = 1 \ldots m_j,\ j \in J$, the function

$$h_{ij}(\lambda_j, \nu_j, \varepsilon) \equiv ((1-\varepsilon)\lambda_j + \varepsilon) g_{ij}\left(\frac{\nu_j}{(1-\varepsilon)\lambda_j + \varepsilon} \right) - \varepsilon g_{ij}(0)(1-\lambda_j) \tag{18}$$

is convex over the set $0 \le \lambda_j \le 1$.

Proof: By assumption, the function $g_{ij}, i = 1 \ldots m_j,\ j \in J$, is a proper closed convex function. Therefore, the perspective function of g_{ij},

$$\tilde{h}_{ij}(u_j, \nu_j) = \begin{cases} u_j g_{ij}\left(\nu_j / u_j\right) & \text{if } u_j > 0 \\ +\infty & \text{if } u_j \le 0, \end{cases} \tag{19}$$

is convex. Now let $u_j = (1-\varepsilon)\lambda_j + \varepsilon$ for $0 \le \lambda_j \le 1$ and $0 < \varepsilon < 1$. Then, from (19), $\tilde{h}_{ij}(u_j, \nu_j) = ((1-\varepsilon)\lambda_j + \varepsilon) g_{ij}\left(\frac{\nu_j}{(1-\varepsilon)\lambda_j + \varepsilon} \right)$ since $0 < u_j \le 1$, and the resulting function is convex since the pre-composition of a convex function ($\tilde{h}_{ij}$) with an affine one (u_j) retains convexity (Proposition 2.1.4 p.88 in [32]). Finally, since $\varepsilon g_{ij}(0)(1-\lambda_j)$ is linear, then (18) is convex.

■

Proposition 2 For any $0<\varepsilon<1$, $eps-rel\ F(\varepsilon)$ is a compact convex set.

Proof: If $eps-rel\ F(\varepsilon)=\varnothing$, then the result trivially follows. Now assume that $eps-rel\ F(\varepsilon)\neq\varnothing$. Then, $eps-rel\ F(\varepsilon)$ is a convex set as constraints (5), (7) and (8) are linear, and from Lemma 1, the LHS of constraint (6) is convex.

It remains to be proven that $eps-rel\ F(\varepsilon)$ is a compact set for any $0<\varepsilon<1$. We first show that $eps-rel\ F(\varepsilon)$ is closed for any $0<\varepsilon<1$. From (7) and (8), $0\le\lambda_j\le 1,\ j\in J$, and so belongs to a compact and thus closed set. Now, if $\lambda_{j'}=0$ for some $j'\in J$, then $\nu_{j'}=0$. Thus $\nu_{j'}$ belongs to a closed set for $\lambda_{j'}=0$. If $0<\lambda_{j'}\le 1$ for some $j'\in J$, then for some $0<\varepsilon'<1$, the constraints (6) reduce to

$$g_{ij'}\left(\frac{\nu_{j'}}{\alpha_{j'}}\right)\le c_{ij'},\quad i=1\ldots m_{j'},\ j'\in J, \tag{20}$$

where $\varepsilon'<\alpha_{j'}=(1-\varepsilon')\lambda_{j'}+\varepsilon'\le 1$ and $c_{ij'}=\dfrac{\varepsilon' g_{ij'}(0)(1-\lambda_{j'})}{((1-\varepsilon')\lambda_{j'}+\varepsilon')}$ is a scalar. Since $g_{ij'}, i=1\ldots m_{j'}$ are closed functions by assumption, then their (lower) level sets $L_c(g_{ij'}):=\left\{x\in\mathbb{R}^n\,\middle|\,g_{ij'}(x)\le c\right\}, i=1\ldots m_{j'}$ are closed as well for any scalar c. Thus, the set $\left\{\left(\dfrac{\nu_{j'}}{\alpha_{j'}}\right)\middle|\,g_{ij'}\left(\dfrac{\nu_{j'}}{\alpha_{j'}}\right)\le c_{ij'},\quad i=1\ldots m_{j'},\ j'\in J\right\}$ is closed for any $c_{ij'}, i=1\ldots m_{j'}$, and $\nu_{j'}$ belongs to a closed set for $0<\lambda_{j'}\le 1$. Therefore, $\nu_{j'}$ belongs to a closed set for $0\le\lambda_{j'}\le 1$. Finally, as a finite union of closed sets is closed, then from (5), x must belong to a closed set. Therefore, $eps-rel\ F(\varepsilon)$ is a closed set.

Next, we prove that $eps-rel\ F(\varepsilon)$ for any $0<\varepsilon<1$ is bounded. Since $eps-rel\ F(\varepsilon)$ is closed, it suffices to show that the recession cone of $eps-rel\ F(\varepsilon)$ is the {0} set in order to prove that $eps-rel\ F(\varepsilon)$ is bounded; i.e. that $eps-rel\ F(\varepsilon)_\infty=\left\{d\,\middle|\,(x+\alpha d,\nu+\alpha d,\lambda+\alpha d)\in eps-rel\ F(\varepsilon),\forall(x,\nu,\lambda)\in eps-rel\ F(\varepsilon),\forall\alpha\ge 0\right\}=\{0\}$. We do this by proving (by contradiction) that no recession direction $d\neq 0$ exists for $eps-rel\ F(\varepsilon)$ (since $d=0$ is always feasible for $eps-rel\ F(\varepsilon)_\infty$). For assume that there

is. Then there exists some $d' \neq 0$ such that for every $(x,\nu,\lambda) \in eps-rel\ F(\varepsilon)$ and every $\alpha \geq 0$, $(x+\alpha d', \nu+\alpha d', \lambda+\alpha d') \in eps-rel\ F(\varepsilon)$. Now let $0<\varepsilon'<1$. If we choose the vector (x,ν,λ) such that $(x, \nu_{j'}, \lambda_{j'}, \nu_j, \lambda_j) = (x', x', 1, 0, 0),\ j \in J \setminus \{j'\}$ for some $j' \in J$, then clearly it belongs to $eps-rel\ F(\varepsilon')$ and constraints (5) – (8) reduce to $g_{ij'}(x') \leq 0, i = 1 \ldots m_{j'}$. But by assumption, the sets $C_j,\ j \in J$ are compact. Therefore, $\left\{x \middle| g_{ij'}(x') \leq 0, i = 1 \ldots m_{j'}\right\}$ is compact and thus bounded, and there exists some $(x,\nu,\lambda) \in eps-rel\ F(\varepsilon')$, namely $(x, \nu_{j'}, \lambda_{j'}, \nu_j, \lambda_j) = (x', x', 1, 0, 0),\ j \in J \setminus \{j'\}$, such that no $d' \neq 0$ can exist. Therefore $eps-rel\ F(\varepsilon)$ is bounded.

■

The projection of $eps-rel\ F(\varepsilon)$ onto the x space is $proj_{(x)}(eps-rel\ F(\varepsilon)) \equiv \left\{x \in \mathbb{R}^n \middle| (x,\nu,\lambda) \in eps-rel\ F(\varepsilon)\right\}$. In the next proposition, we show that not only does $proj_{(x)}(eps-rel\ F(\varepsilon))$ constitute a valid relaxation of F for any $0<\varepsilon<1$, but that furthermore, it is the tightest possible relaxation as $\varepsilon \to 0$.

Proposition 3 For any $0<\varepsilon<1$, $proj_{(x)}(eps-rel\ F(\varepsilon)) \supseteq F$. Furthermore, $\lim_{\varepsilon \to 0} proj_{(x)}(eps-rel\ F(\varepsilon)) = clconv\ F$.

Proof: From Proposition 1, for any $0<\varepsilon<1$, $proj_{(x)}(eps-MIP\ F(\varepsilon)) = F$. Since $proj_{(x)}(eps-rel\ F(\varepsilon))$ represents its continuous relaxation, then clearly $proj_{(x)}(eps-rel\ F(\varepsilon)) \supseteq F$.

We now show that $\lim_{\varepsilon \to 0} proj_{(x)}(eps-rel\ F(\varepsilon)) = clconv\ F$. Clearly,

$$\lim_{\varepsilon \to 0} h_{ij}(\lambda_j > 0, \nu_j, \varepsilon) = \lim_{\varepsilon \to 0}((1-\varepsilon)\lambda_j + \varepsilon) g_{ij}\left(\frac{\nu_j}{(1-\varepsilon)\lambda_j + \varepsilon}\right) - \varepsilon g_{ij}(0)(1-\lambda_j) = \lambda_j g_{ij}\left(\frac{\nu_j}{\lambda_j}\right).$$

Furthermore, as we have already established in Proposition 1, $\lambda_j = 0 \Rightarrow \nu_j = 0, \forall j \in J$. Therefore, $\lim_{\varepsilon \to 0} h_{ij}(0, \nu_j, \varepsilon) = \lim_{\varepsilon \to 0} h_{ij}(0,0,\varepsilon) = \lim_{\varepsilon \to 0} 0 = 0$. Therefore,

$$\lim_{\varepsilon \to 0} h_{ij}(\lambda_j, \nu_j, \varepsilon) = \begin{cases} \lambda_j g_{ij}\left(\nu_j / \lambda_j\right) & \text{if } \lambda_j > 0 \\ 0 & \text{if } \lambda_j = 0 \end{cases} . \tag{21}$$

Now when C_j is compact, $\lambda_j = 0 \Rightarrow v_j = 0$ in (3) since $(cl\ \tilde{G}_j)\ (0, v_j) \leq 0 \Leftrightarrow G'_{\infty j}(\nu) \leq 0$ from (1), and the recession cone of the set C_j, denoted as $C_{\infty j}$, is $C_{\infty j} = \left\{\nu \in \mathbb{R}^n \middle| G'_{\infty j}(\nu) \leq 0\right\}$ [32, Section B, Proposition 3.2.4], and $C_{\infty j} = \{0\}$ when C_j is compact [32, Section A Proposition 2.2.3]. Furthermore, $(cl\ \tilde{G}_j)\ (0,0) = 0$ [32, Section B, Remark 2.2.3]. Therefore, when the sets C_j are compact:

$$(cl\ \tilde{g})\ (\lambda, \nu) = \begin{cases} \lambda g\left(\frac{\nu}{\lambda}\right) & \text{if } \lambda > 0 \\ 0 & \text{if } \lambda = 0 \end{cases} .$$

But this corresponds precisely to (21), and from Theorem 1, the result follows.

■

Let us now define $BM - rel\ F$ to be the set of all those $(x, \lambda) \in \mathbb{R}^{n+|J|}$ that satisfy the following set of constraints:

$$g_{ij}(x) - M_{ij}(1 - \lambda_j) \leq 0, \quad i = 1 \ldots m_j,\ j \in J \tag{22}$$

$$\sum_{j \in J} \lambda_j = 1 \tag{23}$$

$$\lambda_j \geq 0, \quad j \in J , \tag{24}$$

where the parameters M_{ij} need to be large enough in order for $BM - rel\ F$ to be feasible (this can be accomplished, for example, by setting $M_{ij} = \max_{x \in C_j} g_{ij}(x)$). If we now define $BM - MIP\ F \equiv \left\{(x, \lambda) \in BM - rel\ F \middle|\ \lambda_j \in \{0,1\},\ j \in J\right\}$, then the projection of $BM - MIP\ F$ onto the x space is $proj_{(x)}(BM - MIP\ F) \equiv \left\{x \in \mathbb{R}^n \middle| (x, \lambda) \in (BM - MIP\ F)\right\}$. Clearly, $proj_{(x)}(BM - MIP\ F) = F$, and its continuous relaxation $proj_{(x)}(BM - rel\ F) \supseteq F$.

In the following proposition, we give a necessary and sufficient condition for $proj_{(x)}(BM-rel\ F) \supseteq proj_{(x)}(eps-rel\ F(\varepsilon))$ for some $0<\varepsilon<1$. Although this condition is difficult to ascertain in practice given the computational expense associated with demonstrating that it holds, it is still a useful theoretical result as it shows when we are guaranteed to have a formulation at least as strong as the Big-M formulation (of course, at the limit, since $\lim_{\varepsilon\to 0} proj_{(x)}(eps-rel\ F(\varepsilon)) = clconv\ F$ per Proposition 3, the formulation will always be at least as tight as the Big-M formulation; however, that is not necessarily true when $0<\varepsilon<1$).

Proposition 4 Let $NSC_{ij}(\varepsilon) = \left\{\max g_{ij}(x) - M_{ij}(1-\lambda_j) \middle| (x,\nu,\lambda) \in eps-rel\ F(\varepsilon)\right\}$ for some $0<\varepsilon<1$. Then $proj_{(x)}(BM-rel\ F) \supseteq proj_{(x)}(eps-rel\ F(\varepsilon))$ iff $NSC_{ij}(\varepsilon) \le 0, \forall i = 1\ldots m_j, \forall j \in J$.

Proof: Let $NSC_{ij}(\varepsilon') \le 0, \forall i = 1\ldots m_j, \forall j \in J$ for some $0<\varepsilon'<1$. Then for any $(x',\nu',\lambda') \in eps-rel\ F(\varepsilon)$, $g_{ij}(x') - M_{ij}(1-\lambda'_j) \le 0$, $i = 1\ldots m_j$, $j \in J$, $\sum_{j\in J}\lambda'_j = 1$ and $\lambda'_j \ge 0$, $j \in J$. But this corresponds precisely to (22) – (24), and therefore $BM-rel\ F \supseteq \left\{(x,\lambda) \in eps-rel\ F(\varepsilon) \mid \nu \in eps-rel\ F(\varepsilon)\right\}$. It immediately follows that $proj_{(x)}(BM-rel\ F) \supseteq proj_{(x)}(eps-rel\ F(\varepsilon))$.

We now prove the converse. Let $proj_{(x)}(BM-rel\ F) \supseteq proj_{(x)}(eps-rel\ F(\varepsilon'))$ for some $0<\varepsilon'<1$. Then every $(x,\lambda) \in eps-rel\ F(\varepsilon')$ must also belong to $BM-rel\ F$, and by implication, satisfy the constraints $g_{ij}(x) - M_{ij}(1-\lambda_j) \le 0$, $i = 1\ldots m_j$, $j \in J$. Therefore, $NSC_{ij}(\varepsilon') \le 0, \forall i = 1\ldots m_j, \forall j \in J$.

■

4. Illustrative Computational Results

In this section, we present illustrative computational results using our novel formulation discussed in Section 3 to generate ε-approximate representations of the convex hull of several classes of disjunctive convex sets within the context of Generalized Disjunctive Programming (or equivalently, 0-1 indicator-induced MINLPs). We focus on the so-called Synthesis, Retrofit-Synthesis and Constrained Layout problems, although we note that this ε-approximate representation can be applied to explicitly describe the convex hull of any nonlinear disjunctive convex set described by (2), assuming the condition in (4) holds. For details about the general disjunctive programming formulations of these problems, we refer the interested reader to the Appendix, as well as to Sawaya's thesis [44].

Our approach here consists of converting the disjunctive convex set F into the ε-approximate set $eps - MIP\ F(\varepsilon) \equiv \left\{(x, \nu, \lambda) \in eps - rel\ F(\varepsilon) \middle|\ \lambda_j \in \{0,1\},\ j \in J\right\}$ as described in Section 3. We note that if the terms of the disjunctive set in GDP form do not contain bounds, then we add those explicitly to ensure that the condition in (4) is met (this is often the easiest way to ensure that (4) holds and is typically the approach used within the GDP community). If the problem contains more than one disjunction $k \in K$, then every disjunctive set F is converted into its ε-approximate form and the resulting algebraic sets are intersected to obtain, following Balas' nomenclature in [3] for linear disjunctive sets, the so-called "hull reformulation" $eps - MIP^{HR}(\varepsilon) \equiv \bigcap_{k \in K} eps - MIP\ F_k(\varepsilon)$.

We conduct our computational experiments as a function of the parameter ε in order to assess the sensitivity of the reformulation to this parameter. We also compare our reformulation against the Big-M formulation, where every disjunctive convex set F is converted into the set $BM - MIP\ F \equiv \left\{(x, \lambda) \in BM - rel\ F \middle|\ \lambda_j \in \{0,1\},\ j \in J\right\}$ to have a point of reference when assessing the strength of our reformulation's relaxation. We note that the approach taken in these computational experiments is one of many approaches that could have been taken using our reformulation. Alternatively, a hybrid approach of converting only some disjunctive sets into their ε-approximate form while converting

others to their Big-M form could have been used; or intersecting certain disjunctive sets first via basic steps (see [3] for details on basic steps) before converting them into their ε-approximate forms (to strengthen the resulting relaxation) could have been performed; or even using our new ε-approximate formulation within the context of a nonlinear cut-generating program (whose cutting planes can be added to strengthen the big-M formulation) could have been entertained (see [44, 47] for preliminary efforts in that direction). As such, we emphasize that these computational experiments are merely illustrative and are meant to show that this new reformulation "works" in practice. A comprehensive computational study comparing the various approaches that could be used with our reformulation against other approaches in the literature is out of scope for this paper, but remains interesting future work.

The computations were performed using a nonlinear programming-based branch-and-bound method (GAMS/SBB with CONOPT as the NLP solver) [10] with a 5 hour time limit on a 2.4 GHz / 8GB RAM Linux PC. The computational time is reported in seconds for 52 total instances, including 24 Synthesis, 22 Retrofit Synthesis and 6 Constrained Layout problems. In the computations, values for ε ranging from 10^{-10} to 0.99 are tested using the new ε-approximation formulation and compared with the classic Big-M formulation (the values of the Big-M parameters were chosen for each nonlinear inequality by solving for $M_{ijk} = \max_{x \in C_{jk}} g_{ijk}(x)$ for every $i \in I, j \in J, k \in K$). We also note that all instances solved in this section were submitted to MINLPLib [41] and are available online.

Tables 2 through 4 show the number of instances from each class of problems that were solvable, and Table 5 combines all of them into a single solution set. As can be seen, using a value for ε of 10^{-10} results in an increased number of numerical failures in the NLP sub-problems due to CONOPT not being able to converge to a feasible solution, most likely due to the ill-conditioning of the Hessian of the approximation function as ε tends to zero. On the other hand, using a value for ε above 0.1 tends to reach the time limit for some of the Retrofit problems, most likely due to having weaker relaxations (similarly for the Big-M formulations). This suggests that choosing a value of ε that is not too small, to avoid potential numerical difficulties, nor too large, to avoid "loose" relaxations that cause problems to "time-out", is best. Indeed, by choosing a value of ε

between 0.1 and 10^{-4} , we solve all instances of our problems with no numerical difficulties. The more general question of choosing the "best" ε is left as part of a more comprehensive computational study.

Table 2: Number of Instances Solved for Constrained Layout Problem

Epsilon	Failure Type: Numerical	Failure Type: NLP	Failure Type: Time Out	Total Solved	Percent Solved
1E-10	1	2	0	3	50%
0.000001	0	0	0	6	100%
0.00001	0	1	0	5	83%
0.0001	0	0	0	6	100%
0.001	0	0	0	6	100%
0.01	0	0	0	6	100%
0.1	0	0	0	6	100%
0.5	0	0	0	6	100%
0.9	0	0	0	6	100%
0.99	0	0	0	6	100%
Big-M	0	0	0	6	100%

Table 3: Number of Instances Solved for Synthesis Problem

Epsilon	Failure Type: Numerical	Failure Type: NLP	Failure Type: Time Out	Total Solved	Percent Solved
1E-10	8	0	0	16	67%
0.000001	0	0	0	24	100%
0.00001	0	0	0	24	100%
0.0001	0	0	0	24	100%
0.001	0	0	0	24	100%
0.01	0	0	0	24	100%
0.1	0	0	0	24	100%
0.5	0	0	0	24	100%
0.9	0	0	0	24	100%
0.99	0	0	0	24	100%
Big-M	0	0	6	18	75%

Table 4: Number of Instances Solved for Retrofit-Synthesis Problem

Epsilon	Failure Type: Numerical	Failure Type: NLP	Failure Type: Time Out	Total Solved	Percent Solved
1E-10	18	0	0	4	18%
0.000001	1	0	0	21	95%
0.00001	0	0	0	22	100%
0.0001	0	0	0	22	100%
0.001	0	0	0	22	100%
0.01	0	0	0	22	100%
0.1	0	0	0	22	100%
0.5	0	0	3	19	86%
0.9	0	0	3	19	86%
0.99	0	0	2	20	91%
Big-M	0	0	18	4	18%

Table 5: Number of Instances Solved for All Problem Classes

Epsilon	Failure Type: Numerical	Failure Type: NLP	Failure Type: Time Out	Total Solved	Percent Solved
1E-10	27	2	0	23	44%
0.000001	1	0	0	51	98%
0.00001	0	1	0	51	98%
0.0001	0	0	0	52	100%
0.001	0	0	0	52	100%
0.01	0	0	0	52	100%
0.1	0	0	0	52	100%
0.5	0	0	3	49	94%
0.9	0	0	3	49	94%
0.99	0	0	2	50	96%
Big-M	0	0	24	28	54%

Table 6 presents the results of solving the root relaxation of all instances $eps - MIP^{HR}(\varepsilon)$ for various values of ε (i.e. we are solving the NLP $eps - rel^{HR}(\varepsilon) \equiv \bigcap_{k \in K} eps - rel\ F_k(\varepsilon)$). The CONOPT solver was used to solve this resulting NLP, and the relaxation gap, which is defined as

$$GAP(\varepsilon) = \frac{Optimal\ Value\left(eps - MIP^{HR}(\varepsilon)\right) - Optimal\ Value\left(eps - rel^{HR}(\varepsilon)\right)}{Optimal\ Value\left(eps - MIP^{HR}(\varepsilon)\right)}$$, is reported.

The *GAP* is a useful indicator of the strength of the formulation, as a smaller *GAP* typically leads to shorter solution times. As expected, the use of the ε-approximation formulation leads to a relaxation at least as strong as the Big-M formulation for all instances, and one that is significantly stronger for instances of the Synthesis and Retrofit problems. However, we note that for ε values of 10^{-10}, 12 of 22 retrofit instances experienced numerical errors in solving the root relaxation. Furthermore, a clear increase in *GAP* is evident as ε increases for the synthesis and retrofit instances. Finally, Constrained Layout instances show no difference in relaxation strength between ε and Big-M reformulations.

Table 6: Solution of the Root Relaxation

	Constrained Layout			Synthesis			Retrofit-Synthesis		
Epsilon	Avg Gap	Low Gap	High Gap	Avg Gap	Low Gap	High Gap	Avg Gap	Low Gap	High Gap
1E-10	100%	100%	100%	2%	0%	17%	6%	1%	10%
0.000001	100%	100%	100%	2%	0%	17%	4%	0%	10%
0.00001	100%	100%	100%	2%	0%	17%	4%	0%	10%
0.0001	100%	100%	100%	2%	0%	17%	4%	0%	10%
0.001	100%	100%	100%	2%	0%	17%	4%	0%	10%
0.01	100%	100%	100%	2%	0%	17%	4%	1%	10%
0.1	100%	100%	100%	2%	0%	21%	5%	2%	10%
0.5	100%	100%	100%	5%	0%	44%	12%	2%	25%
0.9	100%	100%	100%	7%	0%	66%	17%	3%	38%
0.99	100%	100%	100%	8%	0%	72%	19%	3%	40%
Big-M	100%	100%	100%	435%	19%	2608%	320%	57%	876%

Table 7 compares cumulative results for the subset of instances in which the Big-M formulation and the $eps - MIP^{HR}(\varepsilon)$ formulation for all ε values were solvable within the time limit; these instances are referred to as "shared" instances in the table, for which we report the cumulative number of branch-and-bound nodes and the cumulative run time. For Synthesis and Retrofit instances, the Big-M formulations require substantially longer run-times than those with the ε-approximation formulation due to much weaker relaxations. Furthermore, values of ε in the range of 0.1 to 0.001 appear to have the best

range of performance. However, for the Constrained Layout instances, the Big-M formulations outperform the ε-approximation formulations given the fact that we are solving a smaller problem whose relaxation is no worse than the ε-approximation formulations.

Table 7: Cumulative Results for Instances Solvable by all Approaches

	Constrained Layout				Synthesis				Retrofit-Synthesis			
Epsilon	Shared	Nodes	Time (sec)	Nodes/sec	Shared	Nodes	Time (sec)	Nodes/sec	Shared	Nodes	Time (sec)	Nodes/sec
1E-10	3	3882750	3797	1023	16	108	8	13	4	1328	16	81
0.000001	3	4767347	5491	868	16	113	6	18	4	1265	11	117
0.00001	3	3018376	2760	1094	16	143	6	23	4	1362	10	135
0.0001	3	1797986	1467	1226	16	188	8	25	4	1252	9	143
0.001	3	2204085	1861	1184	16	190	7	27	4	1267	7	183
0.01	3	2204085	1859	1185	16	190	7	27	4	1330	7	186
0.1	3	2497305	1642	1521	16	192	7	27	4	1442	7	193
0.5	3	1424466	727	1960	16	250	8	30	4	3263	19	174
0.9	3	1799602	947	1900	16	408	11	37	4	5594	36	157
0.99	3	1579994	834	1895	16	522	14	38	4	6105	43	143
Big-M	3	382858	43	9006	16	1682571	17801	95	4	990966	1566	633

5 Conclusion

In this paper, we have developed an explicit algebraic representation for general disjunctive convex sets using the perspective function that yields tight relaxations, while avoiding the computational challenges resulting from the functional form of the perspective function. We have shown that this new algebraic representation can be used to generate Mixed-Integer Programming reformulations that are exactly equivalent to the original disjunctive convex set. Furthermore, we have shown that this algebraic representation is equivalent to the closure of the convex hull of the disjunctive convex set at its limit. Finally, and importantly, this representation facilitates implementation in general purpose algebraic modeling languages and uses general purpose solvers. We also note that this representation can be used in the generation of cutting planes for problems that contain disjunctive convex sets as part of their formulation or from disjunctions that are generated from an MINLP formulation, although the details of these approaches are

not covered in this paper. A logical next step is to compare the use of our reformulation within the context of different algorithmic approaches (e.g. its performance as an explicit algebraic representation of the disjunctive convex set versus its use in a nonlinear cut-generating program) against the various other approaches in the literature previously discussed in order to assess which methods work best on which classes of problems.

Appendix. Description of nonlinear GDP problem classes

Synthesis of Process Networks

These problems consist in determining those process units Y_k to be included in the design of a process network such that the structure and operating conditions of this network will meet certain design specifications, while minimizing the sum of fixed costs c_k and variable costs $a^T x$ of the overall network (the variable x represents material flow). The following example represents a GDP model of a 5-process network, and can be found in Sawaya's thesis [44]. This example is a slightly modified version of the original form proposed by Duran and Grossmann [13] (MINLP form), Turkay and Grossmann [48] and Lee and Grossmann [37] (GDP forms).

$$\min Z = \sum_{k=1}^{5} c_k + a^T x$$

s.t.

$$x_1 - x_2 - x_3 = 0 \tag{25}$$

$$x_6 - x_4 - x_5 = 0 \tag{26}$$

$$x_6 - x_7 - x_8 - x_{11} = 0 \tag{27}$$

$$x_8 - x_9 - x_{10} - x_{11} = 0 \tag{28}$$

$$\begin{bmatrix} Y_1 \\ \exp(x_4) - 1 - x_2 \le 0 \\ c_1 = 5 \end{bmatrix} \vee \begin{bmatrix} \neg Y_1 \\ x_4 = x_2 = 0 \\ c_1 = 0 \end{bmatrix} \tag{29}$$

$$\begin{bmatrix} Y_2 \\ \exp(x_5 / 1.2) - 1 - x_3 \le 0 \\ c_2 = 8 \end{bmatrix} \vee \begin{bmatrix} \neg Y_2 \\ x_3 = x_5 = 0 \\ c_2 = 0 \end{bmatrix} \tag{30}$$

$$\begin{bmatrix} Y_3 \\ x_{13} - 0.75 x_9 = 0 \\ c_3 = 6 \end{bmatrix} \vee \begin{bmatrix} \neg Y_3 \\ x_9 = x_{13} = 0 \\ c_3 = 0 \end{bmatrix} \tag{31}$$

$$\begin{bmatrix} Y_4 \\ \exp(x_{14} / 1.5) - 1 - x_{10} \le 0 \\ c_4 = 10 \end{bmatrix} \vee \begin{bmatrix} \neg Y_4 \\ x_{10} = x_{14} = 0 \\ c_4 = 0 \end{bmatrix} \tag{32}$$

$$\begin{bmatrix} Y_5 \\ x_{15} - x_{11} = 0 \\ x_{15} - 0.5 x_{12} = 0 \\ c_5 = 6 \end{bmatrix} \vee \begin{bmatrix} \neg Y_5 \\ x_{11} = x_{12} = x_{15} = 0 \\ c_5 = 0 \end{bmatrix} \tag{33}$$

$$x_1 \le 10,\ x_{12} \le 7 \tag{34}$$

$$\underline{Y_1 \vee Y_2} \tag{35}$$

$$a^T = [0,0,0,0,0,0,5,0,0,0,0,-2,200,250,300]$$

$$x_j,\ c_k \ge 0,\ Y_k \in \{True, False\} \text{ for } k=1,2,...5 \text{ ; } j=1,2,...15$$

Equations (25)-(28) represent linear mass balances around nodes N1-N4. Disjunctions (29)-(33) embody the discrete dichotomy of process selection, where a unit, along with its in-and-out flows (connected to one another via an exponential relationship) and fixed cost, is selected for inclusion in the final network only if its corresponding Boolean variable $Y_k = True$; otherwise, the unit is not selected, and its flows and fixed cost are set to 0. Equation (34) represents upper bounds on certain flows, and finally,

logic equation (35) imposes the condition that either process 1 or 2 must be selected (but not both).

Retrofit-Synthesis of Process Networks

These problems consist in simultaneously redesigning part of an existing plant and synthesizing (from scratch) part of a new one. Specifically, one is interested in determining whether certain units should be included in the design of the new plant, and whether certain modifications such as improvements in yield, capacity and energy reduction should be performed on the existing plant. In addition it is required that economic potential be maximized given a certain time horizon and limited capital investments. The nonlinearities in this set of problems stem from the synthesis portion of the model, and correspond to logarithmic functions. Below, we only show the retrofit portion of the model (since the Synthesis portion was presented above), which is a modification of work by Jackson and Grossmann [33] and which appeared in [44]:

$$Min \quad \sum_{\forall t\in T}\sum_{\forall s\in S_{prod}} PR_s^{\,t} mf_s^{\,t} - \sum_{\forall t\in T}\sum_{\forall s\in S_{raw}} PR_s^{\,t} mf_s^{\,t} - \sum_{\forall t\in T} PRSTqst^t - \sum_{\forall t\in T} PRWTqwt^t \quad (36)$$
$$- \sum_{\forall t\in T}\sum_{\forall p\in P} fc_p^{\,t} - \sum_{\forall t\in T} ec^t$$

$$s.t. \quad mf_s^{\,t} = f_s^{\,t} MW_s \qquad \forall s\in S, \forall t\in T \quad (37)$$

$$mf_s^{\,t} \geq DEM_s^{\,t} \qquad \forall s\in S_{prod}, \forall t\in T \quad (38)$$

$$mf_s^{\,t} \leq SUP_s^{\,t} \qquad \forall s\in S_{raw}, \forall t\in T \quad (39)$$

$$\sum_{\forall s\in S_{n_{in}}} mf_s^{\,t} = \sum_{\forall s\in S_{n_{out}}} mf_s^{\,t} \qquad \forall n\in N,\ \forall t\in T \quad (40)$$

$$\sum_{\forall s\in S_{p_{in}}} mf_s^{\,t} = \sum_{\forall s\in S_{p_{out}}} mf_s^{\,t} + unrct_p^{\,t} \qquad \forall p\in P, \forall t\in T \quad (41)$$

$$\bigvee_{\forall m\in M}\begin{bmatrix} Y_{pm}^{\,t} \\ f_s^{\,t} = f_{p_{lmt}}^{\,t}\left(\frac{GMA_s^{\,t}}{GMA_{p_{lmt}}}\right) ETA_{pm}^{\,t} \\ \sum_{\forall s\in S_{p_{in}}} mf_s^{\,t} \leq CAP_{pm}^{\,t} \end{bmatrix} \qquad \forall s\in S_{p_{out}}, \forall p\in P, \forall t\in T \quad (42)$$

$$\bigvee_{\forall m\in M}\begin{bmatrix} W_{pm}^{\,t} \\ fc_p^{\,t} = FC_{pm}^{\,t} \end{bmatrix} \qquad \forall p\in P, \forall t\in T \quad (43)$$

$$q_{sk}^{\,t} = mf_s^{\,t} CP_s (T_{s_{out\,k}}^{\,t} - T_{s_{in\,k}}^{\,t}) \qquad \forall s\in S_{cold}, \forall k\in K, \forall t\in T \quad (44)$$

$$q_{sk}^{\,t} = mf_s^{\,t} CP_s (T_{s_{in\,k}}^{\,t} - T_{s_{out\,k}}^{\,t}) \qquad \forall s\in S_{hot}, \forall k\in K, \forall t\in T \quad (45)$$

$$\begin{bmatrix} X_1^{\,t} \\ qst^t = \sum_{\forall k\in K}\sum_{\forall s\in S_{cold}} q_{sk}^{\,t} \\ qwt^t = \sum_{\forall k\in K}\sum_{\forall s\in S_{hot}} q_{sk}^{\,t} \end{bmatrix} \vee \begin{bmatrix} X_2^{\,t} \\ r_k^{\,t} - r_{k-1}^{\,t} - qst_k^{\,t} + qwt_k^{\,t} = \sum_{\forall s\in S_{hot}} q_{sk}^{\,t} - \sum_{\forall s\in S_{cold}} q_{sk}^{\,t} \\ qst^t = \sum_{\forall k\in K} qst_k^{\,t} \\ qwt^t = r_{|K|}^{\,t} + \sum_{\forall k\in K} qwt_k^{\,t} \end{bmatrix} \quad \forall k\in K, \forall t\in T \quad (46)$$

$$\begin{bmatrix} V_1^t \\ ec^t = EFC_1^t \end{bmatrix} \vee \begin{bmatrix} V_2^t \\ ec^t = EFC_2^t \end{bmatrix} \qquad \forall t \in T \qquad (47)$$

$$\sum_{\forall p \in P} fc_p^t + ec^t + \sum_{\forall s \in S_{raw}} PR_s^t mf_s^t + PRSTqst^t + PRWTqwt^t \le INV^t \qquad \forall t \in T \qquad (48)$$

$$Y_{pm}^t \rightarrow \bigwedge_{\tau > t} Y_{pm}^\tau \qquad \forall p \in P, \forall t \in T, \forall m \in M \setminus m_1 \qquad (49)$$

$$W_{pm}^t \rightarrow \bigwedge_{\tau \neq t} W_{p1}^t \qquad \forall p \in P, \forall t \in T, \forall m \in M \setminus m_1 \qquad (50)$$

$$Y_{p1}^t \rightarrow W_{p1}^t \qquad \forall p \in P, \forall t \in T \qquad (51)$$

$$Y_{pm}^t \bigwedge_{\tau < t} \neg Y_{pm}^\tau \rightarrow W_{pm}^t \qquad \forall p \in P, \forall t \in T, \forall m \in M \setminus m_1 \qquad (52)$$

$$X_j^t \rightarrow \bigwedge_{\tau > t} X_j^\tau \qquad \forall t \in T, \forall j \in J \setminus j_1 \qquad (53)$$

$$V_j^t \rightarrow \bigwedge_{\tau \neq t} V_1^t \qquad \forall t \in T, \forall j \in J \setminus j_1 \qquad (54)$$

$$X_1^t \rightarrow V_1^t \qquad \forall t \in T \qquad (55)$$

$$X_j^t \bigwedge_{\tau < t} \neg X_j^\tau \rightarrow V_j^t \qquad \forall t \in T, \forall j \in J \setminus j_1 \qquad (56)$$

$$mf_s^t,\ f_s^t \in \mathbb{R}^1_+ \qquad \forall s \in S, \forall t \in T$$

$$f_{lmt,p}^t,\ unrct_p^t,\ fc_p^t \in \mathbb{R}^1_+ \qquad \forall p \in P, \forall t \in T$$

$$q_{sk}^t \in \mathbb{R}^1_+ \qquad \forall s \in S, \forall k \in K, \forall t \in T$$

$$qst^t, qwt^t, ec^t \in \mathbb{R}^1_+ \qquad \forall t \in T$$

$$qst_k^t, qwt_k^t, r_k^t \in \mathbb{R}^1_+ \qquad \forall k \in K, \forall t \in T$$

$$Y_{pm}^t, W_{pm}^t \in \{True, False\} \qquad \forall p \in P, \forall t \in T, \forall m \in M$$

$$X_j^t, V_j^t \in \{True, False\} \qquad \forall j \in J, \forall t \in T$$

The objective function (36) includes revenues from sales, costs of raw material, utility costs, as well as capital costs fc_p^t and energy costs ec^t over time periods $t \in T$. Equation (37) represents an equivalence relation between mass and molar flow rates, equations (38) and (39) ensure that mass flow rates for products and raw materials are respectively bounded by demand and supply parameters, and equations (40) and (41) serve as mass balances around nodes $n \in N$ and processes $p \in P$, respectively. The first set of disjunctions (42) selects one of the operating modes for the retrofit project $m \in M$, for every process $p \in P$, in every time period $t \in T$, where projects *m* include modifying either nothing at all ($m_1 \in M$), process conversion ($m_2 \in M$), capacity ($m_3 \in M$) or both ($m_4 \in M$). The second set of disjunctions (43) enforces the cost of the aforementioned

modifications, where capital costs are set to zero ($fc_p^t = 0$) if nothing is modified. Equations (44) and (45) serve as equivalence relations between energy and mass flow rate variables, while disjunction(s) (46) select the appropriate operating mode X_j^t $\forall j \in J$ so that X_1^t corresponds to no energy integration and X_2^t enforces the transshipment equations. Through Boolean variables V_j^t, the set of disjunctions (47) enforce the cost associated with energy reduction, where these costs are set to zero ($ec^t = 0$) if nothing is modified (V_1^t = True). Equation (48) limits the expenses for the retrofit project. Equations (51) and (52), (55) and (56) are logical conditions that connect, respectively, disjunctions (42) to (43) and disjunctions (44) to (45) with each other, and equations (49) and (50), (53) and (54) impose logical conditions between disjuncts in every set of corresponding disjunctions. Essentially, these logical equations constrain the problem such that costs associated with conversion and/or capacity are enforced exactly once for every process $p \in P$ in every time period $t \in T$, and such that costs associated with energy reduction are enforced exactly once per time period $t \in T$.

Constrained Layout

In these problems, which first appeared in [45] (see also [44]), non-overlapping process units represented by rectangles must be placed within the confines of certain designated areas formulated as circular nonlinear constraints, such that the cost of connecting these units is minimized. The nonlinearities in this set of problems are all quadratic and correspond to Euclidean-distance constraints, and the integrality gap for all instances presented is equal to 100%. Note that these problem are intentionally poorly modeled in order to have a large integrality gap and no feasible solution near the optimal solution of the continuous relaxation.

$$Min \quad Q = \sum_i \sum_j c_{ij}(delx_{ij} + dely_{ij}) \tag{57}$$

s.t.

$$delx_{ij} \geq x_i - x_j \qquad \forall i, j \in N, i < j \tag{58}$$

$$delx_{ij} \geq x_j - x_i \qquad \forall i, j \in N, i < j \tag{59}$$

$$dely_{ij} \geq y_i - y_j \qquad \forall i, j \in N, i < j \tag{60}$$

$$dely_{ij} \geq y_j - y_i \qquad \forall i, j \in N, i < j \tag{61}$$

$$\begin{bmatrix} Z^1_{ij} \\ x_i + L_i/2 \leq x_j - L_j/2 \end{bmatrix} \vee \begin{bmatrix} Z^2_{ij} \\ x_j + L_j/2 \leq x_i - L_i/2 \end{bmatrix} \vee \begin{bmatrix} Z^3_{ij} \\ y_i + H_i/2 \leq y_j - H_j/2 \end{bmatrix} \vee \begin{bmatrix} Z^4_{ij} \\ y_j + H_j/2 \leq y_i - H_i/2 \end{bmatrix} \quad \forall i, j \in N, i < j \tag{62}$$

$$\bigvee_{area \in J_i} \begin{bmatrix} W_{area,i} \\ (x_i - L_i/2 - xbar_{area})^2 + (y_i + H_i/2 - ybar_{area})^2 \leq r_{area}^2 \\ (x_i - L_i/2 - xbar_{area})^2 + (y_i - H_i/2 - ybar_{area})^2 \leq r_{area}^2 \\ (x_i + L_i/2 - xbar_{area})^2 + (y_i + H_i/2 - ybar_{area})^2 \leq r_{area}^2 \\ (x_i + L_i/2 - xbar_{area})^2 + (y_i - H_i/2 - ybar_{area})^2 \leq r_{area}^2 \end{bmatrix} \quad \forall i \in N \tag{63}$$

$$x_i \leq UB^1_i \qquad \forall i \in N \tag{64}$$

$$x_i \geq LB^1_i \qquad \forall i \in N \tag{65}$$

$$y_i \leq UB^2_i \qquad \forall i \in N \tag{66}$$

$$y_i \geq LB^2_i \qquad \forall i \in N \tag{67}$$

$$delx_{ij}, dely_{ij} \in \mathbb{R}^1_+, Z^1_{ij}, Z^2_{ij}, Z^3_{ij}, Z^4_{ij}, W_{area,i} \in \{True, False\} \qquad \forall i, j \in N, i < j$$

Every process unit is represented by a rectangle $i \in N$ that has length L_i, height H_i and coordinates (x_i, y_i), where the point of reference corresponds to the center of every rectangle. By constraining every pair of rectangles (i,j) where $(i,j \in N, i < j)$ such that no overlap occurs, we obtain a series of disjunctions with four terms each – equations (62) – where each term represents the position of rectangle i in relation to rectangle j. Furthermore, per equations (63), every rectangle i must be placed within some circular constrained area centered at $(xbar, ybar)_{area}$ with radius r_{area}, and must also satisfy the upper and lower bounds represented by inequalities (64)-(67). Finally, there is a cost c_{ij} that needs to be paid between every pair of rectangles (i,j). The objective of the problem, then, is to minimize the overall cost of laying out the rectangles (represented by the objective function (57) and the inequalities (58)-(61)) such that no two rectangles overlap and every rectangle is placed within some constrained circular area.

Bibliography

1. Akturk S., Atamturk A., Gurel S., "A strong conic quadratic reformulation for machine-job assignment with controllable processing times", *Operations Research Letters*, 37, pp. 187–191 (2009).

2. Balas E., “Disjunctive Programming”, *Annals of Discrete Mathematics*, 5, 3-51, (1979).

3. Balas E., “Disjunctive Programming and a Hierarchy of Relaxations for Discrete Continuous Optimization Problems”, SIAM J. Alg. Disc. Meth., Vol. 6, No. 3 (1985).

4. Balas E., “Disjunctive Programming: Properties of the Convex Hull of Feasible Points”, Invited paper, with a foreword by G. Cornuejols and W. Pulleybank, Discrete Applied Mathematics, 89, 3-44, (1998). (Originally MSRR #348, Carnegie Mellon University, July 1974).

5. Bertsekas D., Gallager R., “Data Networks”, Prentice-Hall, Englewood Cliffs, NJ, (1987).

6. Bonami P., “Lift-and-Project Cuts for Mixed Integer Convex Programs”, IPCO 2011, *Lecture Notes in Computer Science* 6655, pp. 52-64 (2011)

7. Bonami P., Lodi A., Tramontani A., Wiese S., "On mathematical programming with indicator constraints." *Mathematical Programming* 151.1 pp. 191-223 (2015)

8. Bonami P., Tramontani A., “Advances in CPLEX for Mixed Integer Nonlinear Optimization”, Presentation at ISMP Pittsburgh (2015)

9. Boorstyn R., Frank H., “Large-scale network topological optimization”, IEEE *Transactions on Communications*, 25 pp. 29–47 (1977).

10. Brooke A., Kendrick D., Meeraus A., Raman R., “GAMS language guide”, Version 98, GAMS Development Corporation, Washington D.C.
SBB: https://www.gams.com/latest/docs/S_SBB.html
CONOPT: https://www.gams.com/latest/docs/S_CONOPT.html

11. Castro P.M., Grossmann I.E., “Generalized Disjunctive Programming as a Systematic Modeling Framework to Derive Scheduling Formulations,” *Ind. Eng. Chem. Res* 51, 5781−5792 (2012).

12. Ceria S., Soares J., “Convex programming for disjunctive optimization”, *Mathematical Programming*, **86** (3), 595-614 (1999).

13. Duran M.A., Grossmann I.E., "An Outer Approximation Algorithm for a Class of Mixed Integer Nonlinear Programs," *Mathematical Programming* **36**, 307-339, (1986)

14. Elhedhli S., “Service system design with immobile servers, stochastic demand, and congestion”, *Manufacturing and Service Operations Management*, 8 pp. 92–97 (2006).

15. Ferris M.C., Dirkse S.P., Jagla J.H. and Meeraus A., "An Extended Mathematical Programming Framework", Computers and Chemical Engineering. Vol. 33 pp. 1973-1982 (2009); see also https://www.gams.com/latest/docs/UG_EMP.html

16. Frangioni A., Gentile C., "Perspective cuts for a class of convex 0-1 mixed integer programs", *Mathematical Programming* 106, 225–236, (2006)

17. Frangioni A., Gentile C., "SDP diagonalizations and perspective cuts for a class of nonseparable MIQP", *Operations Research Letters* 35(2), 181–185 (2007)

18. Frangioni A., Gentile C., "A computational comparison of reformulations of the perspective relaxation: SOCP vs. cutting planes", *Operations Research Letters* 37(3), 206–210 (2009)

19. Frangioni A., Gentile C., Grande E., Pacifici A., "Projected Perspective Reformulations with Applications in Design Problems" *Operations Research* 59:5, 1225-1232 (2011)

20. Furman K.C., "Private Communication to Sawaya" (2005)

21. Furman K.C., Sawaya N.W., Grossmann I.E., "An exact MINLP formulation for nonlinear disjunctive programs based on the convex hull", Presentation at 20th International Symposium on Mathematical Programming (2009)

22. Furman K.C., Sawaya N.W., Grossmann I.E., "A useful algebraic representation of convex sets using the perspective function", Presentation at MINLP Pittsburgh (2014)

23. Grossmann I.E., Westerberg A.W., Biegler L.T., "Retrofit Design of Chemical Processes," *Proceedings of Foundations of Computer Aided Process Operations* (Eds. G.V. Reklaitis and H.D. Spriggs, Elsevier), 403 (1987).

24. Grossmann I.E., Caballero J.A, Yeomans H., "Mathematical Programming Approaches for the Synthesis of Chemical Process Systems," *Korean J. Chem.Eng.*, 16, 407-426 (1999).

25. Grossmann I.E., Lee S., "Generalized Disjunctive Programming: Nonlinear Convex Hull Relaxation and Algorithms", *Computational Optimization and Applications*, 26, 83-100, (2003).

26. Gunluk O., Lee J., Weismantel R., "MINLP strengthening for separable convex quadratic transportation-cost UFL", Tech. Rep. RC24213 (W0703-042), IBM Research Division, (March 2007).

27. Gunluk O., Linderoth J., "Perspective relaxation of mixed integer nonlinear programs with indicator variables", in: A. Lodi, A. Panconesi, G. Rinaldi (eds.) Integer

Programming and Combinatorial Optimization, *Lecture Notes in Computer Science*, vol. 5035, pp. 1–16. Springer Berlin / Heidelberg (2008)

28. Gunluk O., Linderoth J., "Perspective reformulations of mixed integer nonlinear programs with indicator variables", *Mathematical Programming* 124, 183–205 (2010)

29. Gunluk, O., Linderoth J., "Perspective reformulation and Applications", Mixed Integer Nonlinear Programming, The IMA Volumes in Mathematics and its Applications Volume 154, pp 61-89, (2012)

30. Hart W.E., Laird C.D., Watson J.P., Woodruff D.L., Hackebeil G.A., Nicholson B.L., Siirola J.D. "Generalized Disjunctive Programming. In: Pyomo — Optimization Modeling in Python", *Springer Optimization and Its Applications*, vol 67, Springer (2017)

31. Hijazi H., Bonami P., Cornujols G., Ouorou A.: "Mixed Integer Non Linear Programs featuring "On/Off" constraints: convex analysis and applications" *Electronic Notes in Discrete Mathematics* 36, 1153–1160 (2010). ISCO 2010 - International Symposium on Combinatorial Optimization

32. Hiriart-Urruty J., Lemaréchal C., "Fundamentals of Convex Analysis", 2nd edition, Springer-Verlag, (2004).

33. Jackson J., Grossmann I.E., "High-Level optimization model for the retrofit planning of process networks", *Ind. Eng. Chem. Res.*, 41, 3762-3770, (2002).

34. Jackson J., Grossmann I.E., "A Disjunctive Programming Approach for the Optimal Design of Reactive Distillation Columns", *Computers and Chemical Engineering*, 25, 1661-1673, (2001).

35. Jeroslow R.G., "Representability in Mixed Integer Programming, I: Characterization Results", *Discrete Applied Mathematics*, 17, 223-243, (1987).

36. Kilinc M., Linderoth J., Luedtke J., "Lift-and-Project Cuts for Convex Mixed Integer Nonlinear Programs", *Mathematical Programming Computation*, 9, 499-526 (2017)

37. Lee S., Grossmann I.E., "New Algorithms for Nonlinear Generalized Disjunctive Programming", *Computers and Chemical Engineering*, 24, 2125-2141, (2000).

38. Lee S., Grossmann I.E.: "Erratum to "New Algorithms for Nonlinear Generalized Disjunctive Programming", *Comp. and Chem. Eng*. 24, 1153 (2001)

39. Lee S., Grossmann I.E.: "Logic-Based Modeling and Solution of Nonlinear Discrete/Continuous Optimization Problems" *Annals of Op. Res.* 139, 267–288 (2005)

40. Mendez C.A., Cerdá J., Grossmann I.E., Harjunkoski I., Fahl M., "State-Of-The-Art Review of Optimization Methods for Short-Term Scheduling of Batch Processes," *Computers and Chemical Engineering* 30, 913-946 (2006).

41. MINLPLib, A Library of Mixed-Integer and Continuous Nonlinear Programming Instances, available at http://www.minlplib.org/index.html

42. Raman R., Grossmann I.E., "Modeling and Computational Techniques for Logic Based Integer Programming", *Computers and Chemical Engineering*, 18 (7), 563-578, (1994).

43. Ravemark E. "Optimization models for design and operation of chemical batch processes", Ph.D. Thesis, ETH Zurich, (1995)

44. Sawaya N.W. "Reformulations, Relaxations and Cutting Planes for Generalized Disjunctive Programming", Ph.D. Thesis, Carnegie Mellon University (2006)

45. Sawaya N.W., Grossmann I.E., "Computational Implementation of Non-linear Convex Hull Reformulation", *Computers and Chemical Engineering*, 31, 856-866, (2007).

46. Stubbs R., Mehrotra S., "A Branch-and-Cut Method for 0-1 Mixed Convex Programming", *Mathematical Programming* 86, 515-532, (1999)

47. Trespalacios F., Grossmann I.E., "Cutting Plane Algorithm for Convex Generalized Disjunctive Programs", *INFORMS Journal on Computing*, 28, 209-222 (2016)

48. Turkay M., Grossmann I.E., "Logic-Based MINLP Algorithms for the Optimal Synthesis of Process Networks", *Computers and Chemical Engineering*, **20** (8), 959-978, (1996).

49. Vecchietti A., "LOGMIP 2.0 User Manual" (2011) http://www.logmip.ceride.gov.ar/files/pdfs/newUserManual.pdf

50. Vecchietti A., Lee S., Grossmann I.E., "Modeling of Discrete/Continuous Optimization Problems: Characterization and Formulation of Disjunctions and Their Relaxations", *Comp. and Chem. Eng.* 27, 433–448 (2003)

51. Wood A., Wollemberg B., "Power Generation Operation and Control", John Wiley and Sons, (1996).

52. Wu H., Wen H., Zhu Y., "Branch-and-Cut Algorithmic Framework for 0-1 Mixed-Integer Convex Nonlinear Programs", *Ind. and Eng. Chem. Res.* 48, 9119–9127 (2009)

53. Zhu Y., Kuno T., “A Disjunctive Cutting-Plane Based Branch-and-Cut Algorithm for 0-1 Mixed-Integer Convex Nonlinear Programs”, *Ind. and Eng. Chem. Res.* 45(1), 187–196 (2006)