



HAL
open science

Handling separable non-convexities using disjunctive cuts

Claudia d'Ambrosio, Jon Lee, Daphne Skipper, Thomopoulos Dimitri

► **To cite this version:**

Claudia d'Ambrosio, Jon Lee, Daphne Skipper, Thomopoulos Dimitri. Handling separable non-convexities using disjunctive cuts. Lecture Notes in Computer Science, 2020. hal-03008730

HAL Id: hal-03008730

<https://hal.science/hal-03008730v1>

Submitted on 16 Nov 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Handling separable non-convexities using disjunctive cuts^{*}

Claudia D’Ambrosio¹, Jon Lee², Daphne Skipper³, and Dimitri Thomopulos⁴

¹ LIX CNRS, École Polytechnique, Institut Polytechnique de Paris, Palaiseau, France
`dambrosio@lix.polytechnique.fr`

² Dept. of IOE, University of Michigan, Ann Arbor, Michigan, USA
`jonxlee@umich.edu`

³ Dept. of Mathematics, U.S. Naval Academy, Annapolis, Maryland, USA
`skipper@usna.edu`

⁴ Dept. of Energy, Process and System Engineering, Università di Pisa, Italy
`dimitri.thomopulos@unipi.it`

Abstract. D’Ambrosio, Lee, and Wächter (2009, 2012) introduced an algorithmic approach for handling separable non-convexities in the context of global optimization. That algorithmic framework calculates lower bounds (on the optimal min objective value) by solving a sequence of convex MINLPs. We propose a method for addressing the same setting, but employing disjunctive cuts (generated via LP), and solving instead a sequence of convex NLPs. We present computational results which demonstrate the viability of our approach.

Keywords: MINLP, separable nonconvexity, disjunctive cuts

Introduction

We consider non-convex global-optimization problems of the form

$$\begin{aligned} \min \quad & f_0(\mathbf{x}) + \sum_{j=1}^n g_{0,j}(\mathbf{x}_j) \\ \text{subject to:} \quad & \\ & f_i(\mathbf{x}) + \sum_{j=1}^n g_{i,j}(\mathbf{x}_j) \leq 0, \text{ for } i = 1, \dots, m; \\ & L_j \leq \mathbf{x}_j \leq U_j, \text{ for } j = 1, \dots, n, \end{aligned}$$

where the f_i are convex ($i = 0, \dots, m$) and the $g_{i,j}$ are univariate but not necessarily convex ($i = 0, \dots, m; j = 1, \dots, n$). So all of the non-convexity is assumed to be separable. We assume that all of the functions are continuous

^{*} C. D’Ambrosio and D. Thomopulos were supported by a public grant as part of the Investissement d’avenir project, reference ANR-11-LABX-0056-LMH, LabEx LMH. This research benefited from the support of the FMJH Program PGMO and from the support of EDF. J. Lee was supported in part by ONR grant N00014-17-1-2296 and LIX, École Polytechnique. D. Skipper was supported in part by ONR grant N00014-18-W-X00709.

and sufficiently smooth. We may have that some of the variables are restricted to be integers, but this does not directly matter for our approach.

An algorithm for this class of problems was studied in [8,9]. A key aspect of that algorithm is to develop and refine a convex Mixed Integer Nonlinear Program (MINLP) relaxation. The algorithm is based on making a piecewise-convex under-estimator for each $g_{i,j}$, by identifying the concave intervals and using secant under-estimation on them, to get a convex MINLP relaxation — binary variables are used to manage the piecewise functions. Refinement of the piecewise-convex under-estimator is carried out by adding further breakpoints on the concave intervals. After each new breakpoint, a convex MINLP is solved.

Our goal is to use the same starting relaxation, but to relax its integrality, resulting in a continuous, convex Nonlinear Program (NLP). At each iteration, rather than using further breakpoints to tighten the relaxation, we use a Linear Program (LP) to generate a cut, an inequality that is not in the original formulation, but is valid for the formulation and tightens the relaxation. In particular, we iteratively introduce disjunctive cuts as a much more efficient means of improving the NLP relaxation. The efficiency is realized by solving an LP and a convex NLP at each iteration, rather than an expensive convex MINLP.

In §1, we describe our piecewise-convex under-estimation model. For ease of exposition and economy of notation, we confine our attention to a single univariate function g on domain $[L, U]$. So what we propose applies separately to each of the $g_{i,j}$ defined above. In §2, we describe a method for tightening our under-estimator using disjunctive cuts, plus some improvements to our basic approach. In §3, we describe our successful computational results. In §4, we make some brief conclusions and indicate some plans for future work.

1 A piecewise-convex under-estimator

Our framework is similar to that of [8,9], with our focus being on optimization models in which all of the non-convexity is separable, with each (univariate) summand being continuous on a finite interval and piecewise sufficiently smooth. In our development, we focus on how to handle each such non-convex summand.

Toward that end, we consider treating a non-convex univariate function $g : [L, U] \rightarrow \mathbb{R}$, where $L < U$ are real. We assume that g is continuous on $[L, U]$ and g is piecewise-defined over a finite set of $T \geq 1$ closed subintervals of $[L, U]$ so that g is twice continuously differentiable on each associated open subinterval. To formalize our notation, we assume that g is sub-divided by points p_i ,

$$L =: p_0 < p_1 < p_2 < \dots < p_T := U,$$

so that g is twice continuously differentiable on (p_{i-1}, p_i) , for $i = 1, 2, \dots, T$.

In this section, we develop a convex under-estimator for g (essentially the same as was used in [9]). Toward this end, we assume that g is either convex or concave on each interval $I_i := [p_{i-1}, p_i]$, for $i = 1, 2, \dots, T$. We note that g could be convex or concave on consecutive intervals. Let $H := \{1, 2, \dots, T\}$ be the set of indices of these intervals, which we partition into $\bar{H} := \{i \in H :$

g is convex on I_i and $\hat{H} := \{i \in H : g \text{ is not convex on } I_i\}$. Note that g is concave on interval I_i , for $i \in \hat{H}$; and if g is linear on interval I_i , then $i \in \hat{H}$.

We employ a set of binary variables z_i , $i = 1, \dots, T-1$, in order to express g as a separable function of continuous “interval variables” δ_i , for $i = 1, \dots, T$, using the so-called “delta method” (see [4, pp. 282–283] and [18, Chapter 3], for example). We want to write the variable $x \in [L, U]$ as a function of the interval variables in a certain disciplined manner. Specifically,

$$x = p_0 + \sum_{i=1}^T \delta_i, \quad (1)$$

where for each $i = 1, \dots, T$:

$$0 \leq \delta_i \leq p_i - p_{i-1}, \text{ and} \quad (2)$$

$$\delta_i > 0 \implies \delta_j = p_j - p_{j-1}, \text{ for } 1 \leq j < i. \quad (3)$$

Condition (3) dictates that if δ_i is positive, then all δ_j “to the left” (i.e., $j < i$) should be at their upper bounds (as specified by (2)). In this way, every value of $x \in [L, U]$ is associated with a unique solution of (1).

We accomplish this with the constraints:

$$z_1(p_1 - p_0) \leq \delta_1 \leq p_1 - p_0; \quad (4)$$

$$z_i(p_i - p_{i-1}) \leq \delta_i \leq z_{i-1}(p_i - p_{i-1}), \text{ for } i = 2, 3, \dots, T-1; \quad (5)$$

$$0 \leq \delta_T \leq z_{T-1}(p_T - p_{T-1}); \quad (6)$$

$$z_i \in \{0, 1\}, \text{ for } i = 1, \dots, T-1, \quad (7)$$

which drive the correct behavior of the δ_i variables. Note that for a particular $\bar{x} \in [L, U]$, the values of the (ordered) binary z variables form a sequence of 1s followed by a sequence of 0s, where the index $i_{\bar{x}}$ of the first 0 indicates the interval $I_{i_{\bar{x}}}$ that contains \bar{x} , so that

$$\delta_i = \begin{cases} p_i - p_{i-1}, & \text{for } i < i_{\bar{x}}; \\ \bar{x} - p_{i-1}, & \text{for } i = i_{\bar{x}}; \\ 0, & \text{for } i > i_{\bar{x}}. \end{cases}$$

Now we can express $g(x)$ as a separable function of the δ -variables,

$$g(x) = \sum_{i=1}^T g(p_{i-1} + \delta_i) - \sum_{i=1}^{T-1} g(p_i),$$

giving us access to g on the individual subintervals of $[L, U]$. Our goal now is to give a piecewise-convex under-estimator for g . So we use g as its own convex under-estimator on the subintervals where g is convex, and replace g with a secant under-estimator on the subintervals where g is not convex. Our piecewise-convex under-estimator for g is

$$\underline{g}(x) = \sum_{i=1}^T y_i + g(p_0), \quad (8)$$

with

$$y_i \geq \left(\frac{g(p_i) - g(p_{i-1})}{p_i - p_{i-1}} \right) \delta_i, \quad \text{for } i \in \hat{H}, \quad (\text{secant under-estimator}) \quad (9)$$

and

$$y_i \geq g(p_{i-1} + \delta_i) - g(p_{i-1}), \quad \text{for } i \in \check{H}. \quad (\text{under-estimation by } g \text{ itself}) \quad (10)$$

Notice that in the secant under-estimation (9), the right-hand side is linear in δ_i , and is 0 for $\delta_i = 0$ and is $g(p_i) - g(p_{i-1})$ when $\delta_i = p_i - p_{i-1}$.

[10] uses a related approach where concave separable quadratics are extracted from indefinite non-separable quadratics, and then under-estimated by secants.

2 Disjunctive cuts

Our under-estimator \underline{g} can be quite far from g on the concave subintervals of $[L, U]$. In [9], the algorithmic strategy is to branch at points interior to I_i , for $i \in \hat{H}$, to get tighter lower bounds on the concave segments of g . In doing so, we are faced with solving a convex MINLP at every stage of the overall algorithm.

Our strategy is to use disjunctive cuts to iteratively tighten our formulation, with the goal of reducing the number of branching nodes that would be required for globally optimizing. In contrast to the algorithm of [9], we solve convex NLPs rather than convex MINLPs, thereby decreasing the computational burden as we seek to improve the global lower bound in the overall solution process.

2.1 Disjunctive cuts in general

We take an aside now to review disjunctive cuts (see [2]). This sub-section is self contained, and the notation is not meant to coincide with its earlier usage. For example, here we have $\mathbf{x} \in \mathbb{R}^n$. Let

$$\begin{aligned} \mathcal{P} &:= \{\mathbf{x} \in \mathbb{R}^n : A\mathbf{x} \leq b\}, \\ \mathcal{D}_1 &:= \{\mathbf{x} \in \mathbb{R}^n : D^1\mathbf{x} \leq f^1\}, \\ \mathcal{D}_2 &:= \{\mathbf{x} \in \mathbb{R}^n : D^2\mathbf{x} \leq f^2\}. \end{aligned}$$

In its typical usage, \mathcal{P} would be (a subset of) the inequalities describing a polyhedral relaxation of a non-convex model, and $\mathcal{D}_1 \vee \mathcal{D}_2$ would be a disjunction (i.e., $\mathbf{x} \in \mathcal{D}_1$ or $\mathbf{x} \in \mathcal{D}_2$) that is valid for the non-convex model.

It is well known that we can characterize the set of all linear cuts $\alpha^\top \mathbf{x} \leq \beta$ that are valid for $\mathbb{X} := \text{conv}((\mathcal{P} \cap \mathcal{D}_1) \cup (\mathcal{P} \cap \mathcal{D}_2))$, via the cone

$$\mathbb{K} := \left\{ \begin{pmatrix} \alpha \\ \beta \end{pmatrix} \in \mathbb{R}^{n+1} : \exists \pi_1, \pi_2, \gamma_1, \gamma_2 \geq \mathbf{0}, \text{ with} \right. \\ \left. \alpha^\top = \pi_1^\top A + \gamma_1^\top D^1, \alpha^\top = \pi_2^\top A + \gamma_2^\top D^2, \beta \geq \pi_1^\top b + \gamma_1^\top f^1, \beta \geq \pi_2^\top b + \gamma_2^\top f^2 \right\}.$$

Notice how, for $\pi_1, \pi_2, \gamma_1, \gamma_2 \geq \mathbf{0}$, the inequality $\pi_\ell^\top (A\mathbf{x} \leq b) + \gamma_\ell^\top (D^\ell \mathbf{x} \leq f^\ell)$, which we can alternatively write as $(\pi_\ell^\top A + \gamma_\ell^\top D^\ell)\mathbf{x} \leq (\pi_\ell^\top b + \gamma_\ell^\top f^\ell)$, is valid for $\mathcal{P} \cap \mathcal{D}_\ell$, for $\ell = 1, 2$. This implies that if $\alpha^\top = \pi_\ell^\top A + \gamma_\ell^\top D^\ell$ and $\beta \geq \pi_\ell^\top b + \gamma_\ell^\top f^\ell$, then $\alpha^\top \mathbf{x} \leq \beta$ is valid for $\mathcal{P} \cap \mathcal{D}_\ell$.

The so-called ‘‘cut-generating linear program’’ with respect to an $\bar{\mathbf{x}} \in \mathcal{P}$ is

$$\max \left\{ \alpha^\top \bar{\mathbf{x}} - \beta : \begin{pmatrix} \alpha \\ \beta \end{pmatrix} \in \mathbb{K} \right\}. \quad (\text{CGLP})$$

Because \mathbb{K} is a cone, CGLP either has a maximum value of 0 or is unbounded. If the maximum value is 0, then $\bar{\mathbf{x}} \in \mathbb{X}$. Otherwise, the LP is unbounded, and any direction $\begin{pmatrix} \alpha \\ \beta \end{pmatrix}$ with $\alpha^\top \bar{\mathbf{x}} - \beta > 0$ gives a violated valid cut $\alpha^\top \mathbf{x} \leq \beta$.

Typically, we bound \mathbb{K} by using a so-called ‘‘normalization constraint’’, so that CGLP always has a finite optimum. See [11] for an understanding of how to properly solve CGLP using an appropriate normalization. For an example of the use of a CGLP that is very relevant to our setting, see [15,16,17] for treating indefinite non-separable quadratics, and [3] for an implementation of some of those ideas more broadly.

2.2 Our disjunction

Now we return to our setting (see §1). To improve upon the secant under-estimators on the concave portions of g , we generate cuts based on disjunctions of the following form. Choosing $\psi \in (0, p_k - p_{k-1})$ for some $k \in \hat{H}$, we let

$$\begin{aligned} \mathcal{D}_1^k &:= \left\{ (y_k, \delta_k, *) : \underbrace{\delta_k \leq \psi}_{\lambda_1}, \underbrace{y_k \geq \left(\frac{g(p_{k-1} + \psi) - g(p_{k-1})}{\psi} \right) \delta_k}_{\omega_{1,k}} \right\} \text{ and} \\ \mathcal{D}_2^k &:= \left\{ (y_k, \delta_k, *) : \underbrace{\delta_k \geq \psi}_{\lambda_2}, \right. \\ &\quad \left. \underbrace{y_k \geq \left(\frac{g(p_k) - g(p_{k-1} + \psi)}{p_k - (p_{k-1} + \psi)} \right) (\delta_k - \psi) + g(p_{k-1} + \psi) - g(p_{k-1})}_{\omega_{2,k}} \right\}, \end{aligned}$$

where ‘‘*’’ is a place-holder for additional variables in the CGLP (the red annotations define variables that will play the role of the γ -variables in the CGLP)

This disjunction corresponds to making a pair of secant under-estimators, one on $[p_{k-1}, p_{k-1} + \psi]$ and one on $[p_{k-1} + \psi, p_k]$. But note that we are not advocating for refining the set of intervals à la [9]; rather, we want to include

some of the lower-bounding power of such a branching refinement (which would come at the substantial cost of solving a further convex MINLP) with a cut (which only leads to a further convex NLP).

An important consideration is which additional variables and constraints to include in our CGLP, playing the role of \mathcal{P} in the previous section. We also need to address the selection of k and ψ .

Suppose that we have a “current solution” to a relaxation, including values

$$\bar{\delta}_1, \bar{\delta}_2, \dots, \bar{\delta}_T; \bar{z}_1, \bar{z}_2, \dots, \bar{z}_{T-1}; \bar{y}_1, \bar{y}_2, \dots, \bar{y}_T. \quad (11)$$

We choose $k \in \hat{H}$ so that the secant approximation for g on I_k is “bad” at the current solution. This means that $g(p_{k-1} + \bar{\delta}_k) \gg \bar{y}_k$, so we choose $k \in \hat{H}$ so that the difference $g(p_{k-1} + \bar{\delta}_k) - \bar{y}_k$ is maximized. Then we choose $\psi = p_{k-1} + \bar{\delta}_k$, in the context of the disjunction $\mathcal{D}_1^k \vee \mathcal{D}_2^k$.

Now, we turn to the more subtle topic of describing which variable and inequalities comprise $\mathcal{P} = \{\mathbf{x} : A\mathbf{x} \leq b\}$. The variable-space for the CGLP should include all δ -, z -, and y -variables corresponding to g and the following inequalities (the red annotations define variables that will play the role of the π -variables in the CGLP):

$$\begin{aligned} z_i(p_i - p_{i-1}) - \delta_i &\leq 0, & \text{for } i = 1, 2, \dots, T-1; & \mu_{\ell,i} \\ -\delta_T &\leq 0; & & \mu_{\ell,T} \\ \delta_1 &\leq p_1 - p_0; & & \nu_{\ell,1} \\ -z_{i-1}(p_i - p_{i-1}) + \delta_i &\leq 0, & \text{for } i = 2, 3, \dots, T; & \nu_{\ell,i} \\ -z_1 &\leq 0; & & \rho_{\ell,0} \\ z_{T-1} &\leq 1; & & \rho_{\ell,1} \\ \left(\frac{g(p_i) - g(p_{i-1})}{p_i - p_{i-1}}\right) \delta_i - y_i &\leq 0, & \text{for } i \in \hat{H} \setminus \{k\}. & \omega_{\ell,i} \end{aligned}$$

Note that we omit the secant inequality for I_k because it is implied by our disjunctive secants.

We also need to include something to represent the convex pieces of g . That is, we would like to use y_i for $i \in \hat{H}$ in a constraint similar to (10), but it should be linearized at the point $(\bar{\delta}_i, \bar{y}_i)$ for use in an LP:

$$g(p_{i-1} + \bar{\delta}_i) - g(p_{i-1}) + g'(p_{i-1} + \bar{\delta}_i)(\delta_i - \bar{\delta}_i) - y_i \leq 0, \quad \text{for } i \in \hat{H}. \quad \omega_{\ell,i} \quad (12)$$

Note that we use this linearization only in the CGLP. We propose to solve the overall model relaxation as a convex NLP, including (10), directly.

Additional possibilities for inclusion in \mathcal{P} for forming CGLP are:

- linearizations of (10) at other points;
- the variables (δ , z , and y) and constraints corresponding to other univariate functions in the formulation, but maybe only those univariate functions that operate on the same x -variable as g .

The CGLP is designed to seek to separate the current solution (to our relaxation) using a linear inequality. In particular, we have values (11), and so we may seek an inequality of the form

$$\sum_{i=1}^T a_i \delta_i + \sum_{i=1}^{T-1} b_i z_i + \sum_{i=1}^T c_i y_i \leq \beta \quad (*)$$

that is violated by (11). We do this by solving our version of the CGLP:

$$\begin{aligned} \max \quad & \sum_{i=1}^T a_i \bar{\delta}_i + \sum_{i=1}^{T-1} b_i \bar{z}_i + \sum_{i=1}^T c_i \bar{y}_i - \beta \\ \text{subject to:} \end{aligned} \quad (13)$$

$$a_i = -\mu_{\ell,i} + \nu_{\ell,i} + \left(\frac{g(p_i) - g(p_{i-1})}{p_i - p_{i-1}} \right) \omega_{\ell,i}, \quad \text{for } i \in \hat{H} \setminus \{k\}, \ell = 1, 2; \quad (14)$$

$$a_i = -\mu_{\ell,i} + \nu_{\ell,i} + g'(p_{i-1} + \bar{\delta}_i) \omega_{\ell,i}, \quad \text{for } i \in \check{H}, \ell = 1, 2; \quad (15)$$

$$a_k = -\mu_{1,k} + \nu_{1,k} + \left(\frac{g(p_{k-1} + \psi) - g(p_{k-1})}{\psi} \right) \omega_{1,k} + \lambda_1; \quad (16)$$

$$a_k = -\mu_{2,k} + \nu_{2,k} + \left(\frac{g(p_k) - g(p_{k-1} + \psi)}{p_k - p_{k-1} - \psi} \right) \omega_{2,k} - \lambda_2; \quad (17)$$

$$b_1 = (p_1 - p_0) \mu_{\ell,1} - (p_2 - p_1) \nu_{\ell,2} - \rho_{\ell,0}, \quad \text{for } \ell = 1, 2; \quad (18)$$

$$b_i = (p_i - p_{i-1}) \mu_{\ell,i} - (p_{i+1} - p_i) \nu_{\ell,i+1}, \quad \text{for } i = 2, \dots, T-2, \ell = 1, 2; \quad (19)$$

$$b_{T-1} = (p_{T-1} - p_{T-2}) \mu_{\ell,T-1} - (p_T - p_{T-1}) \nu_{\ell,T} + \rho_{\ell,1}, \quad \text{for } \ell = 1, 2; \quad (20)$$

$$c_i = -\omega_{\ell,i}, \quad \text{for } i = 1, \dots, T, \ell = 1, 2; \quad (21)$$

$$\beta \geq (p_1 - p_0) \nu_{1,1} + \rho_{1,1} \quad (22)$$

$$- \sum_{i \in \check{H}} (g(p_{i-1} + \bar{\delta}_i) - g(p_{i-1}) - g'(p_{i-1} + \bar{\delta}_i) \bar{\delta}_i) \omega_{1,i} + \psi \lambda_1;$$

$$\beta \geq (p_1 - p_0) \nu_{2,1} + \rho_{2,1} \quad (23)$$

$$- \sum_{i \in \check{H}} (g(p_{i-1} + \bar{\delta}_i) - g(p_{i-1}) - g'(p_{i-1} + \bar{\delta}_i) \bar{\delta}_i) \omega_{2,i}$$

$$- \left(-\psi \frac{g(p_k) - g(p_{k-1} + \psi)}{p_k - (p_{k-1} + \psi)} + g(p_{k-1} + \psi) - g(p_{k-1}) \right) \omega_{2,k} - \psi \lambda_2;$$

$$\mu_{\ell,i} \geq 0, \quad \text{for } i = 1, \dots, T, \ell = 1, 2; \quad (24)$$

$$\nu_{\ell,i} \geq 0, \quad \text{for } i = 1, \dots, T, \ell = 1, 2; \quad (25)$$

$$\rho_{\ell,j} \geq 0, \quad \text{for } j = 0, 1, \ell = 1, 2; \quad (26)$$

$$\omega_{\ell,i} \geq 0, \quad \text{for } i = 1, \dots, T, \ell = 1, 2; \quad (27)$$

$$\lambda_\ell \geq 0, \quad \text{for } \ell = 1, 2. \quad (28)$$

2.3 Possible improvements and another disjunction: inviting the z -variables to the party

In this section, we describe three ideas, the first two of which we have tested, for inviting the z -variables to the disjunctive party.

Improvement to \mathcal{P} . Model relaxations similar to ours can be solved much faster when the convex constraints (10) are linearized (and thus weakened, unfortunately) so that we are in the realm of LP rather than NLP. [6,7] suggests that these constraints can be tightened via a “perspective reformulation” (see

[12]) and then linearized and thus weakened, but hopefully for a net benefit. In particular, (10) becomes:

$$y_i \geq z_{i-1} g(p_{i-1} + \delta_i/z_{i-1}) - z_{i-1} g(p_{i-1}), \quad \text{for } i \in \check{H}, \quad (29)$$

(where $z_0 := 1$ if $1 \in \check{H}$). Inequalities (29) are convex by the perspective-reformulation definition and can be linearized with standard techniques. One could carry this out directly, but we proceed as follows: we linearize (10) as (12) only for the conceptual purpose of using within \mathcal{P} to set up our CGLP; that is, we strengthen (12) via the perspective reformulation, and then linearize only to strengthen our CGLP.

Improvement to $\mathcal{D}_1^k \vee \mathcal{D}_2^k$. We can attempt to improve (i.e., tighten) the representation of our disjunction that was based on secant inequalities. We can see that $\delta_k \leq \psi \implies z_k = 0$, and $\delta_k \geq \psi \implies z_{k-1} = 1$. So we could include $z_k = 0$ in \mathcal{D}_1^k and $z_{k-1} = 1$ in \mathcal{D}_2^k , which implies a slightly different and potentially stronger CGLP.

Another disjunction. Related to the last idea, we can make a direct disjunction on a z_k that we could use to generate disjunctive cuts. That is, we could use the disjunction

$$\mathcal{Z}_0^k := \{(z_k, *) : z_k = 0\} \vee \mathcal{Z}_1^k := \{(z_k, *) : z_k = 1\},$$

choosing k based on \bar{z}_k being fractional. It is well known that a diversity of cuts can be quite effective, so we propose to use disjunctive cuts based on $\mathcal{Z}_0 \vee \mathcal{Z}_1$ as well as ones based on $\mathcal{D}_1 \vee \mathcal{D}_2$. Note that in forming the CGLP based on the disjunction $\mathcal{Z}_0 \vee \mathcal{Z}_1$, we should include in \mathcal{P} the secant inequality for I_k (i.e., (9) for $i = k$).

3 Computational experiments

As a proof of concept, we tested our ideas on challenging non-linear continuous knapsack-type problems of the form:

$$\begin{aligned} & \min f(x_1, \dots, x_n) + g(x_0) \\ & \text{subject to:} \\ & C_L \leq \sum_{j=0}^n r_j x_j \leq C_U ; \\ & 0 \leq x_j \leq 1, \quad j = 0, 1, \dots, n, \end{aligned}$$

where $C_L, C_U, r_j \in \mathbb{R}^+$ for $j = 0, 1, \dots, n$, $f : \mathbb{R}^n \mapsto \mathbb{R}$ is convex, and the single univariate g is a highly non-convex function. In particular, we defined $f(x_1, \dots, x_n)$ as a convex quadratic, namely $\sum_{i=1}^n \sum_{j=1}^n Q_{ij} x_i x_j$, where $Q := \tilde{Q}^\top \tilde{Q}$ and \tilde{Q} is randomly generated. Note that f is convex by the definition of Q . The convexity of f and the minimization objective drive (x_1, \dots, x_n) toward $\mathbf{0}$

at optimality. By choosing $0 < C_L < C_U$ appropriately, we can drive the solution into a range where x_0 is in a concave portion of g , thus stressing our relaxation.

We generated ten test instances. For g , we used two types of functions, designed to be difficult (see Figure 1):

1. $g(x_0) := s \cdot x_0 - \frac{2 \cos(h\pi x_0)}{h\pi} - x_0 \sin(h\pi x_0)$, and
2. $g(x_0) := d(\sin((h\pi x_0) + 2e\pi + \sin^{-1}(\frac{m}{d}))) + m((h\pi x_0) + 2e\pi + \sin^{-1}(\frac{m}{d}))^2 + v((h\pi x_0) + 2e\pi + \sin^{-1}(\frac{m}{d}))$,

where s is randomly generated (with a uniform distribution) on $[-4, +4]$, h on $[7, 15]$, d on $\{100, 200, 300\}$, e on $\{-3, -2\}$, m on $\{-2, -1\}$, v on $\{10, 15, 20\}$. Entries in the vector r are uniformly randomly generated on $[1, 200]$. C_L and C_U are chosen so that the value of x_0 , when $(x_1, \dots, x_n) = \mathbf{0}$, is in a centrally-located concave interval of g . The first type of example only puts local stress on the relaxation; the second type of example puts global stress on the relaxation, as interpolating across the full domain of x_0 can result in a very poor underestimate of g in the range where x_0 is feasible (when $(x_1, \dots, x_n) = \mathbf{0}$).

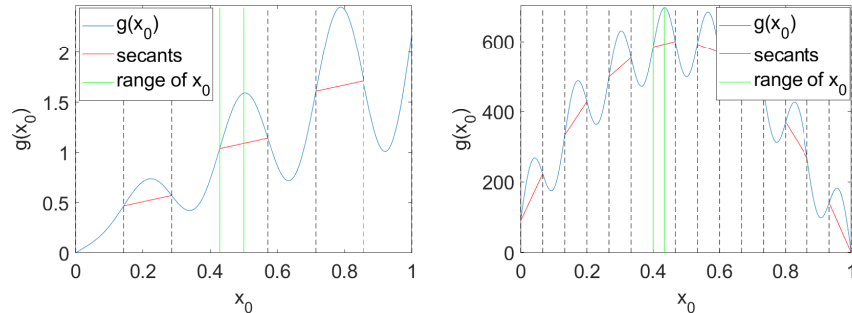


Fig. 1. Example of the function of type 1 (left) and type 2 (right).

We generated 5 instances with $n = 3$ for each of the two types of functions g .

All experiments were performed on a single machine equipped with an Intel Xeon E5649 processor clocked at 2.53 GHz and 50 GB RAM. We used open-source solvers like IPOPT 3.12.8, Bonmin 1.8.6, and Couenne 0.5 to solve convex NLPs, convex MINLPs, and non-convex (MI)NLPs, respectively. As for the MILPs (solved within Bonmin and Couenne), we used IBM CPLEX 12.6.

We tested and compared three strategies: the basic approach we have outlined (called “Alg” in the tables), the improvement of the relaxation with the Perspective Reformulation (29) (“Alg+PC”), and the improvement using $\mathcal{D}_1^k \vee \mathcal{D}_2^k$ (“Alg+IDC”). We have not yet tested the modifications in tandem.

Table 1 summarizes the performance of the algorithm when employing each of the three cut-generating strategies, and applied to each of our ten test instances. In particular, the algorithm iteratively adds violated cuts to the convex NLP relaxation, then solves the strengthened convex MINLP. For each instance and strategy, we present the objective-function value of the convex MINLP, along

with the number of iterations of cut generation (an iteration limit of 300 was set). The strategy Alg+PC most frequently requires the fewest iterations among the three strategies, making it the fastest. Among the five instances for which the Alg+PC strategy reaches the iteration limit, the final solution obtained with this strategy is better (i.e., greater lower bound) in three cases, and equivalent in one of the two remaining instances. Strategy Alg+IDC sometimes finds the best solution, but often requires the most iterations.

inst.	strategy	CMINLP	#iter.	inst.	strategy	CMINLP	#iter.
1	Alg	1.07	18	6	Alg	502.44	300
1	Alg+PC	1.07	12	6	Alg+PC	506.85	300
1	Alg+IDC	1.11	69	6	Alg+IDC	502.44	300
2	Alg	2.09	36	7	Alg	502.48	300
2	Alg+PC	2.15	22	7	Alg+PC	505.81	300
2	Alg+IDC	2.09	38	7	Alg+IDC	502.92	300
3	Alg	2.56	221	8	Alg	246.46	300
3	Alg+PC	2.57	60	8	Alg+PC	252.14	300
3	Alg+IDC	2.66	300	8	Alg+IDC	246.46	300
4	Alg	-2.10	300	9	Alg	504.40	300
4	Alg+PC	-2.10	29	9	Alg+PC	504.40	300
4	Alg+IDC	-2.13	26	9	Alg+IDC	504.40	300
5	Alg	2.70	73	10	Alg	587.70	300
5	Alg+PC	2.72	63	10	Alg+PC	587.70	300
5	Alg+IDC	2.70	300	10	Alg+IDC	589.60	300

Table 1. Results on the solution of the convex MINLP obtained after strengthening the convex NLP relaxation by adding iteratively the cuts based on the different strategies

In Table 2 we compare the strategies by defining a measure of the impact of disjunctive cuts. In particular, for each instance (one per row) and each strategy (one per block of 3 columns), we display the values of $\text{GAP1} := 100 \cdot \frac{\text{GO}-\text{CMINLP}}{\text{GO}-\text{NLP}}$, $\text{GAP2} := 100 \cdot \frac{\text{GO}-\text{MINLP}}{\text{GO}-\text{NLP}}$, and $\text{GAP3} := 100 \cdot \frac{\text{GO}-\text{CNLP}}{\text{GO}-\text{NLP}}$, where GO is the global optimum value of the non-convex MINLP problem, NLP is the optimal value of the convex NLP relaxation, MINLP is the optimal value of the convex MINLP relaxation, CNLP is the optimal value of the convex NLP relaxation after applying the disjunctive cuts, and CMINLP is the optimal value of the convex MINLP relaxation after applying disjunctive cuts.

Most importantly, GAP3 measures the effectiveness of disjunctive cuts on closing the gap in the convex NLP relaxation (initial gap is 100). We can see that with all three strategies substantial gap is often closed using the disjunctive cuts. Comparing across the approaches, Alg+IDC usually gives the greatest effect from the disjunctive cuts.

By imposing integrality and solving a single convex MINLP, we can compare gaps obtained via convex MINLP versus convex NLP: GAP2 vs 100 (no disjunctive cuts) and GAP1 vs GAP3 (with disjunctive cuts). Generally, we obtain much smaller gaps in the first of each pair, but at the cost of solving a convex MINLP. Note that GAP2 is the same for Alg and Alg+IDC, because they use the same NLP relaxation; while Alg+PC uses a different relaxation, so

inst.	Alg			Alg+PC			Alg+IDC		
	GAP1	GAP2	GAP3	GAP1	GAP2	GAP3	GAP1	GAP2	GAP3
1	4.25	4.25	81.43	8.38	8.38	79.01	2.42	4.25	29.67
2	15.62	15.62	80.78	22.70	27.06	79.72	15.62	15.62	79.54
3	4.57	4.57	98.28	8.27	8.88	79.39	0.83	4.57	18.82
4	0.34	1.96	80.37	0.69	3.96	74.37	1.66	1.96	76.91
5	88.34	88.34	99.38	94.36	94.47	99.21	88.34	88.34	93.10
6	7.34	7.34	93.26	12.33	13.63	92.24	7.34	7.34	41.12
7	8.20	8.20	89.62	14.31	14.88	92.74	8.16	8.20	42.62
8	9.77	9.77	93.16	17.02	17.61	90.73	9.77	9.77	50.65
9	7.32	7.32	92.70	15.19	15.19	86.48	7.32	7.32	74.71
10	4.08	4.08	93.85	8.31	8.31	88.87	3.93	4.08	47.93

Table 2. Results per instance using the basic algorithm (Alg), the algorithm with perspective cuts (Alg+PC), and the algorithm with improved disjunctive cuts (Alg+IDC).

GAP2 is different. Comparing GAP1 vs GAP2, we see several instances where the disjunctive cuts close much of the remaining gap when included in the single convex MINLP; e.g., instance 4 using Alg or Alg+PC, and instances 1 and 3, using Alg+IDC. Again, comparing across the approaches, Alg+IDC is usually the most effective strategy of the three.

To conclude, strategy Alg+PC seems to be the most promising both on the speed of convergence and on the strengthening of the convex MINLP problem, while strategy Alg+IDC gives the strongest disjunctive cuts.

As a future direction, we plan to explore a hybrid Alg+PC+IDC strategy, and to experiment with including a diversity of disjunctive cuts (e.g., those produced from the disjunction described at the end of §2). Finally, we have concentrated on lower bounds, but we plan to experiment with incorporating branching to get a complete algorithm for reaching global optimality.

4 Conclusions and outlook

We have introduced a technique for using disjunctive cuts so as to improve a framework for handling mathematical-optimization models where the non-convexities are separable. The efficiency is realized by solving an LP and a convex NLP at each iteration, rather than an expensive convex MINLP. We have presented preliminary computational work to demonstrate the promise of our methodology. Further work will center on enhancing and tuning the method to efficiently handle models coming from real-world applications as the minimization of the ripple effect (see, e.g., [13]) or the optimization of the design of bandpass and low-pass filters (see, e.g., [14]). Optimization models with non-convexities are indeed typical of electromagnetic problems where the outputs are often characterized by a strongly sinusoidal or generally periodic trend (see [1,5]).

References

1. C. K. Alexander and M. N. Sadiku. *Fundamentals of electric circuits*. McGraw-Hill Education, 2000.
2. E. Balas. *Disjunctive programming*. Springer, Cham, 2018.
3. P. Belotti. Disjunctive cuts for nonconvex MINLP. In J. Lee and S. Leyffer, editors, *Mixed Integer Nonlinear Programming*, pages 117–144. Springer, New York, 2012.
4. S.P. Bradley, A.C. Hax, and T.L. Magnanti. *Applied Mathematical Programming*. Addison-Wesley, 1977.
5. M. Ceraolo and D. Poli. *Fundamentals of electric power engineering: from electromagnetics to power systems*. John Wiley & Sons, 2014.
6. C. D’Ambrosio, A. Frangioni, and C. Gentile. Strengthening convex relaxations of mixed integer non linear programming problems with separable non convexities. In A. Rocha, M. Costa, and E. Fernandes, editors, *Proceedings of the XIII Global Optimization Workshop (GOW’16)*, pages 49–52, 2016.
7. C. D’Ambrosio, A. Frangioni, and C. Gentile. Strengthening the sequential convex MINLP technique by perspective reformulations. *Optimization Letters*, 13(4):673–684, 2019.
8. C. D’Ambrosio, J. Lee, and A. Wächter. A global-optimization algorithm for mixed-integer nonlinear programs having separable non-convexity. In A. Fiat and P. Sanders, editors, *Algorithms - ESA 2009, 17th Annual European Symposium, Copenhagen, Denmark, September 7-9, 2009. Proceedings*, pages 107–118, 2009.
9. C. D’Ambrosio, J. Lee, and A. Wächter. An algorithmic framework for MINLP with separable non-convexity. In J. Lee and S. Leyffer, editors, *Mixed Integer Nonlinear Programming*, pages 315–347. Springer, New York, 2012.
10. M. Fampa, J. Lee, and W. Melo. On global optimization with indefinite quadratics. *EURO Journal on Computational Optimization*, 5(3):309–337, 2017.
11. M. Fischetti, A. Lodi, and A. Tramontani. On the separation of disjunctive cuts. *Mathematical Programming*, 128(1):205–230, 2011.
12. A. Frangioni and C. Gentile. Perspective cuts for a class of convex 0-1 mixed integer programs. *Mathematical Programming*, 106(2):225–236, 2006.
13. H. Mahmoudi, M. Aleenejad, and R. Ahmadi. Torque ripple minimization for a permanent magnet synchronous motor using a modified quasi-Z-source inverter. *IEEE Transactions on Power Electronics*, 34(4):3819–3830, April 2019.
14. C. Quendo, E. Rius, C. Person, and M. Ney. Integration of optimized low-pass filters in a bandpass filter for out-of-band improvement. *IEEE Transactions on Microwave Theory and Techniques*, 49(12):2376–2383, Dec 2001.
15. A. Saxena, P. Bonami, and J. Lee. Disjunctive cuts for non-convex mixed integer quadratically constrained programs. In *Integer Programming and Combinatorial Optimization, 13th International Conference, IPCO 2008, Bertinoro, Italy, May 26-28, 2008, Proceedings*, pages 17–33, 2008.
16. A. Saxena, P. Bonami, and J. Lee. Convex relaxations of non-convex mixed integer quadratically constrained programs: extended formulations. *Mathematical Programming*, 124(1-2):383–411, 2010.
17. A. Saxena, P. Bonami, and J. Lee. Convex relaxations of non-convex mixed integer quadratically constrained programs: projected formulations. *Mathematical Programming*, 130(2):359–413, 2011.
18. D. Wilson. *Polyhedral methods for piecewise-linear functions*. PhD thesis, University of Kentucky, 1998.