Math. Program., Ser. B 103, 225–249 (2005)

Digital Object Identifier (DOI) 10.1007/s10107-005-0581-8

Mohit Tawarmalani · Nikolaos V. Sahinidis

A polyhedral branch-and-cut approach to global optimization*

Received: June 19, 2004 / Accepted: November 19, 2004
Published online: May 3, 2005 – © Springer-Verlag 2005

Abstract. A variety of nonlinear, including semidefinite, relaxations have been developed in recent years for nonconvex optimization problems. Their potential can be realized only if they can be solved with sufficient speed and reliability. Unfortunately, state-of-the-art nonlinear programming codes are significantly slower and numerically unstable compared to linear programming software.

In this paper, we facilitate the reliable use of nonlinear convex relaxations in global optimization via a polyhedral branch-and-cut approach. Our algorithm exploits convexity, either identified automatically or supplied through a suitable modeling language construct, in order to generate polyhedral cutting planes and relaxations for multivariate nonconvex problems. We prove that, if the convexity of a univariate or multivariate function is apparent by decomposing it into convex subexpressions, our relaxation constructor automatically exploits this convexity in a manner that is much superior to developing polyhedral outer approximators for the original function. The convexity of functional expressions that are composed to form nonconvex expressions is also automatically exploited.

Root-node relaxations are computed for 87 problems from `globallib` and `minlplib`, and detailed computational results are presented for globally solving 26 of these problems with `BARON 7.2`, which implements the proposed techniques. The use of cutting planes for these problems reduces root-node relaxation gaps by up to 100% and expedites the solution process, often by several orders of magnitude.

Key words. Mixed-integer nonlinear programming – Outer approximation – Convexification – Factorable programming – Convexity identification

1. Introduction

Current implementations of local search methods for nonlinear programming problems are often efficient in practice and thus highly desirable for finding good feasible solutions of nonlinear programs. Yet, the same implementations often suffer from serious numerical issues making them inadmissible for solving convex relaxations in the context of global optimization algorithms.

The idea to construct and solve entirely polyhedral-based relaxations in the context of branch-and-bound for global optimization was first proposed and analyzed by Tawarmalani and Sahinidis in [21], where polyhedral outer approximations of univariate convex functions and bivariate functions were incorporated. Such a relaxation of convex functions often leads to a larger relaxation gap compared to the convex nonlinear relaxation from which it is derived. On the other hand, it considerably reduces the cost

M. Tawarmalani: Krannert School of Management, Purdue University, West Lafayette, IN 47907-1310, USA. e-mail: `mtawarma@mgmt.purdue.edu`

N. V. Sahinidis: Department of Chemical and Biomolecular Engineering, University of Illinois at Urbana-Champaign, Urbana, Illinois 61801, USA. e-mail: `nikos@uiuc.edu`

* The research was supported in part by ExxonMobil Upstream Research Company, the National Science Foundation under awards DMII 0115166 and CTS 0124751, and the Joint NSF/NIGMS Initiative to Support Research in the Area of Mathematical Biology under NIH award GM072023.

for obtaining lower bounds. Furthermore, a polyhedral relaxation can take advantage of highly efficient and reliable state-of-the-art linear programming solvers in order to increase the reliability of the overall approach to global optimization.

In this paper, we go beyond the ideas of [21] in that we address directly the problem of generating polyhedral relaxations of multivariate convex functions. The advance from the univariate to the multivariate case presents significant challenges at the algorithmic level. In particular, whereas sandwich algorithms for constructing outer approximations of univariate functions have long been developed and analyzed, and are known to converge quadratically [17, 21], the asymptotically optimal strategies for outer approximation of multivariate convex functions is a subject of very recent and on-going research [10, 3] and has thus far not resulted in any implementations.

In Section 2 of this paper, we prove that the outer approximation of a decomposed equivalent is tighter than the direct outer approximation of the original multivariate problem. As a consequence, we derive a two-step outer approximation methodology that can be used to improve the outer approximations of several classical convex programming techniques [12, 9, 6]. An important feature of this methodology is that it does not require the explicit identification of convexity in the original problem. Furthermore, it exploits convexity of elementary functional expressions that are composed to form nonconvex expressions. In Section 3, we explore these counterintuitive properties and propose ways to construct polyhedral relaxations in global optimization. We also show that the convexification strategy of Section 2 results in the automatic exploitation of convexity in a way that subsumes known convexity identification rules.

In Section 4, we present an implementation of a branch-and-cut global optimization solver that relies on the generation of cutting planes that are supporting hyperplanes of convex relaxations. While branch-and-cut solvers have been presented in the past for specific classes of global optimization problems (cf. [1, 22]) and Lagrangian relaxations [16], our implementation represents the first general-purpose primal branch-and-cut solver for global optimization of nonlinear and mixed-integer nonlinear programs. The implementation is further enhanced through a modeling language construct that permits the modeler to supply convexity information to the solver. Examples are provided in Section 5 to illustrate the proposed polyhedral relaxations. Preliminary computational results demonstrating the significant potential of these relaxations are presented in Section 6.

2. Polyhedral outer approximations via recursive functional compositions

Let $f : \mathbb{R}^k \mapsto \mathbb{R}^m$ and $g : \mathbb{R}^m \mapsto \mathbb{R}^n$ be vectors of functions such that $f(x) = (f_1(x), \ldots, f_m(x))$ and $g(u) = (g_1(u), \ldots, g_n(u))$, where $x = (x_1, \ldots, x_k) \in \mathbb{R}^k$ and $u = (u_1, \ldots, u_m) \in \mathbb{R}^m$. Let $f_i(x)$ be convex for $i = 1, \ldots, m$ and $g_j(u)$ be convex for $j = 1, \ldots, n$. Let $I = \{1, \ldots, m\}$ and $I_L = \{i \in I \mid f_i(x) \text{ is affine}\}$. Assume that $g_j(u)$, for all j, is non-decreasing in the range of $f_i(x)$ for each $i \in I \backslash I_L$. Define $h = g \circ f$ as the composition of g and f, *i.e.*, $h(x) = g(f(x))$. Then, $h(x)$ is $(h_1(x), \ldots, h_n(x))$, where $h_l = g_l \circ f$ for $l = 1, \ldots, n$. It can be proved easily that each h_l is convex. We include a proof for completeness.

Proposition 1. *For each $l = 1, \ldots, n$, the function h_l is convex.*

Proof. Consider the following convex set:

$$S = \{(\gamma_l, \phi_1, \ldots, \phi_m, x) \mid \gamma_l \geq g_l(\phi_1, \ldots, \phi_m);$$
$$\phi_i \geq f_i(x), \forall i \in I \backslash I_L;\ \phi_i = f_i(x), \forall i \in I_L\}.$$

We argue that the epigraph of $h(x)$ is the projection of S onto the space of γ_l and x variables. A point (γ_l, x) which belongs to the epigraph of $h(x)$ also lies in the projection of S since $(\gamma_l, f_1(x), \ldots, f_m(x), x) \in S$. Now, consider any point $(\gamma_l, \phi_1, \ldots, \phi_m, x) \in S$. Note that $\gamma_l \geq g_l(\phi_1, \ldots, \phi_m)$. Since g_l is non-decreasing in the range of each f_i with $i \in I \backslash I_L$ and ϕ_i equals $f_i(x)$ for each $i \in I_L$, it follows that $g_l(\phi_1, \ldots, \phi_m) \geq g_l(f_1(x), \ldots, f_m(x))$. Therefore, $\gamma_l \geq g_l(f_1(x), \ldots, f_m(x))$, or in other words (γ_l, x) in the epigraph of h_l. Since S is convex and projection preserves convexity, h_l is convex.□

The main purpose of this section is to show that outer-approximating $f(x)$ and $g(u)$ separately leads to a tighter approximation of $h(x)$ than the one obtained by outer-approximating $h(x)$ directly. In the following, f and g will be assumed to be differentiable unless stated otherwise. The case of non-differentiable functions will be addressed in Subsection 2.1. For a vector of differentiable functions, $[f_1(x), \ldots, f_m(x)]$, $x \in \mathbb{R}^k$, we denote by $\nabla f(\bar{x})$ the Jacobian matrix of $f(\cdot)$ at $\bar{x}$. When $m = 1$, we do not distinguish between the vector and the function.

For a differentiable convex function $f_i(x)$, let $f_i(x) \geq f_i(\bar{x}) + \nabla f_i(\bar{x})(x - \bar{x})$ be the subgradient inequality where $\nabla f_i(\bar{x})$ is the gradient of f_i at $\bar{x}$. Given a point $\bar{x}$ and $i \in I \backslash I_L$, by convexity of $f_i(x)$, the following inequality is valid for $\{(\phi_i, x) \mid \phi_i \geq f_i(x)\}$:

$$\phi_i \geq f_i(x^j) + \nabla f_i(x^j)(x - x^j).$$

If $i \in I_L$, by linearity of f_i, $\{(\phi_i, x) \mid \phi_i = f_i(x)\}$ can be rewritten as:

$$\phi_i = f_i(x^j) + \nabla f_i(x^j)(x - x^j).$$

Also, by convexity of $g(f)$, the following inequality is valid for $\{(\gamma_l, \phi) \mid \gamma_l \geq g(\phi)\}$:

$$\gamma \geq g(f(\bar{x})) + \nabla g(f(\bar{x}))(\phi - f(\bar{x})).$$

Theorem 1. *Consider a set of points x^j, $j = 1, \ldots, r$. Let*

$$S_1(h) = \{(\gamma, x) \mid \gamma \geq h(x^j) + \nabla h(x^j)(x - x^j) \quad j = 1, \ldots, r\}$$

and

$$S_2(h) = \left\{ (\gamma, x) \;\middle|\; \begin{array}{ll} \gamma \geq g(f(x^j)) + \nabla g(f(x^j))(\phi - f(x^j)) & j = 1, \ldots, r \\ \phi_i \geq f_i(x^j) + \nabla f_i(x^j)(x - x^j) & j = 1, \ldots, r; i \in I \backslash I_L \\ \phi_i = f_i(x^j) + \nabla f_i(x^j)(x - x^j) & j = 1, \ldots, r; i \in I_L \end{array} \right\}.$$

Then, $S_2(h) \subseteq S_1(h)$.

Proof. Note that $g(f)$ is non-decreasing for each i in $I \backslash I_L$. For $(\gamma^0, x^0) \in S_2(h)$, there exists ϕ^0 such that:

$$\begin{aligned}\gamma^0 &\geq g(f(x^j)) + \nabla g(f(x^j))(\phi^0 - f(x^j)) \\ &\geq g(f(x^j)) + \nabla g(f(x^j))\nabla f(x^j)(x^0 - \bar{x}).\end{aligned}$$

By the chain rule of differentiation:

$$\nabla h(x^j) = \nabla g(f(x^j))\nabla f(x^j).$$

Therefore:

$$\gamma^0 \geq h(x^j) + \nabla h(x^j)(x^0 - x^j).$$

□

It should be apparent that, as r tends to ∞ and the points x^j at which the supporting hyperplanes are constructed cover the feasible space densely, then $S_2(h)$ and $S_1(h)$ tend to the epigraph of $h(x)$.

A convex function h can be outer approximated by subgradient inequalities through a single-step procedure yielding the set $S_1(h)$. Alternatively, an outer approximating set $S_2(h)$ can be derived via the two-step procedure that constructs subgradient inequalities for the functions g and f whose composition results in h. The above theorem shows that the set $S_2(h)$ is contained within the set $S_1(h)$. In the sequel, we investigate the potential advantages and disadvantages of the two alternative outer approximating schemes.

Remark 1. Consider $h^2(x) = g^2(f(x), x)$ and $h^1 = g^1(f(x))$ where $g^2(u, x) = g^1(u) + Ax + b$. Clearly, $h^2(x) = h^1(x) + Ax + b$. Then, $(\gamma^0, x^0) \in S_2(h^1)$ if and only if $(\gamma^0 + Ax^0 + b, x^0) \in S_2(h^2)$. Similarly, $(\gamma^0, x^0) \in S_1(h^1)$ if and only if $(\gamma^0 + Ax^0 + b, x^0) \in S_1(h^2)$.

In fact, if $f(x)$ is linear, the two procedures result in the same approximation:

Proposition 2. *If $f(x) = Ax + b$, where $A \in \mathbb{R}^m \times \mathbb{R}^k$, then $S_1(h) = S_2(h)$.*

Proof. Consider $(\gamma^0, x^0) \in S_1(h)$. Define $\phi^0 = Ax^0 + b$.

$$\begin{aligned}\gamma^0 &\geq h(x^j) + \nabla h(x^j)(x^0 - x^j) \\ &= g(Ax^j + b) + \nabla g(Ax^j + b)(Ax^0 + b - Ax^j - b) \\ &= g(f(x^j)) + \nabla g(f(x^j))(\phi^0 - f(x^j)).\end{aligned}$$

□

Remark 2. For sets $S_2(h)$ and $S_1(h)$ defined in Theorem 1, it is possible that $S_2(h) \subset S_1(h)$. Indeed, consider

$$f(x) = 2x^3 + 4x^2 - 7x + 5 + \delta(x \mid x \geq 0)$$

where $\delta(x \mid x \geq 0)$ equals 1 when $x \geq 0$ and is 0 otherwise. Let $g(u) = e^u$. We outer-approximate $\gamma \geq h(x)$ by constructing subgradient inequalities at $x = 0$ and $x = 1$. Clearly

$$S_2(h) = \{(\gamma, x) \mid \phi \geq 5 - 7x,\ \gamma \geq e^5(f - 4),\ \phi \geq 7x - 3,\ \gamma \geq e^4(f - 3)\},$$

whereas

$$S_1(h) = \{(\gamma, x) \mid \gamma \geq e^5(1 - 7x),\ \gamma \geq e^4(7x - 6)\}.$$

Let $h_{S_2(h)}(x)$ denote $\min\{\gamma \mid (\gamma, x) \in S_2(h)\}$ and $h_{S_1(h)}(x)$ denote $\min\{\gamma \mid (\gamma, x) \in S_1(h)\}$. It follows easily that $h_{S_1(h)}(0.25) = -0.75e^5 < 0.25e^4 = h_{S_2(h)}(0.25)$. Figure 1 provides a pictorial representation of the two outer approximations, clearly demonstrating that $S_2(h) \subset S_1(h)$.

In fact, the two-step approximation $S_2(h)$ is quite often tighter than the single-step approximation $S_1(h)$. A somewhat surprising demonstration of this fact is that the approximation for x^4 in the nonnegative orthant is better if it is treated as a composition of $f(x) = x^2$ and $g(u) = u^2$, and subgradient inequalities are constructed at two points, for example $x = 1$ and $x = 4$. In this case, the two-step approximation produces a lower bound of 15 at $x = 3$ whereas the single-step approximation produces a lower bound of 9. Figure 2 illustrates this difference pictorially.

Notice that, in the examples of Figures 1 and 2, $f(x)$ as well as $g(u)$ are univariate nonlinear functions.

Proposition 3. *If $|I \backslash I_L| = 1$ and $g(u)$ is a linear function, then $S_1(h) = S_2(h)$.*

Proof. Let $g(u) = Au + b$, $(\gamma^0, x^0) \in S_1(h)$, $I \backslash I_L = \{i^0\}$ and

$$k = \operatorname{argmax}\{f_{i^0}(x^j) + \nabla f_{i^0}(x^j)(x^0 - x^j) \mid j = 1, \ldots, r\}. \tag{1}$$

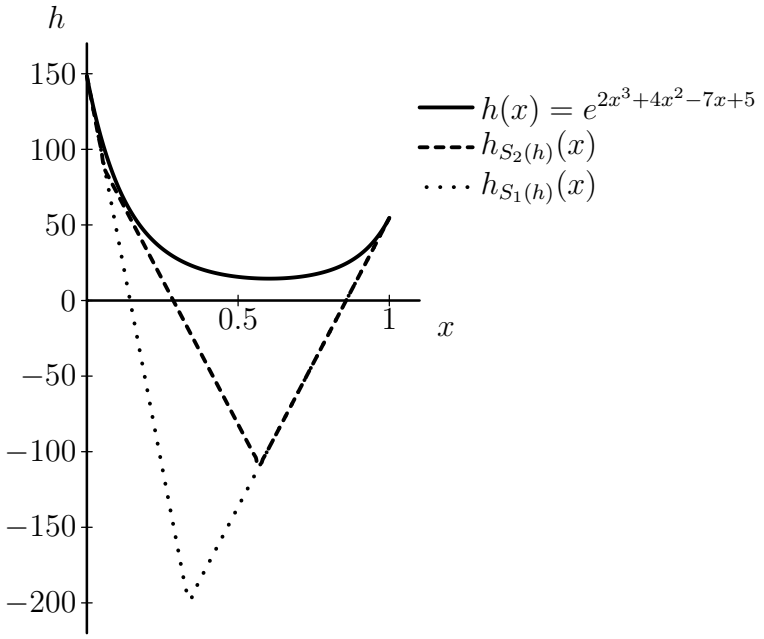

Fig. 1. Outer-approximating composite functions

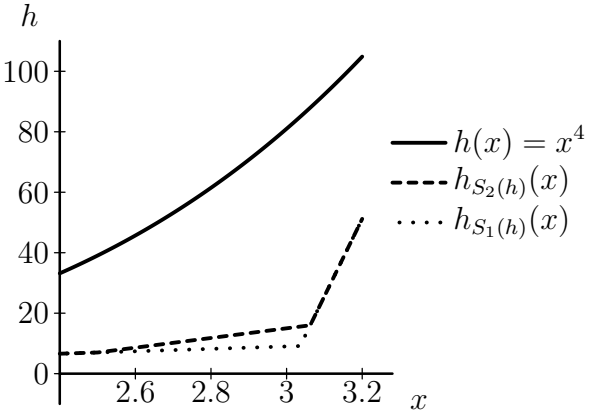

Fig. 2. Outer-approximating x^4 at $x = 1$ and $x = 4$

Define $\phi^0 = f(x^k) + \nabla f(x^k)(x^0 - x^k)$. Then:

$$\begin{aligned}\gamma^0 &\geq h(x^k) + \nabla h(x^k)(x^0 - x^k)\\ &= Af(x^k) + b + A\nabla f(x^k)(x^0 - x^k)\\ &= Af(x^k) + b + A(\phi^0 - f(x^k))\\ &= A\phi^0 + b\\ &= Af(x^j) + b + A(\phi^0 - f(x^j)).\end{aligned}$$

By (1): $\phi^0_{i^0} \geq f_{i^0}(x^j) + \nabla f_{i^0}(x^j)(x^0 - x^j)$. Since $f_i(x)$ is linear for $i \in I_L$, it follows that $\phi^0_i = f_i(x^k) + \nabla f_i(x^k)(x^0 - x^k) = f_i(x^j) + \nabla f_i(x^j)(x^0 - x^j)$. In other words, $(\gamma^0, x^0) \in S_2(h)$. □

Remark 3. If $g(u)$ is a linear function but $|I \backslash I_L| > 1$, $S_2(h)$ may be a strict subset of $S_1(h)$ even when $x \in \mathbb{R}$. For example, consider $f_1(x) = x^2$, $f_2(x) = -\log(x)$, and $g(u_1, u_2) = u_1 + 10u_2$ so that $h = x^2 - 10\log(x)$. We use $x = 2$ and $x = 4$ to construct the outer approximations. If $h_{S_2(h)}(x)$ denotes the minimum value of x when a two-step outer approximation is constructed and $h_{S_1(h)}(x)$ denotes the minimum value of h for a single-step approximation, then it is easily seen that $h_{S_1(h)}(2.9) = 3.1 - 10\log(2) < 10.35 - 10\log(4) = h_{S_2(h)}(2.9)$.

Thus far, we have established that the two-step procedure provides a tighter outer approximation of $h(x)$ than the single-step outer approximation. On the other hand, clearly, $S_2(h)$ requires more linear inequalities than $S_1(h)$. Then, a natural question is whether the approximation $S_2(h)$ can be obtained from $h(x)$ through a single-step procedure that uses the same or a comparable number of subgradient inequalities as does $S_2(h)$.

Proposition 4. *If $g(\cdot)$ is linear and there is only one nonlinear variable in $f_i(\cdot)$, then there exist $|I \backslash I_L|(r-1)+1$ points such that the set $S_1(h)$ developed using these points is contained within $S_2(h)$, when the latter is developed using r points.*

Proof. Let x_1 be the only nonlinear variable. Then, the linear part can be removed from $f_i(x)$ and aggregated as part of $g(u)$ where u is augmented to include the variables $x_2, \ldots, x_n$. By Remark 1, we are only required to treat the case $h(x_1) = \sum_{i=1}^{|I \setminus I_L|} f_i(x_1)$ where each f_i is a univariate nonlinear function. Let $\underline{f}_i(x)$ represent the piecewise-linear outer approximation of $f_i(x_1)$ obtained by constructing subgradient inequalities at $x_1^1, \ldots, x_1^r$. Let $\underline{h}(x_1) = \sum_{i=1}^{|I \setminus I_L|} \underline{f}_i(x)$. Since each $\underline{f}_i$ is piecewise-linear with $r-1$ breakpoints, $\underline{h}_i$ is also piecewise-linear with at most $|I \setminus I_L|(r-1)$ breakpoints and can be represented using $|I \setminus I_L|(r-1)+1$ underestimators. A tighter underestimator of $h(x)$ can then be developed by maintaining the slopes of the underestimating hyperplanes but shifting them so that they support h. □

Proposition 4 is established under fairly restrictive conditions. As we show in the next result, in a more general setting, even polynomially many subgradient inequalities of $h(x)$ may not suffice to reconstruct the underestimator produced by $S_2(h)$.

Proposition 5. *Consider a separable function $f_1(x_1) + \cdots + f_m(x_m)$ where each f_i is strictly convex. Then, the outer approximation $S_2(h)$ formed by constructing supporting hyperplanes at $x^1, \ldots, x^r$ is equivalent to $S_1(h)$ formed by constructing supporting hyperplanes at every x for which there exists $j \in \{1, \ldots, r\}$ such that $x_i = x_i^j$.*

Proof. It can be easily argued that

$$\nabla h(\bar{x}) = [\nabla f_1(\bar{x}_1), \ldots, \nabla f_m(\bar{x}_m)].$$

Then $(\gamma^0, x^0) \in S_2(h)$ if and only if there exists ϕ^0 such that $\gamma^0 \geq \sum_{i=1}^m \phi_i^0$, where $\phi_i \geq f_i(x_i^j) + \nabla f_i(x_i^j)(x_i^0 - x_i^j)$. By Fourier-Motzkin elimination, $(\gamma^0, x^0) \in S_2(h)$ if and only if, for all $j_i \in \{1, \ldots, r\}$,

$$\gamma^0 \geq \sum_{i=1}^m \left(f_i(x_i^{j_i}) + \nabla f_i(x_i^{j_i})(x_i^0 - x_i^{j_i}) \right).$$

These are precisely the subgradient inequalities for $h(x)$ at

$$\{x \mid \exists j \text{ such that } x_i = x_i^j\}.$$

All the subgradient inequalities are distinct due to the strict convexity of each f_i. □

If for all i and for all $j \neq k$ we have $x_i^j \neq x_i^k$, then Proposition 5 states that an exponential number of supporting hyperplanes at r^n points are needed in $S_1(h)$ to be able to yield the approximation obtained from $S_2(h)$ via subgradients at r points. Separable functions are indeed quite common in nonconvex problems, indicating that the two-step approximation has the potential of providing much tighter polyhedral outer approximations than the one-step procedure with a comparable number of subgradient inequalities.

2.1. Outer approximation of non-differentiable functions

Using the chain rule for subdifferentials, the ideas presented above will now be extended to compositions of non-differentiable functions. For example, if $h(x)$ is a separable function of the form

$$f_1(x_1) + \cdots + f_m(x_m),$$

it can be easily shown that:

$$\partial h(\bar{x}) = \partial f_1(\bar{x}_1) \times \cdots \times \partial f_m(\bar{x}_m),$$

where $\partial f(x)$ is the subdifferential of f at x.

Consider the non-differentiable function

$$h(x) = \max\{f_1(x), \ldots, f_m(x)\},$$

where each $f_i(\cdot)$ is a strictly convex differentiable function. Consider a point $\bar{x}$ in the domain of h. If $I(\bar{x}) = \{i | f_i(\bar{x}) = h(\bar{x})\}$, then the subdifferential of $h(\bar{x})$ is the set of convex combinations of the subgradients of $f_i(\bar{x})$, where $i \in I(\bar{x})$. Consider $j \notin I(\bar{x})$. The subgradients of $h(\bar{x})$ do not depend on the subgradient of $f_j(\cdot)$ at $\bar{x}$. However, the subgradient inequality $h(x) \geq f_j(\bar{x}) + \nabla f_j(x)(x - \bar{x})$ is valid. In fact, this inequality may not be dominated by the inequalities obtained using the subgradients of $h(\bar{x})$. For example, consider $h(x) = \max\{(x-3)^2, 2\}$. If we underestimate $h(\cdot)$ using the subgradient at $x = 0$, we obtain $h \geq 9 - 6x$. At $x = 3$, this underestimator is much weaker than the underestimator $h \geq 2$.

We now consider the case of general non-differentiable functions. Under the assumption of differentiability, the chain rule of differentiation was used to prove Theorem 1. Fortunately, the chain rule can be generalized to the non-differentiable case as follows:

Lemma 1 (Theorem 4.3.1 in [11]). *Let h, f and g be such that $h = g \circ f$, f is a vector of convex functions $f_1, \ldots, f_m$ and g is convex non-decreasing in f_i. For all $x \in \mathbb{R}^n$,*

$$\partial(g \circ f)(x) = \left\{ \sum_{i=1}^{m} \rho_i s_i \mid (\rho_1, \ldots, \rho_m) \in \partial g(f(x)), s_i \in \partial f_i(x), i = 1, \ldots, m \right\}.$$

With the above result, we can provide an analogous result to Theorem 1 for the non-differentiable case.

Theorem 2. *Consider a list of points x^j, $j = 1, \ldots, r$ (repetitions allowed). Let*

$$S_1(h) = \{(\gamma, x) \mid \gamma \geq h(x^j) + a^j(x - x^j) \quad j = 1, \ldots, r\},$$

where $a^j \in \partial(g \circ f)(x^j)$. Then, there exist $\rho^j \in \partial g(f(x^j))$ and $s_i^j \in \partial f_i(x^j)$ for $i = 1, \ldots, m$ and $j = 1, \ldots, r$ such that $S_2(h)$ defined as

$$S_2(h) = \left\{ (\gamma, x) \left| \begin{array}{ll} \gamma \geq g(f(x^j)) + \rho^j(\phi - f(x^j)) & j = 1, \ldots, r \\ \phi_i \geq f_i(x^j) + s_i^j(x - x^j) & j = 1, \ldots, r \end{array} \right. \right\},$$

is contained in $S_1(h)$.

Proof. Lemma 1 implies that, for each $a^j \in \partial(g \circ f)(x^j)$, there exist $s_i^j \in \partial f_i(x^j)$ and $\rho^j \in \partial g(f(x^j))$ such that $a^j = \sum_{i=1}^m \rho_i^j s_i^j$. Then, consider a point $(\gamma, x) \in S_2(h)$. Clearly,

$$\begin{aligned}\gamma &\geq \sum_{i=1}^m \rho_i^j(\phi_i - f_i(x^j)) + g(f(x^j)) \\ &\geq \sum_{i=1}^m \rho_i^j s_i^j(x - x^j) + g(f(x^j)) \\ &= a^j(x - x^j) + g(f(x^j)).\end{aligned}$$

Therefore $(\gamma, x) \in S_1(h)$. □

Theorem 2 generalizes Theorem 1. However, due to the non-uniqueness of the subgradient, S_2 is not defined constructively but is obtained using an existence result for the corresponding subgradients of g and f_i provided by Lemma 1.

3. Exploiting convexity in global optimization

3.1. Automatic exploitation of convexity in factorable programs

Branch-and-bound algorithms for global optimization often construct recursive decompositions of factorable functions [14, 21]. In particular, factorable functions can be decomposed as sums and products of univariate functions. Then, the convexity/concavity properties of the primitive univariate functions can be automatically identified and exploited in order to develop subgradient inequalities. Thus, global optimization methods based on factorable decompositions can naturally exploit the benefits of polyhedral outer approximations of the type analyzed in the previous section. The question addressed in this subsection is how such an approach compares to the automatic identification of convexity for the original factorable functions. With this goal in mind, we would like to identify the range of functions that can be recognized as convex under the assumptions in Section 2 that led to the proof of Proposition 1.

Interest in detecting convexity automatically with the hope of exploiting it in optimization algorithms has been surging recently [5, 13, 7]. At the present time, it is difficult to conceive of a prover/disprover of convexity that is both efficient and reliable. Yet, there are many known conditions under which convexity/concavity can be characterized [2].

In the following, we discuss many of the rules for detecting convexity (or concavity) presented in [13] and show that these properties are subsumed in the composition rule of Section 2:

1. If g is univariate convex and f is linear, then $g \circ f$ is convex. This is a special case of the multivariate composition rule.
2. If g is univariate convex, f is convex, and g is increasing in the range of f, then $g \circ f$ is convex. This is a special case of the multivariate composition rule.

3. If g is univariate convex, f is concave, and g is decreasing in the range of f, then $g \circ f$ is convex. This follows from the composition rule since $-f$ is convex and $g(-\cdot)$ is increasing convex.
4. If $a_i \geq 0$ and f_i is convex for $i = 1, \ldots, n$, then $\sum_{i=1}^{n} a_i f_i(x) + c$ is convex. This follows from the composition rule since $g(u) = \sum_{i=1}^{n} a_i u_i + c$ is a convex increasing function.
5. If f_i is convex for $i = 1, \ldots, m$, then $\max\{f_1(x), \ldots, f_m(x)\}$ is convex since $g(u) = \max\{u_1, \ldots, u_m\}$ is an non-decreasing convex function.
6. If $f(x)$ is convex, then $e^{f(x)}$ is convex since $g(u) = e^u$ is increasing convex.
7. If $f(x) > -p$ and $f(x)$ is concave, then $-\log(f(x) + p)$ is convex since $g(u) = -\log(-u)$ is increasing convex.
8. If $f(x)$ is nonnegative and convex, then $f(x)^2$ is convex since $g(u) = u^2$ is an increasing convex function when $u \geq 0$.

Clearly, there are reasons to automatically detect convexity of functions in global optimization. For instance, convexity detection techniques can concentrate on functions whose convexity is not apparent from the composition rule of Proposition 2. Deduction of convexity based on compositions of univariate nonlinear functions is beneficial in that it can lead to smaller relaxations of similar quality (see Proposition 4). Yet, the above analysis shows that automatic detection of convexity is not essential for constructing polyhedral outer approximators of convex functions recognizable using Proposition 2. In the latter case, it suffices to construct polyhedral outer approximators of the convex functions of the decomposed equivalent, wherein convexity of the primitive univariate functions is known *a priori*.

3.2. Cutting planes for convex relaxations of primitive nonconvex functions

Certain primitive operations, like x/y, are neither convex nor concave. As such, they cannot be exploited as discussed above. However, we can use nonlinear convex relaxations for such functions to develop a polyhedral relaxation for global optimization purposes. For instance, let us assume that $(x, y) \in [x^L, x^U] \times [y^L, y^U]$, where $x^L > 0$ and $y^L > 0$. Then, it was shown in [24, 23, 18, 19] that x/y can be underestimated by the following convex function:

$$\frac{x}{y} \geq \max\{f_1(x, y), f_2(x, y), f_3(x, y)\} \tag{2}$$

where

$$f_1(x, y) = \frac{\left(x + \sqrt{x^L x^U}\right)^2}{y\left(\sqrt{x^L} + \sqrt{x^U}\right)^2},$$
$$f_2(x, y) = x/y^L + x^U(1/y - 1/y^L), \text{ and}$$
$$f_3(x, y) = x/y^U + x^L(1/y - 1/y^U).$$

For any given point $(\bar{x}, \bar{y})$, a cutting plane that supports the underestimator in (2) can be easily obtained by choosing the inequality amongst:

$$\frac{x}{y} \geq \frac{2\left(\bar{x}+\sqrt{x^L x^U}\right)}{\left(\sqrt{x^L}+\sqrt{x^U}\right)^2 \bar{y}} x - \frac{\left(\bar{x}+\sqrt{x^L x^U}\right)^2}{\bar{y}^2\left(\sqrt{x^L}+\sqrt{x^U}\right)^2} y + 2\frac{\left(\bar{x}+\sqrt{x^L x^U}\right)\sqrt{x^L x^U}}{\bar{y}\left(\sqrt{x^L}+\sqrt{x^U}\right)^2},$$

$$\frac{x}{y} \geq \frac{\bar{x}}{y^L} - \frac{x^U}{\bar{y}^2} y - \frac{\left(\bar{y}-2y^L\right)x^U}{\bar{y}y^L}, \text{ and}$$

$$\frac{x}{y} \geq \frac{\bar{x}}{y^U} - \frac{x^L}{\bar{y}^2} y - \frac{\left(\bar{y}-2y^U\right)x^L}{\bar{y}y^U},$$

that is most violated. Concave envelopes of x/y over the positive orthant are already linear [18] and can be included as such in the polyhedral relaxation.

3.3. Improved relaxations of nonconvex terms via exploitation of convexity

Exploiting convexity for functional expressions has further indirect benefits in nonconvex optimization. In addition to identifying and approximating convex functions closely, relaxations of nonconvex terms improve as a result of better approximation of their convex subexpressions. For example, consider a nonconvex function $h(x) = f_1(x)f_2(x)$, where $f_1(x)$ and $f_2(x)$ are nonnegative convex functions. Let $f_1(x) \in [f_1^L(x), f_1^U(x)]$ and $f_2(x) \in [f_2^L(x), f_2^U(x)]$ in the domain of interest. Then, the following underestimator of $h(x)$ is well known [14]:

$$\begin{aligned} h(x) \geq \max\{&f_1(x)f_2^L(x) + f_2(x)f_1^L(x) - f_1^L(x)f_2^L(x), \\ &f_1(x)f_2^U(x) + f_2(x)f_1^U(x) - f_1^U(x)f_2^U(x)\}. \end{aligned} \tag{3}$$

Assume that the recursive algorithm for constructing relaxations for factorable functions introduces variables ϕ_1 and ϕ_2 such that $\phi_1 = f_1(x)$, $\phi_2 = f_2(x)$ and relaxes $\phi_1\phi_2$ using the bilinear envelopes of [14]. Further, assume that the convexity of $f_1(x)$ and $f_2(x)$ is identified by the algorithm and that domain reduction schemes [21] are able to identify that $\phi_1 \in [f_1^L(x), f_1^U(x)]$ and $\phi_2 \in [f_2^L(x), f_2^U(x)]$. Then, the underestimator (3) is automatically exploited in the relaxation.

The above discussion has established that, if the convexity/concavity of a function follows by expressing it as a composition of primitive functions whose convexity/concavity is either exploited automatically (for example x^2, $\sum$, exp, log) or identified by the modeler, then such convexity/concavity is implicitly exploited using the subgradient inequalities, as long as the latter are constructed for all primitive functions in the decomposition of the original function. As a result, it is not necessary to identify if a function is non-decreasing.

More concretely, for nonconvex terms of the form $h(x) = g(f_1(x), \ldots, f_m(x))$, where g and f_i, $i = 1, \ldots, m$ are convex, cutting planes are able to partially exploit the structure in constructing relaxations. Note that $h(x)$ is, in general, nonconvex since g is not required to be a non-decreasing function. In the following, we investigate the

relaxation of $h(x)$ that results when cutting planes are generated to exploit the convexity of g and each f_i.

For an arbitrary function $f : \mathbb{R}^n \mapsto \mathbb{R}$, we say that a function φ minorizes f if $\varphi(x) \leq f(x)$ for all $x \in \mathbb{R}^n$. Similarly, φ majorizes f if $\varphi(x) \geq f(x)$ for all $x \in \mathbb{R}^n$. For our analysis, we will mainly be interested in functions φ that are lower-semicontinuous (l.s.c.), convex, and non-decreasing. We assume that all the functions have an affine minorizer.

Proposition 6. *Consider a function $f : \mathbb{R}^n \mapsto \mathbb{R}$. There exists an l.s.c. function L_f which is convex, non-decreasing, minorizes $f(x)$, and majorizes every other l.s.c., convex, non-decreasing minorizer of $f(x)$.*

Proof. Clearly $h(x) = -\infty$ is an l.s.c. convex non-decreasing function that minorizes $f(x)$. If $g_1(x)$ and $g_2(x)$ are two convex non-decreasing functions that minorize $f(x)$, then $\max\{g_1(x), g_2(x)\}$ is another such function that majorizes $g_1(x)$ and $g_2(x)$. □

Proposition 7. $L_f = \sup_{s \geq 0} \langle s, x \rangle - f^*(s)$.

Proof. Clearly, L_f is convex (a supremum of linear functions) and minorizes $f(x)$ (By definition, $f^*(s) \geq \langle s, x \rangle - f(x)$). We check that L_f is non-decreasing by noting that for $x'' \geq x'$:

$$\begin{aligned} L_f(x'') &= \sup_{s \geq 0} \{\langle s, x'' - x' \rangle + \langle s, x' \rangle - f^*(s)\} \\ &\geq \sup_{s \geq 0} \{\langle s, x' \rangle - f^*(s)\} \\ &= L_f(x'). \end{aligned}$$

The proof is by contradiction. Assume that there exists a function $h(x)$ that is l.s.c., convex, and non-decreasing. In addition, assume that $h(x)$ minorizes $f(x)$ and majorizes $L_f(x)$. Since $h(x) \leq f(x)$, it follows that $h^*(s) \geq f^*(s)$. Then,

$$\begin{aligned} h^{**}(x) &= \sup_{s} \{\langle s, x \rangle - h^*(s)\} \\ &= \sup_{s \geq 0} \{\langle s, x \rangle - h^*(s)\} \\ &\leq \sup_{s \geq 0} \{\langle s, x \rangle - f^*(s)\} \\ &= L_f(x). \end{aligned}$$

The supremum above was restricted to nonnegative s because, if $s_i < 0$, then $h^*(s) = \sup_x \{\langle s, x \rangle - h(x)\}$ is infinite as is easily seen by reducing x_i to $-\infty$ and noticing that $h(x)$ is a non-decreasing function. However, since $h(x)$ is l.s.c. and convex, $h^{**}(x) = h(x)$ and we have a contradiction of the assumption. □

Theorem 3. $\{(x, \gamma) \mid \gamma \geq g^{**}(\phi), \phi \geq f(x)\} = \{(x, \gamma) \mid \gamma \geq L_g(f(x))\}$.

Proof. Let

$$\begin{aligned} S &= \{(x, \gamma) \mid \gamma \geq g^{**}(\phi), \phi \geq f(x)\} \\ &= \{(x, \gamma) \mid \gamma \geq \sup_{s} \{\langle s, \phi \rangle - g^*(s)\}, \phi \geq f(x)\}. \end{aligned}$$

We argue first that we can restrict our attention to nonnegative s while taking the supremum of $\langle s, \phi \rangle - g^*(s)$. This follows because, for a given x and an s which has at least one coordinate s_i strictly less than zero, ϕ_i can be increased without bound, taking $\langle s, \phi \rangle - g^*(s)$ to $-\infty$. Therefore:

$$\begin{aligned} S &= \{(x, \gamma) \mid \gamma \geq \sup_{s \geq 0}\{\langle s, \phi \rangle - g^*(s)\}, \phi \geq f(x)\} \\ &= \{(x, \gamma) \mid \gamma \geq L_g(\phi), \phi \geq f(x)\} \\ &= \{(x, \gamma) \mid \gamma \geq L_g(f(x))\}, \end{aligned}$$

where the first equality follows from Proposition 7 and the second equality follows from the fact that L_g is non-decreasing. □

Consider, for example, $f(x) = x^2 - 1$ and $g(u) = u^2$. Then, $L_g(u) = \max(u, 0)^2$. As is evident from Figure 3, $L_g(f(x)) = \max(x^2 - 1, 0)^2$ is the convex envelope of $(x^2 - 1)^2$.

Clearly, the cutting planes proposed here do not always generate the convex envelope. Even in the one-dimensional example of Figure 3, if x is restricted to be nonnegative, the resulting relaxation is not the convex envelope. In fact, the relaxation obtained by using the proposed cutting planes may not imply the convex envelope even in the absence of bounds. For example, consider

$$f(x) = \begin{cases} e^{-x} - 1 & \text{if } x \leq 0, \\ -x & \text{otherwise,} \end{cases}$$

and $g(u) = u^2$. Then, the convex envelope of $g(f(x))$ equals x^2 for $x \geq 0$. However, $L_g(f(x)) = 0$ when $x \geq 0$. Nevertheless, as the next result demonstrates, the cutting planes are able to exploit the convex envelope of $g(f(x))$ when f and g are one-dimensional coercive convex functions.

Proposition 8. *If $f(x) : \mathbb{R} \mapsto \mathbb{R}$ and $g(u) : \mathbb{R} \mapsto \mathbb{R}$ are coercive convex functions, then the convex envelope of $g(f(x))$ is $L_g(f(x))$.*

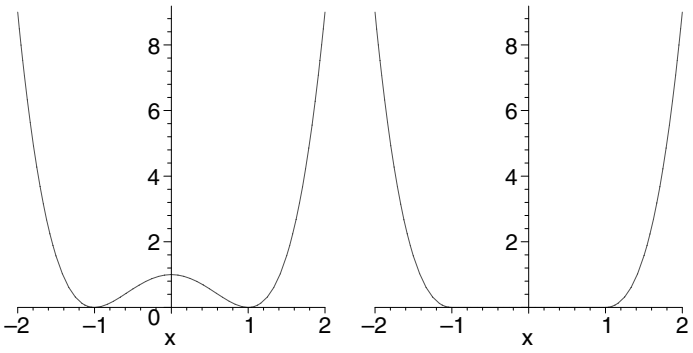

Fig. 3. Relaxation of $(x^2 - 1)^2$

Proof. Since $g(u)$ is coercive, it tends to infinity as $u \to \infty$ and as $u \to -\infty$. Let $u^* = \text{argmin}\{g(u)\}$. Such a u^* exists since coercivity allows us to restrict attention to a bounded interval. Then, it follows from the monotonicity of the subgradient mapping of g that $g(u)$ is non-increasing to the left of u^* and non-decreasing to the right of u^*. Therefore,

$$L_g(u) = \begin{cases} g(u^*) & \text{if } u \leq u^*, \\ g(u) & \text{otherwise.} \end{cases}$$

Since $f(x)$ is coercive and convex, the level set $\{x \mid f(x) \leq u^*\}$ is a bounded interval, say $[a, b]$. By continuity and coercivity of $f(x)$, $f(a) = u^*$ and $f(b) = u^*$. Clearly, $L_g(f(x)) = g(f(x))$ if $x \leq a$ or $x \geq b$. For $x \in [a, b]$, $L_g(f(x)) = g(u^*) = g(f(a)) = g(f(b))$. Since $L_g(f(x))$ is convex, it must be the convex envelope of $g(f(x))$. □

3.4. Polyhedral approximations of convex bodies not recognized by the composition rule

The discussion of Section 3.1 establishes that the main advantage of detecting convexity automatically would be in producing smaller rather than tighter relaxations. Unfortunately, convexity of many functions including $\text{logexp}(x) = \log(e^{x_1} + \cdots + \cdots e^{x_n})$ and $x^2 + y^2 - xy$ cannot be gleaned using the composition rule of Section 2. For such a function, a specialized proof that expresses the quadratic form generated by the Hessian as a sum of squares is often needed to prove convexity. If the convexity of such functions is determined, then the increasing nature is exploited automatically by the recursive decomposition scheme allowing recognition of the convexity of functions like $\text{logexp}(f_1(x), \ldots, f_m(x))$, where $f_i(x)$ is convex.

A user of the global optimization algorithm and/or an automatic detector of convexity developed in the modeling interface can thus inform the optimization algorithm of the convexity characteristics of expressions that are not automatically recognized. In order to exploit this convexity using a polyhedral approach, we are interested in circumscribing the convex body K formed by the epigraph of the convex function. For the univariate case, there are well-established sandwich algorithms for achieving this [17, 21]. The multivariate case has been studied in the literature for a long time but has not resulted in any implementations. Nonetheless, recent progress in the direction of approximating compact convex sets has been remarkable. Gruber [10] developed asymptotic estimates for the approximation properties of the polytope P_n^{best} formed by constructing supporting hyperplanes at the best n points on the boundary of the smooth convex body with positive Gaussian curvature. In particular, Gruber proved that $\text{Volume}(P_n^{\text{best}}) - \text{Volume}(K) = O(n^{\frac{2}{d-1}})$. More recently, Böröczky and Reitzner [3] showed that, if random circumscribed polytopes P_n are formed by choosing points according to a continuous density function d_K, then the expected volume of P_n is $\text{EV}(P_n) = \text{Volume}(K) + O(n^{\frac{2}{d-1}})$. In particular, $\text{EV}(P_n)$ is minimized if $d_K(x)$ is proportional to $H(x)^{\frac{1}{d+1}}$, where $H(x)$ is the Gaussian curvature of K at x.

The insights gained by these results can be exploited computationally in a variety of ways. Computing the cumulative probability distribution generated by using the Gaussian curvature of the convex body is fairly involved. Instead, we propose to choose a random set of points from a grid on the surface of the epigraph to build an initial convex outer approximation. Subgradient inequalities at the optimal solution of the relaxation can be used to iteratively improve the quality of this initial relaxation.

4. Implementing polyhedral branch-and-cut

Our implementation exploits convexity using a three-pronged approach that realizes the strategies discussed in the previous section. The first approach automatically exploits the convexity/concavity of the univariate functions that appear in the recursive decomposition of multivariate convex and nonconvex factorable functions. The second approach exploits recently developed convex underestimators for simple expressions such as x/y. The third approach relies on a modeling language construct that allows the modeler to provide knowledge of univariate or multivariate convex structures to the global optimization solver for the construction of polyhedral outer approximators.

The ideas are implemented in `BARON 7.2`, which employs automatic differentiation [8] to generate subgradient inequalities for constraints that fall in one of the above three categories. An initial set of linear underestimators is first developed based on a crude outer approximation of user-identified convex functions and the univariate functions in the decomposition of the original program. For each univariate function, after extensive experimentation, we decided to use four supporting hyperplanes located by angular trisection [21]. Subsequently, subgradient inequalities are generated iteratively at the current optimal solution of the LP relaxation if the solution is found to violate a known valid inequality from any of the three classes discussed above. These inequalities are used to solve all relaxation and probing subproblems [20]. The cuts generated are used locally and not passed on to the descendants of the branch-and-cut tree. In this sense, our implementation, albeit preliminary, resembles the use of separation techniques to generate cuts in a branch-and-cut algorithm for integer programming.

Finally, in light of the discussion of Subsection 3.4, it seems pertinent at the time of this writing to enhance the modeling interface of a global optimization solver in a manner that provides the modeler with the flexibility of explicitly identifying convex expressions. In our current implementation of `BARON`'s modeling language, this is enabled by the construct:

```
CONVEX_EQUATIONS <constraint names>;
```

5. Illustrative examples

Two examples are presented next to illustrate the use of the above concepts. In both cases, we detail the improvement in the lower bound with each round of cutting plane generation as implemented in `BARON 7.2`.

5.1. Example 1

The following example illustrates the automatic exploitation of convexity of factorable programs and relaxations of x/y. Consider:

$$\begin{aligned} \text{(P)}\ \min\ & x^2 - 100x + y^2 - 30y + 1000\frac{x}{y} \\ \text{s.t.}\ & 0 \le x \le 1000 \\ & 1 \le y \le 1000 \end{aligned}$$

It can be easily verified that the objective function for the above problem is nonconvex. Using (2) and assuming that $x \in [x^L, x^U]$ and $y \in [y^L, y^U]$, a convex relaxation for this problem is easily constructed as follows:

$$\begin{aligned} \text{(R)}\ \min\ & x^2 - 100x + y^2 - 30y + 1000 \max\{f_1(x, y), f_2(x, y), f_3(x, y)\} \\ \text{s.t.}\ & 0 \le x \le 1000 \\ & 1 \le y \le 1000 \end{aligned}$$

For P, using a variety of domain reduction techniques, `BARON 7.2` establishes that $x \in [0, 129.77]$ and $y \in [1, 129.77]$. An initial lower bounding linear program is then constructed by underestimating x^2 and y^2 at four points and using bilinear envelopes for $wy = x$. Then, in an iterative fashion, subgradient inequalities for x^2, y^2, and $\max\{f_1(x, y), f_2(x, y), f_3(x, y)\}$ are constructed as necessary. The following lower bounds are generated at the root node in an actual `BARON 7.2` run:

Iteration	Lower bound	Relaxation optimal solution
1	−7500.9	$x_1 = (65.3, 66.5)$
2	−3832.2	$x_2 = (33.1, 34.1)$
3	−2839.5	$x_3 = (49.2, 19.0)$
4	−2325.7	$x_4 = (41.1, 25.6)$
5	−2057.5	$x_5 = (37.1, 22.3)$
6	−2041.1	$x_6 = (39.1, 23.9)$

The objective function of P at x_6 equals −2034.1, which is an upper bound on the optimal value of R. Therefore, R has been approximately solved to optimality. The true optimal objective function value of R is approximately −2036.60. The optimal solution of P is approximately (34.317, 31.881) with an objective function value of −1117.67. By constructing subgradient inequalities using the convex expressions appearing in R, `BARON 7.2` reduces the root-node relaxation gap by 85.5% in 6 rounds of cutting plane generation. If the convex envelope for x/y described in [18] is used as the underestimating function for x/y, one can derive cutting planes leading to an improved lower bound of −2023.2 for P (this underestimator is not currently implemented in `BARON 7.2`). Using a relative tolerance of 10^{-6}, `BARON 7.2` solves P in 7 nodes when convexity is exploited to generate cutting planes but takes 47 nodes to solve without cutting planes.

5.2. Example 2

$$\begin{aligned}(\mathrm{Q})\ \min\ & 100\log(e^{x_1}+e^{x_2}+e^{x_3})+x_1^2-40x_1\\ & +x_2x_3-10\log(x_1)+x_2^2-20x_2-50x_3\\ \text{s.t.}\ & 1\le x_1\le 10\\ & 1\le x_2\le 10\\ & 1\le x_3\le 10\end{aligned}$$

A relaxation for Q can be easily constructed using bilinear envelopes for x_2x_3 over $[x_2^L, x_2^U]\times[x_3^L, x_3^U]$. However, BARON 7.2 is not able to exploit the convexity of $\log(e^{x_1}+e^{x_2}+e^{x_3})$ since the latter cannot be derived from the convexity/concavity of $\exp(\cdot)/\log(\cdot)$ using the composition rule of Section 2. In the following table, we compare the bound generated by BARON 7.2 while solving the root-node relaxation when convexity of $\log(e^{x_1}+e^{x_2}+e^{x_3})$ is not identified by the user (Lower bound 1) and when convexity of $\log(e^{x_1}+e^{x_2}+e^{x_3})$ has been identified through the CONVEX_EQUATIONS construct (Lower bound 2):

Iteration	Lower bound 1	Lower bound 2
1	−586.9	62.5
2	−514.9	78.9
3	−445.8	79.6
4	−436.2	80.2
5	−432.9	80.6

Considering that the optimal solution has an objective function value of approximately 83.28, the identification of the convexity of $\log(e^{x_1}+e^{x_2}+e^{x_3})$ reduces the root-node relaxation gap by 99.5% in comparison to an automated polyhedral outer approximation scheme that relies only on convexity/concavity of the primitive operations. Subsequently, BARON 7.2 is able to prove optimality within a relative tolerance of 10^{-6} after exploring 35 nodes as opposed to 1793 nodes when the convexity is not specified by the modeler.

6. Computational results with automatic convexity exploitation

The goals of this section are twofold. First, to provide computational evidence that the proposed cutting plane generation techniques are successful in significantly reducing relaxation gaps, even at the root node, for a wide variety of problems. Second, to demonstrate that incorporating the polyhedral cutting planes in every node of a branch-and-cut algorithm results in significant reductions in the computational effort for solving nonlinear and mixed-integer nonlinear optimization problems to global optimality. To achieve these goals, we consider a large number of problems from globallib [15] and minlplib [4]. No convexity information was provided to the solver for the models used in these computations.

All computations reported below were done on a Dell Precision 530 workstation with a 1.7 GHz Pentium IV Xeon processor. All problems were solved in minimization form.

6.1. The test set

From `globallib` and `minlplib`, we eliminated all those problems that were trivial, were too large, had functions that `BARON` cannot currently handle (other than product, power, logarithmic, and exponential), were bilinear or concave quadratic programs for which the procedures developed in this paper do not generate any cutting planes, or were posed over unbounded domains and thus could potentially lead to unreliable lower bound calculations. A total of 40 problems from `globallib` and 47 problems from `minlplib` were selected and are listed in Tables 1 and 2, respectively. In Table 3, we provide some statistics on the size of these problems in terms of the numbers of constraints (m), variables (n), discrete variables (n_{d}), nonzero elements in the constraints and objective (nz), and nonlinear nonzero elements in the constraints and objective (nnz). Most of the problems are sparse with a significant linear component and a significant nonlinear component.

6.2. Root-node bound improvements for `globallib` and `minlplib` problems

For the problems solved, we first computed the root-node relaxation gap, defined as the difference between the calculated upper and lower bounds for the problem at the end of

Table 1. Problems from `globallib` in the test set

(1) alkyl	(11) ex4_1_3	(21) ex8_4_7	(31) qp2
(2) arki0001	(12) ex4_1_4	(22) ex8_4_8	(32) sambal
(3) chain100	(13) ex4_1_6	(23) gsg_0001	(33) srcpm
(4) chain25	(14) ex4_1_7	(24) gtm	(34) st_cqpjk1
(5) chain400	(15) ex5_3_3	(25) hhfair	(35) st_e03
(6) chain50	(16) ex5_4_4	(26) linear	(36) st_e04
(7) chakra	(17) ex8_1_7	(27) meanvar	(37) st_e19
(8) etamac	(18) ex8_1_8	(28) pindyck	(38) st_e20
(9) ex4_1_1	(19) ex8_4_1	(29) process	(39) st_iqpbk1
(10) ex4_1_2	(20) ex8_4_4	(30) qp1	(40) st_iqpbk2

Table 2. Problems from `minlplib` in the test set

(1) batch	(11) ex1223	(21) nvs03	(31) nvs21	(41) synthes3
(2) batchdes	(12) ex1223b	(22) nvs08	(32) nvs23	(42) tls12
(3) cecil_13	(13) ex1224	(23) nvs12	(33) nvs24	(43) tls2
(4) contvar	(14) ex3	(24) nvs13	(34) ortez	(44) tls4
(5) csched1	(15) ex4	(25) nvs14	(35) procsel	(45) tls5
(6) du-opt	(16) fac1	(26) nvs15	(36) ravem	(46) tls6
(7) du-opt5	(17) fac2	(27) nvs17	(37) spectra2	(47) tls7
(8) eniplac	(18) fac3	(28) nvs18	(38) stockcycle	
(9) enpro48	(19) gkocis	(29) nvs19	(39) synthes1	
(10) enpro56	(20) meanvarx	(30) nvs20	(40) synthes2	

Table 3. Size statistics for test set

	problems from `globallib`				problems from `minlplib`				
	m	n	nz	nnz	m	n	n_{d}	nz	nnz
min	1	2	2	1	2	3	2	7	2
max	514	1031	3814	1203	899	841	668	7205	530
average	45	81	276	85	76	99	57	495	59

the root node of the branch-and-cut search. These relaxation gaps were computed for two relaxation strategies: without and with cutting planes. Up to 10 rounds of cutting planes were allowed in the cutting plane strategy. Fewer rounds were used when the relaxation optimal solution did not sufficiently violate the cutting planes generated in the previous round. In each round, one violated cutting plane was generated for each nonlinear term in the univariate decomposition of multivariate functions.

Results for problems from `globallib` are shown in Figures 4 and 5. For all the 40 problems solved, the cutting plane strategy leads to root-node relaxation gap reductions. These reductions range from 0.05% to 100%, with an average of 48%. In other words, the cutting planes eliminated almost half of the root-node relaxation gap for these problems. As Figure 4 indicates, there were seven problems for which the gap was essentially entirely eliminated thanks to cutting planes.

Figure 5 presents the lower bound improvement, defined as the difference between the lower bounds of the two strategies (with and without cutting planes) divided by the absolute value of the lower bound of the strategy without cutting planes. Figure 5 shows that, for five of the problems, the lower bound improved by less than 1%. For the remaining 35 problems, the lower bound improved significantly. Over the entire set of problems, there was a maximum lower bound improvement of 2000%, with an average improvement of 115%.

Similar computations were performed with the 47 problems from `minlplib`. For these problems, the cutting plane strategy leads to improvements to the root-node relaxation gaps that average 20%. As Figure 6 indicates, there is one problem with a significant deterioration in the quality of the root-node relaxation gap when cutting planes are used. This is because the relaxation solution gave a starting point for local search that led to a much worse upper bound. Excluding this problem, all other problems exhibited

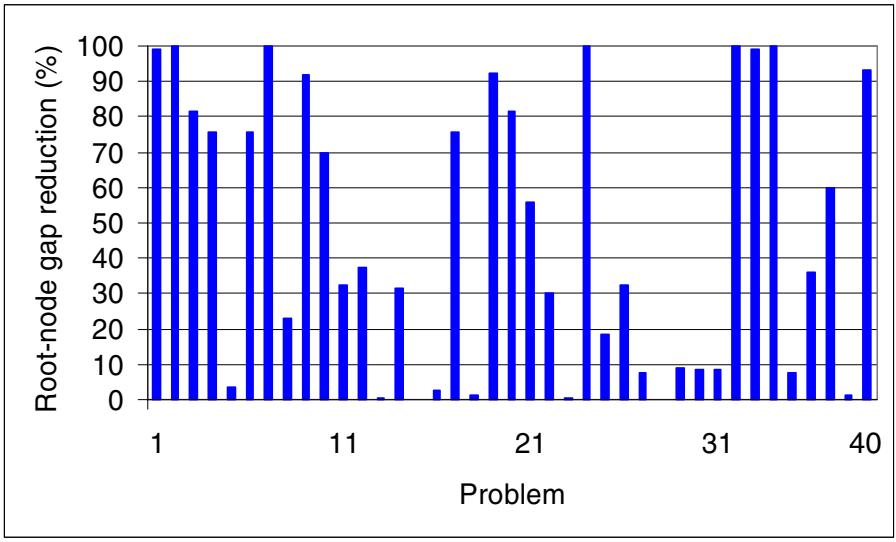

Fig. 4. Root-node gap reductions due to cutting planes for `globallib` problems

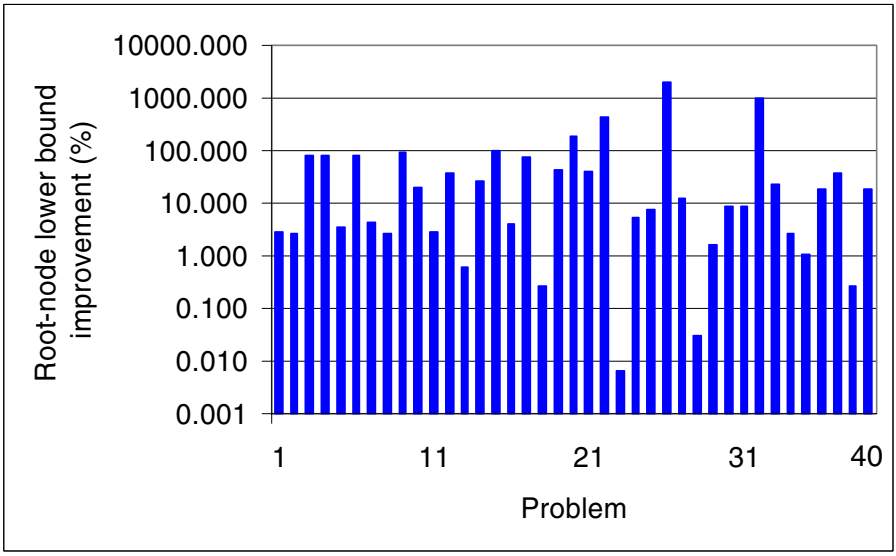

Fig. 5. Root-node lower bound improvements due to cutting planes for `globallib` problems

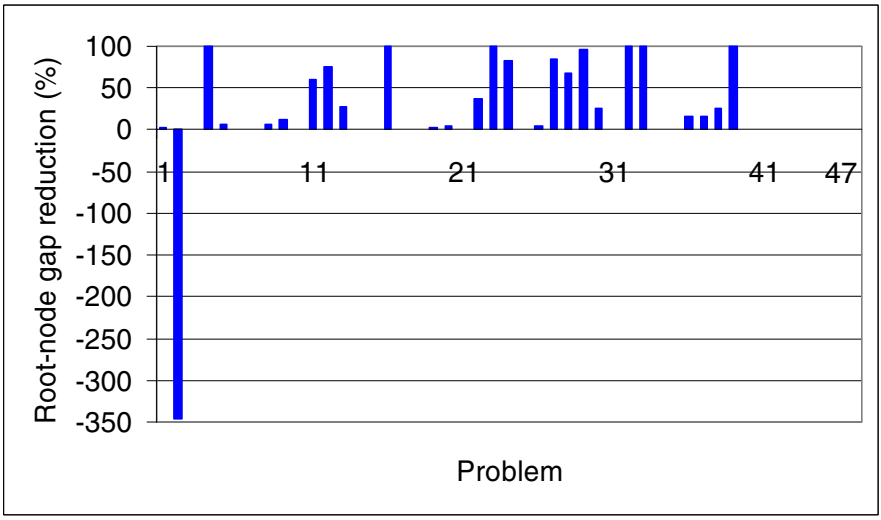

Fig. 6. Root-node gap reductions due to cutting planes for `minlplib` problems

root-node gap reductions, with an average reduction of 27%. As Figure 6 indicates, there are seven problems for which the relaxation gap is essentially entirely eliminated at the root node thanks to the generation of cutting planes.

Figure 7 presents improvements to the lower bounds due to the cutting planes for the 47 problems from `minlplib`. For six of these problems, the improvements

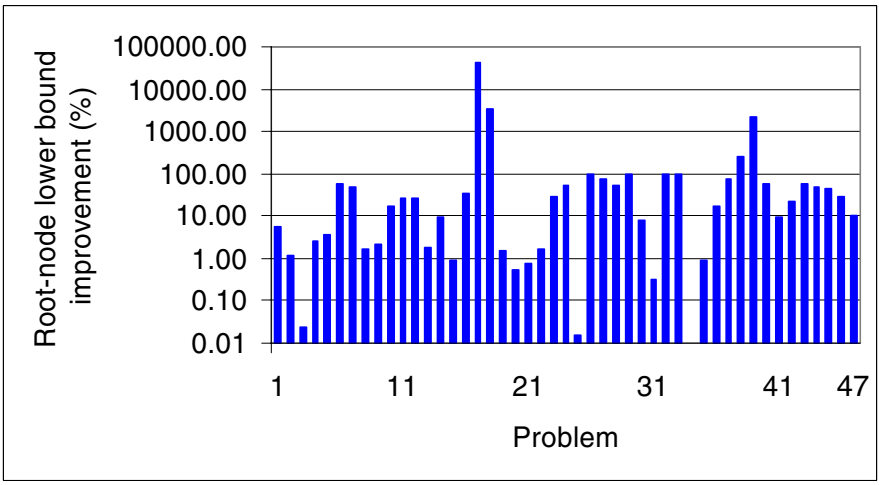

Fig. 7. Root-node lower bound improvements due to cutting planes for `minlplib` problems

are below 1% but, for the majority of the problems, the root-node lower bound improves significantly, often by orders of magnitude. The maximum observed lower bound improvement is 41357%, while improvements average 1032% over this collection of problems.

6.3. Solution to global optimality of problems from `globallib` *and* `minlplib`

The previous subsection demonstrated that automatic exploitation of convexity leads to significant reduction of relaxation gaps for 87 problems from `globallib` and `minlplib`. We now solve to global optimality many of the problems reported in the previous subsection. In particular, we restrict attention to 26 of these problems, each of which was *nontrivial* (required over 5 CPU s) and partial branch-and-bound/cut tree computations indicated that it might be solvable to global optimality within a few CPU hours by at least one of the two strategies considered: without and with cutting planes. All problems are solved using absolute and relative optimality tolerances of 10^{-6} and 10^{-9}, respectively. Termination occurs if either one of these tolerances is met.

Results are presented in Table 4 to compare the two solution strategies. For each problem, this table shows the numbers of constraints (m), variables (n), and discrete variables (n_{d}). Then, for each of the two solution strategies, we present the total number of branch-and-bound/cut iterations (N_{t}), the maximum number of nodes that had to be stored in memory (N_{m}), and the CPU seconds for solution. All runs were restricted to no more than seven CPU hours (25200 s). Problems that reached this limit were not solved to global optimality by the corresponding strategy. This happened for nine problems when no cutting planes were used. On the other hand, the branch-and-cut strategy successfully solved all problems in this set.

The results of Table 4 are analyzed in Table 5 to further quantify the effect of cutting planes on the algorithm. The first line of Table 5 provides the number, and in parentheses the percentage, of problems for which polyhedral cutting planes lead to at least a factor of two improvement of the algorithm in terms of the total number of iterations (N_t), memory requirements (N_m), and CPU time required for termination. The remainder of the table provides similar counts for problems for which the algorithm was improved by at least 30% but no more than an order of magnitude, problems for which there was no significant performance change after addition of cutting planes, and problems for which there was some deterioration in performance because of cutting planes.

It can be seen from Table 5 that, for about two thirds of the problems in this test set, branch-and-bound exhibits at least an order of magnitude improvement because of cutting planes. Only for six of the problems in this test set branch-and-cut took more time than branch-and-bound. Looking at the results of Table 4, it is apparent that five of these problems are amongst the easiest problems in the entire test set in the sense that both strategies take less than 100 seconds to solve them. The only nontrivial problem for which the cutting planes lead to an increase in the computational requirements of branch-and-bound is problem tls4. This problem is mostly linear and the cutting planes, apparently, reduce the size of the search tree at the expense of larger and more time consuming LP relaxations.

Table 4. Computational results with `BARON 7.2` on problems from `globallib` and `minlplib`

				Without cuts			With cuts		
Problem	m	n	n_d	N_t	N_m	CPU s	N_t	N_m	CPU s
arki0001	513	1030		261	157	25200	1	1	137
chakra	41	62		164238	84712	25200	1	1	0
du-opt	9	20	13	82520	44734	25200	79	25	157
du-opt5	9	20	12	106299	33954	25200	78	15	92
elf	38	54	24	866	91	31	698	89	95
enpro48	215	154	92	4215	434	165	702	92	84
enpro56	192	128	73	4609	477	182	1354	99	101
ex1233	64	52	12	28122	652	922	2198	310	1246
ex5_4_4	19	27		971	93	33	843	80	46
ex6_2_14	2	4		739	65	7	2713	115	38
ex8_4_1	10	22		926853	58084	17864	13	4	4
ex8_4_4	12	17		2655	154	63	101	12	7
ex8_4_7	40	62		12497	7021	25200	10011	6658	1883
fac1	18	22	6	78055	8842	143	1	1	0
fac2	33	66	12	4653901	101202	25200	27	8	1
fac3	33	66	12	11516667	101172	25200	24	7	0
gsg_0001	112	78		145	22	17	89	10	21
gtm	24	63		3229450	87622	25200	1	1	0
himmel16	21	18		1211	152	27	915	116	81
linear	20	24		1904473	19706	24403	59472	955	2772
parallel	115	205	25	651	65	198	555	72	194
ravem	186	112	53	778	134	30	246	52	18
sambal	10	17		9279	436	119	1	1	0
spectra2	73	70	30	1963	330	54	43	10	6
stockcycle	98	481	432	108260	67671	25200	1573	231	1152
tls4	64	105	89	191760	4357	4106	172015	4807	12295
Average	76	115	63	885825	23936	10583	9760	530	786

Table 5. Effect of polyhedral cutting planes

Effect of adding cuts	Iterations	Memory	CPU time
Better by a factor at least two	18 (69%)	19 (73%)	15 (58%)
Between 30% and 100% better	2 (8%)	1 (4%)	3 (12%)
Difference smaller than 30%	5 (19%)	5 (19%)	2 (8%)
Between 30% and 100% worse	1 (4%)	1 (4%)	6 (23%)
Worse by a factor at least two	0 (0%)	0 (0%)	0 (0%)

As indicated by the last row of Table 4, cutting plane generation reduces CPU times by an average of 93% for the entire collection. The overall time reduces despite increased processing time for each node because of significant reduction in the size of the trees due to the improved lower bounds. The reductions in the number of iterations and memory requirements due to cutting planes are equally dramatic as CPU time reductions: 99% and 98%, respectively.

Finally, Table 6 provides the objective function values for the problems solved. Problems marked with a * have not been reported as solved before in `globallib` and `minlplib`. For problem ex8_4_7, the objective function value of 28.6032 corresponds to a point that satisfies `BARON`'s default feasibility tolerance of 10^{-5}. With a feasibility tolerance of 10^{-6}, this point is no longer reported as feasible by `BARON` and the solution obtained is a nearby point with an objective function value of 29.04731 and which is identical with solutions reported in the literature for this problem. All constraints of this problem are nonconvex equalities, thus making the problem numerically sensitive to tolerances. For problem tls4, the previous best known solution, as reported in `minlplib` [4], had a value of 9.3. As Table 6 indicates, this problem's globally optimal solution corresponds to an objective function value of 8.3. The global solution corresponds to a combination of integer variables that is very different to that of the earlier best known solution.

Table 6. Global objectives

Problem	Objective		Problem	Objective	
arki0001	40.7129	*	fac1	160912612.3500	
chakra	-179.1336	*	fac2	331837498.1770	
du-opt	3.5563		fac3	31982309.8480	
du-opt5	8.0737		gsg_0001	2378.1605	*
elf	0.1917		gtm	543.5651	*
enpro48	187276.7080		himmel16	-0.8660	
enpro56	263427.2212		linear	88.9994	*
ex1233	155010.6713		parallel	924.2956	
ex5_4_4	10077.7754	*	ravem	269589.5584	
ex6_2_14	-0.6954		sambal	3.9682	
ex8_4_1	0.6185	*	spectra2	13.9783	
ex8_4_4	0.2125		stockcycle	119948.6883	
ex8_4_7	28.6032	†*	tls4	8.3000	¶

†: Infeasible solution point returned.
*: No solutions reported for these problems in `globallib` and `minlplib`.
¶: Better solution than the one reported earlier in `minlplib`.

7. Conclusions

This paper demonstrates that the generation of cutting planes from convex constraints or convex relaxations of nonconvex programs significantly accelerates a branch-and-bound global optimization algorithm by enhancing its lower bounding capabilities. These cutting planes reduced branch-and-bound solution time and memory requirements by over 93% on a collection of 26 nontrivial problems from `globallib` and `minlplib`. Nine of these problems were not solvable by previous techniques within seven CPU hours.

Outer approximation of a decomposed equivalent automatically exploits convexity without requiring explicit strategies for identification of convexity. As such decompositions are becoming commonplace in global optimization systems, exploiting convexity in the manner described in this paper is natural and rewarding.

The potential impact of the proposed functional decomposition and convexification strategy on classical convex programming techniques is worth investigating in the future. Several extensions of this work are possible for nonconvex programming as well. For instance, much remains to be done to identify the optimal mix of cutting and branching in terms of frequency and number of generated cutting planes. More important would be the extension of the branch-and-cut framework to supplement the convexity cuts used here by additional partial convex hull representations obtained from feasibility and optimality arguments.

Acknowledgements. The authors would like to thank two anonymous referees for comments and suggestions that improved the quality of this manuscript.

References

1. Audet, C., Hansen, P., Jaumard, B.,Savard, G. : A branch and cut algorithm for nonconvex quadratically constrained quadratic programming. Math. Prog. **87**, 131–152 (2000)
2. Avriel, M., Diewert, W.E., Schaible, S., Zang, I.: Generalized Concavity. Plenum Press, 1988
3. Böröczky K.Jr., Reitzner,M.: Approximation of smooth convex bodies by random circumscribed polytopes. Annals of Applied Probability **14**, 239–273 (2004)
4. Bussieck, M.R.: MINLP World. `http://www.gamsworld.org/minlp/index.htm` 2002
5. Chinneck, J.W.: Discovering the characteristics of mathematical programming via sampling. Optimization Methods and Software **17**, 319–352 (2002)
6. Duran, M.A., Grossmann, I.E.: An outer-approximation algorithm for a class of mixed-integer nonlinear programs. Math. Prog. **36**, 307–339 (1986)
7. Fourer, R., Moré, J., Munson, T., Sarich, J.: Next-generation servers for optimization as an internet resource. Available at `http://iems.nwu.edu/~4er` 2004
8. Griewank, A.: Evaluating derivatives. Principles and Techniques of Algorithmic Differentiation, vol **19** of Frontiers in Applied Mathematics. SIAM, Philadelphia, PA, 2000
9. Griffith, R.E., Stewart, R.A.: A nonlinear programming technique for the optimization of continuous processing systems. Management Science **7**, 379–392 (1961)
10. Gruber, P.M.: Asymptotic estimates for best and stepwise approximation of convex bodies II. Forum Mathematicum **5**, 521–538 (1993)
11. Hiriart-Urruty, J.-B., Lemaréchal, C.: Convex Analysis and Minimization Algorithms I. Springer-Verlag, Berlin, 1993
12. Kelley, J.E.: The cutting plane method for solving convex programs. Journal of the SIAM **8**, 703–712 (1960)
13. Maheshwari, C., Neumaier, A., Schichl, H.: Convexity and concavity detection. Available at `http://www.mat.univie.ac.at/~herman/techreports/D12convconc.ps` 2003
14. McCormick, G.P.: Computability of global solutions to factorable nonconvex programs: Part I—Convex underestimating problems. Math. Prog. **10**, 147–175 (1976)

15. Meeraus, A.: GLOBAL World. http://www.gamsworld.org/global/index.htm 2002
16. Nowak, I.: Relaxation and Decomposition Methods for Mixed Integer Nonlinear Programming. Habilitation thesis Humboldt-Universität zu Berlin, Germany, 2004
17. Rote, G.: The convergence rate of the sandwich algorithm for approximating convex functions. Computing **48**, 337–361 (1992)
18. Tawarmalani, M., Sahinidis, N.V.: Semidefinite relaxations of fractional programs via novel techniques for constructing convex envelopes of nonlinear functions. Journal of Global Optimization **20**, 137–158 (2001)
19. Tawarmalani, M., Sahinidis, N.V.: Convex extensions and convex envelopes of l.s.c. functions. Mathematical Programming **93**, 247–263 (2002)
20. Tawarmalani, M., Sahinidis, N.V.: Convexification and Global Optimization in Continuous and Mixed-Integer Nonlinear Programming: Theory, Algorithms, Software, and Applications. Kluwer Academic Publishers, Dordrecht, 2002
21. Tawarmalani, M., Sahinidis, N.V.: Global optimization of mixed-integer nonlinear programs: A theoretical and computational study. Math. Prog. **99**, 563–591 (2004)
22. Vandenbussche, D.: Polyhedral Approaches to Solving Nonconvex Quadratic Programs. PhD thesis, Georgia Institute of Technology, Department of Indystrial and Systems Engineering, Atlanta, GA, 2003
23. Zamora, J.M., Grossmann, I.E.: A global MINLP optimization algorithm for the synthesis of heat exchanger networks with no stream splits. Computers & Chemical Engineering **22**, 367–384 (1998)
24. Zamora, J.M., Grossmann, I.E.: A branch and contract algorithm for problems with concave univariate, bilinear and linear fractional terms. Journal of Global Optimization **14**, 217–249 (1999)