Math. Program., Ser. A 100: 517–535 (2004)

Digital Object Identifier (DOI) 10.1007/s10107-004-0503-1

Padmanaban Kesavan · Russell J. Allgor · Edward P. Gatzke · Paul I. Barton[*]

Outer approximation algorithms for separable nonconvex mixed-integer nonlinear programs

Received: May 5, 1999 / Accepted: December 26, 2003
Published online: 21 May 2004 – © Springer-Verlag 2004

Abstract. A rigorous decomposition approach to solve separable mixed-integer nonlinear programs where the participating functions are nonconvex is presented. The proposed algorithms consist of solving an alternating sequence of Relaxed Master Problems (mixed-integer linear program) and two nonlinear programming problems (NLPs). A sequence of valid nondecreasing lower bounds and upper bounds is generated by the algorithms which converge in a finite number of iterations. A Primal Bounding Problem is introduced, which is a convex NLP solved at each iteration to derive valid outer approximations of the nonconvex functions in the continuous space. Two decomposition algorithms are presented in this work. On finite termination, the first yields the global solution to the original nonconvex MINLP and the second finds a rigorous bound to the global solution. Convergence and optimality properties, and refinement of the algorithms for efficient implementation are presented. Finally, numerical results are compared with currently available algorithms for example problems, illuminating the potential benefits of the proposed algorithm.

Key words. Mixed-integer nonconvex nonlinear programming – Decomposition algorithms – Global solution

1. Introduction

A general class of optimization problems involving integer and continuous variables can be defined as:

$$\begin{aligned} &\min_{x,y} f(x,y) \\ \text{s.t.}\quad & h(x,y) = 0 \\ & g(x,y) \leq 0 \\ & x \in X \subset \mathbb{R}^n \\ & y \in Y = \{0,1\}^q. \end{aligned} \tag{1}$$

This problem is essentially finding the minimum of a real valued function (f) subject to constraints defined by vector valued functions (g and h) in the continuous-discrete (x-y) space. Integer and discrete valued variables with given lower and upper bounds may always be represented by sets of binary variables [13]. Problems of this type are generally termed Mixed-Integer Nonlinear Programming (MINLP) problems. MINLP

P. Kesavan: Sensitron Semiconductors, Deer Park NY

R.J. Allgor: Amazon Com Inc., Seattle, WA

E.P. Gatzke: Department of Chemical Engineering, University of South Carolina.

P.I. Barton: Department of Chemical Engineering, Massachusetts Institute of Technology

[*] Author to whom all correspondence should be addressed. e-mail: pib@mit.edu, Ph: (617) 253-6526

has applications in several disciplines: design and scheduling of multipurpose batch plants [29], topology optimization in transportation networks [18], optimal unit allocation in electrical networks [9], computer aided molecular design [27] and process systems synthesis [21] are some examples.

Significant advances have been made in the last four decades in solving problems defined by equation 1 by exploiting the special problem structures that result under certain assumptions. A class of such algorithms known collectively as *decomposition strategies* are derived based on the principles of projection, outer approximation and relaxation. Generalized Benders Decomposition (GBD) [16] and its variants use dual information for outer approximation and the Outer Approximation (OA) algorithm and its variants are based on using optimal primal information for outer approximation [10] [12] [38]. The GBD algorithms are valid under the main assumptions that the functions $f : X \times U \to \mathbb{R}$ and $g : X \times U \to \mathbb{R}^p$ are convex, and $h : X \times U \to \mathbb{R}^m$ is linear in the continuous variable $x \in X \subset \mathbb{R}^n$ for each fixed $y \in Y = \{0, 1\}^q$, $Y \subset U = [0, 1]^q$. Several variants of GBD have been developed under further assumptions such as f, g and h being separable in x and y, and the y variables appear linearly in the problem defined by equation 1. The original OA algorithm [10] can find the global solution of problem 1 when the functions (f and g) are convex and separable in x and y and are linear in y, in the absence of equality constraints (h). The OA algorithm developed by Fletcher and Leyffer [12] can solve the problem defined by equation 1 to find the global optimum in the absence of nonlinear equality constraints (the algorithm allows linear equality constraints as long as they are included in the definition of $X \times U$) and when f and g are convex. An excellent review of currently existing algorithms when f, g and h are convex is given in Floudas [13].

Many practical optimization problems often involve nonconvex functions f, g and h, and this class of problem is generally termed a nonconvex MINLP. Global solutions of these problems cannot be guaranteed by the aforementioned algorithms. The reason is that linearizations derived at the solutions (even global) of the nonconvex primal NLPs can cut off portions of the feasible space, and hence a global solution cannot be guaranteed by OA. Similarly, in the GBD algorithms the support functions derived can also cut off portions of the feasible space [31]. However, more recently, deterministic algorithms based on branch and bound strategies have been developed (e.g., branch-and-reduce [30], GMIN-αBB and SMIN-αBB [1] and spatial branch and bound [34]) to find the global solutions of nonconvex MINLPs. All of these algorithms are based on solving a lower bounding Relaxed Master Problem and an upper bounding Primal Problem at each node of the branch and bound tree. They differ only in the formulation of the Master and Primal problem, and the heuristic branching strategy. A brief review of these algorithms is given in the following paragraphs.

The Master Problem in all the algorithms is derived using the principle of convex relaxation of a nonconvex function defined on a specified domain. The concept of constructing the convex hull of the feasible region and convex envelope of the objective function to solve nonconvex NLPs has long been established [11] [19] [35]. The major difficulty in implementation however has been in deriving the same efficiently in practice. There have been significant developments in deriving convex relaxations of a given function of special structure in the last couple of decades. For example, convex envelopes for bilinear functions [25] [6], univariate concave, trilinear, fractional and fractional trilinear functions [24] have all been derived. A method for constructing convex underestimators for factorable functions was presented in [25]. More recently, a valid convex underestimator for arbitrary general twice continuously differentiable nonconvex functions has been derived [3] [23]. It should be noted that while elucidating the convex envelope of a given function is as hard as solving the nonconvex optimization problem, deriving a convex underestimator (not necessarily the convex envelope) of a given function can be achieved with polynomial complexity for a broad class of functions [3] [25]. This concept is pivotal to this paper because any valid convex underestimating function will serve the desired theoretical purpose, at the cost of a potential loss of efficiency.

The branch-and-reduce algorithm [30] is applicable to nonconvex MINLPs of the general type defined by equation 1. The lower bounding Master problem is obtained by relaxing the binary variables in the nonconvex MINLP to be continuous, adding constraints that force these relaxed binary variables to take discrete values, convexifying this NLP, and then solving the resulting convex NLP at each node. Any procedure (which is valid) could be used to find the upper bound. A set of branching rules has been developed.

The SMIN-αBB algorithm [1] is applicable to nonconvex MINLPs in which the integer variables participate linearly or in bilinear mixed integer terms in equation 1. A branching strategy that allows branching on a combination of integer and continuous variables has been developed. For each partition, a valid convex MINLP is constructed based on the principles of αBB [3] [23] for twice continuously differentiable functions. The solution of the convex MINLP and the (rather unpredictable) result produced by applying a standard procedure such as OA and GBD to the original nonconvex MINLP provide the lower and upper bounds respectively, restricted to the current partition (node of the branch-and-bound tree). The upper and lower bounding MINLPs can be solved by GBD or OA. The GMIN-αBB algorithm [1] is applicable to nonconvex MINLPs described by equation 1 and where the functions f, g and h are twice continuously differentiable. The algorithm is very similar to SMIN-αBB except for the generation of the lower bounds, which are obtained in each region of interest (each node of the branch-and-bound tree) by first relaxing the integer variables to be continuous, and then solving the resulting nonconvex NLP using the α-BB algorithm [3].

A heuristic based set of branching rules to prune the feasible space have been suggested in each of these algorithms. On the other hand, the development of decomposition algorithms to solve nonconvex MINLPs has been limited due to the difficulty in constructing valid support functions/linearizations, as mentioned previously. However, empirical experience indicates that decomposition algorithms perform better on average than Branch and Bound approaches [13] for solving convex MINLPs, and this motivates the development of decomposition approaches to solve nonconvex MINLPs. In this paper, we show that the convex underestimators employed by the branch-and-bound approaches may also be exploited to develop rigorous decomposition algorithms for nonconvex MINLPs. The theoretical and algorithmic development is detailed in the sections that follow.

2. Problem description and reformulation

The class of nonconvex MINLPs considered in the present work conform to the following formulation.

$$\mathrm{P} = \begin{cases} \min\limits_{x,y} c_1^T y + f(x) \\ \text{s.t. } g_1(x) + B_1 y \leq 0 \\ \quad\; g_2(x) + B_2 y \leq 0 \\ \quad\; x \in X \subset \mathbb{R}^n \\ y \in Y = \{0, 1\}^q \end{cases} \tag{2}$$

where $f : X \to \mathbb{R}$ and $g_1 : X \to \mathbb{R}^{p_1}$ are continuous but nonconvex, and $g_2 : X \to \mathbb{R}^{p_2}$ is convex on the nonempty, compact, convex set defined by $X = \{x : x \in \mathbb{R}^n,\ D_1 x \leq c_2\}$. B_1, B_2, D_1 and c_1, c_2 are matrices and vectors of conformable dimensions respectively. The problem as defined by equation 2 will be referred to as P hereafter. These assumptions are sufficient to guarantee that either a minimum exists or the problem is infeasible.

Two different algorithms are developed in this paper to solve P as defined above. On finite termination, the first algorithm finds the global solution of P and the second algorithm finds rigorous bounds bracketing the global solution of P. The algorithms presented here are based on the construction of the following subproblems:

Lower Bounding Convex MINLP the solution of which yields a valid lower bound to the global solution of problem P.

Master Problem a MILP, the solution of which represents a valid lower bound to the global solution of P.

Relaxed Master Problem a MILP, the solution of which represents a valid lower bound on that subset of Y not yet explored by the algorithm.

Primal Problem which is a nonconvex NLP obtained by fixing the binary variables (y) in P. Any feasible solution yields a rigorous upper bound to the solution of nonconvex MINLP (P).

Primal Bounding Problem which is a convex NLP and the solution of which provides a valid and a tighter lower bound to the Primal problem for each fixed binary realization y than that provided by the Relaxed Master problem that generates y.

In the remainder of this section, the problems defined above will be derived. In Section 3 the algorithms are presented and their properties are discussed. In Section 4.1 implementation considerations and refinements of the algorithms are presented. Finally, in Sections 4.2–4.3, preliminary results are presented with comparison to currently available algorithms.

2.1. *Lower bounding convex MINLP*

Problem P reduces to a convex MINLP if the functions f and g_1 are convex on X, which can be solved to global optimality using GBD or OA. Indeed, our algorithm reduces to OA in this case because the Primal and Primal Bounding Problems become equivalent, assuming the primal bounding problem convex underestimators coincide with the convex function. Otherwise, the aforementioned algorithms can yield invalid support functions or linearizations which cut off portions of the feasible region, and hence convergence to the global solution of P cannot be guaranteed, and convergence to an arbitrary suboptimal point is more likely [1] [31].

Since problem P is separable in the continuous and the integer variables, the continuous and discrete feasible spaces can be individually characterized [10]. In order to construct a valid lower bounding convex MINLP for Problem P, it suffices to convexify and underestimate the nonconvex functions defined in the continuous variables (f and g_1). The convex envelopes of the functions f and g_1 are not necessary and any valid convex underestimator is sufficient. However, the closer this underestimator is to the convex envelope, the tighter the lower bound thus generated. Several methods have been developed for relatively broad classes of problems. Hence with the current state-of-the-art, given a twice continuously differentiable or factorable MINLP as defined by equation 2, a lower bounding convex MINLP can be constructed. Moreover, we refrain from restricting problem P to twice continuously differentiable functions, since any advances that yield convex underestimators for broader classes of functions can be exploited by the algorithms presented in this paper. Thus, we merely require continuity of the participating functions and existence of convex underestimators.

Let L_1 and L_2 represent the convex underestimators of f and g_1 on X respectively. Conceptually, this is equivalent to underestimating the objective function and overestimating the continuous feasible space of Problem P. The lower bounding convex MINLP (referred to as P1 hereafter) is:

$$\text{P1} = \begin{cases} \min\limits_{x,y} c_1^T y + L_1(x) \\ \text{s.t. } L_2(x) + B_1 y \leq 0 \\ \qquad g_2(x) + B_2 y \leq 0 \\ \qquad x \in X \subset \mathbb{R}^n \\ \quad y \in Y = \{0, 1\}^q. \end{cases} \tag{3}$$

Every element of the feasible set of P for each y is in the feasible set of P1 and there may be elements of feasible set of P1 that are not in P. Therefore P1 contains the feasible set of P for each $y \in Y$, and the objective function of P1 underestimates the objective function of P for each $y \in Y$. Hence the solution of P1 represents a valid lower bound to the global solution of P by construction.

In order to derive P1 from P additional assumptions are required on Problem P which depend upon the specific nature of the problem being solved, as discussed below:

1. The functions $f(x)$ and $g_1(x)$ can be represented as sums of terms of special structure (for example, bilinear, univariate, concave, etc.) only, for which once continuously differentiable convex underestimators can be derived without any assumptions on the differentiability of functions f and g_1. In this case no additional assumptions on P are necessary. P as defined is then very general and covers a broad class of problems. Furthermore, by the principles of symbolic reformulation [32] a wide variety of factorable problems can be reformulated automatically into problems involving terms of the special structure. A detailed discussion is presented in Smith [33].
2. The functions $f(x)$ and $g_1(x)$ involve general nonconvex terms. At present, a valid convex underestimating function can be derived for a given nonconvex elementary function that is twice continuously differentiable [3]. Therefore an additional assumption that general nonconvex terms in the functions f and g_1 be twice continuously differentiable is required to be satisfied by P in this case.

The following assumptions on P1 are necessary:

1. The functions L_1, L_2 and g_2 are once continuously differentiable in an open neighborhood of the KKT points of each subproblem (convex NLP) obtained by fixing the binary variables y in P1.
2. A constraint qualification holds at the solution of every NLP subproblem obtained by fixing the binary variables y in P1.

Remark 1. Assumptions 1 and 2 guarantee that the KKT conditions are both necessary and sufficient to identify the global minimum of the Primal Bounding problems at each iteration of the proposed algorithms.

Remark 2. Assumption 1 is slightly different from that assumed by OA [10] [12] in that the functions L_1, L_2 and g_2 are not required to be once continuously differentiable over the entire domain. However, the OA algorithm is derived based on stronger assumptions to ensure that assumptions 1 and 2 are satisfied. Therefore the OA algorithm with the above weaker assumptions is still valid to find the global minimum of convex MINLPs.

Remark 3. The convex underestimators derived for P consisting of bilinear, trilinear, univariate concave, fractional and fractional trilinear terms are once continuously differentiable [3]. Hence L_1 and L_2 are once continuously differentiable by construction. Also, if f and g_1 contain terms of generic structure that are twice continuously differentiable, then the corresponding terms derived in L_1 and L_2 are also twice continuously differentiable by construction [3].

2.2. *Primal problem*

The Primal Problem (NLP(y^j)) is a nonconvex NLP obtained by fixing the binary variables ($y = y^j$) in P:

$$\text{NLP}(y^j) = \begin{cases} \min\limits_x c_1^T y^j + f(x) \\ \text{s.t.}\ \ g_1(x) + B_1 y^j \le 0 \\ \qquad g_2(x) + B_2 y^j \le 0 \\ \qquad\qquad x \in X. \end{cases} \tag{4}$$

Any solution (local or global) of the Primal Problem is a valid upper bound (UBD) on the global solution of P. It is assumed that this problem can be solved to ϵ optimality in a finite number of steps using deterministic methods. In some cases for global NLP problems of special structure, $\epsilon = 0$ convergence can be guaranteed in a finite number of steps.

2.3. *Primal bounding problem*

Fixing the integer variables ($y = y^j$) in Problem P1 yields:

$$\text{NLPB}(y^j) = \begin{cases} \min\limits_x c_1^T y^j + L_1(x) \\ \text{s.t.}\ \ L_2(x) + B_1 y^j \le 0 \\ \qquad g_2(x) + B_2 y^j \le 0 \\ \qquad\qquad x \in X. \end{cases} \tag{5}$$

For a fixed realization of y, the feasible set of NLPB(y^j) overestimates the feasible set of NLP(y^j) and underestimates the objective function of NLP(y^j). Hence the solution of NLPB(y^j) represents a valid lower bound to the solution of NLP(y^j). Furthermore, the solution of NLPB(y^j) is greater than or equal to the solution of the Relaxed Master Problem that generated y^j (this will be established below). Hence NLPB(y^j) is a Primal Bounding Problem for P.

2.4. Master problem

Problem P1 is a MINLP which is linear in the discrete (y) variables and in which the functions L_1, L_2 and g_2 are convex and once continuously differentiable (by construction). Therefore, the outer approximation development of Duran and Grossmann [10], later modified by Fletcher and Leyffer [12], can be used to derive the equivalent MILP (M) which is given below:

$$
\mathrm{M} = \left\{ \begin{array}{l}
\min\limits_{x,y,\eta} \ \eta \\
\text{s.t.} \\
\left.\begin{array}{l}
\eta \geq L_1(x^j) + \nabla L_1(x^j)^T(x - x^j) + c_1^T y \\
L_2(x^j) + \nabla L_2(x^j)^T(x - x^j) + B_1 y \leq 0 \\
g_2(x^j) + \nabla g_2(x^j)^T(x - x^j) + B_2 y \leq 0
\end{array}\right\} \forall j \in T \\
\\
\left.\begin{array}{l}
L_2(x^k) + \nabla L_2(x^k)^T(x - x^k) + B_1 y \leq 0 \\
g_2(x^k) + \nabla g_2(x^k)^T(x - x^k) + B_2 y \leq 0
\end{array}\right\} \forall k \in S \\
x \in X, y \in Y.
\end{array}\right. \tag{6}
$$

where,
$T = \{\, j$: NLPB(y^j) is feasible and x^j is the optimal solution to NLPB(y^j) $\}$
$S = \{\, j$: NLPB(y^j) is infeasible and x^j is the optimal solution to $F(y^j)$ (as defined in [12]) $\}$

Remark 4. Derivation of M requires the assertion of a constraint qualification at the KKT points of the Primal Bounding Problems on which the outer approximation is based [12]. While any suitable constraint qualification can be employed, the treatment of equality constraints appearing in P by the constraint qualification merits some comments. Any nonlinear equalities will be relaxed to a pair of convex inequalities in the Primal Bounding Problem, and will be treated as inequalities in the constraint qualification. On the other hand, any linear equalities will appear as a linearly dependent pair of linear inequalities in the Primal Bounding Problem. Such linear pairs should be treated as the original equality in any suitable constraint qualification.

Theorem 1 of Fletcher and Leyffer [12] establishes the relation between P1 and M.

Theorem 1. *M is equivalent to P1 in the sense that x^*, y^* solves P1 iff it solves M.*

Since P1 is a relaxation of P, the following Corollary establishes the relationship between P and M.

Corollary 1. *If (x^*, y^*) is the global optimal solution of P then it is feasible in M.*

Proof. The proof of theorem 1 establishes that the feasible region of M is a relaxation of that of P1. Similarly, by construction the feasible region of P1 is a relaxation of that of P. Hence (x^*, y^*) is feasible in M. □

2.5. *Relaxed master problem*

The solution of the Master Problem (identical to the solution of P1) represents a valid lower bound to the solution of P. This requires solution of all the NLPBs and is impractical to solve. Therefore, relaxations of M are solved at each iteration (similar to the original OA algorithm). Since the Primal Bounding Problem is a convex NLP and is a valid underestimator of the nonconvex NLP (Primal Problem) for each integer realization, and since the integer variables appear linearly and are separable, valid linearizations of the constraints can be derived at the solution of the Primal Bounding Problem at each iteration. The solution of the Primal Problem at each iteration is therefore *not* used to derive the linearizations required to construct the current Relaxed Master Problem. Integer cuts [8] that exclude the previously examined integer realizations are added to the Relaxed Master problem. The solution of the Relaxed Master Problem yields a new integer realization and the iteration is repeated. The solution of the Primal Problem however is required to update the upper bound to the global solution of P (UBD). This is done by solving only those Primal Problems corresponding to the integer realizations with a Primal Bounding solution less than or equal to the global solution of P. The algorithm terminates when the minimum of the Primal Bounding solutions (UBDPB) is greater than or equal to the current UBD or the Relaxed Master is infeasible.

At each iteration in both the algorithms, a new integer realization is chosen and the Primal Bounding (NLPB(y^k)) problem is solved. Either a feasible solution of the Primal Bounding Problem x^k is obtained or it is infeasible in which case the feasibility problem $F(y^k)$ [12] is solved. Note that, if NLPB(y^k) is infeasible then NLP(y^k) is also infeasible. The sets T and S are replaced in the algorithms by:
$T^k = \{ j | j \leq k : \text{NLPB}(y^j)$ is feasible and x^j is the optimal solution to $\text{NLPB}(y^j) \}$
$S^k = \{ j | j \leq k : \text{NLPB}(y^j)$ is infeasible and x^j is the optimal solution to $F(y^j) \}$.
An additional constraint to prevent the previously examined integer realization (y^k) from becoming a solution is also added at each iteration to the Relaxed Master Problem. If $k \in S^k$, the linearizations derived from the solution of the feasibility problem exclude y^k [12] and hence no additional constraint is necessary. If $k \in T^k$, then an integer cut [8] that excludes y^k is added to the Relaxed Master problem. The Relaxed Master Problem solved at iteration k then is:

$$M^k = \begin{cases} \min\limits_{x,y,\eta} \ \eta \\ \text{s.t.} \\ \left.\begin{array}{l} \eta \geq L_1(x^j) + \nabla L_1(x^j)^T(x - x^j) + c_1^T y \\ L_2(x^j) + \nabla L_2(x^j)^T(x - x^j) + B_1 y \leq 0 \\ g_2(x^j) + \nabla g_2(x^j)^T(x - x^j) + B_2 y \leq 0 \end{array}\right\} \forall j \in T^k \\ \\ \left.\begin{array}{l} L_2(x^i) + \nabla L_2(x^i)^T(x - x^i) + B_1 y \leq 0 \\ g_2(x^i) + \nabla g_2(x^i)^T(x - x^i) + B_2 y \leq 0 \end{array}\right\} \forall i \in S^k \\ \sum_{i \in B^j} y_i - \sum_{i \in NB^j} y_i \leq |B^j| - 1, \forall j \in T^k \\ B^j = \{i : y_i^j = 1\}, NB^j = \{i : y_i^j = 0\} \\ x \in X, y \in Y. \end{cases} \tag{7}$$

Remark 5. Whilst the constraint $\eta < UBD$ is sufficient to exclude y^j, $j \in T^k$ in the Relaxed Master problem derived by Fletcher and Leyffer [12] for convex MINLPs, in the nonconvex MINLP case UBD is unrelated to previous solutions of the Primal Bounding Problem and thus an integer cut is necessary.

Remark 6. The solution of M^k represents a lower bound to the corresponding Primal Bounding Problem due to the convexity of P1.

Remark 7. The Relaxed Master Problem in Outer Approximation with Equality Relaxation and Augmented Penalty (OA/ER/AP) proposed by Viswanathan and Grossmann [37] to solve problems which conform to the form defined by P, relaxes the linearizations derived at the local solution of the Primal Problem. Even though the feasible region is expanded by such an approach, no theoretical guarantees can be given about the solution since the algorithm may still cut off portions of the feasible region and converge to a suboptimal solution. On the other hand, in Algorithms 1 and 2 valid linearizations are derived at the solution of the Primal Bounding Problem. The feasible region is overestimated and convergence to the optimal solution of P (Algorithm 1) or a rigorous bound to the global solution (Algorithm 2) is guaranteed (if P is feasible) by the decomposition strategy proposed in this paper.

3. The Algorithms

3.1. Algorithm 1: Global solution of nonconvex MINLPs

```
Initialize:
1. Iteration counter $k = 0, l = 1, T^0 = \emptyset, S^0 = \emptyset, U^0 = \emptyset$.
2. UBD = $+\infty$, UBDPB = $+\infty$.
3. Integer combination $y^1$ is given.
REPEAT
  IF ($k = 0$ or ($M^k$ is feasible and $\eta^k <$ UBDPB and $\eta^k <$ UBD)) THEN
    REPEAT
      Set $k = k + 1$
      1. Solve the Primal Bounding Problem (NLPB($y^k$)). If NLPB($y^k$) is infeasible,
         solve the feasibility problem $F(y^k)$. Let the solution be $x^k$.
      2. Linearize the objective and active constraint functions of P1 about ($x^k$, $y^k$).
         Set ($S^k = S^{k-1}$ and $T^k = T^{k-1} \cup \{k\}$ ) or ($S^k = S^{k-1} \cup \{k\}$ and $T^k = T^{k-1}$)
         as the case may be.
      3. If NLPB ($y^k$) is feasible and $c_1^T y^k + L_1(x^k) <$UBDPB, update $x^* = x^k$, $y^*$
         $= y^k$, $k^* = k$ and UBDPB = $L_1(x^k)$+$c_1^T y^k$.
      4. Solve the current relaxation $M^k$ (solution $\eta^k$) of P yielding a new integer
         assignment $y^{k+1}$.
    UNTIL $\eta^k \geq$ UBDPB or $M^k$ is infeasible.
  ENDIF
  IF (UBDPB < UBD) THEN
    1. Solve the Primal Problem NLP($y^*$) and find the global minimum. Set
    $U^l = U^{l-1} \cup k^*$. If NLP($y^*$) is feasible, let the solution be $x_p^k$, and if $f(x_p^k)+c_1^T y^* <$ UBD, update $x_p^* = x_p^k$, $y_p^* = y^*$ and UBD = $f(x_p^*)+c_1^T y_p^*$.
  2. If $T^k \backslash U^l \neq \emptyset$, update UBDPB = min $(c_1^T y^m + L_1(x^m))$, $m \in T^k \backslash U^l$ (UBDPB corresponds to the Primal Bounding solution of $(x^s, y^s)$). Update $x^* = x^s$, $y^* = y^s$, $k^* = s$. Set $l = l + 1$. Otherwise, set UBDPB = $+\infty$.
 ENDIF
UNTIL UBDPB $\geq$ UBD and $\{M^k$ is infeasible or $\eta^k \geq$ UBD$\}$. The global solution of P is given by the current UBD, $x_p^*$, $y_p^*$.
```

3.2. Algorithm 2: Rigorous bound on the global solution of nonconvex MINLPs

This Algorithm produces rigorous upper and lower bounds on the global solution of nonconvex MINLPs. This Algorithm may also be viewed as a heuristic method for determination of a feasible upper bounding solution. This algorithm follows Algorithm 1. Replace step:

- Solve the Primal Problem NLP(y^*) and find the global minimum.

in Algorithm 1 with the following step:

- Solve the Primal Problem NLP(y^*) for any feasible solution, global solution not necessary.

A potential suboptimal solution of P is given by the current UBD, x_p^*, y_p^*. For this Algorithm, LBD = $\min(z_{PB}^f, ...z_{PB}^j, ..., z_{PB}^n)$ where f is the iteration at which the first feasible Primal Bounding solution z_{PB}^f is attained, z_{PB}^j is the solution of the Primal Bounding Problem at iteration j, and z_{PB}^n is the last feasible Primal Bounding solution before Algorithm 2 terminated. The distance between the global solution of P and the solution found (x_p^*, y_p^*) will be less than or equal to UBD - LBD.

Remark 8. If the current minimum of the Primal Bounding solutions corresponds to more than one integer realization, then any one of them is selected arbitrarily to update UBDPB.

Remark 9. Algorithms 1 and 2 may be reformulated by adding an integer cut excluding only those integer realizations for which the Primal Problem are solved and adding a constraint based on the UBDPB (similar to the constraint based on UBD) to exclude all the other previously visited integer realizations. In this case, a Primal Problem will be solved whenever the Relaxed Master becomes infeasible. This may however be inefficient since the current Relaxed Master will be resolved with the updated UBDPB.

Remark 10. Algorithms 1 and 2 can be refined to solve the Primal Problem whenever the current UBDPB is equal to the solution of the Relaxed Master Problem in the previous iteration. This will eliminate solving exactly one Relaxed Master Problem in the unusual case when the solution of the Relaxed Master, Primal and Primal Bounding Problems are all equal (objective value) to the global solution of P for a particular integer realization.

3.3. *Theoretical properties of the algorithms*

The following Corollary is based on Theorem 2 of Fletcher and Leyffer [12] and establishes the convergence properties of Algorithms 1 and 2.

Corollary 2. *If assumptions with respect to problems P and P1 hold, $|Y| < \infty$, and ϵ optimality can be achieved in a finite number of steps for the Primal Problems (NLP(y^j)), then Algorithm 1 terminates in a finite number of steps providing an optimal solution of P, or with an indication that P is infeasible, and Algorithm 2 will terminate in a finite number of steps providing a rigorous bracket containing the global solution of P, or with an indication that P is infeasible.*

Proof. First it is shown that no integer realization is visited twice. Let $l \leq k$. If $l \in S^k$, it follows from Lemma 1 of Fletcher and Leyffer [12] that the constraints added from the solution of the feasibility problem $F(y^l)$ eliminates y^l from further consideration. If $l \in T^k$, then the integer cuts added to the Relaxed Master Problem exclude y^l from further consideration. Since the set Y is finite, Algorithms 1 and 2 therefore terminate in a finite number of iterations.

Next it is shown that Algorithm 1 always terminates with the optimal solution of P if P is feasible. Let a global solution of P be given by (x_p^*, y_p^*) with optimal value $c_1^T y_p^* + f(x_p^*)$. Assume that Algorithm 1 terminates with the solution $(x^{'}, y^{'})$ and optimal value UBD $= c_1^T y^{'} + f(x^{'}) > c_1^T y_p^* + f(x_p^*)$. This would imply that the current Relaxed Master does not have a solution less than $c_1^T y^{'} + f(x^{'})$ and UBDPB $\geq c_1^T y^{'} + f(x^{'})$. Let the Primal Bounding solution corresponding to y_p^* be x_{pb} with the optimal value, $L_1(x_{pb}) + c_1^T y_p^*$. Then, $L_1(x_{pb}) + c_1^T y_p^* \leq c_1^T y_p^* + f(x_p^*) <$ UBDPB (on termination). Let the Relaxed Master yielding the integer realization y_p^* be M^* with the solution η^*. Since the solution of the Relaxed Master Problem represents a valid lower bound for the corresponding Primal Bounding Problem, $\eta^* \leq L_1(x_{pb}) + c_1^T y_p^* <$ UBD, which implies that y_p^* was the solution of the Relaxed Master at some iteration before termination (in which case an integer cut excludes y_p^* from the current Relaxed Master) or is feasible in the current Relaxed Master. In the former case, since UBDPB is updated as the minimum of the Primal Bounding solutions, UBDPB would have corresponded to $L_1(x_{pb}) + c_1^T y_p^*$ at some iteration before termination and the Primal Problem corresponding to y_p^* would have been solved. In the latter case the solution of the current Relaxed Master will correspond to y_p^*. The Primal Bounding solution $L_1(x_{pb}) + c_1^T y_p^* <$ UBD and therefore the Primal Problem corresponding to y_p^* will be solved yielding the global solution of P. Since the UBD is updated as the infimum of the Primal solutions, the algorithm terminating with UBD $> c_1^T y_p^* + f(x_p^*)$ is a contradiction.

Since the feasible region of P1 is obtained by overestimating the feasible region of P (i.e., P1 is a relaxation of P), if the Problem P is infeasible, P1 may or may not be feasible. If P1 is infeasible, then all the Primal Bounding Problems are infeasible, the algorithms never update UBDPB, do not solve a single Primal Problem, and terminate with an UBD = $+\infty$. On the other hand if P1 is feasible, the algorithms solve those Primal Problems for which the Primal Bounding Problems are feasible but never updates the UBD (since none of the Primal Problems are feasible) and exits with an UBD = $+\infty$. □

Remark 11. If only ϵ optimality can be achieved in a finite number of steps for the Primal Problem (NLP(y^j)), then Algorithm 1 will only yield ϵ optimality in a finite number of steps.

4. Results

4.1. Implementation

A variety of general purpose components have been developed to implement Algorithms 1 and 2. Results are presented using MINOS 5.5 [26] for solution of NLP problems with tolerances of 1e-4 to ensure convergence of the local noconvex problems and nonlinear convex problems. Primal Problems were solved to the global solution with a relative tolerance of 1e-3. The branch-and-reduce of [30] was used as described in [15]. The computational platform used was a 733 MHz Pentium III processor with 128 MB RAM running Linux 2.2.14.

4.1.1. Primal bounding problem The Primal Bounding Problem (as well as the global solution of the Primal Problem) rely upon creation of a convex relaxation of the nonconvex functions $f(x)$ and $g_1(x)$. A convexification method for factorable nonconvex functions is presented in [25] [33]. This method proceeds by recursively transforming the original nonconvex function into a set of nonconvex functions involving only simple nonlinear equality constraints for which the convex envelopes are known. The simple nonlinear equality expressions can then be relaxed, forming a convex problem. This procedure requires the addition of new variables and constraints into the problem formulation. The functionality of DAEPACK [36], originally developed for automatic differentiation and sparsity pattern generation, has now been extended [15] to generate automatically information required for convexification of nonconvex expressions. The resulting convex Primal Bounding Problem can readily be solved and linearized for a given binary realization. This convexification can also be used in the global solution of the Primal Problem using the branch-and-reduce method.

4.1.2. Primal problem For some fixed binary realizations of a given problem, the nonconvex Primal Problem must be solved to global optimality, searching over the possible values for the continuous variables. A branch-and-reduce optimization routine has been developed based on previous methods [30] [33] [2]. This method can use the current upper bound from Algorithm 1 to fathom nodes, resulting in reduced computational effort as Algorithm 1 proceeds. The branch-and-reduce method applies a variety of bounds tightening techniques in order to reduce the size of a given partition, resulting in tighter lower bounds for the considered region and improved convergence characteristics. Bounds tightening techniques based upon interval analysis of nonlinear constraints and linear constraints have been established. Additionally, Lagrangian based methods can be used for bounds tightening. For results presented here, up to 5 rounds of tightening are used for each convex lower bounding problem. No optimization based bounds tightening methods were used for the convex problem at the root node or during solution of subsequent convex problems in the branch-and-bound search.

Table 1. Results for simple MINLP example problems.

Problem	A	B	C	D
Continuous Variables	2	3	2	2
Binary Variables	3	8	6	3
Constraints	5	7	6	5
Average solution time (CPU s)	0.8	0.3	0.1	0.4
Primal Problems Started, N_P (avg/min/max)	7 / 7 / 7	5 / 5 / 5	1 / 1 / 1	3 / 3 / 3
Primal Nonconvex NLPs (avg/min/max)	61 / 61 / 61	5 / 5 / 5	1 / 1 / 1	3 / 3 / 3
Primal Convex NLPs (avg/min/max)	61 / 61 / 61	5 / 5 / 5	1 / 1 / 1	9 / 9 / 9
Primal Bounding NLPs, N_{PB} (avg/min/max)	7.13 / 7 / 8	6.92 / 6 / 8	2.04 / 1 / 3	3.5 / 3 / 4
MILP Iterations, N_R (avg/min/max)	7.13 / 7 / 8	6.92 / 6 / 8	2.04 / 1 / 3	3.5 / 3 / 4
Feasibility Problems Solved (avg/min/max)	0.13 / 0 / 1	0.72 / 0 / 1	0.64 / 0 / 2	0 / 0 / 0

Branching variables are selected from the set of possible variables using the ratio rule [4]. The set of possible branching variables can be user specified or derived from the variables participating in the nonconvex constraint that is most relaxed when evaluated at the convex solution. This implementation has been tested on a variety of examples taken from literature test problem compilations [14] [30].

4.1.3. Relaxed master problem The Relaxed Master Problem is derived from linearizations of solutions to the Primal Bounding Problem. DAEPACK is used to provide Jacobian values for nonlinear functions. The resulting MILP is solved using CPLEX 7.0 callable libraries.

4.1.4. Convex feasibility problem A convex feasibility problem must be solved for cases where a binary realization results in an infeasible Primal Bounding Problem. Positive slack variables augment the problem to relax all constraints. The sum of all slacks is then minimized and the resulting solution is used to construct a linearization of the Feasibility Problem.

4.2. Simple examples

Four MINLP problems are taken from Chapter 12 of [14]. Some of these problems needed slight transformations to establish constraint functions in the form of Problem 2. In some cases, a binary realization was infeasible for the convex Primal Bounding Problem, requiring solution of a feasibility problem. All elements of Y were enumerated as initial starting points y^1 for these problems, converging to the global solution in all cases. Results are shown in Table 1. The number of Primal Bounding Problems solved equals the number of MILP Relaxed Master Problems solved. The number of nonconvex Primal Problems started is reported. Statistics are shown for the number of NLPs solved in the Primal Problem branch-and-reduce procedure, including both nonconvex local solutions and convex solutions.

Problem A.

$$\min_{x,y} \; 2x_1 + 3x_2 + 1.5y_1 + 2y_2 - 0.5y_3$$
$$x_1^2 + y_1 = 1.25$$

$$\begin{aligned} x_2^{1.5} + 1.5y_2 &= 3 \\ x_1 + y_1 &\le 1.6 \\ 1.333x_2 + y_2 &\le 3 \\ -y_1 - y_2 + y_3 &\le 0 \\ x_1,\ x_2,\ x_3 &\ge 0 \\ y_1,\ y_2,\ y_3 &\in \{0,\ 1\} \end{aligned}$$

In order to construct a convex relaxation of the problem, all variables must be bounded. Results are presented with $x_1, x_2 \le 10$. The global minimum of 7.667 is attained at $x = (1.12, 1.31)^T$, $y = (0, 1, 1)^T$.

Problem B.

$$\min_{x,y} -x_1x_2x_3$$

$$\begin{array}{rcl} -\ln(1-x_1) + \ln(0.1)y_1 + \ln(0.2)y_2 + \ln(0.15)y_3 & = & 0 \\ -\ln(1-x_2) + \ln(0.05)y_4 + \ln(0.2)y_5 + \ln(0.15)y_6 & = & 0 \\ -\ln(1-x_3) + \ln(0.02)y_7 + \ln(0.06)y_8 & = & 0 \\ -y_1 - y_2 - y_3 & \le & -1 \\ -y_4 - y_5 - y_6 & \le & -1 \\ -y_7 - y_8 & \le & -1 \\ 3y_1 + y_2 + 2y_3 + 3y_4 + 2y_5 + y_6 + 3y_7 + 2y_8 & \le & 10 \\ 0 \le & x_1 & \le 0.9970 \\ 0 \le & x_2 & \le 0.9985 \\ 0 \le & x_3 & \le 0.9988 \\ & y \in \{0, 1\}^8 & \end{array}$$

The minimum objective function value is -0.94347 at $y = (0, 1, 1, 1, 0, 1, 1, 0)$ $x = (0.970, 0.993, 0.980)$. In this problem, the Primal Problem obtained by fixing the binary variables can be solved by a simple function evaluation. As a result, the global branch-and-reduce method readily finds the solution at the root node for each Primal Problem. A tailored algorithm could solve the Primal Problem using a function evaluation. This insight, which completely eliminates branching on continuous variables, would be lost using branch-and-bound procedures branching on both continuous and binary variables.

Problem C.

$$\min_{x,y} 7x_1 + 10x_2$$

$$\begin{aligned} x_1^{1.2}x_2^{1.7} - 7x_1 - 9x_2 &\le -24 \\ -x_1 - 2x_2 &\le 5 \\ -3x_1 + x_2 &\le 1 \\ 4x_1 - 3x_2 &\le 11 \\ -x_1 + y_1 + 2y_2 + 4y_3 &= 0 \end{aligned}$$

$$\begin{aligned} -x_2 + y_4 + 2y_5 + y_6 &= 0 \\ 1 \leq x_1, x_2 &\leq 5 \\ y &\in \{0, 1\}^6 \end{aligned}$$

The minimum objective function value is 31 at $y = (1, 1, 0, 1, 0, 0)$ $x = (3, 1)$. In this nonconvex integer problem, the Primal Problem can also be solved trivially by a function evaluation.

Problem D.

$$\min_{x,y} -5x_1 + 3x_2$$

$$\begin{aligned} 2x_2^2 - 2x_2^{0.5} - 2x_1^{0.5}x_2^2 + 11x_2 + 8x_1 &\leq 39 \\ x_1 - x_2 &\leq 3 \\ 3x_1 + 2x_2 &\leq 24 \\ -x_2 + y_1 + 2y_2 + 4y_3 &= 0 \\ 1 \leq x_1 &\leq 10 \\ 1 \leq x_2 &\leq 6 \\ y &\in \{0, 1\}^3 \end{aligned}$$

The minimum objective function value is -17 at $y = (1, 0, 0)$ $x = (4, 1)$.

4.3. Large scale problem

The heat exchanger network synthesis (HENS) optimization problem as formulated by Yee and Grossmann [39] [40] is considered here as an example to analyze Algorithm 1. This problem represents the design of a heat exchanger network. The optimization problem conforms to the formulation necessary for Algorithm 1, with nonconvexity only in the objective function. There are 44 linear inequality constraints and 20 linear equality constraints. The problem involves 12 binary variables corresponding to the presence / absence of heat exchangers. The problem includes 40 continuous variables arising from heat duty for heat exchangers, temperatures for process streams, and temperature differences.

The global optimum of this problem has been previously reported [5] [2] using a branch-and-bound based algorithm. This solution coincides with the solution found using Algorithm 1. The results obtained by employing the OA/ER algorithm [20] for this problem have also been reported previously [5]. The solution attained varied with the starting point employed (initial guess). Since the problem is a nonconvex MINLP, the OA/ER algorithm may cut off portions of the feasible space, resulting in convergence to suboptimal solutions as expected.

In [2], a modified formulation of the original problem [39] [40] was used. This modification, derived from physical insight specific to the problem, tightens some variable bounds based on current bounds of other variables. For example, the maximum heat duty of a single exchanger may depend upon bounds of the relevant temperature streams. Similarly, a given heat exchanger may not be used for a given binary realization, resulting in a 0 heat duty. This type of problem specific bounds tightening can significantly impact algorithm performance. For these results, only bounds tightening methods were used in the Primal Problem solution that could be derived from the original mathematical formulation of [39] [40]. For a given binary realization, some heat exchangers may not be used, resulting in some heat duty variables being forced to 0 by linear constraints. Additionally, variables representing temperature differences can be unconstrained (but bounded) for a binary realization, having no affect on the problem solution. These variables should not be branched upon, and can be constrained to any single value within their original bounds without affecting Primal Problem solution. This bounds tightening application is equivalent to reformulating the Primal Problem at each iteration, removing variables that cannot affect the solution of the Primal Problem.

Selection of branching variable also impacts performance of the solution of the Primal Problem. Typically, the nonconvex constraint having a loose underestimate is identified, and a branching variable is chosen from the set of variables involved in the constraint [30] [33] [2]. A heuristic was presented in [2] for branching variable selection using a physical insight specific to the problem: rather than branch on variables involved in the nonconvex expressions (the duty for heat exchangers and differences in temperature streams) only variables representing temperatures for process streams should be used. Results using this branching rule are presented in Table 2 as constraint case A. Without using this branching rule, our current branch-and-reduce implementation will eventually solve the Primal Problem for a binary iteration, but may require extended amounts of time and many thousands of convex lower bounding problem solutions.

An initial binary realization must be found. Ideally, the binary realization for the global solution should be used in order to help reduce solution time in Algorithm 1. For the presented HENS problem results, 25 randomly chosen starting points were selected as the initial binary realization. Alternatively, Algorithm 2 could be used to generate a potentially suboptimal initial binary realization.

Table 2. HENS Results using outer approximation methods. Algorithm 1 provides a deterministic global solution, while Algorithm 2 only provides a potentially sub-optimal solution and bounds on the global solution. Constraint case A uses the original problem formulation, while constraint case B adds additional valid constraints. 25 random starting points were selected for initialization of the algorithm.

Algorithm	2	1	2	1
Constraint Case	A	A	B	B
Potential Branching Variables	0	8	0	40
Average Solution Time (CPU s)	32	228	39	96
Primal Problems Started, N_P (avg/min/max)	60.08/60/61	60.08/60/61	58.08/58/59	58.08/58/59
Primal Nonconvex NLPs (avg/min/max)	60.08/60/61	1042.08/1048/1049	58.08/58/59	638.08/638/639
Primal Convex NLPs (avg/min/max)	0	2258.12/2258/2259	0	396.88/396/397
MILP Iterations, N_R (avg/min/max)	60.96/60/61	60.96/60/61	60.96/60/61	60.96/60/61
Feasibility Problems (avg/min/max)	0.88/0/1	0.88/0/1	0.88/0/1	0.88/0/1

The original formulation of [39] [40] uses propositional logic constraints for enforcing equality relationships dependent on the presence of a heat exchanger in the form: $z_{ij} = 1 \Rightarrow \Delta T_{ij} = T_i - T_j$. This can be expressed as two inequality constraints (with M large):

$$\Delta T_{ij} - T_i + T_j \leq M(1 - z_{ij})$$
$$-\Delta T_{ij} + T_i - T_j \leq M(1 - z_{ij})$$

The original formulation uses only one of the inequality constraints. This is justified in that ΔT_{ij} variables will always be maximized for this problem [40]. As a result, changes in the bounds on ΔT_{ij} will not affect both the upper and lower bounds of T_i and T_j using interval-based linear bounds tightening techniques. Inclusion of both valid inequality constraints allows full propagation of changes in bounds using linear tightening methods. All variables can then be used for branching variable selection rather than a heuristic subset. Results using the additional inequality constraints while potentially branching on *all* variables are presented in Table 2 as constraint case B. Note that this formulation reduces the total solution time.

The solution statistics for solving this problem using SMIN-αBB algorithm [2] on a HP-C160 have been reported. The CPU times and the number of iterations varied with the heuristic employed. The best case reported solution time was 315 CPU s, while most heuristics converge in 500 - 700 CPU s. Heuristics considered different branching rules and methods of optimization based bounds tightening application. The CPU times previously reported were an order of magnitude larger than that now reported from Algorithm 2. However, since Algorithm 2 solves the Primal Problem locally, the results cannot be directly compared. Use of Algorithm 1 takes an average of 228 CPU s using the same problem constraints and branching variable selection heuristics, without the addition of problem specific bounds-tightening procedures, optimization based bounds tightening techniques, or feasibility based bounds tightening. Use of additional valid constraints and branching on all variables results in a total solution time of 96 CPU s. The HP C160 has a SPECfp95 of 16.3, while the Pentium III at 733 MHz has an approximate SPECfp95 result of 20.1 (published result of 15.1 for 550 MHz), approximately 25 % faster than the HP for floating point operations. As compared to published branch-and-bound results for this example problem, Algorithm 1 provides a global solution using comparable CPU time.

5. Conclusions

A decomposition strategy based on outer approximation to solve a class of nonconvex Mixed-Integer Nonlinear Programming (MINLP) problems has been developed in the current paper and two algorithms are presented. The convergence and optimality properties of the algorithms were discussed. The computational performance of the proposed strategy is illustrated by example problems. Numerical results indicate that the proposed decomposition strategy is more efficient as regards to the computational time required when compared to the currently available Branch and Bound algorithms for solving nonconvex MINLPs. Furthermore, the most computationally demanding step in the presented MINLP solution algorithm is typically the global solution of the Primal Problem, a nonconvex NLP. However, in some cases the resulting Primal Problem for a fixed binary realization requires only a function evaluation or solution of a convex problem, as demonstrated in two of the presented example problems. Additionally, future developments in branch-and-reduce or interval methods for nonconvex global optimization will benefit the presented MINLP method.

Acknowledgements. This work was supported by the National Science Foundation under grants CTS-9703623 and CTS-0238663 and the National Computational Science Alliance. The authors would also like to thank the anonymous reviewers for their contributions to this work.

References

1. Adjiman, C.S., Androulakis, I.P., Floudas, C.A.: Global Optimization of MINLP Problems in Process Synthesis and Design." Comput. Chem. Eng. **21**, S445–S450 (1997)
2. Adjiman, C.S., Androulakis, I.P., Floudas, C.A.: Global Optimization Method of Mixed-Integer Nonlinear Problems." AIChE Journal **46** (9), 1769–1797 (2000)
3. Adjiman, C.S., Dallwig, S., Floudas, C.A., Neumaier, A.: A Global Optimization Method, α-BB, for General Twice-Differentiable Constrained NLPs-I. Theoretical Advances." Comput. Chem. Eng. **22**, 1137–1158 (1998)
4. Adjiman, C.S., Dallwig, S., Floudas, C.A., Neumaier, A.: A Global Optimization Method, α-BB, for General Twice-Differentiable Constrained NLPs-I. Implementation and Computational Results." Comput. Chem. Eng. **22** (9), 1159–1179 (1998)
5. Adjiman, C.S., Schweiger, C.A., Floudas, C.A.: "Mixed-Integer Nonlinear Optimization in Process Synthesis." In: D.Z. Du and P.M. Pardalos, (eds)., Handbook of Combinatorial Optimization (Kluwer Academic Publishers Inc., Dordrecht, 1998)
6. Al-Khayyal, F.A., Falk, J.E.: Jointly Constrained Biconvex Programming." Mathematics Operations Research **8**, 273–286 (1983)
7. Allgor, R.J., Barton, P.I.: Mixed-Integer Dynamic Optimization I: Problem Formulation." Comput. Chem. Eng. **23** (4), 567–584 (1999)
8. Balas, E., Jeroslow, R.: Canonical Cuts on the Unit Hypercube." SIAM J. Appl. Math. **23**, 61–79 (1972)
9. Bertsekas, D.L., Lower, G.S., Sandell, N.R., Posbergh, T.A.: Optimal Short Term Scheduling of Large-Scale Power Systems." IEEE Trans. Automatic Control **AC-28**, 1–11 (1983)
10. Duran, M.A., Grossmann, I.E.: An Outer-Approximation Algorithm for a Class of Mixed-Integer Nonlinear Programs." Math. Programming **36**, 307–339 (1986)
11. Falk, J.E., Soland, R.M.: An Algorithm for Separable Nonconvex Programming Problems. " Manage. Sci. **15** (9), 550–569 (1969)
12. Fletcher, R., Leyffer, S.: Solving Mixed Integer Nonlinear Programs by Outer Approximation. " Math. Programming **66**, 327–349 (1994)
13. Floudas, C.A.: Nonlinear and Mixed Integer Optimization (Oxford University Press, New York 1995)
14. Floudas, C.A. et. al: Handbook of Test Problems in Local and Global Optimization (Kluwer Academic Publishers, Boston, 1999) http://titan.princeton.edu/TestProblems/chapter12.html
15. Gatzke, E.P., Tolsma, J.E., Barton, P.I.: Construction of Convex Function Relaxations Using Automated Code Generation Techniques." Optimization and Engineering **3** (3), 305–326 (2002)
16. Geoffrion, A.M.: Generalized Benders Decomposition." Journal of Optimization Theory Appl. **10**, 237–262 (1972)
17. Glover, F.: Improved Integer Programming Formulations of Nonlinear Integer Problems. " Manage. Sci. **22** (4), 455–460 (1975)
18. Hoang, H.H.: Topological Optimization of Networks: A Nonlinear Mixed-Integer Model Employing Generalized Benders Decomposition." IEEE Trans. Automatic Control **AC-27**, 164–169 (1982)
19. Kleibohm, K.: Remarks on the Nonconvex Programming Problem." Unternehmensf **11**, 49–60 (1967)
20. Kocis, G.R., Grossmann, I.E.: Relaxation Strategy for the Structural Optimization of Process Flow Sheets." Ind. Eng. Chem. Res. **26** (9), 1869–1880 (1987)
21. Kocis, G.R., Grossmann, I.E.: Global Optimization of Nonconvex MINLP Problems in Process Synthesis." Ind. Eng. Chem. Res. **27** (8), 1407–1421 (1988)
22. Laursen, P.S.: Simple Approaches to Parallel Branch and Bound." Parallel Comput. **19**, 143–152 (1993)
23. Maranas, C.D., Floudas, C.A.: Global Minimum Potential Energy Conformations of Small Molecules." J. Global Optimization **4**, 135–170 (1994)

24. Maranas, C.D., Floudas, C.A.: Finding All Solutions of Nonlinearly Constrained Systems of Equations." J. Global Optimization **7** (2), 143–182 (1995)
25. McCormick, G.P.: Computability of Global Solutions to Factorable Nonconvex Programs: Part I - Convex Underestimating Problems." Math. Programming **10**, 147–175 (1976)
26. Murtagh, B.A., Saunders, M.A.: "MINOS 5.1 Users Guide," Technical Report SOL 83-20R, Systems Optimization Laboratory, Department of Operations Research, Stanford University, 1987
27. Odele, O., Macchietto, S.: Computer Aided Molecular Design: A Novel Method for Optimal Solvent Selection." Fluid Phase Equilibria **82**, 47–54 (1993)
28. Queseda, I., Grossmann, I.E.: Global Optimization Algorithm for Heat Exchanger Networks." Ind. Eng. Chem. Res. **32**, 487–499 (1993)
29. Reklaitis, G.V.: Scheduling of Multipurpose Batch Chemical Plants with Resource Constraints." Ind. Eng. Chem. Res. **32**, 3037–3050 (1993)
30. Ryoo, H.S., Sahinidis, N.V.: Global Optimization of Nonconvex NLPs and MINLPs with Applications in Process Design." Comput. Chem. Eng. **19** (5), 551–566 (1995)
31. Sahinidis, N.V., Grossmann, I.E.: Convergence Properties of Generalized Benders Decomposition." Comput. Chem. Eng. **15** (7), 481–491 (1991)
32. Smith, E.M.B., Pantelides, C.C.: "Global Optimization of General Process Models." In: I.E. Grossmann, (ed.), Global Optimization in Engineering Design (Kluwer Academic Publishers Inc., Dordrecht, 1996, pp. 355-386
33. Smith, E.M.B.: "On the Optimal Design of Continuous Processes." Ph.D. Thesis, University of London (October 1996)
34. Smith, E.M.B., Pantelides, C.C.: Global Optimization of Nonconvex MINLPs." Comput. Chem. Eng. **21** (796), S791–S (1997)
35. Soland, R.M.: An Algorithm for Separable Nonconvex Programming Problems II: Nonconvex Constraints." Manage. Sci. **17** (11), 759–773 (1971)
36. Tolsma, J.E., Barton, P.I.: DAEPACK: An Open Modeling Environment for Legacy Code." Ind. Eng. and Chem. Res. **6**, 1826–1839 (2000)
37. Viswanathan, J., Grossmann, I.E.: A Combined Penalty Function and Outer-Approximation Method for MINLP Optimization." Comput. Chem. Eng. **14** (7), 769–782 (1990)
38. Westerlund, T., Pettersson, F.: An Extended Cutting Plane Method for Solving Convex MINLP Problems." Comput. Chem. Eng. **19** (S), 131–136 (1995)
39. Yee, T.F., Grossmann, I.E.: "Simultaneous Optimization Model for Heat Exchanger Network Synthesis." In: I.E. Grossmann, (ed.), Chemical Engineering Optimization Models with GAMS, CACHE Design Case Studies Series 6 1991. www.gamsworld.org/minlp/minlplib/synheat.htm
40. Yee, T.F., Grossmann, I.E.: Simultaneous Optimization Models for Heat Integration -II. Heat Exchanger Network Synthesis." Comput. Chem. Eng. **14** (10), 1165–1184 (1990)
41. Zamora, J.M., Grossmann, I.E.: A Global Optimization Algorithm for the Synthesis of Heat Exchanger Networks with No Stream Splits." Comput. Chem. Eng. **22** (3), 367–384 (1998)