, , 1–27 (1995)

αBB: A Global Optimization Method for General Constrained Nonconvex Problems

I.P. ANDROULAKIS, C.D. MARANAS AND C.A. FLOUDAS*
Department of Chemical Engineering, Princeton University, Princeton, NJ 08544–5263

Editor:

Abstract. A branch and bound global optimization method, αBB, for general continuous optimization problems involving nonconvexities in the objective function and/or constraints is presented. The nonconvexities are categorized as being either of special structure or generic. A convex relaxation of the original nonconvex problem is obtained by (i) replacing all nonconvex terms of special structure (i.e. bilinear, fractional, signomial) with customized tight convex lower bounding functions and (ii) by utilizing the α parameter as defined in [17] to underestimate nonconvex terms of generic structure. The proposed branch and bound type algorithm attains finite ϵ–convergence to the global minimum through the successive subdivision of the original region and the subsequent solution of a series of nonlinear convex minimization problems. The global optimization method, αBB, is implemented in C and tested on a variety of example problems.

Keywords: Global optimization, constrained optimization, convex relaxation

1. Introduction

A significant effort has been spent in the last five decades studying theoretical and algorithmic aspects of local optimization algorithms and their applications in engineering and science. Comparatively, there has been traditionally much less attention devoted to global optimization methods. However, in the last decade the area of global optimization has attracted a lot of interest from the operations research, engineering and applied mathematics communities. This recent surge of interest can be attributed to the realization that there exists an abundance of optimization problems for which existing local optimization approaches cannot consistently locate the global minimum solution. Furthermore, the steady improvement in the performance of computers constantly extends the scope of problems which are tractable with global optimization approaches.

Existing global optimization algorithms, based on their convergence properties, can be be partitioned into deterministic and stochastic. The deterministic approaches include Lipschitzian methods [12], [13]; branch and bound procedures [2], [15], [1]; cutting plane methods [32]; difference of convex functions and reverse convex methods [31]; outer approximation approaches [14]; primal–dual methods [29], [6], [7], [33], [4]; reformulation–linearization [27], [28]; and interval methods [8]. Stochastic approaches, encompass among others simulated annealing [26], genetic algorithms [10], [3], and clustering methods [25]. A number of books [23], [30], [24], [16], [5], [11] summarize the latest developments in the area.

* Author to whom all correspondence should be addressed.

Deterministic approaches typically provide mathematical guarantees for convergence to an ϵ–global minimum in finite number of steps for optimization problems involving certain mathematical structure. On the other hand, stochastic methods offer asymptotic convergence guarantees only at infinity for a very wide class of optimization problems. It is the objective of this work to extend deterministic guarantees for convergence to a very general class of continuous optimization problems and implement this procedure in the αBB global optimization package. In the next section, a description of the global optimization problem addressed in this paper is presented.

2. Problem Definition

The optimization problem addressed in this paper can be formulated as the following constrained nonlinear optimization problem involving only continuous variables.

$$\min_{\mathbf{x}} \quad f(\mathbf{x}) \qquad \textbf{(P0)}$$

$$\begin{aligned} \text{subject to} \quad h_j(\mathbf{x}) &= 0, \quad j = 1, \ldots, M \\ g_k(\mathbf{x}) &\leq 0, \quad k = 1, \ldots, K \\ A\mathbf{x} &\leq \mathbf{c} \\ \mathbf{x}^L &\leq \mathbf{x} \leq \mathbf{x}^U \end{aligned}$$

Here $\mathbf{x}$ denotes the vector of variables, $f(\mathbf{x})$ is the nonlinear objective function, $h_j(\mathbf{x})$ is the set of nonlinear equality constraints and $g_k(\mathbf{x}),\ k = 1, \ldots, K$ is the set of nonlinear inequality constraints. Formulation **(P0)** in general corresponds to a nonconvex optimization problem possibly involving multiple local and disconnected feasible regions. It has been observed in practice that existing path–following techniques cannot consistently locate the global minimum solution of **(P0)** even if a multi-start procedure are utilized. For special cases of **(P0)** involving bilinear or polynomial terms [6], [7], signomial terms [18], efficient algorithms have been proposed for locating the global minimum solution. For the general case, however, of minimizing a nonconvex function subject to a set of nonconvex equality and inequality constraints there has been comparatively little work in deriving global optimization methods and tools.

Our approach is based on the convex relaxation of the original nonconvex formulation **(P0)**. This requires the convex lower bounding of all nonconvex expressions appearing in **(P0)**. These terms can be partitioned into three classes:

(i) convex,

(ii) nonconvex of special structure,

(iii) nonconvex of generic structure.

Clearly, no convex lower bounding action is required for convex functions. For nonconvex terms of special structure (e.g. bilinear, univariate concave functions), tight specialized convex lower bounding schemes already exist and therefore can be utilized. Based on this partitioning of different terms appearing in the objective function and constraints, formulation **(P0)** is rewritten equivalently as follows:

$$\min_{\mathbf{x}} \quad C^0(\mathbf{x}) + \sum_{k \in \mathcal{K}^0} NC_k^0(\mathbf{x}) + \sum_{i=1}^{N-1} \sum_{i'=i+1}^{N} b_{i,i'}^0 x_i x_{i'} \qquad \textbf{(P)}$$

$$\text{subject to} \quad C^j(\mathbf{x}) + \sum_{k \in \mathcal{K}^j} NC_k^j(\mathbf{x}) + \sum_{i=1}^{N-1} \sum_{i'=i+1}^{N} b_{i,i'}^j x_i x_{i'}, \quad \leq \ 0$$

$$j = 1, \ldots, (2M + K)$$

$$A\mathbf{x} \ = \ \mathbf{c}, \quad \mathbf{x}^L \ \leq \ \mathbf{x} \ \leq \ \mathbf{x}^U$$

$$\text{where} \quad NC_k^j(\mathbf{x}) \text{ with } \mathbf{x} \ \in \ \left\{ x_i \ : \ i \in \mathcal{N}_k^j \right\}, \ \ j = 0, \ldots, (2M + K)$$

Note that all nonlinear equality constraints $h_j(\mathbf{x}) = 0$ appearing in **(P0)** have been replaced by two inequalities in **(P)**. $C^0(\mathbf{x})$ is the convex part of the objective function; $NC_k^0(\mathbf{x})$ is the set of $\mathcal{K}^0$ generic nonconvex terms appearing in the objective function; $\mathcal{N}_k^0$ is the subset of variable $\mathbf{x}$ participating in each generic nonconvex term k in the objective; and $b_{i,i'}^0 x_i x_{i'}$ the bilinear terms. Similarly, for each constraint j, there exists a convex part $C^j(\mathbf{x})$, $\mathcal{K}^j$ generic nonconvex terms $NC_k^j(\mathbf{x})$, with $\mathcal{N}_k^j$ variables $\mathbf{x}$ per term, and the bilinear terms $b_{i,i'}^j x_i x_{i'}$. Additionally, linear equality constraints and variable bounding constraints appear explicitly in the model **(P)**. Clearly, for each optimization problem that falls within formulation **(P0)** there exist several ways of reformulating it into **(P)**. In the current implementation of αBB the only nonconvex terms recognized as having special structure are the bilinear terms. Work is currently under way to include in the set of nonconvex terms of special structure additional nonconvex functions such as univariate concave, signomial functions, and products of univariate functions [19]. In the next section the derivation of a convex relaxation **(R)** of **(P)** is discussed.

3. Convex Relaxation

A convex relaxation of **(P)** can be constructed by replacing each generic nonconvex term, $NC_k^j(\mathbf{x})$, and each bilinear term, $b_{i,i'}^j x_i x_{i'}$, $j = 0, \ldots, (2M + K)$, with one or more convex lower bounding functions.

3.1. Nonconvex Terms of Special Structure

As it is shown in [2], the tightest possible convex lower bounding of a bilinear term $b_{i,i'}x_ix_{i'}$ inside some rectangular domain $\left[x_i^L, x_i^U\right] \times \left[x_{i'}^L, x_{i'}^U\right]$ (convex envelope) corresponds to the maximum of the following two linear cuts.

$$\begin{aligned} b_{i,i'}x_ix_{i'} \;\geq\; s_{i,i'}(x_i, x_{i'}) \;&=\; \max\left(Y_i^L x_{i'} + Y_{i'}^L x_i - Y_i^L Y_{i'}^L,\right. \\ &\qquad \left. Y_i^U x_{i'} + Y_{i'}^U x_i - Y_i^U Y_{i'}^U\right) \end{aligned}$$

$$\text{where} \qquad \begin{aligned} Y_i^L &= \min\left(b_{i,i'}x_i^L, b_{i,i'}x_i^U\right), \\ Y_{i'}^L &= \min\left(b_{i,i'}x_{i'}^L, b_{i,i'}x_{i'}^U\right), \\ Y_i^U &= \max\left(b_{i,i'}x_i^L, b_{i,i'}x_i^U\right), \\ Y_{i'}^U &= \max\left(b_{i,i'}x_{i'}^L, b_{i,i'}x_{i'}^U\right) \end{aligned}$$

$s_{i,i'}(x_i, x_{i'})$ is the convex envelope of $b_{i,i'}x_ix_{i'}$ inside the rectangular domain $\left[x_i^L, x_i^U\right] \times \left[x_{i'}^L, x_{i'}^U\right]$ and therefore, it can become arbitrarily close to $b_{i,i'}x_ix_{i'}$ for a small enough rectangular domain.

It can be shown that the maximum separation between $b_{i,i'}x_ix_{i'}$ and $s_{i,i'}$ inside the domain $\left[x_i^L, x_i^U\right] \times \left[x_{i'}^L, x_{i'}^U\right]$ can be at most one fourth of the area of the rectangular domain multiplied by the absolute value of $b_{i,i'}$:

$$|b_{i,i'}| \,\frac{\left(x_i^U - x_i^L\right)\left(x_{i'}^U - x_{i'}^L\right)}{4}.$$

This maximum separation occurs at the middle point

$$x_i^m \;=\; \frac{x_i^L + x_i^U}{2}, \quad x_{i'}^m \;=\; \frac{x_{i'}^L + x_{i'}^U}{2}.$$

Lemma 1 *The maximum separation of the bilinear term xy from its convex envelope, $\max\left(x^Ly + xy^L - x^Ly^L,\ x^Uy + xy^U - x^Uy^U\right)$, inside the rectangle $\left[x^L, x^U\right] \times \left[y^L, y^U\right]$ occurs at the middle point*

$$x^m \;=\; \frac{x^L + x^U}{2}, \quad y^m \;=\; \frac{y^L + y^U}{2}.$$

and is equal to one fourth of the area of the rectangular domain,

$$\frac{\left(x^U - x^L\right)\left(y^U - y^L\right)}{4}.$$

Proof: This can be formulated as the following optimization problem.

$$\max_{x,y} \quad xy - \max\left(x^Ly + xy^L - x^Ly^L,\ x^Uy + xy^U - x^Uy^U\right)$$

$$\text{subject to} \quad \begin{aligned} x^L &\le x \le x^U \\ y^L &\le y \le y^U \end{aligned}$$

By substituting in the objective function $x = x^L$ or $x = x^U$ or $y = y^L$ or $y = y^U$ the maximum separation becomes zero. This implies that for any point in the perimeter of the rectangle $\left[x^L, x^U\right] \times \left[y^L, y^U\right]$ xy matches its convex envelope and thus the point where the maximum separation occurs must be an interior point. After replacing the $\max_{x,y}(\cdot)$ operator with the equivalent $-\min_{x,y} -(\cdot)$ operator and eliminating the $\max$ over the two linear cuts at the expense of two new inequality constraints we have:

$$-\min_{x,y} \quad -xy + z$$

$$\text{subject to} \quad \begin{aligned} z &\ge x^L y + x y^L - x^L y^L \\ z &\ge x^U y + x y^U - x^U y^U \\ \\ x^L &\le x \le x^U \\ y^L &\le y \le y^U \end{aligned}$$

Let $\mu_1, \mu_2 \ge 0$ be the multipliers associated with the two new inequality constraints. Clearly, the multipliers associated with the variable bound constraints are zero since the solution will be an interior point. The KKT conditions yield the following stationarity conditions:

$$\begin{aligned} \mu_1 + \mu_2 - 1 &= 0 \\ \mu_1 x^L + \mu_2 x^U - x &= 0 \\ \mu_1 y^L + \mu_2 y^U - y &= 0 \\ \left(-z + x^L y + x y^L - x^L y^L\right) \mu_1 &= 0 \\ \left(-z + x^U y + x y^U - x^U y^U\right) \mu_2 &= 0 \\ \mu_1, \mu_2 &\ge 0 \end{aligned}$$

Clearly, at least one of μ_1, μ_2 must be nonzero, leading to the following three cases:

(i) $\mu_1 = 0,\ \mu_2 = 1$

(ii) $\mu_1 = 1,\ \mu_2 = 0$

(iii) $\mu_1 > 0,\ \mu_2 > 0$

If $\mu_1 = 1$ or $\mu_2 = 1$ then we have $-z + x^L y + x y^L - x^L y^L = 0$ or $-z + x^U y + x y^U - x^U y^U = 0$ respectively. Both cases (i) and (ii) lead to a zero maximum separation implying that they correspond to local minima. The single remaining case (iii) yields the following linear system of equations in μ_1, μ_2, x, y, z

$$\mu_1 + \mu_2 - 1 = 0$$

$$\begin{aligned}
\mu_1 x^L + \mu_2 x^U - x &= 0 \\
\mu_1 y^L + \mu_2 y^U - y &= 0 \\
-z + x^L y + x y^L - x^L y^L &= 0 \\
-z + x^U y + x y^U - x^U y^U &= 0
\end{aligned}$$

Solution of this system gives:

$$x = \frac{x^L + x^U}{2}, \quad y = \frac{y^L + y^U}{2}, \quad z = \frac{x^U y^L + x^L y^U}{2}, \quad \mu_1 = \mu_2 = \frac{1}{2}$$

The maximum separation therefore is

$$xy - z = \frac{\left(x^U - x^L\right)\left(y^U - y^L\right)}{4}.$$

■

3.2. Nonconvex Terms of Generic Structure

The convex lower bounding of the generic nonconvex terms NC_k^j is motivated by the approach introduced in [17]. For each one of the generic nonconvex functions,

$$\begin{aligned}
& NC_k^j(\mathbf{x}), \ \ j = 0, \ldots, (2M + K), \ \ k \in \mathcal{K}^j \\
\text{where} \quad & NC_k^j(\mathbf{x}) \text{ with } \mathbf{x} \in \left\{ x_i \ : \ i \in \mathcal{N}_k^j \right\}, \ \ j = 0, \ldots, (2M + K)
\end{aligned}$$

a convex lower bounding function $NC_k^{j,conv}$ can be defined by augmenting the original nonconvex expression with the addition of a separable convex quadratic function of $(x_i, \ i \in \mathcal{N}_k^j)$.

$$\begin{aligned}
NC_k^{j,conv}(\mathbf{x}) &= NC_k^j(\mathbf{x}) \\
&+ \sum_{i \in \mathcal{N}_k^j} \alpha_{i,k}^j(\mathbf{x}^L, \mathbf{x}^U)\left(x_i^L - x_i\right)\left(x_i^U - x_i\right), \ \ j = 0, \ldots, (2M + K), \ \ k \in \mathcal{K}^j
\end{aligned}$$

$$\text{where} \quad \alpha_{i,k}^j(\mathbf{x}^L, \mathbf{x}^U) \geq \max\left\{0, -\frac{1}{2} \min_{\mathbf{x}^L \leq \mathbf{x} \leq \mathbf{x}^U} \lambda(\mathbf{x})\right\}$$

Note that $\alpha_{i,k}^j$ are nonnegative parameters which must be greater or equal to the negative one half of the minimum eigenvalue of the Hessian matrix of $NC_k^{j,conv}$ over $x_i^L \leq x_i \leq x_i^U, \ i \in \mathcal{N}_k^j$. These parameters $\alpha_{i,k}^j$ can be estimated either through the solution of an optimization problem or by using the concept of the measure of a matrix [17]. The effect of adding the extra separable quadratic term on the generic nonconvex terms is to construct new convex functions by overpowering the nonconvexity characteristics of the original nonconvex terms with the addition of the terms $2\alpha_{i,k}^j$ to all of their eigenvalues. These new functions $NC_k^{j,conv}$ defined over the rectangular domains $x_i^L \leq x_i \leq x_i^U,\ i \in \mathcal{N}_k^j$ involve a number of important properties. These properties are as follows:

Property 1:. $NC_k^{j,conv}$ is a valid *underestimator* of NC_k^j.

$$\forall\, x_i \in \left[x_i^L, x_i^U\right],\ i \in \mathcal{N}_k^j \text{ we have } NC_k^{j,conv}(\mathbf{x}) \leq NC_k^j(\mathbf{x}).$$

Proof: For every $i = 1, \ldots, N$ we have $\left(x_i^L - x_i\right)\left(x_i^U - x_i\right) \leq 0$ and also by definition $\alpha_{i,k}^j(\mathbf{x}^L, \mathbf{x}^U) \geq 0$. Therefore, $\forall\, \mathbf{x} \in \left[\mathbf{x}^L, \mathbf{x}^U\right],\ NC_k^{j,conv}(\mathbf{x}) \leq NC_k^j(\mathbf{x})$. ■

Property 2:. $NC_k^{j,conv}(\mathbf{x})$ matches NC_k^j at all corner points.

Proof: Let $\mathbf{x}^c$ be a corner point of $\left[\mathbf{x}^L, \mathbf{x}^U\right]$ then for every $i = 1, \ldots, N$ $(x_i^L - x_i^c) = 0$ or $(x_i^U - x_i^c) = 0$. Therefore, $NC_k^{j,conv}(\mathbf{x^c}) = NC_k^j(\mathbf{x}^c)$ in either case. ■

Property 3:. $NC_k^{j,conv}(\mathbf{x})$ is *convex* in $x_i \in \left[x_i^L, x_i^U\right],\ i \in \mathcal{N}_k^j$.

Proof: It is a direct consequence of the definition of the parameters $\alpha_{i,k}^j(\mathbf{x}^L, \mathbf{x}^U)$, (See [17]). ■

Property 4:. The maximum separation between the nonconvex term of generic structure $NC_k^{j,conv}$ and its convex relaxation NC_k^j is *bounded* and proportional to the positive parameters $\alpha_{i,k}^j$ and to the square of the diagonal of the current box constraints.

$$\begin{aligned}
&\max_{\mathbf{x}^L \leq \mathbf{x} \leq \mathbf{x}^U} \left(NC_k^j(\mathbf{x}) - NC_k^{j,conv}(\mathbf{x})\right) \\
&\qquad = \frac{1}{4} \sum_{i \in \mathcal{N}_k^j} \alpha_{i,k}^j(\mathbf{x}^L, \mathbf{x}^U)\left(x_i^U - x_i^L\right)^2
\end{aligned}$$

Proof:

$$\begin{aligned}
&\max_{\mathbf{x}^L \leq \mathbf{x} \leq \mathbf{x}^U} \left(NC_k^j(\mathbf{x}) - NC_k^{j,conv}(\mathbf{x})\right) \\
&= \max_{\mathbf{x}^L \leq \mathbf{x} \leq \mathbf{x}^U} - \sum_{i \in \mathcal{N}_k^j} \alpha_{i,k}^j(\mathbf{x}^L, \mathbf{x}^U)\left(x_i^L - x_i\right)\left(x_i^U - x_i\right) \\
&= -\min_{\mathbf{x}^L \leq \mathbf{x} \leq \mathbf{x}^U} \sum_{i \in \mathcal{N}_k^j} \alpha_{i,k}^j(\mathbf{x}^L, \mathbf{x}^U)\left(x_i^L - x_i\right)\left(x_i^U - x_i\right) \\
&= \frac{1}{4} \sum_{i \in \mathcal{N}_k^j} \alpha_{i,k}^j(\mathbf{x}^L, \mathbf{x}^U)\left(x_i^U - x_i^L\right)^2
\end{aligned}$$

■

Property 5:. The underestimators constructed over supersets of the current set are always *less tight* than the underestimator constructed over the current box constraints for every point within the current box constraints.

Proof: See [17]. ■

Clearly, the smaller the values of the positive parameters $\alpha^j_{i,k}$, the narrower the separation between the original nonconvex terms and their respective convex relaxations will be. Therefore fewer iterations will also be required for convergence. To this end, customized α parameters are defined for each variable, term and constraint. Furthermore, an updating procedure for the α's as the size of the partition elements decreases is currently under investigation.

This type of convex lower bounding is utilized for nonconvex functions which lack any specific structure that might enable the construction of customized convex lower bounding functions. Clearly, the α–based convex lower bounding can be applied to bilinear terms as well without having to introduce additional variable and constraints. However, in this case the maximum separation will be larger than the one based on the linear cuts. More specifically, the maximum separation for the α convex lower bounding scheme is,

$$\frac{\left(x^U - x^L\right)^2 + \left(y^U - y^L\right)^2}{8}.$$

This is always greater than

$$\frac{\left(x^U - x^L\right)\left(y^U - y^L\right)}{4}$$

unless $x^U - x^L = y^U - y^L$. Based on the aforementioned convex lower bounding procedures for bilinear terms and generic nonconvex terms, a convex relaxation **(R)** of **(P)** is proposed.

$$\begin{aligned}
\min_{\mathbf{x}} \quad & C^0(\mathbf{x}) + \sum_{k \in \mathcal{K}^0} NC^0_k(\mathbf{x}) \\
& + \sum_{i \in \mathcal{N}^0_k} \alpha^0_{i,k}(\mathbf{x}^L, \mathbf{x}^U)\left(x^L_i - x_i\right)\left(x^U_i - x_i\right) + s^0_{i,i'} \qquad \textbf{(R)} \\
\text{subject to} \quad & C^j(\mathbf{x}) + \sum_{k \in \mathcal{K}^j} NC^j_k(\mathbf{x}) \\
& + \alpha^j_{i,k}(\mathbf{x}^L, \mathbf{x}^U) \sum_{i \in \mathcal{N}^j_k} \left(x^L_i - x_i\right)\left(x^U_i - x_i\right) + s^j_{i,i'} \le 0, \\
& j = 1, \ldots, (2M + K)
\end{aligned}$$

$$s^j_{i,i'} \;\geq\; \max\Big(Y_i^{j,L} x_{i'} + Y_{i'}^{j,L} x_i - Y_i^{j,L} Y_{i'}^{j,L},$$

$$Y_i^{j,U} x_{i'} + Y_{i'}^{j,U} x_i - Y_i^{j,U} Y_{i'}^{j,U},\Big), \quad j = 0, \ldots, (2M+K)$$

where
$$\begin{aligned} Y_i^{j,L} &= \min\left(b^j_{i,i'} x_i^L, b^j_{i,i'} x_i^U\right), \\ Y_{i'}^{j,L} &= \min\left(b^j_{i,i'} x_{i'}^L, b^j_{i,i'} x_{i'}^U\right), \\ Y_i^{j,U} &= \max\left(b^j_{i,i'} x_i^L, b^j_{i,i'} x_i^U\right), \\ Y_{i'}^{j,U} &= \max\left(b^j_{i,i'} x_{i'}^L, b^j_{i,i'} x_{i'}^U\right) \end{aligned}$$

$$A\mathbf{x} \;=\; \mathbf{c}, \quad \mathbf{x}^L \;\leq\; \mathbf{x} \;\leq\; \mathbf{x}^U$$

and
$$NC_k^j(\mathbf{x}) \text{ with } \mathbf{x} \;\in\; \left\{x_i \;:\; i \in \mathcal{N}_k^j\right\}, \quad j = 0, \ldots, (2M+K)$$

Formulation **(R)** is a convex programming problem whose global minimum solution can be routinely found with existing local optimization solvers such as MINOS5.4 [22]. Formulation **(R)** is a relaxation of **(P)** and therefore its solution is a valid lower bound on the global minimum solution of **(P)**.

In the next section, we will see how this convex lower bounding formulation **(R)** can be utilized in a branch and bound framework for locating the global minimum solution of **(P)**.

4. Global Optimization Algorithm, αBB

A global optimization procedure, αBB, is proposed for locating the global minimum solution of **(P)** based on the refinement of converging lower and upper bounds. Lower bounds are obtained through the solution of convex programming problems **(R)** and upper bounds based on the solution of **(P)** with local methods.

As it has been discussed in the previous subsection, the maximum separation between the generic and bilinear nonconvex terms and their respective convex lower bounding functions is bounded. For the generic nonconvex terms this maximum separation is proportional to the square of the diagonal of the rectangular partition element and for the bilinear terms proportional to the area of the rectangular domain. Furthermore, as the size of the rectangular domains approaches zero, these maximum separations go to zero as well. This implies that as the current box constraints $[\mathbf{x}^L, \mathbf{x}^U]$ collapse into a point; (i) the maximum separation between the original objective function of **(P)** and its convex relaxation in **(R)** becomes zero; and (ii) by the same argument, the maximum separation between the original constraint set in **(P)** and the one in **(R)** goes to zero as well. This implies that for every positive number ϵ_f and $\mathbf{x}$ there always exists a positive number δ such that by reducing the rectangular region $[\mathbf{x}^L, \mathbf{x}^U]$ around $\mathbf{x}$ so as $||\mathbf{x}^U - \mathbf{x}|| \leq \delta$ differences between the feasible region of the original problem **(P)** and its convex relaxation **(R)** become less than ϵ_f. Therefore, any feasible point $\mathbf{x}^c$ of problem **(R)** (even the global minimum solution) becomes at least ϵ_f–feasible for problem **(P)** by sufficiently tightening the bounds on $\mathbf{x}$ around this point.

The next step, after establishing an upper and a lower bound on the global minimum, is to refine them. This is accomplished by successively partitioning the initial rectangular region into smaller ones. The number of variables along which subdivision is required is equal to the number of variables $\mathbf{x}$ participating in at least one nonconvex term in formulation **(P)**. The partitioning strategy involves the successive subdivision of a rectangle into two subrectangles by halving on the middle point of the longest side of the initial rectangle (bisection). Therefore, at each iteration a lower bound of the objective function of **(P)** is simply the minimum over all the minima of problem **(R)** in every subrectangle composing the initial rectangle. Therefore, a straightforward (bound improving) way of tightening the lower bound is to halve at each iteration, only the subrectangle responsible for the infimum of the minima of **(R)** over all subrectangles, according to the rules discussed earlier. This procedure generates a *nondecreasing* sequence for the lower bound. An *nonincreasing* sequence for the upper bound is derived by solving locally the nonconvex problem **(P)** and selecting it to be the minimum over all the previously recorded upper bounds. Clearly, if the single minimum of **(R)** in any subrectangle is greater than the current upper bound we can safely ignore this subrectangle because the global minimum of **(P)** cannot be situated inside it (fathoming step).

Because the maximum separations between nonconvex terms and their respective convex lower bounding functions are bounded and continuous functions of the size of rectangular domain, arbitrarily small ϵ_f feasibility and ϵ_c convergence tolerances are reached for a finite size partition element.

The basic steps of the proposed global optimization algorithm are as follows:

STEP 1 - *Initialization*

A convergence tolerance, ϵ_c, and a feasibility tolerance, ϵ_f, are selected and the iteration counter $Iter$ is set to one. Current variable bounds $\mathbf{x}^{L,Iter}, \mathbf{x}^{U,Iter}$ for the first iteration are set to be equal to the global ones $\mathbf{x}^{LBD}, \mathbf{x}^{UBD}$. Lower and upper bounds LBD, UBD on the global minimum of **(P)** are initialized and an initial current point $\mathbf{x}^{c,Iter}$ is selected.

STEP 2 - *Local Solution of Nonconvex NLP and Update of Upper Bound* G_0^{UBD}

The nonconvex optimization problem **(P)** is solved locally within the current variable bounds $\mathbf{x}^{LBD}, \mathbf{x}^{UBD}$. If the solution f_{local}^{Iter} of **(P)** is ϵ_f–feasible the upper bound UBD is updated as follows,

$$UBD \;=\; \min\left(UBD, f_{local}^{Iter}\right)$$

STEP 3 - *Partitioning of Current Rectangle*

The current rectangle $\left[\mathbf{x}^{L,Iter}, \mathbf{x}^{U,Iter}\right]$ is partitioned into the following two rectangles $(r = 1, 2)$:

$$\begin{bmatrix} x_1^{L,Iter} & x_1^{U,Iter} \\ \vdots & \vdots \\ x_{l^{Iter}}^{L,Iter} & \frac{\left(x_{l^{Iter}}^{L,Iter} + x_{l^{Iter}}^{U,Iter}\right)}{2} \\ \vdots & \vdots \\ x_N^{L,Iter} & x_N^{U,Iter} \end{bmatrix}, \qquad \begin{bmatrix} x_1^{L,Iter} & x_1^{U,Iter} \\ \vdots & \vdots \\ \frac{\left(x_{l^{Iter}}^{L,Iter} + x_{l^{Iter}}^{U,Iter}\right)}{2} & x_{l^{Iter}}^{U,Iter} \\ \vdots & \vdots \\ x_N^{L,Iter} & x_N^{U,Iter} \end{bmatrix}$$

where l^{Iter} corresponds to the variable with the longest side in the initial rectangle,

$$l^{Iter} \;=\; \arg\max_i \left(x_i^{U,Iter} \;-\; x_i^{L,Iter}\right)$$

STEP 4 - *Update of* $\alpha_{i,k}^j$*'s inside both subrectangles r=1,2*

The positive parameters $\alpha_{i,k}^j(\mathbf{x}^{U,Iter}, \mathbf{x}^{L,Iter})$ are updated inside both rectangles r=1,2.

STEP 5 - *Solution of* **(R)** *inside both subrectangles r=1,2*

The convex optimization problem **(R)** is solved inside both subrectangles $(r = 1, 2)$ using any convex nonlinear solver (e.g. MINOS5.4 [22]). If a solution $l_{sol}^{r,Iter}$ is less than the current upper bound, UBD then it is stored along with the solution point $\mathbf{x}_{sol}^{r,Iter}$.

STEP 6 - *Update Iteration Counter* $Iter$ *and Lower Bound* LBD
The iteration counter is increased by one,

$$Iter \;\longleftarrow\; Iter \;+\; 1$$

and the lower bound LBD is updated to the minimum solution over the stored ones from previous iterations. Furthermore, the selected solution is erased from the stored set.

$$\begin{aligned} LBD \;&=\; l_{sol}^{r',Iter'} \\ \text{where} \quad l_{sol}^{r',Iter'} \;&=\; \min_{r,I}\; l_{sol}^{r,I}, \quad r = 1, 2, \quad I = 1, \ldots, Iter - 1. \end{aligned}$$

STEP 7 - *Update Current Point* $\mathbf{x}^{c,Iter}$ *and Current Bounds* $\mathbf{x}^{L,Iter}, \mathbf{x}^{U,Iter}$ *on* $\mathbf{x}$
The current point is selected to be the solution point of the previously found minimum solution in **STEP 6**,

$$\mathbf{x}^{c,Iter} \;=\; \mathbf{x}_{sol}^{r',Iter'}$$

and the current rectangle becomes the subrectangle containing the previously found solution,

$$\left[\mathbf{x}^{L,Iter}, \mathbf{x}^{U,Iter}\right] = \begin{bmatrix} x_1^{L,Iter'} & x_1^{U,Iter'} \\ \vdots & \vdots \\ x_{l_{Iter'}}^{L,Iter'} & \dfrac{\left(x_{l_{Iter'}}^{L,Iter'} + x_{l_{Iter'}}^{U,Iter'}\right)}{2} \\ \vdots & \vdots \\ x_N^{L,Iter'} & x_N^{U,Iter'} \end{bmatrix}, \quad \text{if} \quad r' = 1$$

$$\left[x_i^{L,Iter}, x_i^{U,Iter}\right] = \begin{bmatrix} x_1^{L,Iter'} & x_1^{U,Iter'} \\ \vdots & \vdots \\ \dfrac{\left(x_{l_{Iter'}}^{L,Iter'} + x_{l_{Iter'}}^{U,Iter'}\right)}{2} & x_{l_{Iter'}}^{U,Iter'} \\ \vdots & \vdots \\ x_N^{L,Iter'} & x_N^{U,Iter'} \end{bmatrix}, \quad \text{if} \quad r' = 2$$

STEP 8 - *Check for Convergence*

IF $(UBD - LBD) > \epsilon_c$, then return to **STEP 2**

Otherwise, ϵ_c–convergence has been reached and the global minimum solution, and solution point are:

$$\begin{aligned} f^* &\longleftarrow f^{c,Iter''} \\ \mathbf{x}^* &\longleftarrow \mathbf{x}^{c,Iter''} \\ \text{where} \quad Iter'' &= \arg_I \left\{ f^{c,I} \right) = UBD, \quad I = 1, \ldots, Iter \right\}. \end{aligned}$$

A mathematical proof that the proposed global optimization algorithm converges to the the global minimum is based on the analysis of standard deterministic global optimization algorithms presented in [16] as shown in [17] and [18].

5. Implementation of αBB

One of the key characteristics of the αBB method is that it is a generic global optimization method for constrained optimization problems involving only continuous variables. The algorithm is implemented in C and at this point the user has the capability of selecting from four different types of functional forms to define the optimization model. These forms include (i) linear, (ii) convex, (iii) bilinear, and (iv) nonconvex terms. The original data are pre–processed so that any linear part in the model, (i.e. linear constraints and linear cuts), are identified at the very beginning thus reducing the amount of time that is needed to set up the problem in subsequent stages of the algorithm. The user has the capability to supply the values for the parameters α which are defined for each variable $i = 1, \ldots, N$ participating in term $k \in \mathcal{K}^j$ and constraint (or objective function) $j = 0, \ldots, M$. In principle, tailoring the α parameters for each variable, term and constraint generates tighter convex underestimators than by simply defining a single generic α for all the variables and nonconvex terms. Furthermore, the user also decides along which variables branching will be performed. These variables are typically the ones that appear in at least one nonconvex term.

The information required by the user, in the current implementation, consists of an input file and a set of user specified functions.

- *Input File* : This file provides, in a user–friendly format, information such as (i) the number of variables and constraints; (ii) the number of different functional forms (i.e. linear, convex, bilinear, and nonconvex) appearing in the model; (iii) the actual linear and bilinear entries; (iv) values for the parameter $\alpha^j_{i,k}$ for each variable, term, and constraint or objective function; and finally (v) the variables along which branching will be performed.

- *User Specified Functions* : The nonlinear, (i.e. convex and nonconvex), terms of the formulation have to be explicitly provided by the user in a form of a C or F77 subroutine. Here the user specifies, for each function (as defined in the input file), the convex and nonconvex terms.

An efficient parsing phase which would significantly simplify the problem input and declaration is currently under development and is going to be incorporated in the version of αBB. Further work is in progress towards the evaluation of customized parameters α for different partition elements.

6. Computational Studies

The αBB method has been tested on a variety of nonlinear optimization problems which are described in the following subsections. The selected convergence tolerance is 10^{-4} and computational requirements are reported for an HP–730 workstation.

6.1. Bilinearly Constrained Optimization Problems

The simplest type of non–linearities present in the formulation are bilinear terms in either the objective or the constraint set. The first three examples to be considered are the *Haverly Pooling Problems* [9]. The three instances of of the Haverly Pooling problems are the following:

- Case I :

$$max \quad 9x + 15y - 6A - 16B - 10(c_x + c_y)$$

$$
\begin{aligned}
s.t. \quad P_x + P_y - A - B &= 0 \\
x - P_x - C_x &= 0 \\
y - P_y - C_y &= 0 \\
pP_x + 2C_x - 2.5x &\leq 0 \\
pP_y + 2C_y - 1.5y &\leq 0 \\
pP_x + pP_y - 3A - B &= 0 \\
x &\leq 100 \\
y &\leq 200
\end{aligned}
$$

In this first instance, there are three linear equality constraints, two bilinear inequalities and one bilinear equality. The three bilinear constraints will be underestimated using linear cuts [2]. There is a total of 9 continuous variables, however, branching is required on only three of them, (i.e. p, P_x, P_y), which participate in the bilinear terms. The algorithm converges to the global minimum in about 2.7 seconds and a total of 89 nodes of the complete binary tree are expanded. This means that 89 lower bounding problems were solved to meet the selected convergence tolerance of 10^{-3}. The global minimum solution is located at : $p = 1, B = P_y = C_y = 100, y = 200, C_x = A = 0$.

- Case II :

  This problem is identical to Case I, except that the upper bound on variable x is changed from 100 to 600. The global minimum is now at: $p = 3, A = P_x = C_x = 300, x = 600, C_y = B = 0$. The solution is found in about 3.0 seconds and a total of 97 nodes are investigated.

- Case III :

  In Case III, the value of the coefficient of B in the objective function is changed from 16 to 13. The solution, located in about 2.2 seconds, is $p = 1.5, A = 50, B = 150, P_y = 200, y = 200, P_x = x = 0$, and a total of 91 nodes needed to be investigated.

6.2. Bilinearly Constrained with Bilinear Objective Optimization Problems

The next degree of difficulty is to consider bilinearities in both in the objective as well as in the constraint set. As such an example we will consider the following formulation which describes the optimal design of a separation system involving three mixers, a splitter, a flash unit, and a column. The optimization problem is defined as follows :

$$
\begin{aligned}
\min \quad & - 87.5x_1 - 316.5625x_1x_3 - 352.3438x_1x_4 - 143.5x_1x_5 - 175x_2 \\
& - 271.875x_2x_3 - 307.8125x_2x_4 - 62.5x_2x_5 + 1250x_6 + 50x_7
\end{aligned}
$$

$$
\begin{aligned}
s.t.\quad & 13.9375x_1x_3 + 13.9688x_1x_4 - x_4 + 25x_1x_5 \\
& +13.125x_2x_3 + 12.8125x_2x_4 + 25x - 2x_5 \leq 15 \\
& -25x_1x_3 - 25x_2x_3 + 62.5x_7 \leq 0 \\
& 25x_1x_3 + 25x_2x_3 - 62.5x_7 \leq 0 \\
& -25x_1x_4 - 25x_2x_4 + 62.5x_6 \leq 0 \\
& 25x_1x_4 + 25x_2x_4 - 62.5x_6 \leq 0 \\
& 31.25x_1x_5 - 2.6875x_1x_3 - 11.1563x_1x_4 \\
& +37.5x_2x_5 - 0.625x_2x_3 - 9.6875x_2x_4 \leq 0 \\
& -30x_1x_5 + 25x_2 + 29.375x_2x_3 - 35.9375x_2x_4 - 25x_2x_5 \leq 0 \\
& 25x_1 - 13.9375x_1x_3 - 13.9688x_1x_4 - 25x_1x_5 \\
& +25x_2 - 13.125x_2x_3 - 12.8125x_2x_4 - 25x_2x_5 \leq 18
\end{aligned}
$$

This problem involves seven variables and branching is required in all of them. Convergence to the global minimum solution $(x_1 = 0.3200, x_2 = 1.0000, x_3 = 0.7920, x_4 = 0.0629, x_5 = 0.0000, x_6 = 0.0033, x_7 = 0.0418)$, takes 28.5 seconds and requires the solution of 153 linear programming subproblems.

6.3. Nonlinear Unconstrained Optimization Problems

The next degree of difficulty consists of optimization problems with nonconvexities in the objective function and simple variable bound constraints. An example corresponding to a robust control synthesis problem which has been very challenging to solve for the local solver MINOS5.4 is addressed. The problem is formulated as follows:

$$\min_{x,w} -T = -\sqrt{(\frac{M_1}{M_2})}$$

$$
\begin{aligned}
where\quad M_1 &= (\mu_2\mu_3 - \mu_1\mu_4)^2 + (\mu_1\mu_3 + \mu_2\mu_4)^2 \\
M_2 &= \{1 + (\mu_2 - \mu_5)\mu_3 - (\mu_1 - \mu_6)\mu_4\}^2 + \{\mu_2 - \mu_5)\mu_4 \\
&+ (\mu_1 - \mu_6)\mu_3\}^2 \\
\mu_6 &= \frac{2\{sin(-5w) - 3wcos(-5w)\}}{(3w)^2 + 1} \\
\mu_5 &= \frac{2\{cos(-5w) + 3wsin(-5w)\}}{(3w)^2 + 1} \\
\mu_4 &= 0.5\{\frac{3w - \epsilon w}{(\epsilon w)^2 + 1}\} \\
\mu_3 &= 0.5\{\frac{1 + 3\epsilon w^2}{(\epsilon w)^2 + 1}\} \\
\mu_1 &= x_1\frac{sin(-x_3w) + x_2wcos(-x_3w)}{(x_2w)^2 + 1}
\end{aligned}
$$

$$
\begin{aligned}
\mu_2 &= x_1 \frac{cos(-x_3 w) - x_2 w sin(-x_3 w)}{(x_2 w)^2 + 1} \\
0.00 &\leq w \leq 1.00 \\
1.00 &\leq x_1 \leq 3.00 \\
2.00 &\leq x_2 \leq 4.00 \\
4.00 &\leq x_3 \leq 6.00
\end{aligned}
$$

The following two cases for the parameter ϵ are considered.

1. $\underline{\epsilon = 3.00 :}$ The problem involves only four variables, however, after 100 multi-start runs using the local solver MINOS5.4 [22], the global minimum was identified only 5 times. The method αBB consistently located the global minimum solution $x_1 = 3, x_2 = 2, x_3 = 4, w = 0.6670$ with an objective function value of -2.8765. Computational requirements for different values of α are shown in Table 1. Apparently, there is a very strong local minimum solution with a value of -2.7072, and a corresponding solution vector $x_1 = 3, x_2 = 2, x_3 = 6, w = 0.0$ which was most of the time the convergence point of the local solver MINOS5.4.

Table 1. Results for the robust control synthesis problem $\epsilon = 3.00$.

α	N_{iter}	CPU (s.)
0.50	16	0.61
0.75	17	0.62
1.00	18	0.67

2. $\underline{\epsilon = 10.5101 :}$ This selection for the parameter ϵ makes the problem even more difficult for the local solver MINOS5.4 to generate the global minimum solution. In fact, in only one out of 100 times was the local solver able to find the global minimum located at $x_1 = 3, x_2 = 4, x_3 = 6, w = 0.0959$ with an objective function value of -1.0507. A very strong local minimum of -1.000 located at $x_1 = 2.806842, x_2 = 3.126284, x_3 = 4.183029, w = 0.0$ again dominated the reported solutions by the local solver. Computational results are shown in Table 2.

6.4. Linearly Constrained Nonlinear Optimization Problems

The examples of this section are taken from [20]. They correspond to three very challenging phase equilibrium problems, and are defined as follows: *Given i components participating in up to k potential phases under isothermal and isobaric conditions find the mole vector n that minimizes the value of the Gibbs free energy while satisfying appropriate material balance constraints.*

Table 2. Results for the robust control synthesis problem $\epsilon = 10.5101$.

α	N_{iter}	CPU (s.)
0.50	203	5.94
0.75	488	14.9
1.00	511	16.2

- Problem I:

  The first physical system describes the phase equilibrium of a systems containing *n-Butyl-Acetate – Water.* The formulation is as follows :

$$
\begin{aligned}
\min\ \hat{G}_I \ =\ & n_1^1 ln\, n_1^1 + n_2^1 ln\, n_2^1 - [n_1^1 + n_2^1] ln\, [n_1^1 + n_2^1] \\
+\ & n_1^2 ln\, n_1^2 + n_2^2 ln\, n_2^2 - [n_1^2 + n_2^2] ln\, [n_1^2 + n_2^2] \\
+\ & G_{12}\tau_{12} n_1^1 \frac{n_2^1}{n_2^1 + G_{12} n_2^1} + G_{21}\tau_{21} n_2^1 \frac{n_1^1}{n_1^1 + G_{21} n_2^1} \\
+\ & G_{12}\tau_{12} n_1^2 \frac{n_2^2}{n_2^2 + G_{12} n_1^2} + G_{12}\tau_{12} n_1^1 \frac{n_2^1}{n_2^1 + G_{12} n_2^1} \\
s.t.\quad & n_1^1 + n_1^2 = 0.5 \\
& n_2^1 + n_2^2 = 0.5 \\
& 0 \le n_1^1, n_2^1, n_1^2, n_2^2 \le 0.5
\end{aligned}
$$

  The terms of the form $nln(n)$ have been shown to be convex [20]. Therefore, the optimization problem contains an objective function that has a convex term and four additional non–convex terms.

  The global minimum solution is presented in Table 3. Computational results are shown in Table 4.

Table 3. Global minimum of example I.

	obj	n_1^1	n_1^2	n_2^1	n_2^2
Global	-0.00202	0.00071	0.49929	0.15588	0.34412

- Problem II:

  The second system describes the phase equilibrium of the ternary system *n-Propanol – n-Butanol – Water.* The minimization of the Gibbs free energy takes the form :

Table 4. Results for the first phase equilibrium example.

α	N_{iter}	CPU (s.)
0.10	18	0.41
0.25	49	0.99
0.50	105	2.46

$$\begin{aligned}
\min\ \hat{G}_I \ &= \ n_1^1 ln\, n_1^1 + n_2^1 ln\, n_2^1 + n_3^1 ln\, n_3^1 - [n_1^1 + n_2^1 + n_3^1] ln\, [n_1^1 + n_2^1 + n_3^1] \\
&+ \ n_1^2 ln\, n_1^2 + n_2^2 ln\, n_2^2 + n_3^2 ln\, n_3^2 - [n_1^2 + n_2^2 + n_3^2] ln\, [n_1^2 + n_2^2 + n_3^2] \\
&+ \ n_1^1 [G_{12}\tau_{12} \frac{n_2^1}{n_2^1 + G_{12} n_1^1 + G_{32} n_3^1} + G_{13}\tau_{13} \frac{n_3^1}{n_3^1 + G_{13} n_1^1 + G_{23} n_2^1}] \\
&+ \ n_2^1 [G_{21}\tau_{21} \frac{n_1^1}{n_1^1 + G_{21} n_2^1 + G_{31} n_3^1} + G_{23}\tau_{23} \frac{n_3^1}{n_3^1 + G_{13} n_1^1 + G_{23} n_2^1}] \\
&+ \ n_3^1 [G_{31}\tau_{31} \frac{n_1^1}{n_1^1 + G_{21} n_2^1 + G_{31} n_3^1} + G_{32}\tau_{32} \frac{n_2^1}{n_2^1 + G_{12} n_1^1 + G_{32} n_3^1}] \\
&+ \ n_1^2 [G_{12}\tau_{12} \frac{n_2^2}{n_2^2 + G_{12} n_1^2 + G_{32} n_3^2} + G_{13}\tau_{13} \frac{n_3^2}{n_3^2 + G_{13} n_1^2 + G_{23} n_2^2}] \\
&+ \ n_2^2 [G_{21}\tau_{21} \frac{n_1^2}{n_1^1 + G_{21} n_2^2 + G_{31} n_3^2} + G_{23}\tau_{23} \frac{n_3^2}{n_3^2 + G_{13} n_1^2 + G_{23} n_2^2}] \\
&+ \ n_3^2 [G_{31}\tau_{31} \frac{n_1^2}{n_1^1 + G_{21} n_2^2 + G_{31} n_3^2} + G_{32}\tau_{32} \frac{n_2^2}{n_2^2 + G_{12} n_1^2 + G_{32} n_3^2}] \\
s.t. \qquad & n_1^1 + n_1^2 = n_1^T \ , \ \ 0 \le n_1^1, n_1^2 \le n_1^T \\
& n_2^1 + n_2^2 = n_2^T \ , \ \ 0 \le n_2^1, n_2^2 \le n_2^T \\
& n_3^1 + n_3^2 = n_3^T \ , \ \ 0 \le n_3^1, n_3^2 \le n_3^T
\end{aligned}$$

The extremely difficult instance with $n_i^T = \{0.148, 0.052, 0.800\}$, due to the very small objective value difference between the global and a local solution, was successfully solved.

The global minimum solution is presented in Table 5.

Table 5. Global minimum of example II.

	obj	n_1^1	n_1^2	n_2^1	n_2^2	n_3^1	n_3^2
Global	-0.27081404	0.0456	0.0063	0.6550	0.1450	0.1280	0.0200

Computational results are presented in Table 6.

Table 6. Results for the second phase equilibrium problem.

α	N_{iter}	CPU (s.)
0.025	208	7.10
0.05	320	10.4
0.10	772	24.6

- Problem III :

  The final example describes the phase equilibrium of *Toluene – Water.*

$$z_1^R = \frac{5q_1 - 1}{r_1} = 3.53316 \qquad z_2^R = \frac{5q_2 - 1}{r_2} = 6.52174$$

$$z_M^R = \min\{\, z_1^R, z_2^R \,\} = z_1^R = 3.53316$$

$$z^A = z_M^R + \left(z_1^R - z_M^R\right) + \left(z_2^R - z_M^R\right) = z_2^R = 6.52174$$

$$z_1^B = z_2^R - z_M^R = 2.98858 \qquad z_2^B = z_1^R - z_M^R = 0$$

$$\varphi_1 = q_1' + r_1 \cdot z^A = 28.53522$$
$$\varphi_2 = q_2' + r_2 \cdot z^A = 7.0$$

$$\begin{aligned}
\min\ \hat{G}_I \;=\; & n_1^T \left\{ -z_1^R r_1 \ln r_1 + \frac{z}{2} q_1 \ln q_1 \right\} n_2^T \left\{ -z_2^R r_2 \ln r_2 + \frac{z}{2} q_2 \ln q_2 \right\} \\
+ \; & z^A \left[r_1 n_1^1 + r_2 n_2^1 \right] \ln \left[r_1 n_1^1 + r_2 n_2^1 \right] + z_1^B r_1 n_1^1 \ln \frac{n_1^1}{r_1 n_1^1 + r_2 n_2^1} \\
+ \; & z^A \left[r_1 n_1^2 + r_2 n_2^2 \right] \ln \left[r_1 n_1^2 + r_2 n_2^2 \right] + z_1^B r_1 n_1^2 \ln \frac{n_1^2}{r_1 n_1^2 + r_2 n_2^2} \\
+ \; & \frac{z}{2} q_1 n_1^1 \ln \frac{n_1^1}{q_1 n_1^1 + q_2 n_2^1} + \frac{z}{2} q_2 n_2^1 \ln \frac{n_2^1}{q_1 n_1^1 + q_2 n_2^1} \\
+ \; & \frac{z}{2} q_1 n_1^2 \ln \frac{n_1^2}{q_1 n_1^2 + q_2 n_2^2} + \frac{z}{2} q_2 n_2^2 \ln \frac{n_2^2}{q_1 n_1^2 + q_2 n_2^2} \\
+ \; & \left[q_1' n_1^1 + q_2' n_2^1 \right] \ln \left[q_1' n_1^1 + q_2' n_2^1 \right] + q_1' n_1^1 \ln \frac{n_1^1}{q_1' n_1^1 + \tau_{21} q_2' n_2^1} \\
+ \; & q_2' n_2^1 \ln \frac{n_2^1}{\tau_{12} q_1' n_1^1 + q_2' n_2^1}
\end{aligned}$$

$$
\begin{aligned}
&+ \ \left[q_1' n_1^2 + q_2' n_2^2\right] \ln \left[q_1' n_1^2 + q_2' n_2^2\right] + q_1' n_1^2 \ln \frac{n_1^2}{q_1' n_1^2 + \tau_{21} q_2' n_2^2} \\
&+ \ q_2' n_2^2 \ln \frac{n_2^2}{\tau_{12} q_1' n_1^2 + q_2' n_2^2} \\
&- \ \varphi_1 \left[n_1^1 \ln n_1^1 + n_1^2 \ln n_1^2\right] - \varphi_2 \left[n_2^1 \ln n_2^1 + n_2^2 \ln n_2^2\right]
\end{aligned}
$$

s.t.

$$
\begin{aligned}
& n_1^1 + n_1^2 = 0.5 \\
& n_2^1 + n_2^2 = 0.5 \\
& 0 \leq n_1^1, n_2^1, n_1^2, n_2^2 \leq 0.5
\end{aligned}
$$

The global minimum solution is presented in Table 7.

Table 7. Global minimum of example III.

	obj	n_1^1	n_1^2	n_2^1	n_2^2
Global	-0.01975944	0.0004	0.4996	0.4772	0.0228

Computational results are shown in Table 8.

Table 8. Results for the third phase equilibrium problem.

α	N_{iter}	CPU (s.)
0.50	51	1.95
0.75	55	2.08
1.00	63	2.27

6.5. Nonlinearly Constrained Nonlinear Optimization Problems

The problems of this section involve nonconvex terms of generic structure in both their objective function and constraints. Two examples will be presented, a small one in order to illustrate all possible combinations of functional forms that can be present in a general nonconvex optimization problem, and a larger one in order to illustrate the applicability of the method on a real–world problem.

- Example I: This example is taken from the manual of MINOS5.4 [22]. The formulation is as follows :

$$
\min \ (x_1 - 1)^2 + (x_1 - x_2)^2 + (x_2 - x_3)^3 + (x_3 - x_4)^4 + (x_4 - x_5)^4
$$

$$
\begin{aligned}
s.t. \quad x_1 + x_2^2 + x_3^3 &= 3\sqrt{2} + 2 \\
x_2 - x_3^2 + x_4 &= 2\sqrt{2} - 2 \\
x_1 x_5 &= 2
\end{aligned}
$$

The αBB input file for this problem is shown in the appendix. This examples involves linear, convex, bilinear, and non–convex terms. The global minimum solution along with four local solutions are shown in Table 9. Computational requirements for different values of α are presented in Table 10.

Table 9. Global and local minima of example I.

	obj	x_1	x_2	x_3	x_4	x_5
Global	0.0293	1.1166	1.2204	1.5378	1.9728	1.7911
Local 1	27.8719	-1.2731	2.4104	1.1949	-0.1542	-1.5710
Local 2	44.0221	-0.7034	2.6357	-0.0963	-1.7980	-2.8434
Local 3	52.9026	0.7280	-2.2452	0.7795	3.6813	2.7472
Local 4	64.8740	4.5695	-1.2522	0.4718	2.3032	4.3770

Table 10. Computational results for example I.

α	N_{iter}	CPU (s.)
0.05	35	13.87
0.25	81	30.80
0.50	335	99.76

- Example II: This test example addresses an *optimal blank nesting* problem involving important industrial applications. The objective is to minimize the “scrap” metal and the problem is formulated as follows:

$$
\min_{\theta_1, \theta_2, \Delta x, \Delta y, w, p} \quad wp
$$

$$
\begin{aligned}
s.t. \quad w &= w_1 - w_2 \\
w_1 &\geq y_i^a \ , \quad a = 1,2 \\
w_2 &\leq y_i^a \ , \quad a = 1,2 \\
(r_i + r_j)^2 &\leq (x_i^a - x_j^b)^2 + (y_i^a - y_j^b)^2 \\
r_i \leq y_i^1 &\leq w - r_i \\
r_i \leq y_i^2 &\leq w - r_i \\
\sum_{i=1}^{N_s} x_i^1 &\leq \sum_{i=1}^{N_s} x_i^2 \leq \sum_{i=1}^{N_s} x_i^{1+}
\end{aligned}
$$

$$\begin{aligned}
\forall a, b &\in \{1, 1^+, 2\}, \forall i, j \in \{1 \ldots N_s\} \\
where \quad x_i^1 &= c_1 x_i - s_1 y_i \\
y_i^1 &= s_1 x_i + c_1 y_i \\
x_i^{1+} &= c_1 x_i - s_1 y_i + p \\
y_i^{1+} &= s_1 x_i + c_1 y_i \\
x_i^2 &= c_2 x_i - s_2 y_i + \Delta x \\
y_i^2 &= s_2 x_i + c_2 y_i + \Delta y \\
c_1 &= cos(\theta_1), \quad s_1 \quad = \quad sin(\theta_1) \\
c_2 &= cos(\theta_2), \quad s_2 \quad = \quad sin(\theta_2)
\end{aligned}$$

The results of local minimization runs from 50 randomly generated starting points using MINOS 5.4 [22] are shown in Table 11. The global solution is identified by α**BB** in 250 iterations and 3,153 seconds of CPU time. It is shown, along with some local solutions in Table 12.

Table 11. Sample Local Runs

Run No.	obj. fun.	Run. No.	obj. fun.
0	299.230856 (0)	1	299.230856 (0)
2	Failure	3	350.065316 (0)
4	Failure	5	344.095172 (0)
6	350.065316 (0)	7	350.065316 (0)
8	344.095172 (0)	9	Failure
10	350.065316 (0)	11	Failure
12	Failure	13	350.065316 (0)
14	350.065316 (0)	15	Failure
16	350.065316 (0)	17	350.065316 (0)
18	350.065316 (0)	19	Failure
20	Failure	21	350.065316 (0)
22	Failure	23	Failure
24	344.095172 (0)	25	235.721608 (0)
26	299.230856 (0)	27	Failure
28	Failure	29	Failure
30	Failure	31	235.721608 (0)
32	Failure	33	350.065316 (0)
34	350.065316 (0)	35	350.065316 (0)
36	350.065316 (0)	37	Failure
38	344.095172 (0)	39	350.065316 (0)
40	350.065316 (0)	41	Failure
42	350.065316 (0)	43	350.065316 (0)
44	350.065316 (0)	45	Failure
46	350.065316 (0)	47	350.065316 (0)
48	Failure	49	Failure

As can be seen from these results, in only two out of the 50 runs the global minimum is identified.

Table 12. Local and global solutions

f	θ_1	θ_2	dx	dy	w	p
235.721608	2.755873	5.897465	-12.660181	-3.62188	8.394903	28.079134
299.230856	2.783586	5.925179	-14.000000	0.000000	13.283868	22.525884
344.095172	2.315182	5.456775	-13.767864	0.000000	15.000000	22.939678
350.065316	2.362680	5.504273	-14.000000	0.000000	14.004254	24.997070

7. Conclusions

In this paper, the global optimization method αBB, is introduced for solving continuous constrained nonlinear optimization problems with nonconvexities both in the objective function and constraints. These nonconvexities are partitioned as either of special structure, if there exist tight convex lower bounding functions for them, or otherwise generic. A convex relaxation of the original problem is then constructed by (i) replacing all nonconvex terms of special structure (i.e. bilinear) with customized tight convex lower bounding functions and (ii) by utilizing the α parameter, as defined in [17], to underestimate nonconvex terms of generic structure. αBB attains finite ϵ–convergence to the global minimum solution through the successive partitioning of the feasible region coupled with the solution of a series of nonlinear convex minimization problems. The key feature of αBB is that it is applicable to a large number of optimization problems. Comparisons with other methods on test problems indicate the efficiency of αBB.

Acknowledgments

Financial support from the National Science Foundation under Grants CBT–8857013, CTS–9221411, the Air Force Office of Scientific Research, as well as Exxon Co., Amoco Chemicals Co., Mobil Co., and Tennessee Eastman Co. is gratefully acknowledged.

Appendix

Sample αBB Input File

```
# Note:
#   (1) Lines starting with a "#" or "!" are comment
#       lines.
#   (2) Empty lines are ignored.
#   (3) Fields can be separated by any non-zero
#       number of spaces or tabs.
#
BEGIN     mhw4d        # Problem name
OBJTYPE   Nonconvex    # Type of objective function
```

```
CONTYPE   Nonconvex    # Type of constraints
NXVAR     5            # Number of X variables
NCON      5            # Total number constraints
EPSA      1.0E-05      # Absolute conv. tolerance
EPSR      1.0E-03      # Relative conv. tolerance
YSTART   -1            # Type of starting point desired
OBJCONST  0            # Constants in the objective
USERFUNC  default
#
# Rows section
# this section will perform a complete description
# of all the rows in the problem.
# A row is defined by
# 1. a number
# 2. a relationship type ( <=, >= , == )
# 3. a RHS value
# 4. number of linear terms
# 5. number of convex terms
# 6. number of bilinear terms
# 7. number of nonconvex terms
# All constraints should appear in this section
# and all constraints are assumed to have a default
# linear. This will help the consistency
# of the model and the handling of the different
# formulations.
# Format :
# Row_No Row_sense RHS_Value Linear_terms
#        Convex_terms Bilinear_terms Nonconvex_terms
#
ROWS 5
0  -1  0.0       1  1  0  1  0
1  -1  6.242641  1  1  0  1  0
2  -1 -6.242641  1  0  0  1  0
3  -1  0.828427  2  0  0  1  0
4  -1 -0.828427  2  1  0  0  0
5   0  2.0       0  0  1  0  0
#
# Columns section
# Format :
#   Row_Number   X-Index   Y-Index   Coefficient
#
# The objective function is denoted by Row Number 0.
#
COLUMNS 0
  0   1   0   0.0
  1   1   0   1.0
  2   1   0  -1.0
  3   2   0   1.0
```

```
  3   4   0   1.0
  4   2   0  -1.0
  4   4   0  -1.0
  5   1   5   1.0
#
# Bounds Section
# Format:
#    Bound_Type  X-Index  Y-Index   Value
#
BOUNDS  10  # Total number of bounds specified = 2
  L   1   0   -6.5
  L   2   0   -6.5
  L   3   0   -6.5
  L   4   0   -6.5
  L   5   0   -6.5
  U   1   0    6.5
  U   2   0    6.5
  U   3   0    6.5
  U   4   0    6.5
  U   5   0    6.5
#
# Alpha Section
# Format:
# Obj. Function / Constraint X-Index  Value
#
#
#  Case (a) : all alphas will correspond to the
#             same fixed values
# ALPHA -1
#  0   0   10.0
#  Case (b) : user specified alphas

ALPHA 9
T1 1 1 0.25
T1 1 2 0.25
T1 1 3 0.25
T1 1 4 0.25
T1 1 5 0.25
T2 2 3 0.25
T3 3 2 0.25
T3 3 3 0.25
T4 4 3 0.25
BRANCH 5
V1 1
V2 2
V3 3
V4 4
V5 5
```

```
END
```

References

1. F.A. Al-Khayyal. Jointly Constrained Bilinear Programs and Related Problems. *Int. J. Comp. Math.*, 19(11):53, 1990.
2. F.A. Al-Khayyal and J.E. Falk. Jointly constrained biconvex programming. *Math. Opers. Res.*, 8:523, 1983.
3. I.P. Androulakis and V. Venkatasubramanian. A Genetic Algorithmic Framework for Process Design and Optimization. *Comp. chem. engng.*, 15(4):217, 1991.
4. A. Ben-Tal, G. Eiger, and V. Gershovitz. Global Optimization by Reducing the Duality Gap. *in press, Math. Prog.*, 1994.
5. C.A. Floudas and P.M. Pardalos. *Recent advances in global optimization*. Princeton Series in Computer Science. Princeton University Press, Princeton, New Jersey, 1992.
6. C.A. Floudas and V. Visweswaran. A global optimization algorithm (GOP) for certain classes of nonconvex NLPs: I. Theory. *Comput. chem. Engng.*, 14(12):1397, 1990.
7. C.A. Floudas and V. Visweswaran. A primal-relaxed dual global optimization approach. *J. Opt. Th. Appl.*, 78(2):187, 1993.
8. E.R. Hansen. *Global Optimization Using Interval Analysis*. Marcel Dekkar, New York, NY, 1992.
9. C.A. Haverly. Studies of the Behavior of Recursion for the Pooling Problem. *SIGMAP Bulletin*, 25:19, 1978.
10. D.E. Goldberg. *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison–Wesley, New York, NY, 1989.
11. R. Horst and P.M. Pardalos. *Handbook of Global Optimization: Nonconvex Optimization and Its Applications*. Kluwer Academic Publishers, 1994.
12. P. Hansen, B. Jaumard, and S. Lu. Global Optimization of Univariate Lipschitz Functions: I. Survey and Properties. *Math. Prog.*, 55:251, 1992a.
13. P. Hansen, B. Jaumard, and S. Lu. Global Optimization of Univariate Lipschitz Functions: II. New Algorithms and Computational Comparison. *Math. Prog.*, 55:273, 1992b.
14. R. Horst, N.V. Thoai, and J. De Vries. A New Simplicial Cover Technique in Constrained Global Optimization. *J. Global Opt.*, 2:1, 1992.
15. R. Horst and H. Tuy. On the Convergence of Global Methods in Multiextremal Optimization. *Opt. Th. Appl.*, 54:283, 1987.
16. R. Horst and H. Tuy. *Global Optimization*. Springer-Verlag, Berlin, Germany, 1990.
17. C.D. Maranas and C.A. Floudas. Global Minimum Potential Energy Conformations of Small Molecules. *J. Glob. Opt.*, 4:135, 1994a.
18. C.D. Maranas and C.A. Floudas. Global Optimization in Generalized Geometric Porgramming. *submitted to Comp. chem. Engnr.*, 1994c.
19. C.D. Maranas and C.A. Floudas. Finding All Solutions of Nonlinearly Constrained Systems of Equations. *submitted to J. Global Opt.*, 1995a.
20. C.M. McDonald and C.A. Floudas. Decomposition Based and Branch and Bound Global Optimization Approaches for the Phase Equilibrium Problem. *J. Global Opt.*, 5:205, 1994.
21. R. Moore, E. Hansen and A. Leclerc. Rigorous Methods for Global Optimization. In *Recent advances in global optimization*. Princeton Series in Computer Science. Princeton University Press, Princeton, New Jersey, 1992.
22. B.A. Murtagh and M.A. Saunders. *MINOS5.0 Users Guide*. Systems Optimization Laboratory, Dept. of Operations Research, Stanford University, CA., 1983. Appendix A: MINOS5.0, Technical Report SOL 83-20.
23. P.M. Pardalos and J.B. Rosen. *Constrained global optimization: Algorithms and applications*, volume 268 of *Lecture Notes in Computer Science*. Springer Verlag, Berlin, Germany, 1987.
24. H. Ratschek and J. Rokne. *New Computer Methods for Global Optimization*. Halsted Press, Chichester, Great Britain, 1988.
25. A.H.G. Rinnoy-Kan and G.T. Timmer. Stochastic Global Optimization Methods. Part I: Clustering Methods. *Math. Prog.*, 39:27, 1987.

26. C.D. Gelatt S. Kirkpatrick and M.P. Vecchi. *Science*, 220:671, 1983.
27. H.D. Sherali and A. Alameddine. A New Reformulation Linearization Technique for Bilinear Programming Problems. *J. Global Opt.*, 2(4):379, 1992.
28. H.D. Sherali and C.H. Tuncbilek. A Global Optimization Algorithm for Polynomial Programming Using a Reformulation–Linearization Technique. *J. Global Opt.*, 2:101, 1992.
29. N.Z. Shor. Dual Quadratic Estimates in Polynomial and Boolean Programming. *Annals of Operations Research*, 25:163, 1990.
30. A. Torn and A. Zilinskas. *Global Optimization*, volume 350 of *Lecture Notes in Computer Science*. Springer-Verlag, Berlin, Germany, 1989.
31. H. Tuy. Global Minimum of the Difference of Two Convex Functions. *Mathematical Programming Study*, 30:150, 1987.
32. H. Tuy, T.V. Thieu, and N.Q. Thai. A Conical Algorithm for Globally Minimizing A Concave Function over a Closed Convex Set. *Mathematics of Operations Research*, 10:498, 1985.
33. V. Visweswaran and C.A. Floudas. New properties and computational improvement of the GOP algorithm for problems with quadratic objective function and constraints. *J. Global Opt.*, 3(3):439, 1993.