J Glob Optim (2008) 41:163–186
DOI 10.1007/s10898-007-9203-8

A Lagrangean based branch-and-cut algorithm for global optimization of nonconvex mixed-integer nonlinear programs with decomposable structures

Ramkumar Karuppiah · Ignacio E. Grossmann

Received: 5 July 2006 / Accepted: 20 June 2007 / Published online: 25 July 2007

Abstract In this work we present a global optimization algorithm for solving a class of large-scale nonconvex optimization models that have a decomposable structure. Such models, which are very expensive to solve to global optimality, are frequently encountered in two-stage stochastic programming problems, engineering design, and also in planning and scheduling. A generic formulation and reformulation of the decomposable models is given. We propose a specialized deterministic branch-and-cut algorithm to solve these models to global optimality, wherein bounds on the global optimum are obtained by solving convex relaxations of these models with certain cuts added to them in order to tighten the relaxations. These cuts are based on the solutions of the sub-problems obtained by applying Lagrangean decomposition to the original nonconvex model. Numerical examples are presented to illustrate the effectiveness of the proposed method compared to available commercial global optimization solvers that are based on branch and bound methods.

Keywords Global optimization · Lagrangean decomposition · Cuts · Two-stage stochastic programming

1 Introduction

Many real-world optimization problems lead to nonconvex problems [2,17,28,42]. The mathematical models for such optimization problems include nonlinearities and/ or discrete variables, which give rise to the nonconvexity of the model. Due to the presence of these nonconvexities, sub-optimal solutions may be obtained with local solvers. There are many instances where the global solution of a problem is required [33], and for these problems deterministic global optimization techniques can be used to find the solution. These techniques

R. Karuppiah · I. E. Grossmann (✉)
Department of Chemical Engineering, Carnegie Mellon University,
Pittsburgh, PA 15213, USA
e-mail: grossmann@cmu.edu

guarantee global optimality for problems with special structures, and usually involve some form of a branch and bound search.

Deterministic global optimization techniques for solving nonconvex nonlinear programming (NLP) problems with special mathematical structures have been proposed by several authors. For instance, Sherali and Alameddine [39] have presented an algorithm based on a reformulation linearization technique for optimizing bilinear programming models, while Quesada and Grossmann [36], Ryoo and Sahinidis [37] and Zamora and Grossmann [45] have made extensions for handling linear fractional and concave functions within a spatial branch and bound method. For handling nonlinear problems with factorable constraints which do not have an explicit representation, a global optimization algorithm has been proposed by Meyer and Floudas [31]. Zhu and Kuno [46] have developed a global optimization method called *QBB* for twice-differentiable nonconvex NLPs, where quadratic lower bounding functions are used in constructing relaxations. Excellent reviews on global optimization methods for solving nonconvex NLP problems are given in Horst and Tuy [20] and in Floudas [15].

For mixed-integer nonlinear programming (MINLP) models, algorithms such as Outer Approximation (OA) by Duran and Grossmann [11] and by Fletcher and Leyffer [14] yield globally optimal solutions only if the feasible space and the objective function of the relaxed NLP problem are convex (see [17]). Pörn and Westerlund [35] have presented an Extended Cutting Plane algorithm for globally optimizing MINLPs with a pseudo-convex objective function and pseudo-convex inequality constraints. For solving nonconvex MINLPs to global optimality, a Branch and Reduce algorithm has been proposed by Sahinidis [38], on which the commercial solver BARON is based. Adjiman et al. [3] have presented the αBB algorithm for globally optimizing nonconvex MINLPs that allows handling of nonlinear functions with no special structures. A finitely convergent decomposition algorithm based on Outer-Approximation that relies on underestimators has been proposed by Kesavan et al. [25] for the same purpose. Bergamini et al. [5] have presented a global optimization algorithm for solving Generalized Disjunctive Programming (GDP) problems where spatial branch and bound is avoided by using successive piecewise linear approximations for the nonconvex terms. Global optimization of dynamic systems, which involve a set of first order differential equations in the constraint set, has been dealt with by Papamichail and Adjiman [34] and by Chachuat et al. [10]. Finally, a stochastic branch and bound method that uses stochastic upper and lower bounds has been developed by Norkin and Pflug [32] for solving stochastic global optimization problems. A recent paper by Floudas et al. [16] reviews recent advances in deterministic global optimization for NLPs and MINLPs among other classes of mathematical problems. It is worth mentioning that solving nonconvex NLPs and MINLPs to global optimality are NP-hard problems [41]. Therefore, the major challenge lies in developing tight bounds and relaxations that will allow the solutions of these problems in reasonable computational times.

An important type of a large-scale problem is one where a number of nonconvex models are combined into a single model. In particular, problems with decomposable structures arise in *two-stage stochastic programming problems* [7] for optimization under uncertainty. The uncertain parameters in the stochastic programming model are often assumed to obey a discrete distribution so that the problem can be equivalently formulated as a large-scale deterministic multiscenario mathematical model. Here, the *first stage decision variables* (to be decided prior to the appearance of the uncertainty) link together the *second stage variables* or *recourse variables*, which are decided upon after the uncertainty has been revealed. Other examples of problems with a decomposable structure are design of engineering systems and planning and scheduling problems [6]. We focus on such decomposable problems (model structure given in Sect. 2) and propose a global optimization algorithm for solving

these problems since they tend to be very expensive to solve to global optimality given that given existing global optimization algorithms scale poorly with problem size. The proposed algorithm is aimed at producing tight lower bounds and uses a branch-and-cut framework involving cuts that are derived from Lagrangean decomposition where the smaller decomposed sub-problems are solved to global optimality.

Lagrangean relaxation techniques have been used by Takriti et al. [40] and Carøe and Schultz [9] to solve stochastic integer programs. Furthermore, decomposition techniques based on Lagrangean relaxation have been used to optimize large models appearing in planning and scheduling applications. For instance, a midterm planning problem has been solved by Gupta and Maranas [19] using a hierarchical Lagrangean relaxation approach. A Lagrangean based decomposition algorithm has been presented by van den Heever et al. [43] to optimize models pertaining to the long-term design and planning of offshore hydrocarbon field infrastructures. Jackson and Grossmann [21] have developed spatial and temporal decomposition schemes to find good local solutions to multisite production planning models. Another notable instance where the Lagrangean based approach has been used is the pooling problem where Adhya et al. [1] have developed a special global optimization method. In this work they use a branch and bound framework where they use Lagrangean relaxation based lower bounds, and prove that the lower bounds they find are at least as strong as those obtained by solving a linear relaxation with convex estimators [30] for their pooling problem. Kuno and Utsunomiya [27] have proposed a branch and bound algorithm where Lagrangean relaxation has been used to obtain bounds on the global optimum for solving production-transportation problems that have concave cost functions in the objective. In most of the work done previously on stochastic programming and planning and scheduling involving nonconvexities, the emphasis has not been on global optimization since it is very hard to globally optimize such large models. Also, in previous literature, there does not exist a generic algorithm for globally optimizing such a class of decomposable problems that involve nonlinearities and discrete variables.

In this work, we first provide a generic formulation for a class of problems with decomposable structures that include binary variables and nonconvex nonlinear terms. We then propose a spatial branch-and-cut algorithm for globally optimizing such large-scale problems. The proposed method includes cuts derived from Lagrangean decomposition to generate tight relaxations and to find good feasible solutions. Numerical examples are presented to illustrate the performance of the proposed algorithm in solving such decomposable models.

The remainder of the paper is organized as follows. Section 2 gives the problem formulation for the class of decomposable models considered. The technique for generating bound strengthening cuts to be included in the convex relaxation of the original nonconvex model, along with the method for generating feasible solutions and the proposed algorithm are given in Sect. 3. Section 4 presents the examples on which the algorithm was applied, and finally, conclusions are given in Sect. 5.

2 Problem description

2.1 Mathematical model

The class of decomposable mixed-integer nonlinear programming problems considered in this paper can be described by the following formulation:

$$
\begin{aligned}
\min \;\; & z = s(x, y) + \sum_{n=1}^{N} r_n(u_n, v_n) \\
s.t. \;\; & h_n(u_n, v_n) = 0 && n = 1, \ldots, N \\
& g_n(u_n, v_n) \leq 0 && n = 1, \ldots, N \\
& h'_n(x, y, u_n, v_n) = 0 && n = 1, \ldots, N \\
& g'_n(x, y, u_n, v_n) \leq 0 && n = 1, \ldots, N \\
& x^L \leq x \leq x^U \\
& y \in \{0, 1\}^J \\
& u_n^L \leq u_n \leq u_n^U \quad n = 1, \ldots, N \\
& v_n \in \{0, 1\}^{m_{v_n}} \quad n = 1, \ldots, N \\
& x \in R^I, \; u_n \in R^{m_{u_n}}
\end{aligned}
\tag{P}
$$

where the functions $h_n : R^{m_{u_n}+m_{v_n}} \rightarrow R^{q_{h_n}}$, $g_n : R^{m_{u_n}+m_{v_n}} \rightarrow R^{q_{g_n}}$, $h'_n : R^{I+J+m_{u_n}+m_{v_n}} \rightarrow R^{q_{h'_n}}$ and $g'_n : R^{I+J+m_{u_n}+m_{v_n}} \rightarrow R^{q_{g'_n}}$ may be convex or nonconvex. The vectors $x = [x_i] \quad i = 1, \ldots, I$ and $y = [y_j] \quad j = 1, \ldots, J$ correspond to the vectors of the continuous and binary linking variables, respectively, while the sets of vectors $\{u_n\}_{n=1,\ldots,N}$ and $\{v_n\}_{n=1,\ldots,N}$ are the sets of the non-linking continuous and binary variables, respectively. The constraints $h_n(.) = 0$ and $g_n(.) \leq 0$ are the set of constraints which are particular to a sub-model n, while $h'_n(.) = 0$ and $g'_n(.) \leq 0$ are the linking constraints that couple the various sub-models, and are written in terms of both the non-linking and the linking variables. It should be noted that some of the linking constraints may not involve the non-linking variables. For the sake of reformulation, such inequalities are assumed to be written in duplicate in each of $g'_n(x, y, u_n, v_n) \leq 0 \quad n = 1, \ldots, N$, while equalities with no non-linking variables are assumed to appear in duplicate in each of $h'_n(x, y, u_n, v_n) = 0 \quad n = 1, \ldots, N$. The objective function consists of two parts, where the linking and non-linking variables contribute to the objective function through two separate functions. All the continuous and discrete variables in the model are assumed to lie within pre-specified bounds. The terms u_n^L and u_n^U are the lower and upper bound vectors, respectively, for the vector variable u_n. Similarly, x^L and x^U correspond to the lower and upper bounds, respectively, on the variable x. It is assumed that convex under- and/ or over-estimators can be constructed for all the nonconvex terms present in the functions $g_n(.) \leq 0$, $g'_n(.) \leq 0$, $h_n(.) = 0$ and $h'_n(.) = 0$ (see [38]). The feasible region of problem (P) is denoted by D^P. This kind of problem formulation is quite generic and so the algorithm developed for solving (P) can be applied to special cases of (P), which would include nonconvex NLPs. This model can be brought to a form that can be decomposed into at most N independent sub-models. In order to bring the model to a decomposable form, we first reformulate the model as shown in the following section.

2.2 Model reformulation

We create identical copies of the linking variables, x and y, that are given by the duplicate variables $\{x^1, x^2, \ldots, x^N\}$ and $\{y^1, y^2, \ldots, y^N\}$. The linking variables in (P) are replaced with these newly created duplicate variables and the following coupling constraints, which state that the linking variables are the same across all the sub-problems, are introduced into (P):

$$x^1 = x^2 = \cdots = x^N \tag{1}$$

$$y^1 = y^2 = \cdots = y^N \tag{2}$$

We then re-write model (P) as:

$$
\begin{aligned}
\min \quad & z^{RP} = \sum_{n=1}^{N} w_n s(x^n, y^n) + \sum_{n=1}^{N} r_n(u_n, v_n) \\
s.t. \quad & h_n(u_n, v_n) = 0 && n = 1, \ldots, N \\
& g_n(u_n, v_n) \leq 0 && n = 1, \ldots, N \\
& h'_n(x^n, y^n, u_n, v_n) = 0 && n = 1, \ldots, N \\
& g'_n(x^n, y^n, u_n, v_n) \leq 0 && n = 1, \ldots, N \\
& x^n - x^{n+1} = 0 && n = 1, \ldots, N-1 \\
& y^n - y^{n+1} = 0 && n = 1, \ldots, N-1 \\
& x^L \leq x^n \leq x^U && n = 1, \ldots, N \\
& y^n \in \{0,1\}^J && n = 1, \ldots, N \\
& u_n^L \leq u_n \leq u_n^U && n = 1, \ldots, N \\
& v_n \in \{0,1\}^{m_{v_n}} && n = 1, \ldots, N \\
& x^n \in R^I, \quad u_n \in R^{m_{u_n}}
\end{aligned}
\tag{RP}
$$

where $w_n$is a parameter that has to be set so that $\sum_{n=1}^{N} w_n = 1 \qquad 0 \leq w_n \leq 1$

It is to be noted that there are multiple ways to represent the coupling constraints (1) and (2) and we have chosen to express them as Eqs. 3 and 4 in the model (RP).

$$x^n - x^{n+1} = 0 \qquad n = 1, \ldots, N-1 \tag{3}$$

$$y^n - y^{n+1} = 0 \qquad n = 1, \ldots, N-1 \tag{4}$$

The coupling constraints in (RP) are the new linking constraints. The bounds of the non-linking variables in (RP) are the same as those in the model (P), while the bounds of the duplicate variables introduced in model (RP) are the same as those of the corresponding linking variables in (P). Finally, the reason for reformulating (P) to obtain (RP) is that it helps in getting stronger relaxations, after it has been decomposed into independent sub-models.

The objective of this work is to globally optimize model (P) (or equivalently (RP)). Generally, some form of branch and bound search is used to carry out the global optimization of a nonconvex mathematical model. In such branch and bound procedures, bounds on the global optimum are obtained by solving relaxations that are constructed by convexifying the nonconvex terms in the model (e.g. see [41]). These relaxations are often very weak, which in turn lead to weak bounds, and hence to poor performance of the algorithm. The fact that the model (P) has a decomposable structure can be exploited to derive tight bounds on the global optimum of (P). The basic idea in this paper is to propose a cutting plane technique based on Lagrangean decomposition to produce tight relaxations for problem (P) for its global optimization. In particular, we propose a branch-and-cut framework for solving problem (P) to global optimality wherein we solve a convex relaxation of the original nonconvex model with cuts added to it in order to obtain a strong lower bound on the solution at every node of the search tree. These relaxation strengthening cuts are generated using the solution of the sub-problems that are obtained by decomposing (P) using Lagrangean decomposition. Feasible solutions, which are upper bounds on the solution of problem (P), are obtained using a heuristic at every node of the search tree. These lower and upper bounds are then converged to lie within a specified tolerance in the branch-and-cut algorithm.

3 Solution methodology

We propose a branch-and-cut framework in which lower and upper bounds on the global optimum are converged to within a specified tolerance. A lower bound at a particular node in the tree is obtained by solving a convex relaxation of (P) with some cuts added to it. An upper bound at a given node is found by locally optimizing (P) over the domain space of the node.

3.1 Generation of tight relaxations

A convex relaxation of the nonconvex MINLP model (P) can be obtained by replacing the nonconvex terms by convex under- and over-estimators appearing in the model (P). Various techniques for constructing such estimators have been proposed by Maranas and Floudas [29], Quesada and Grossmann [36], Ryoo and Sahinidis [37], Tawarmalani and Sahinidis [41] and Zamora and Grossmann [45], among other authors. For concave separable and bilinear functions, linear under- and over-estimators that correspond to convex envelopes are used for convexifying the nonconvex terms in the model (P) yielding a Mixed Integer Linear Programming (MILP) relaxation. A derivation of linear estimators for factorable functions is given in McCormick [30]. These convex relaxations can be weak for large-scale nonconvex problems like (P), particularly when the bounds on the variables present in the model are far apart. Due to this reason, the use of these relaxations in a branch and bound algorithm often leads to inefficient performance of the algorithm. To speed up the search for the global optimum, we need to construct stronger relaxations for the branch and bound procedure. In order to do this, we propose to construct a convex relaxation of (P) with certain cuts added to it. The solution of this convex relaxation provides a tight lower bound on the solution at every node of the branch and bound search tree. These cuts are derived based on a decomposition of (P) that is described in the following section.

3.1.1 Conventional Lagrangean decomposition

The most basic way of decomposing the model (RP) is to remove the coupling constraints from the constraint set. We use the concept of Lagrangean decomposition [18] to decompose the model (RP).[1] In this approach, the coupling constraints are dualized. That is, they are multiplied by fixed values of Lagrange multipliers, and are transferred to the objective function to give a relaxation of model (RP), denoted by (LRP), which can be decomposed into N independent sub-models.

$$\min \; z^{LRP} = \sum_{n=1}^{N} w_n s(x^n, y^n) \; + \; \sum_{n=1}^{N} r_n(u_n, v_n) + \sum_{n=1}^{N-1} (\bar{\lambda}_n^x)^T (x^n - x^{n+1}) + \sum_{n=1}^{N-1} (\bar{\lambda}_n^y)^T (y^n - y^{n+1})$$

[1] The similar idea of "variable splitting" has been introduced by Jornsten et al. [22].

$$
\begin{array}{lll}
s.t. & h_n(u_n, v_n) = 0 & n = 1, \ldots, N \\
& g_n(u_n, v_n) \le 0 & n = 1, \ldots, N \\
& h'_n(x^n, y^n, u_n, v_n) = 0 & n = 1, \ldots, N \\
& g'_n(x^n, y^n, u_n, v_n) \le 0 & n = 1, \ldots, N \\
& x^L \le x^n \le x^U & n = 1, \ldots, N \\
& y^n \in \{0,1\}^J & n = 1, \ldots, N \\
& u_n^L \le u_n \le u_n^U & n = 1, \ldots, N \\
& v_n \in \{0,1\}^{m_{v_n}} & n = 1, \ldots, N \\
& x^n \in R^I,\ u_n \in R^{m_{u_n}} &
\end{array}
\tag{LRP}
$$

where the vectors of Lagrange multipliers are defined as follows: $\overline{\lambda}_n^x = \left[\overline{\lambda}_n^{x_1}\ \overline{\lambda}_n^{x_2}\ \ldots\ \overline{\lambda}_n^{x_I}\right]^T$ $n = 1, \ldots, N-1$, and $\overline{\lambda}_n^y = \left[\overline{\lambda}_n^{y_1}\ \overline{\lambda}_n^{y_2}\ \ldots\ \overline{\lambda}_n^{y_J}\right]^T \quad n = 1, \ldots, N-1$.

In this work, we analyze the problem of decomposition, generating relaxations and feasible solutions and updating multipliers based on the fact that (RP) is decomposed into N sub-problems.

We now decompose (LRP) into the following sub-problems ($\mathrm{SP_n}$), $n = 1, \ldots, N$,

$$
\left.
\begin{array}{ll}
\min & z_n = w_n s(x^n, y^n) + r_n(u_n, v_n) + (\overline{\lambda}_n^x - \overline{\lambda}_{n-1}^x)^T (x^n) \\
& \quad + (\overline{\lambda}_n^y - \overline{\lambda}_{n-1}^y)^T (y^n) \\
s.t. & h_n(u_n, v_n) = 0 \\
& g_n(u_n, v_n) \le 0 \\
& h'_n(x^n, y^n, u_n, v_n) = 0 \\
& g'_n(x^n, y^n, u_n, v_n) \le 0 \\
& x^L \le x^n \le x^U \\
& y^n \in \{0,1\}^J \\
& u_n^L \le u_n \le u_n^U \\
& v_n \in \{0,1\}^{m_{v_n}} \\
& x^n \in R^I,\ u_n \in R^{m_{u_n}}
\end{array}
\right\} \quad n = 1, \ldots, N
\tag{$\mathrm{SP_n}$}
$$

where $\overline{\lambda}_0^x = 0$, $\overline{\lambda}_0^y = 0$, $\overline{\lambda}_N^x = 0$and $\overline{\lambda}_N^y = 0$.

Each of these sub-models has $(I + J + m_{u_n} + m_{v_n})$ variables and $(q_{h_n} + q_{g_n} + q_{h'_n} + q_{g'_n})$ constraints, while the original model (P) has $(I + J + \sum_{n=1}^N m_{u_n} + \sum_{n=1}^N m_{v_n})$ variables and $(\sum_{n=1}^N q_{h_n} + \sum_{n=1}^N q_{g_n} + \sum_{n=1}^N q_{h'_n} + \sum_{n=1}^N q_{g'_n})$ constraints, and hence each of these smaller sub-models can be assumed to be easier to solve than the full space model (P). We globally minimize each of these sub-problems to obtain a set of solutions $z_n^*(n = 1, \ldots, N)$. In conventional Lagrangean decomposition, the sum $\sum_{n=1}^N z_n^* = z^{LB}$ yields a valid lower bound on the global optimum of (P) over a particular region of space. Such a technique has been used by Carøe and Schultz [9] for MILPs and by Westerberg and Shah [44] for solving nonconvex NLPs. It should be mentioned here that the tightest possible lower bound over a particular region is obtained from the solution of the Lagrangean dual (involving the set of Lagrange multipliers $\overline{\lambda}$), which is given by:

$$
z^D = \max_{\overline{\lambda}} z^{LB} \tag{D}
$$

Since this dual problem is in itself very hard to solve, we use a heuristic method given in Fisher [12,13] where we iterate with different values of the Lagrange multipliers to generate

valid lower bounds. This author also provides other techniques based on sub-gradient optimization to solve the dual problem. Furthermore, a code has been developed by Kiwiel [26] for solving this problem with such a method, but it is not widely available. The method used in our work to iteratively update the Lagrange multipliers is discussed in the appendix.

Remark In practice we are able to find the global optimum of nonconvex MINLPs with ε-tolerance for the gap between the upper and lower bounds, and so we use z_n^{L*} in our computation instead of the exact global optimum z_n^*, where z_n^{L*} is the highest valued lower bound on the global optimum of sub-problem (SP_n).

3.1.2 Optimality based cutting planes

We propose to use the decomposition of (LRP) discussed above to derive bound strengthening cuts. These cuts are written in terms of the variables appearing in the objective function of (P) and the Lagrange multipliers. Using the globally optimal solutions of the sub-problems obtained by decomposing (LRP) we can generate valid cuts in the space of the original linking and non-linking variables, which are given below in Eq. ($\mathrm{C_n}$), $n = 1, \ldots, N$. A particular cut ($\mathrm{C_n}$) is obtained by replacing the duplicate variables in the objective function of the sub-problem ($\mathrm{SP_n}$) with the original linking variables and enforcing the condition that the resulting expression has to be greater than or equal to the global optimum (z_n^*) of ($\mathrm{SP_n}$).

$$z_n^* \leq w_n s(x, y) + r_n(u_n, v_n) + (\bar{\lambda}_n^x - \bar{\lambda}_{n-1}^x)^T(x) + (\bar{\lambda}_n^y - \bar{\lambda}_{n-1}^y)^T(y) \qquad (\mathrm{C_n})$$

Theorem 1 *The cuts* (C_n)*, n = 1,…,N are valid, and do not cut off any portion of the feasible region of (RP) (or equivalently* (*P*)).

Proof A cut ($\mathrm{C_n}$) can be written as follows in the space of the duplicate variables appearing in model (RP):

$$z_n^* \leq w_n s(x^n, y^n) + r_n(u_n, v_n) + (\bar{\lambda}_n^x - \bar{\lambda}_{n-1}^x)^T(x^n) + (\bar{\lambda}_n^y - \bar{\lambda}_{n-1}^y)^T(y^n)$$

Assume that a cut ($\mathrm{C_n}$) chops off a part of the mixed-integer feasible region of (RP). This implies that there exists a mixed-integer point (x^{n*}, y^{n*}, u_n^*, v_n^*) for which that particular cut ($\mathrm{C_n}$) is violated. Written in terms of the duplicate variables, the violated cut is as follows:

$$z_n^* > w_n s(x^{n*}, y^{n*}) + r_n(u_n^*, v_n^*) + (\bar{\lambda}_n^x - \bar{\lambda}_{n-1}^x)^T(x^{n*}) + (\bar{\lambda}_n^y - \bar{\lambda}_{n-1}^y)^T(y^{n*}) \qquad (\mathrm{VC})$$

This means that the feasible region of the problem (RP) contains a point (x^{n*}, y^{n*}, u_n^*, v_n^*), where the right-hand side of the violated cut (VC) takes a value of z_n', which is less than z_n^* (which is the globally optimal solution of sub-problem ($\mathrm{SP_n}$)). This is not possible, since the feasible region of the sub-problem ($\mathrm{SP_n}$) is relaxed and therefore larger than the feasible region of (RP), and the global optimum of ($\mathrm{SP_n}$) has to be less or equal to the value taken by the expression on the right side of cut ($\mathrm{C_n}$) in the problem (RP). Hence, z_n' has to be greater than z_n^*, which contradicts the construction (VC) and therefore none of the constraints ($\mathrm{C_n}$), $n = 1, \ldots, N$ are violated, and thus all them are valid. □

In practice, z_n^* is replaced by z_n^{L*} in ($\mathrm{C_n}$), $n = 1, \ldots, N$. These cuts are then added to the model (P). Futhermore, the Lagrange multipliers can be updated in order to derive additional cuts to add to the original problem (P). This procedure of updating the multipliers and adding cuts can be performed an arbitrary number of times. The problem (P) with the cuts added, is denoted as (P′), where for the sake of simplicity in the presentation, only one set of cuts

derived from fixed Lagrange multipliers is shown, although multiple sets of cuts can easily be generated and used.

$$
\begin{aligned}
\min\ & z^{P'} = s(x,y) + \sum_{n=1}^{N} r_n(u_n,v_n) \\
s.t.\ & h_n(u_n,v_n) = 0 && n=1,\ldots,N \\
& g_n(u_n,v_n) \le 0 && n=1,\ldots,N \\
& h'_n(x,y,u_n,v_n) = 0 && n=1,\ldots,N \\
& g'_n(x,y,u_n,v_n) \le 0 && n=1,\ldots,N \\
& z_n^* \le w_n s(x,y) + r_n(u_n,v_n) + (\bar{\lambda}_n^x - \bar{\lambda}_{n-1}^x)^T(x) \\
& \quad +(\bar{\lambda}_n^y - \bar{\lambda}_{n-1}^y)^T(y) \quad n=1,\ldots,N \\
& x^L \le x \le x^U \\
& y \in \{0,1\}^J \\
& u_n^L \le u_n \le u_n^U \quad n=1,\ldots,N \\
& v_n \in \{0,1\}^{m_{v_n}} \quad n=1,\ldots,N \\
& x \in R^I,\ u_n \in R^{m_{u_n}}
\end{aligned}
\tag{P'}
$$

When problem (P′) is convexified by replacing the nonconvex nonlinear terms by valid under- and over-estimators, the resulting relaxation denoted by model (R) is then solved to predict a valid lower bound on the solution of problem (P). For specific nonconvex terms, special convex estimators can be selected (see [41]). The relaxation (R) is as follows,

$$
\begin{aligned}
\min\ & z^R = \bar{s}(x,y) + \sum_{n=1}^{N} \bar{r}_n(u_n,v_n) \\
s.t.\ & \bar{h}_n(u_n,v_n) \le 0 && n=1,\ldots,N \\
& \bar{g}_n(u_n,v_n) \le 0 && n=1,\ldots,N \\
& \bar{h'}_n(x,y,u_n,v_n) \le 0 && n=1,\ldots,N \\
& \bar{g'}_n(x,y,u_n,v_n) \le 0 && n=1,\ldots,N \\
& \tilde{s}(x,y) \le 0 \\
& \tilde{r}_n(u_n,v_n) \le 0 && n=1,\ldots,N \\
& z_n^* \le w_n \bar{s}(x,y) + \bar{r}_n(u_n,v_n) + (\bar{\lambda}_n^x - \bar{\lambda}_{n-1}^x)^T(x) \\
& \quad +(\bar{\lambda}_n^y - \bar{\lambda}_{n-1}^y)^T(y) \quad n=1,\ldots,N \\
& x^L \le x \le x^U \\
& y \in \{0,1\}^J \\
& u_n^L \le u_n \le u_n^U \quad n=1,\ldots,N \\
& v_n \in \{0,1\}^{m_{v_n}} \quad n=1,\ldots,N \\
& x \in R^I, u_n \in R^{m_{u_n}}
\end{aligned}
\tag{R}
$$

In model (R), $\bar{s}(.)$ and $\bar{r}_n(.)$ are the convex reformulations of the functions $s(.)$ and $r_n(.)$, respectively. The functions $\bar{h}_n(.) \le 0$, $\bar{g}_n(.) \le 0$, $\bar{h'}_n(.) \le 0$ and $\bar{g'}_n(.) \le 0$ include the convexified forms of the functions $h_n(.) = 0$, $g_n(.) \le 0$, $h'_n(.) = 0$ and $g'_n(.) \le 0$, respectively and also include the convex estimator equations for the nonconvex terms appearing these functions. $\tilde{s}(.)$ and $\tilde{r}_n(.)$ include the under- and over-estimator equations for the nonconvex terms in $s(.)$ and $r_n(.)$, respectively. The convex components of the original set of constraints $h_n(.) = 0$, $g_n(.) \le 0$, $h'_n(.) = 0$ and $g'_n(.) \le 0$, and convex components of the functions $s(.)$ and $r_n(.)$ in the objective are left unchanged.

Proposition 1 *The lower bound obtained by solving (R) is at least as strong as the one obtained by solving a convex relaxation of (P) obtained by convexifying the nonconvex terms*

Proof Let the feasible region of the convex relaxation of (P), which is obtained by convexifying the nonconvex terms in (P), be denoted by D^{CR}.

$$\begin{aligned} D^{CR} = \{(x, y, u_n, v_n) :\ & x \in R^I,\ x^L \le x \le x^U,\ y \in \{0,1\}^J,\ u_n \in R^{m_{u_n}}, \\ & u_n^L \le u_n \le u_n^U,\ v_n \in \{0,1\}^{m_{v_n}},\ \bar{h}_n(u_n, v_n) = 0 \quad \forall n,\ \bar{g}_n(u_n, v_n) \le 0 \\ & \forall n,\ \bar{h'}_n(x, y, u_n, v_n) = 0 \quad \forall n,\ \bar{g'}_n(x, y, u_n, v_n) \le 0 \quad \tilde{s}(x, y) \le 0, \\ & \tilde{r}_n(u_n, v_n) \le 0 \quad \forall n\} \end{aligned}$$

The feasible region of the relaxation (R) (given by D^{R}), which is obtained by adding the cuts,

$$z_n^* \le w_n \bar{s}(x, y) + \bar{r}_n(u_n, v_n) + (\bar{\lambda}_n^x - \bar{\lambda}_{n-1}^x)^T(x) + (\bar{\lambda}_n^y - \bar{\lambda}_{n-1}^y)^T(y) \quad n = 1, \ldots, N$$

to D^{CR}, is more restricted than D^{CR} since $D^R \subseteq D^{CR}$. Therefore, it trivially follows that the solution of (R) yields at least as tight a lower bound as would be obtained by solving the convex relaxation of (P), given by (CR). □

Proposition 2 *The lower bound obtained by solving (R) is at least as strong as the lower bound obtained from Lagrangean decomposition when all N sub-models are solved to global optimality.*

Proof Taking a summation over all n of the derived cuts ($\mathrm{C_n}$), $n = 1, \ldots, N$,we get,

$$\sum_{n=1}^{N} z_n^* \le \sum_{n=1}^{N} [w_n \bar{s}(x, y) + \bar{r}_n(u_n, v_n) + (\bar{\lambda}_n^x - \bar{\lambda}_{n-1}^x)^T(x) + (\bar{\lambda}_n^y - \bar{\lambda}_{n-1}^y)^T(y)]$$

$$\sum_{n=1}^{N} z_n^* \le \bar{s}(x, y) + \sum_{n=1}^{N} \bar{r}_n(u_n, v_n) \qquad \text{(LC)}$$

From the left-hand side of Eq. (LC), we have $\sum_{n=1}^{N} z_n^* = z^{LB}$, obtained from a conventional Lagrangean decomposition as mentioned in Sect. 3.1.1. The right-hand side of (LC) is the objective function of the relaxation (R). Hence, Eq. (LC) can be written as,

$$z^{LB} \le z^R$$

From which it follows that the optimal objective of (R) will always be greater than or equal to the solution obtained from a conventional Lagrangean decomposition when all the N sub-models are solved to global optimality. □

We often obtain stronger lower bounds by solving (R) than from conventional Lagrangean decomposition or by solving a convex relaxation of (P) because the use of the Lagrangean cuts ($\mathrm{C_n}$), $n = 1,\ldots,N$ with the constraints pertaining to the different sub-models in (R) helps in tightening the feasible space of (R).

We should note here that the lower bound obtained by adding convexified bound strengthening cuts to the convex relaxation of (P) is computationally more expensive than Lagrangean decomposition since the relaxation (R) needs to be solved in addition to solving the sub-problems to global optimality at each node of the search tree. However, the tighter lower bounds obtained using this method may greatly reduce the number of nodes in the branch and bound enumeration leading to an overall acceleration of the search.

Remarks

1. In the proposed cut generation technique it is not necessary to solve N global optimization problems to obtain a valid lower bound, as is required in a pure Lagrangean decomposition algorithm. An arbitrary number of cuts can be generated and included in the relaxation to get strong lower bounds.
2. An infinite number of cuts derived from all sets of values of the Lagrange multipliers would yield the tightest possible relaxation over a sub-region of the feasible space of (P).
3. We can take advantage of the different ways to decompose model (RP) in order to construct tight relaxations. A subset of the coupling constraints can be relaxed, leaving the remaining equality constraints in the constraint set. In this way, the model (RP) can be decomposed into fewer than N sub-models. This is analogous to combining some of the N sub-models in (RP) into a single model and letting the remaining sub-models remain independent. The key point here is that relaxing fewer coupling constraints results in tighter relaxations, as then fewer constraints are violated. This is turn leads to stronger cuts. Furthermore, relaxing different subsets of the coupling constraints would lead to different decomposed sub-models obtained from (RP), that in turn would translate into different cuts.
4. *Lower bounding problem in reduced space*: The relaxation (R) that is constructed by adding cuts to (P) and convexifying the resulting model may become computationally expensive to solve. In order to overcome this problem, we can consider another approach based on the concept of conventional Lagrangean decomposition for generating a lower bound on the global optimum of (P) over a domain of space. We select any 'r' sub-models from the model (RP) and combine them into a single problem (SP') and separate the remaining 'n-r' sub-models into independent sub-problems. This is equivalent to not relaxing 'r-1' coupling constraints pertaining to the chosen 'r' sub-models in the model (RP). This means that model (RP) can be now decomposed into 'n − r + 1' sub-models, where we have 'n-r' independent sub-models and another model (SP'). Based on the global optima of the selected 'r' sub-models, we generate 'r' cuts and incorporate them into model (SP') to get model (rSP'), which is solved to global optimality to obtain a global optimum $z^{rSP'}$ using the proposed algorithm in this work. Theoretically, the sum of $z^{rSP'}$ and the global optima of the remaining 'n-r' sub-problems is a lower bound on the global optimum of (P) over the given domain. In this way, we ensure that the size of the largest relaxation that has to be solved does not exceed the combined size of the 'r' chosen sub-models and we avoid solving the full size relaxation that includes N sub-models. The value of the parameter 'r' can be specified by the user to control the size of the problem (rSP').

3.2 Generation of good feasible solutions

It is very important to obtain good feasible solutions that are close to the global solution, early on in the search tree. They also help in generating additional cuts, which leads to stronger relaxations, and in turn leads to improved computational efficiency. We fix the values of the discrete linking variables in (P) to the optimal values obtained from solution of the relaxation (R) and locally optimize the resulting nonconvex NLP using the optimal solution of the continuous variables in (R) as a starting point. If this NLP is globally optimized, we can derive an integer cut involving the binary variables in (R) (see [4]), to be added to the relaxation (R) in the nodes beneath the current node in the branch and bound tree. This is to preclude the occurrence of current combination of the discrete variables in forthcoming nodes. It is to be noted that if the global optimization of the NLP (obtained by fixing the

integer variables in the original MINLP) yields an infeasible solution, we can discard the particular integer solution.

3.3 Global optimization algorithm

We propose a deterministic branch and bound algorithm that makes use of the relaxations and feasible solutions, described in Sects. 3.1 and 3.2, respectively, to solve (P) to global optimality. F denotes a list of problems given by F^l, each problem defined over a certain region of space Ω^l (which is the domain of a node l in the tree). The steps involved in the proposed algorithm are summarized as follows:

Step 1: Initialization—The bounds on the variables in the model are determined by using the numerical data given for a particular problem. Some variables can also be fixed to certain values using these data. The hyper-rectangle formed by the initial bounds of all the variables in (P) is defined by Ω^0. This also is the domain space of the problem (P) at the root node of the tree. The bounds of the duplicate variables in the sub-problems are the same as that of the corresponding linking variables in the model (P). Further in this step, the model (P) is locally optimized to obtain an initial overall upper bound (OUB) on the objective function.
Sequence of steps:

(a) Set $F = F^0(\Omega^0)$
(b) Set $z^U = OUB$

Step 2: Bound contraction (optional)—Bound contraction is an important aspect of global optimization. It serves two purposes. One is that it narrows the search region for finding the global optimum, and the other is that tighter relaxations can be constructed over contracted bounds leading to the acceleration of the convergence of the search algorithm. The upper and lower bounds of the variables appearing in the nonconvex terms in model (P) can be contracted using a simplified version of the bound contraction technique by Zamora and Grossmann [45]. According to the technique mentioned in this work, we solve a set of minimization and maximization problems, which are all LPs or convex NLPs, obtained by constructing convex under- and over-estimators for the nonconvex terms and relaxing the integrality constraints on the discrete variables in model (P).
Step 3: Formulation of Lagrangean relaxation and decomposition—Formulate model (LRP) over domain Ω^l and decompose into N sub-problems (SP_1)–(SP_N).
Sequence of steps:

(a) Solve sub-problems (SP_1)–(SP_N) to global optimality using any deterministic global optimization algorithm to determine solutions $z_n^* \quad n = 1, \ldots, N$.
If the solutions obtained ($x^{n*}, y^{n*}, u_n^*, v_n^* \quad n = 1, \ldots, N$) are feasible for the model ($RP(\Omega^l)$):
if $\sum_{n=1}^{N} z_n^* < z^U$, set $z^U = \sum_{n=1}^{N} z_n^*$, delete $F^l(\Omega^l)$ from F, go to *step 6*, else, delete $F^l(\Omega^l)$ from F, go to *step 6*.
(b) If any sub-model (SP_n) is found to be infeasible, delete $F^l(\Omega^l)$ from F, go to *step 6*. The model (P) is infeasible if this occurs at the root node.
(c) Generate N cuts (C_1)–(C_N) using solutions $z_1^*, z_2^*, \ldots, z_N^*$ as described in Sect. 3.1.2.
(d) (Optional) Update Lagrange multipliers using the procedure given in the appendix and repeat *step 3*.

Step 4: Bounding of the global optimum—Add cuts to the problem (P) defined over a particular node of the tree and solve the convex relaxation of the resulting model, denoted by model (R), yielding a valid lower bound (LB) at that node of the tree.

Sequence of steps:

(a) Add cuts (C_1)–(C_N) to $F^l(\Omega^l)$ and convexify the resulting model to generate relaxation model $R^l(\Omega^l)$.
(b) Solve $R^l(\Omega^l)$ to obtain a lower bound ($z^R(\Omega^l)$) on the solution of $F^l(\Omega^l)$.

Step 5: Obtaining an upper bound (UB)—A local optimum of (P) is found by using the technique described in Sect. 3.2, and is designated an upper bound on the global optimum of (P). If this is found to be better than the current OUB, the OUB is updated to take the value of the improved upper bound, hence making the OUB, the best available feasible solution.
Sequence of steps:

(a) Solve for an upper bound of problem $F^l(\Omega^l)$ over Ω^l and denote it by $z^{UB}(\Omega^l)$.
(b) If $z^{UB}(\Omega^l) < z^U$, set $z^U = z^{UB}(\Omega^l)$.

Step 6: Termination—A node in the branch and bound tree can be fathomed if one of the following criteria is met:

(i) The lower bound found at the node exceeds the overall upper bound.
(ii) The optimality gap at the node is below a specified tolerance, ε_1. The optimality gap ($\text{gap}(\Omega^l)$) at a node is defined as:

$$\text{gap}(\Omega^l) = \begin{cases} \left|\frac{z^U - z^R(\Omega^l)}{z^U}\right| & if\, z^U \neq 0 \\ -z^R(\Omega^l) & if\, z^U = 0 \end{cases}$$

(iii) If an upper bound z^U of $-\infty$ is found, the problem is unbounded and the search is stopped.

The search is terminated when there are no open nodes remaining in the tree.
Sequence of steps:

(a) If $z^U = -\infty$, problem is unbounded, stop search
(b) Delete all sub-problems $F^l(\Omega^l)$ from F where $z^U - z^R(\Omega^l)| \leq \varepsilon_1 |z^U| \quad z^U \neq 0$
(c) If $F = \emptyset$, stop search, solution = z^U

Step 7: Branching—The domain of a particular node in the branch and bound tree corresponds to a certain region of the search space. Since there may be an optimality gap between the lower bounds and the overall upper bound for some regions of the search space, such regions for which the gap is greater than the specified tolerance are each further partitioned into disjoint sub-regions to create two new nodes in the tree, and steps 2–6 are repeated for each of these regions. We use certain heuristics similar to the ones used by Carøe and Schultz [9] for the branching rules in this work. The branching is performed on the linking variables (continuous or binary) present in the linking constraints. If the duplicate variables corresponding to a certain scalar linking variable x_i (or y_j), take the same value in the solution of all the sub-problems, that particular linking variable is not selected as the branching variable. For any other potential branching variable, the dispersion of a scalar linking variable $x_i \in x$ is defined as $\sum_{n=1}^{N} \frac{|x_i^{n*} - \overline{x_i}|}{\left|\max_n\{x_i^{n*}\} - \min_n\{x_i^{n*}\}\right|}$ where x_i^{n*} is the optimal value of the duplicate variable corresponding to x_i in the nth sub-problem (SP_n), and $\overline{x_i} = \frac{\sum_{n=1}^{N} x_i^{n*}}{N}$. The dispersion of a binary variable $y_j \in y$ is similarly defined. This dispersion is calculated from the solution of the sub-problems formed using the set of Lagrange multipliers that provides

the tightest bound on the global optimum in a conventional Lagrangean decomposition setting. The linking variable x_i (or y_j) for which this dispersion is maximum is branched on. For a continuous variable x_i, the average value $\overline{x_i}$ is taken as the branching point while for a binary linking variable y_j, two new branches corresponding to $y_j = 0$ and $y_j = 1$ are created. Theoretically, the branch and bound can be an infinite process if the branching is performed on the continuous variables, but terminates in a finite number of nodes for ε_1-convergence.
Sequence of steps:

(a) Select the last problem $F^l(\Omega^l)$ in F, and based on the solution of the sub-problems $(\mathrm{SP_n}(\Omega^l))$ $n = 1, \ldots, N$, select a branching variable x_i or y_j
(b) Partition Ω^l into two new sub-regions Ω^{l+1} and Ω^{l+2}:

If x_i is selected as the branching variable, create the new sub-regions Ω^{l+1} and Ω^{l+2} by adding the constraints $x_i \leq \bar{x}_i$ and $x_i \geq \bar{x}_i$, respectively to Ω^l. If y_j is selected as the branching variable, create the new sub-regions Ω^{l+1} and Ω^{l+2} by adding the constraints $y_j = 0$ and $y_j = 1$, respectively to Ω^l.

(c) Add $F^{l+1}(\Omega^{l+1})$ and $F^{l+2}(\Omega^{l+2})$ to F, and delete $F^l(\Omega^l)$ from F, go to *step 2*

Convergence: The convergence of the branch-and-cut algorithm is guaranteed by the fact that the search region can be partitioned further into sub-regions and the partitioning of the regions yields a sequence of non-decreasing lower bounds which converge to the global optimum [20]. The relaxations generated in the nodes while moving down the tree are tighter than those constructed in the nodes above them, since the feasible space of the problems is continuously being contracted and tighter estimators are being constructed to approximate the nonconvex functions. While branching down the tree, some stopping criteria is required if branching is carried out on the continuous variables. Hence, if the branching is performed on the continuous variables and parallel to the coordinate axes, we can stop once the l_∞-diameter of the feasible sets of the sub-problems has fallen below a certain value.

Remarks

1. The decomposable structure of the problems allows parallelization of the algorithm. Each of the sub-models $(\mathrm{SP_1})$–$(\mathrm{SP_N})$ can be solved in parallel and so the computational time for getting the lower bounds on the global solution of (P) can be reduced.
2. If the set of Lagrange multipliers in the sub-problems is kept the same in a parent node and its children nodes, then only a subset of the sub-problems need to be solved to global optimality at the children nodes. These sub-problems are selected based on the optimal values of the duplicate variables in the solution of the sub-problems at the parent node. For instance, let us select to branch on a particular linking variable x_i (or y_j) from a parent node with domain Ω^l whose children nodes have domains Ω^{l+1} and Ω^{l+2}. If the optimal value of the duplicate variable x_i^{n*} (or y_j^{n*}) obtained from the solution of a sub-model $(\mathrm{SP_n}(\Omega^l))$ $n \in \{1, 2, \ldots, N\}$ at the parent node lies within the bounds of the corresponding variable x_i^n (or y_j^n) in a child node, then that particular sub-problem $\mathrm{SP_n}$ does not need to be solved at the particular child node where this occurs. The optimal solution and the optimal objective value of these unsolved sub-problems at the child node are taken to be the same as that obtained from the solution of the corresponding sub-models at the parent node with the domain Ω^l.
3. Instead of using the above global optimization algorithm to solve (P) to global optimality, the model (P) with cuts added to it (see Sect. 3.1.2), can be globally optimized directly using commercial solvers (e.g. BARON for NLPs/ MINLPs) that are based on branch and bound.

4. For solving MILPs/convex MINLPs, we can use the branch-and-cut algorithm where adding the Lagrangean cuts to the model tightens the relaxation of the original model, and hence convergence to the optimal solution is faster.
5. The algorithm presented in this section is generic and is also valid for the case when the model (P) is decomposed into N' ($\leq N$) sub-models.

4 Results

Numerical examples have been solved using the proposed algorithm. The examples were formulated using GAMS [8] and solved on an Intel 3.2 GHz Linux machine with 1,024 MB memory. GAMS/CONOPT 3.0 and GAMS/ BARON 7.2.5 were used to solve the NLP problems, GAMS/CPLEX 9.0 was used for the LP and MILP problems, and GAMS/DICOPT and GAMS/ BARON 7.2.5 were employed for solving the MINLP problems.

4.1 Example 1: illustrative problem

We solve a small problem as a first example to demonstrate the concepts proposed in the paper. This model (EP), as shown below, has 1 binary variable, 19 continuous variables, 10 constraints and 15 nonconvex terms. The objective function is linear while the constraints involve bilinear terms.

$$
\begin{aligned}
\min z^{EP} &= 5x + 7y + 2u_{11} + 6u_{14} + 5u_{21} + 9u_{24} + 3u_{31} + 11u_{34} \\
s.t. \quad & \left.\begin{array}{l} u_{11}u_{14} - 3u_{12}u_{15} + 4u_{13}u_{16} + 4 = 0 \\ u_{11}u_{13} - 5 = 0 \\ 5u_{21}u_{24} - u_{22}u_{25} - u_{23}u_{26} + 5 = 0 \\ u_{21}u_{23} + u_{22}u_{24} - 2u_{25}u_{26} - 5 = 0 \\ 3u_{31}u_{34} + 4u_{32}u_{35} - u_{33}u_{36} + 3 = 0 \\ u_{31}u_{33} + u_{32}u_{34} - 4 = 0 \end{array}\right\} \text{Non-linking equations} \\
& \left.\begin{array}{l} x \geq u_{11} \\ x \geq u_{21} \\ x \geq u_{31} \\ 3y \leq x \leq 5y \end{array}\right\} \text{Linking constraints} \\
& 0 \leq x \leq 5 \\
& y \in \{0, 1\} \\
& 1.5 \leq u_{11} \leq 3 \qquad 0.5 \leq u_{21} \leq 4.5 \qquad 1.5 \leq u_{31} \leq 3.5 \\
& 1.3 \leq u_{12} \leq 11 \qquad 0.2 \leq u_{22} \leq 0.5 \qquad 0.5 \leq u_{32} \leq 13 \\
& 1 \leq u_{13} \leq 2.5 \qquad 0 \leq u_{23} \leq 5 \qquad 1 \leq u_{33} \leq 1.9 \\
& 2 \leq u_{14} \leq 4 \qquad 0.25 \leq u_{24} \leq 5 \qquad 0.5 \leq u_{34} \leq 2.5 \\
& 0 \leq u_{15} \leq 3 \qquad 3.2 \leq u_{25} \leq 8.7 \qquad 2 \leq u_{35} \leq 7.2 \\
& 0 \leq u_{16} \leq 10 \qquad 0.15 \leq u_{26} \leq 7.8 \qquad 1.5 \leq u_{36} \leq 9
\end{aligned}
\tag{EP}
$$

This model is a collection of 3 sub-models. The variables x and y are the continuous and binary linking variables, respectively, while the other variables in the model correspond to the three sets of continuous non-linking variables. This model is very small and is solved in a time of the order of a tenth of second using the commercial solver BARON [38], yielding the global optimum of 64.499. The optimal values of the variables are: $x = 3$, $y = 1$, $u_{11} = 2$, $u_{21} = 2.7138$, $u_{31} = 1.7266$, $u_{12} = 1.3$, $u_{22} = 0.5$, $u_{32} = 1.4388$, $u_{13} = 2.5$, $u_{23} = 5$, $u_{33} = 1.9$, $u_{14} = 2$, $u_{24} = 0.25$, $u_{34} = 0.5$, $u_{15} = 2.0513$, $u_{25} = 3.2$, $u_{35} = 2$, $u_{16} = 0$, $u_{26} = 1.3585$, $u_{36} = 9$. Due to the small size of the problem, the computational times of

solving the original problem and its various relaxations are not analyzed for this example. The model (EP) is reformulated to produce a form analogous to (RP) which is relaxed to obtain model (LR-EP):

$$
\begin{aligned}
\min\ & z^{LR-EP} = 5x^1 + 7y^1 + 2u_{11} + 6u_{14} + 5u_{21} + 9u_{24} + 3u_{31} + 11u_{34} \\
& \quad +\bar{\lambda}_1^x(x^1 - x^2) + \bar{\lambda}_2^x(x^2 - x^3) + \bar{\lambda}_1^y(y^1 - y^2) + \bar{\lambda}_2^y(y^2 - y^3) \\
s.t.\ & u_{11}u_{14} - 3u_{12}u_{15} + 4u_{13}u_{16} + 4 = 0 \\
& u_{11}u_{13} - 5 = 0 \\
& 5u_{21}u_{24} - u_{22}u_{25} - u_{23}u_{26} + 5 = 0 \\
& u_{21}u_{23} + u_{22}u_{24} - 2u_{25}u_{26} - 5 = 0 \\
& 3u_{31}u_{34} + 4u_{32}u_{35} - u_{33}u_{36} + 3 = 0 \\
& u_{31}u_{33} + u_{32}u_{34} - 4 = 0 \\
& x^1 \geq u_{11} \\
& x^2 \geq u_{21} \\
& x^3 \geq u_{31} \\
& 3y^1 \leq x^1 \leq 5y^1 \\
& 3y^2 \leq x^2 \leq 5y^2 \\
& 3y^3 \leq x^3 \leq 5y^3 \\
& 0 \leq x^1 \leq 5 \quad 0 \leq x^2 \leq 5 \quad 0 \leq x^3 \leq 5 \\
& y^1, y^2, y^3 \in \{0, 1\} \\
& 1.5 \leq u_{11} \leq 3 \quad 0.5 \leq u_{21} \leq 4.5 \quad 1.5 \leq u_{31} \leq 3.5 \\
& 1.3 \leq u_{12} \leq 11 \quad 0.2 \leq u_{22} \leq 0.5 \quad 0.5 \leq u_{32} \leq 13 \\
& 1 \leq u_{13} \leq 2.5 \quad 0 \leq u_{23} \leq 5 \quad 1 \leq u_{33} \leq 1.9 \\
& 2 \leq u_{14} \leq 4 \quad 0.25 \leq u_{24} \leq 5 \quad 0.5 \leq u_{34} \leq 2.5 \\
& 0 \leq u_{15} \leq 3 \quad 3.2 \leq u_{25} \leq 8.7 \quad 2 \leq u_{35} \leq 7.2 \\
& 0 \leq u_{16} \leq 10 \quad 0.15 \leq u_{26} \leq 7.8 \quad 1.5 \leq u_{36} \leq 9
\end{aligned}
\tag{LR-EP}
$$

Here x^1, x^2 and x^3 are the duplicate variables corresponding to the linking variable x, while y^1, y^2 and y^3 are the duplicate variables corresponding to the linking variable y. The sets of Lagrange multipliers are given by $\bar{\lambda}_1 = \begin{bmatrix} \bar{\lambda}_1^x & \bar{\lambda}_1^y \end{bmatrix}^T$, $\bar{\lambda}_2 = \begin{bmatrix} \bar{\lambda}_2^x & \bar{\lambda}_2^y \end{bmatrix}^T$ and $\bar{\lambda} = \begin{bmatrix} \bar{\lambda}_1 \\ \bar{\lambda}_2 \end{bmatrix}$. We decompose model (LR-EP) into 3 separate sub-problems (EP-S1), (EP-S2) and (EP-S3) and solve each one to global optimality to within a tolerance of 1% between the lower bounds and the global optimum, and use the best possible lower bounds so obtained to generate cutting planes as described in Sect. 3.1.2.

$$
\begin{aligned}
\min\ & z^{EP-S1} = 5x^1 + 7y^1 + 2u_{11} + 6u_{14} + \bar{\lambda}_1^x(x^1) + \bar{\lambda}_1^y(y^1) \\
s.t.\ & u_{11}u_{14} - 3u_{12}u_{15} + 4u_{13}u_{16} + 4 = 0 \\
& u_{11}u_{13} - 5 = 0 \\
& x^1 \geq u_{11} \\
& 3y^1 \leq x^1 \leq 5y^1 \\
& 0 \leq x^1 \leq 5 \\
& y^1 \in \{0, 1\} \\
& 1.5 \leq u_{11} \leq 3 \quad 1.3 \leq u_{12} \leq 11 \quad 1 \leq u_{13} \leq 2.5 \\
& 2 \leq u_{14} \leq 4 \quad 0 \leq u_{15} \leq 3 \quad 0 \leq u_{16} \leq 10
\end{aligned}
\tag{EP-S1}
$$

Table 1 Numerical results for the root node for example 1

Iteration k	Lagrange multipliers ($\bar{\lambda}$)				Bound from Lagrangean decomposition (z^{LB})
	$\bar{\lambda}_1^x$	$\bar{\lambda}_2^x$	$\bar{\lambda}_1^y$	$\bar{\lambda}_2^y$	
Iteration 1	1	0.5	0.5	1	62.358
Iteration 2	0.46475	0.5	0.5	1	63.3324

$$
\begin{aligned}
\min\ & z^{EP-S2} = 5u_{21} + 9u_{24} + (\bar{\lambda}_2^x - \bar{\lambda}_1^x)x^2 + (\bar{\lambda}_2^y - \bar{\lambda}_1^y)y^2 \\
s.t.\ & 5u_{21}u_{24} - u_{22}u_{25} - u_{23}u_{26} + 5 = 0 \\
& u_{21}u_{23} + u_{22}u_{24} - 2u_{25}u_{26} - 5 = 0 \\
& x^2 \geq u_{21} \\
& 3y^2 \leq x^2 \leq 5y^2 \\
& 0 \leq x^2 \leq 5 \\
& y^2 \in \{0, 1\} \\
& 0.5 \leq u_{21} \leq 4.5 \quad 0.2 \leq u_{22} \leq 0.5 \quad 0 \leq u_{23} \leq 5 \\
& 0.25 \leq u_{24} \leq 5 \quad 3.2 \leq u_{25} \leq 8.7 \quad 0.15 \leq u_{26} \leq 7.8
\end{aligned}
\tag{EP-S2}
$$

$$
\begin{aligned}
\min\ & z^{EP-S3} = 3u_{31} + 11u_{34} + (-\bar{\lambda}_2^x)x^3 + (-\bar{\lambda}_2^y)y^3 \\
s.t.\ & 3u_{31}u_{34} + 4u_{32}u_{35} - u_{33}u_{36} + 3 = 0 \\
& u_{31}u_{33} + u_{32}u_{34} - 4 = 0 \\
& x^3 \geq u_{31} \\
& 3y^3 \leq x^3 \leq 5y^3 \\
& 0 \leq x^3 \leq 5 \\
& y^3 \in \{0, 1\} \\
& 1.5 \leq u_{31} \leq 3.5 \quad 0.5 \leq u_{32} \leq 13 \quad 1 \leq u_{33} \leq 1.9 \\
& 0.5 \leq u_{34} \leq 2.5 \quad 2 \leq u_{35} \leq 7.2 \quad 1.5 \leq u_{36} \leq 9
\end{aligned}
\tag{EP-S3}
$$

At the root node of the branch-and-cut tree, we start with arbitrary initial values of the Lagrange multipliers (see Table 4.1) and update them using the method given in the appendix (with $z^U = 64.499$, $\alpha^1 = 0.5$, $z^{LB}(\bar{\lambda}) = 62.358$, $x^1 = 3$, $x^2 = 3$, $x^3 = 5$, $y^1 = 1$, $y^2 = 1$, $y^3 = 1$) to obtain new values of the multipliers that are used to generate new cuts. Due to the two iterations performed, we end up with 2 sets of 3 cuts each. The details of the Lagrange multipliers at the root node and the lower bounds produced by these multipliers on using these in a conventional Lagrangean decomposition method are given in Table 1.

By introducing the cutting planes into model (EP) and then convexifying the resulting model using convex under- and over-estimators for the nonconvex bilinear terms, we obtain relaxation (EPR) which is solved to obtain a lower bound of $z^{EPR} = 64.01$. Comparing this lower bound with the bounds given in Table 1, we can verify that proposition 2 (Sect. 3.1.2) holds. Also, we take the original model (EP) and construct its MILP relaxation by replacing the bilinear terms with McCormick [30] convex envelopes, and solve it to get a lower bound of 61.6289. This shows that proposition 1 also holds true. Then, taking the value of the variable y as obtained from the solution of (EPR), we fix it to the binary variable in (EP) to transform the model (EP) into a nonlinear programming model. We use the solution from (EPR) as a starting point to solve this NLP model using BARON and obtain an upper bound of 64.499, which is also the global optimum. It is to be noted that so far the analysis for this

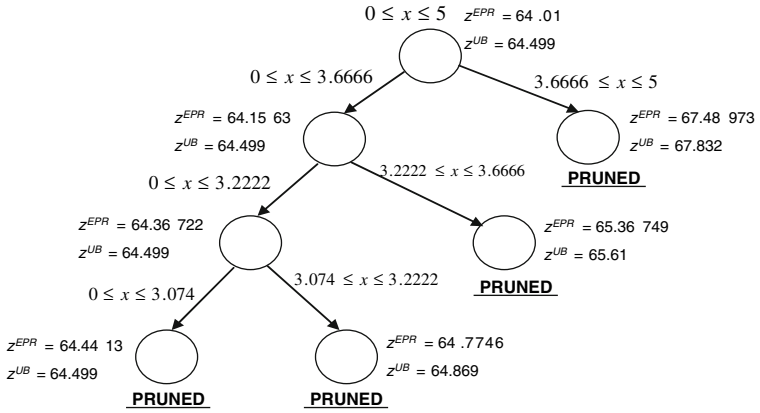

Fig. 1 Branch-and-cut tree for example 1

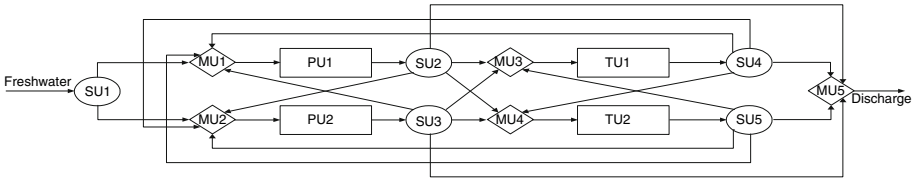

Fig. 2 Superstructure of a 2 Process unit – 2 Treatment unit integrated network

example has been done for the root node of the branch and bound tree. The lower bound obtained is found to be within 0.76% of the upper bound, which is within a tolerance of 1%. In order to reduce the optimality gap to 0.1%, we branch down the tree. The variable x is chosen to be the branching variable and the structure of the branch-and-cut tree is given in Fig. 1.

In all the nodes below the root node, the Lagrange multipliers used when solving the sub-problems were kept the same as those at the root node and were not updated. The sub-problems at all the children nodes were now solved to global optimality within a tolerance of 0.1% and the tightest possible lower bounds on the global optima of the sub-problems were used in updating selected cuts. The updated cuts involved only the Lagrange multipliers used in *Iteration 2* at the root node (see Table 1).

4.2 Example 2: integrated water network problem

As a large-scale example, we consider the synthesis of an integrated water network shown in Fig. 2, consisting of two water using process units, two water treating units and mixers and splitters, operating in 10 scenarios under uncertain contaminant loads and recoveries [23]. The design problem is an example of a two-stage stochastic programming problem, which is formulated as a deterministic multiscenario MINLP problem since the uncertain parameters are assumed to take on a finite number of realizations. In this multiscenario model, each scenario has a separate set of constraints, and there are non-anticipativity constraints that connect together the various scenarios. The set of constraints for a scenario are the mass balances for all the units in the network for that particular scenario and the constraints on the contaminant levels in certain streams in the system for that scenario. The individual

contaminant balances contain the nonconvex bilinear terms. The linking constraints link the first stage design variables, which are the maximum flows allowed in each pipe in the system with the second stage state variables, which are the flowrates in the corresponding pipes in each scenario. The objective is to synthesize a network such that the total costs of designing the network and the expected cost of operating the network optimally over all scenarios is globally minimized. The first stage design costs include the investment cost for piping and the capital cost of each water treatment unit. The operating costs of the network include the cost of obtaining freshwater for use in the process units, the cost of pumping water through the pipes and the operating costs of treating wastewater in the treatment units.

The detailed formulation of the problem is given in Karuppiah and Grossmann [23]. The model and data used for this example can be obtained from these authors. We use a scenario decomposition technique along the lines of the method described in Sect. 3.1.1 to solve the problem to global optimality. The multiscenario MINLP for this example involves 10 different scenarios, 24 binary variables, 972 continuous variables, and 1,136 constraints, and was initially attempted to be solved using GAMS/BARON 7.2.5. The termination criterion used was that the gap between the relaxation and the global optimum should be less than the specified tolerance of 1%. On directly using BARON to solve the problem, the solver could not verify global optimality of the upper bound of \$651,653.1 that it generated, in more than 10 h. The application of the proposed algorithm also yields an objective of \$651,653.1, which is the global solution to the problem. However, on using the proposed algorithm, the lower and upper bounds converge to within a tolerance of 1% at the root node of a branch and bound tree in only 19.3 CPU secs.

A description of the steps involved in applying the proposed algorithm to this example is as follows. Initially, an overall upper bound is found by solving the original nonconvex MINLP model using DICOPT, which yields a local optimum of \$685,466.5. Thereafter, in order to obtain a lower bound, we formulate the Lagrangean relaxation of the original model (model (LRP)) and decompose it into 10 different sub-problems (each sub-problem corresponding to one scenario) as described in Sect. 3. Initially, all the Lagrange multipliers are arbitrarily chosen to be 1. At the root node, each of these sub-problems is solved to global optimality using BARON with 1% tolerance for the gap between the lower bounds and the overall upper bound. Next, we use the best valid lower bounds obtained from the solution of each sub-problem (z_n^{L*}) to generate 10 valid cuts analogous to Eqs. (C_1)–(C_N). The Lagrange multipliers are then updated (with $z^U = 685{,}466.5$, $\alpha^1 = 0.5$ and $z^{LB}(\bar{\lambda}) = 644{,}856.8$) to generate 10 more valid constraints in the same fashion as above. The 20 cutting planes are added to the original nonconvex MINLP model and the resulting model is then convexified to yield a MILP relaxation, which when solved to optimality at the root node gives a solution of \$645,951.6, which corresponds to a valid lower bound on the global optimum of the problem (P). An upper bound of \$651,653.1 is found using GAMS/CONOPT 3.0 using the procedure given in Sect. 3.2 and we find that the gap between the lower and upper bounds lies within a tolerance of 1%. To further reduce the gap between the lower bounds and the global optimum to less than 0.5%, we branch down the tree on a certain continuous design variable that is chosen using the technique given in Sect. 3.3. The tree structure is shown in Fig. 3, while computational times and numerical results for the nodes in the tree are given in Table 2.

It can be seen from Table 2 that at each node of the tree the lower bounds obtained using our cutting plane approach (z^R) are tighter than the best bounds obtained from conventional Lagrangean decomposition technique (z^{LB}) using the given set of Lagrange multipliers and also tighter than the lower bounds obtained from an MILP relaxation of the original nonconvex model (z^{CR}). The total time taken in obtaining the global optimum using the proposed algorithm is 85.5 CPU secs which includes the time for getting an initial overall upper bound

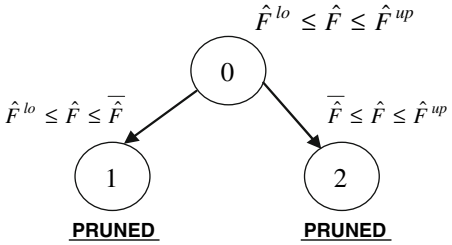

Fig. 3 Branch-and-cut tree for example 2

Table 2 Numerical results for example 2

Node #	Lower bound using proposed algorithm (z^R)	Best bound from Lagrangean decomposition (z^{LB})	Lower bound from MILP relaxation (z^{CR})	Upper Bound (z^{UB})	Total time taken at node[a] (CPU secs)
0 (root node)	645,951.6	644,856.8	610,092.6	651,653.1	19.3
1	648,566.7	647,496.2	610,115.3	672,971.8	4.1
2	648,828.6	648,073.2	610,109.0	661,439.4	61.8

[a] Total time includes time for generating cuts, solving the relaxation problem (R) and generating an upper bound

Table 3 Problem sizes for test cases

Example	Original MINLP model (P)		
	Number of binary variables	Number of continuous variables	Number of constraints
A	77	1,222	1,377
B	48	300	946
C	42	330	994
D	57	381	1,167

using DICOPT (see Table 2 for details of the computational time at each node of the search tree).

The proposed cutting plane approach was applied to the optimization of models arising in other different applications. Table 3 shows the problem sizes of the cases on which the cutting plane technique was used.

4.3 Other numerical results

Example A is a process synthesis problem similar to example 2, which is the design of an integrated water network operating under uncertainty, and is taken from Karuppiah and Grossmann [23]. Examples B, C and D are crude oil scheduling problems taken from Karuppiah et al. [24]. It is to be noted that in examples B, C and D, the idea of the cutting planes developed in this paper was used, but within an outer-approximation algorithm where again tight relaxations are critical. Table 4 provides a comparison of the solution times with and without the use of the proposed cutting planes.

Table 4 Comparison of results with and without proposed cutting planes

Example	Results without using cutting planes			Results on using cutting planes		
	Global optimal solution	Optimality gap (%)	Solution time (CPU secs)	Global optimal solution	Optimality gap (%)	Solution time (CPU secs)
A	1,369,067.5	1.57	2,077.4	1,369,067.5	1.59	193.5
B	282.19	0.37	1,953.6	282.19	0.37	827.7
C	359.48	2.27	14,485.8	359.48	2.27	6,913.9
D	383.69	0	15,875.2	383.69	0	8,928.6

It can be seen from Table 4 that there is significant reduction in solution times using the proposed cutting planes. For example A, the commercial global optimization solver BARON [38] finds the global optimum with 1.57% optimality gap after 2077.4 CPU secs of computation without the proposed cuts. On applying the proposed algorithm to the problem, we obtain the same global optimum with 1.59% optimality gap after 193.5 CPU secs. For examples B, C, and D, the outer-approximation algorithm proposed in Karuppiah et al. [24] takes nearly half the time to solve when the cutting planes were incorporated in the relaxations as compared to when the cuts were not used.

5 Conclusions

In this work we have proposed a decomposition algorithm for the global optimization of large nonconvex mixed-integer nonlinear models that have a decomposable structure. This algorithm involves decomposing the original model using Lagrangean duality and generating smaller sub-models whose global solutions are used in developing cuts which tighten the convex relaxation of the original nonconvex model. Therefore, the main idea in this work is to combine the concepts of Lagrangean decomposition and convex relaxations of nonconvex models in order to generate tight bounds on the global optima of nonconvex models. The resulting relaxations help in accelerating the search for the solution in a branch and bound setting. The algorithm was applied to several example problems to illustrate its computational performance and it was found that significant computational savings can be achieved.

Acknowledgements The authors gratefully acknowledge financial support from the National Science Foundation under Grant CTS-0521769 and from the industrial members of the Center for Advanced Process Decision-making at Carnegie Mellon University.

Appendix

Updating the Lagrange multipliers

The scheme for updating the Lagrange multipliers is quite important since the choice of these parameters has a large impact on the lower bounds predicted from solving the relaxation. We start with an arbitrary initial guess for the values of the Lagrange multipliers and use a sub-gradient method [12] to iteratively update these multipliers. The sequence of multipliers is generated as follows:

$$\begin{bmatrix} \bar{\lambda}_n^x \\ \bar{\lambda}_n^y \end{bmatrix}^{k+1} = \begin{bmatrix} \bar{\lambda}_n^x \\ \bar{\lambda}_n^y \end{bmatrix}^{k} + t^k \begin{bmatrix} (x^{n*})^k - (x^{n+1*})^k \\ (y^{n*})^k - (y^{n+1*})^k \end{bmatrix} \qquad n = 1, \ldots, N-1$$

where t^k is a scalar step size and $(x^{n*})^k$ and $(y^{n*})^k$ are the optimal values of the duplicate variables x^n and y^n, respectively, at the kth iteration, obtained from the solution of the sub-problem ($\mathrm{SP_n}$). We use the following formula to calculate the values of t^k at every iteration k:

$$t^k = \frac{\alpha^k (z^U - z^{LB}(\bar{\lambda}^k))}{\sum_{n=1}^{N-1} \left(\left\| \left(x^{n*}\right)^k - \left(x^{n+1*}\right)^k \right\|^2 + \left\| \left(y^{n*}\right)^k - \left(y^{n+1*}\right)^k \right\|^2 \right)}$$

where α^k is a scalar chosen between 0 and 2, $z^{LB}(\bar{\lambda}^k)$ is the sum of the global optima of the sub-problems ($\mathrm{SP_1}$)–($\mathrm{SP_N}$), when the multipliers are set to $\bar{\lambda}^k = \begin{bmatrix} \left(\bar{\lambda}^x\right)^k \\ \left(\bar{\lambda}^y\right)^k \end{bmatrix}$, where $\left(\bar{\lambda}^x\right)^k = \left[\left(\bar{\lambda}_1^x\right)^k \ \left(\bar{\lambda}_2^x\right)^k \ \ldots \ \left(\bar{\lambda}_{N-1}^x\right)^k \right]^T$, $\left(\bar{\lambda}^y\right)^k = \left[\left(\bar{\lambda}_1^y\right)^k \ \left(\bar{\lambda}_2^y\right)^k \ \ldots \ \left(\bar{\lambda}_{N-1}^y\right)^k \right]^T$ and z^U is the value of the best found feasible solution to (P). The value of α^k is halved at each iteration when z^{LB} fails to improve. The process of updating the multipliers and generating a pool of cuts is iteratively carried out until a pre-specified iteration limit is reached.

References

1. Adhya, N., Tawarmalani, M., Sahinidis, N.V.: A Lagrangian approach to the pooling problem. Ind. Eng. Chem. Res. **38**, 1956–1972 (1999)
2. Adjiman, C.S., Androulakis, I.P., Floudas, C.A.: Global optimization of MINLP problems in process synthesis and design. Comp. Chem. Eng. **21**, S445–S450 (1997)
3. Adjiman, C.S., Androulakis, I.P., Floudas, C.A.: Global optimization of mixed-integer nonlinear problems. Am. Inst. Chem. Eng. J. **46**(9), 1769–1797 (2000)
4. Balas, E., Jeroslow, R.: Canonical cuts on the unit hypercube. SIAM J. Appl. Math. **23**, 61–79 (1972)
5. Bergamini, M.L., Aguirre, P., Grossmann, I.E.: Logic-based outer approximation for globally optimal synthesis of process networks. Comp. Chem. Eng. **29**, 1914–1933 (2005)
6. Biegler, L.T., Grossmann, I.E.: Retrospective on optimization. Comp. Chem. Eng. **28**, 1169–1192 (2004)
7. Birge, J.R., Louveaux, F.V.: Introduction to Stochastic Programming. Springer, New York (1997)
8. Brooke, A., Kendrick, D., Meeraus, A., Raman, R.: GAMS: A user's guide, release 2.50. GAMS Development Corporation (1998)
9. Carøe, C.C., Schultz, R.: Dual decomposition in stochastic integer programming. Oper. Res. Lett. **24**, 37–45 (1999)
10. Chachuat, B., Singer, A.B., Barton, P.I.: Global mixed-integer dynamic optimization. AICHE J. **51**(8), 2235–2253 (2005)
11. Duran, M.A., Grossmann, I.E.: An outer-approximation algorithm for a class of mixed-integer nonlinear programs. Math. Program. **36**, 307–339 (1986)
12. Fisher, M.L.: The Lagrangian relaxation method for solving integer programming problems. Manage. Sci. **27**(1), 1–18 (1981)
13. Fisher, M.L.: An applications oriented guide to Lagrangian relaxation. Interfaces **15**(2), 10–21 (1985)
14. Fletcher, R., Leyffer, S.: Solving mixed integer nonlinear programs by outer approximation. Math. Program. **66**, 327–349 (1994)
15. Floudas, C.A.: Deterministic Global Optimization: Theory, Methods and Applications. Kluwer Academic Publishers, Dordrecht, The Netherlands (2000)
16. Floudas, C.A., Akrotirianakis, I.G., Caratzoulas, S., Meyer, C.A., Kallrath, J.: Global optimization in the 21st century: advances and challenges. Comp. Chem. Eng. **29**, 1185–1202 (2005)

17. Grossmann, I.E.: Review of nonlinear mixed-integer and disjunctive programming techniques. Optim. Eng. **3**, 227–252 (2002)
18. Guignard, M., Kim, S.: Lagrangean decomposition: a model yielding stronger Lagrangean bounds. Math. Program. **39**, 215–228 (1987)
19. Gupta, A., Maranas, C.D.: A hierarchical Lagrangean relaxation procedure for solving midterm planning problems. Ind. Eng. Chem. Res. **38**, 1937–1947 (1999)
20. Horst, R., Tuy, H.: Global Optimization: Deterministic Approaches, 3rd edn. Springer-Verlag, Berlin (1996)
21. Jackson, J.R., Grossmann, I.E.: Temporal decomposition scheme for nonlinear multisite production planning and distribution models. Ind. Eng. Chem. Res. **42**, 3045–3055 (2003)
22. Jornsten, K.O., Nasberg, M., Smeds, P.A.: Variable Splitting – A New Lagrangean Relaxation Approach to some Mathematical Programming Models. Department of Mathematics Report LiTH-MAT-R-85-04, Linkoping Institute of Technology, Sweden (1985)
23. Karuppiah, R., Grossmann, I.E.: Global optimization of multiscenario mixed integer nonlinear programming models arising in the synthesis of integrated water networks under uncertainty. Comp. Chem. Eng. (2007, in press)
24. Karuppiah, R., Furman, K.C., Grossmann, I.E.: Global optimization for scheduling refinery crude oil operations. Comp. Chem. Eng. (2007, submitted)
25. Kesavan, P., Allgor, R.J., Gatzke, E.P., Barton, P.I.: Outer approximation algorithms for separable nonconvex mixed-integer nonlinear programs. Math. Program. **100**, 517–535 (2004)
26. Kiwiel, K.C.: User's Guide for NOA 2.0/3.0: A Fortran Package for Convex Nondifferentiable Optimization. Polish Academy of Sciences, Systems Research Institute, Warsaw, Poland (1993/1994)
27. Kuno, T., Utsunomiya, T.: A Lagrangian based branch-and-bound algorithm for production-transportation problems. J. Global Optim. **18**, 59–73 (2000)
28. Lee, S., Grossmann, I.E.: A global optimization algorithm for nonconvex generalized disjunctive programming and applications to process systems. Comp. Chem. Eng. **25**, 1675–1697 (2001)
29. Maranas, C.D., Floudas, C.A.: Global minimum potential energy conformation of small molecules. J. Global Optim. **4**, 135–170 (1994)
30. McCormick, G.P.: Computability of global solutions to factorable nonconvex programs – part I – convex underestimating problems. Math. Program. **10**, 146–175 (1976)
31. Meyer, C.A., Floudas, C.A., Neumaier, A.: Global optimization with nonfactorable constraints. Ind. Eng. Chem. Res. **41**, 6413–6424 (2002)
32. Norkin, V.I., Pflug, G.Ch., Ruszczynski, A.: A branch and bound method for stochastic global optimization. Math. Program. **83**, 425–450 (1998)
33. Neumaier, A.: Complete search in continuous global optimization and constraint satisfaction. In: Iserles, A. (ed.) Acta Numerica 2004. Cambridge University Press, Cambridge (2004)
34. Papamichail, I., Adjiman, C.S.: Global optimization of dynamic systems. Comp. Chem. Eng. **28**, 403–415 (2004)
35. Pörn, R., Westerlund, T.: A cutting plane method for minimizing pseudo-convex functions in the mixed integer case. Comp. Chem. Eng. **24**, 2655–2665 (2000)
36. Quesada, I., Grossmann, I.E.: A global optimization algorithm for linear fractional and bilinear programs. J. Global Optim. **6**, 39–76 (1995)
37. Ryoo, H.S., Sahinidis, N.: Global optimization of nonconvex NLPs and MINLPs with applications in process design. Comp. Chem. Eng. **19**(5), 551–556 (1995)
38. Sahinidis, N.: BARON: a general purpose global optimization software package. J. Global Optim. **8**(2), 201–205 (1996)
39. Sherali, H.D., Alameddine, A.: A new reformulation linearization technique for bilinear programming problems. J. Global Optim. **2**, 379–410 (1992)
40. Takriti, S., Birge, J.R., Long, E.: A stochastic model of the unit commitment problem. IEEE Trans. Power Syst. **11**, 1497–1508 (1996)
41. Tawarmalani, M., Sahinidis, N.: Convexification and Global Optimization in Continuous and Mixed-Integer Nonlinear Programming: Theory, Algorithms, Software and Applications. Kluwer Academic Publishers, Dordrecht, The Netherlands (2002)
42. Tawarmalani, M., Sahinidis, N.: Global optimization of mixed integer nonlinear programs: a theoretical and computational study. Math. Program. **99**(3), 563–591 (2004)
43. van den Heever, S.A., Grossmann, I.E., Vasantharajan, S., Edwards, K.: A Lagrangean decomposition heuristic for the design and planning of offshore hydrocarbon field infrastructures with complex economic objectives. Ind. Eng. Chem. Res. **40**, 2857–2875 (2001)
44. Westerberg, A.W., Shah, J.V.: Assuring a global optimum by the use of an upper bound on the lower (dual) bound. Comp. Chem. Eng. **2**, 83–92 (1978)

45. Zamora, J.M., Grossmann, I.E.: A branch and bound algorithm for problems with concave univariate, bilinear and linear fractional terms. J. Global Optim. **14**(3), 217–249 (1999)
46. Zhu, Y., Kuno, T.: A global optimization method, QBB, for twice-differentiable nonconvex optimization problem. J. Global Optim. **33**, 435–464 (2005)