3

NEW FORMULATIONS AND BRANCHING STRATEGIES FOR THE GOP ALGORITHM

V. Visweswaran* and C. A. Floudas**

Mobil Research and Development Corporation, Princeton, NJ
*** Department of Chemical Engineering, Princeton University, Princeton, NJ*

ABSTRACT

In Floudas and Visweswaran (1990, 1993), a deterministic global optimization approach was proposed for solving certain classes of nonconvex optimization problems. A global optimization algorithm, **GOP**, was presented for the solution of the problem through a series of *primal* and *relaxed dual* problems that provide valid upper and lower bounds respectively on the global solution. The algorithm was proven to have finite convergence to an ϵ-global optimum. In this paper, a branch-and-bound framework of the GOP algorithm is presented, along with several reduction tests that can be applied at each node of the branch-and-bound tree. The effect of the properties is to prune the tree and provide tighter underestimators for the *relaxed dual* problems. We also present a mixed-integer linear programming (MILP) formulation for the *relaxed dual* problem, which enables an implicit enumeration of the nodes in the branch-and-bound tree at each iteration. Finally, an alternate branching scheme is presented for the solution of the relaxed dual problem through a linear number of subproblems. Simple examples are presented to illustrate the new approaches. Detailed computational results on the implementation of both versions of the algorithm can be found in the companion paper in chapter 4.

1   INTRODUCTION

In recent years, the global optimization of constrained nonlinear problems has received widespread attention. A considerable body of research has focused on the theoretical, algorithmic and computational aspects for identifying the global solution. Comprehensive reviews of the various existing approaches can be found in Dixon and Szego (1975, 1978), Archetti and Schoen (1984), Pardalos and Rosen (1986, 1987),

Torn and Zilinskas (1989), Mockus (1989), Horst and Tuy (1990) and Floudas and Pardalos (1990, 1992).

Floudas and Visweswaran (1990, 1993) proposed a deterministic primal-relaxed dual global optimization approach for solving certain classes of smooth optimization problems. A global optimization algorithm (GOP) was presented for the solution of the nonconvex problem through a series of primal and relaxed dual subproblems that provide upper and lower bounds on the global optimum. The algorithm was shown to attain finite ϵ-convergence and ϵ-global optimality regardless of the starting point. The application of the algorithm to several test problems was detailed in Visweswaran and Floudas (1990). Visweswaran and Floudas (1993) presented properties that vastly improve the efficiency of the algorithm.

The GOP algorithm presented in Floudas and Visweswaran (1990, 1993) follows a cutting plane approach to the solution of the relaxed dual subproblems. While this approach provides tight lower bounds by including all the valid cuts in the relaxed dual subproblems, it renders the implementation of the actual relaxed dual problem more complex. In particular, the identification of valid underestimators at each iteration of the algorithm must be followed with care. Moreover, the algorithm leaves open the questions of (i) an implicit enumeration of all the relaxed dual subproblems, and (ii) the reduction of the number of relaxed dual subproblems from exponential to linear, which would greatly improve the efficiency of the solution procedure.

This paper presents the GOP algorithm in the framework of a branch-and-bound approach. At each node in the branch and bound tree, a primal problem is solved, and the solution of this problem is used to provide a Lagrange function. By branching on the first derivatives of this Lagrange function, several new children nodes are created. This framework has several advantages over the original cutting plane approach, including considerably simplifying the formulation and solution of the relaxed dual problem and allowing for the incorporation of pruning and reduction tests at each node in the tree. While the approach is derived from the same basic properties that motivated the earlier algorithm, it differs sufficiently from the earlier approach so as to merit a complete discussion, which is presented in Section 4.

One of the main advantages of the branch-and-bound framework for the GOP algorithm is that it allows naturally for an implicit enumeration of the relaxed dual subproblems at each level. The introduction of binary variables linked to the sign of the derivatives of the Lagrange function results in mixed integer linear and nonlinear programming formulations that offer considerable scope for incorporation of reduction tests on a per node basis. The resulting GOP/MILP algorithm is discussed in detail in Section 5.

Due to the partitioning of the variable domain using the gradients of the Lagrange function, the GOP algorithm can require, in the worst case, an exponential number of dual subproblems at each iteration. This can lead to large CPU times as the number of variables increases. Therefore, it is worth considering alternate partitioning schemes that can reduce the number of subproblems that need to be solved at each iteration. In Section 6, one such branching scheme is presented that requires only a linear number of subproblems for the determination of the lower bound. A simple example is used to illustrate the new scheme.

In a companion paper (Visweswaran and Floudas, 1995b), a complete implementation of the algorithms presented here, along with comprehensive computational experience on several problems in chemical process design and control, is described.

2   PROBLEM FORMULATION

The general form of the optimization problem addressed in this paper is given as follows:

$$\min_{x,y} \ F(x,y)$$

$$
\begin{aligned}
s.t. \quad G(x,y) &\leq 0 \\
H(x,y) &= 0 \\
x &\in X \\
y &\in Y
\end{aligned}
\tag{3.1}
$$

where X and Y are non-empty, compact, convex sets, $F(x,y)$ is the objective function to be minimized, $G(x,y)$ is a vector of inequality constraints and $H(x,y)$ is a vector of equality constraints. It is assumed that these functions are continuous and piecewise differentiable over $X \times Y$. For the sake of convenience, it will be assumed that the set X is incorporated into the first two sets of constraints. In addition, the problem is also assumed to satisfy the following conditions:

Conditions (A):

(a) $F(x,y)$ and $G(x,y)$ are convex in x for every fixed y, and convex in y for every fixed x,

(b) $H(x,y)$ is affine in x for every fixed y, and affine in y for every fixed x,

(c) $Y \subseteq V$, where $V \equiv \{y : G(x, y) \leq 0, \ H(x, y) = 0, \ \text{for some } x \in X\}$, and

(d) An appropriate constraint qualification (e.g., Slater's qualification) is satisfied for fixed y.

It has been shown (Floudas and Visweswaran, 1990) that the class of problems that satisfies these conditions includes, but is not restricted to, bilinear problems, quadratic problems with quadratic constraints and polynomial and rational polynomial problems. Recently, it has also been shown (Liu and Floudas, 1993; Liu and Floudas, 1995) that a very large class of smooth optimization problems can be converted to a form where they satisfy *Conditions (A)*, and hence are solvable by the GOP algorithm.

3   PRIMAL AND RELAXED DUAL PROBLEMS

The GOP algorithm utilizes primal and relaxed dual subproblemsto obtain upper and lower bounds on the global solution. The primal problem results from fixing the y variables to some value, say y^k, and is defined as follows:

$$\min_{x} \quad F(x, y^k),$$

$$s.t. \quad \begin{aligned} G(x, y^k) &\leq 0 \\ H(x, y^k) &= 0 \end{aligned} \tag{3.2}$$

where $y^k \in Y$. It has been assumed here that any bounds on the x variables are incorporated into the first set of constraints. Notice that because of the introduction of additional constraints by fixing the y variables, this problem provides an upper bound on the global optimum of (3.1). Moreover, $P^k(y^k)$, the solution value of this problem yields a solution x^k for the x variables and Lagrange multipliers λ^k and μ^k for the equality and inequality constraints respectively[1].

The Lagrange function constructed from the primal problem is given as:

$$L^k(x, y, \lambda^k, \mu^k) = F(x, y) + \lambda^{k^T} H(x, y) + \mu^{k^T} G(x, y). \tag{3.3}$$

The x variables that are present in the linearization of the Lagrange function around x^k, and for which the gradients of the Lagrange functions with respect to x at x^k are

---

[1] It is assumed here that the primal problem is feasible for $y = y^k$. See Floudas and Visweswaran (1990, 1993) for the treatment of the cases when the primal problem is infeasible for a given value of y.

functions of the y-variables, are called the *connected* variables. It can easily be shown that the linearization of the Lagrange function around $\boldsymbol{x}^k$ can also be written in the form:

$$L^k(\boldsymbol{x}, y, \lambda^k, \mu^k)\big|_{x^k}^{lin} = L_0^k(y, \lambda^k, \mu^k) + \sum_{i=1}^{NI_C^k} x_i g_i^k(y) \qquad (3.4)$$

where NI_C^k is the number of *connected* variables at the kth iteration (representing the $\boldsymbol{x}$ variables that appear in the Lagrange function), and $L_0^k(y, \lambda^k, \mu^k)$ represents all the terms in the linearized Lagrange function that depend only on y. The positivity and negativity of the functions $g_i^k(y)$ define a set of equations that are called the *qualifying* constraints of the Lagrange function at the kth iteration, and which partition the y variable space into $2^{NI_c^k}$ subregions. In each of these subregions, a Lagrange function can be constructed (using the bounds for the $\boldsymbol{x}$ variables) that underestimates the global solution in the subregion, and can therefore be minimized to provide a lower bound for the global solution in that region.

Consider the first iteration of the GOP algorithm. The initial *parent* region is the entire space $y \in Y$ from the original problem. This region is subdivided into $2^{NI_c^1}$ subregions, and in each of these subregions, a subproblem of the following form is solved:

$$\min_{y \in Y, \mu_B} \mu_B$$

$$s.t. \quad \mu_B \geq L^1(\boldsymbol{x}^{B_l}, y, \lambda^1, \mu^1)\big|_{x1}^{lin},$$

$$\left. \begin{array}{ll} g_i^1(y) \geq 0 & \text{if} \quad x_i^{B_l} = x_i^L \\ g_i^1(y) \leq 0 & \text{if} \quad x_i^{B_l} = x_i^U \end{array} \right\} \quad \forall\, i \in I_c^1,$$

where I_c^1 is the set of *connected* variables at the first iteration, NI_C^k is the number of *connected* variables, and x_i^L and x_i^U are the lower and upper bounds on the ith *connected* variable respectively. This subproblem corresponds to the minimization of the Lagrange function, with the *connected* variables replaced by a combination of their lower and upper bounds. Note the presence of the *qualifying* constraints in the problem. These constraints ensure that the minimization is carried out in a subregion of the *parent* node. If this problem has a value of μ_B that is lower than the current best upper bound obtained from the primal problem, then it is added to the set of candidate lower bounds; otherwise, the solution is fathomed, that is, removed from consideration for further refinement.

Consider a problem with two $\boldsymbol{x}$ and two y variables. In the first iteration, assuming that both $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ are in the set of *connected* variables for the first iteration, there are four relaxed dual subproblems solved. These problems are shown in Figure 1a. It
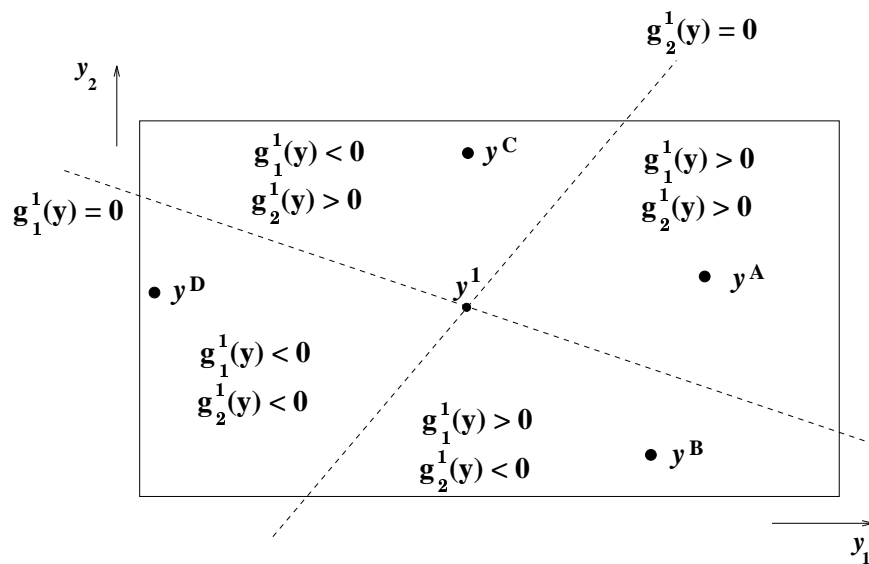
Figure 1a Partition in y for first iteration with two *connected* variables
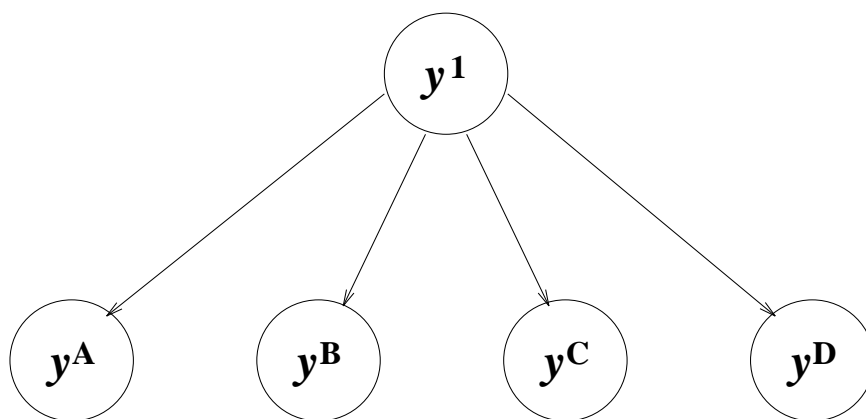

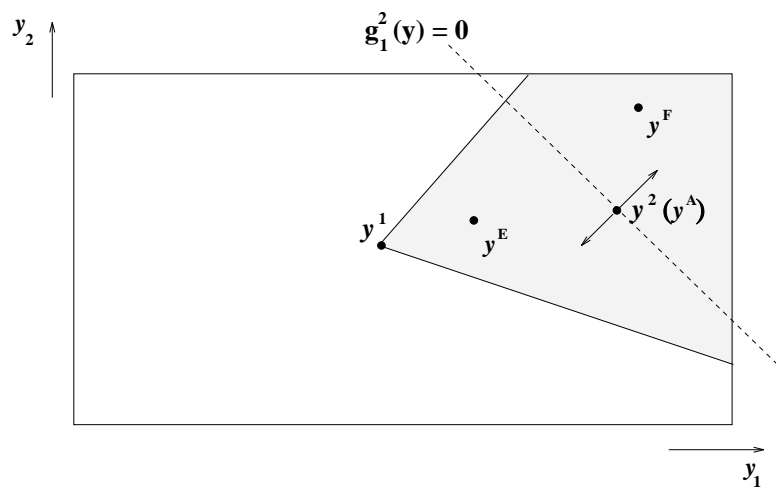**Figure 1b** Branch and bound tree for first iteration

Figure 2a   Partition in y for second iteration with one *connected* variable
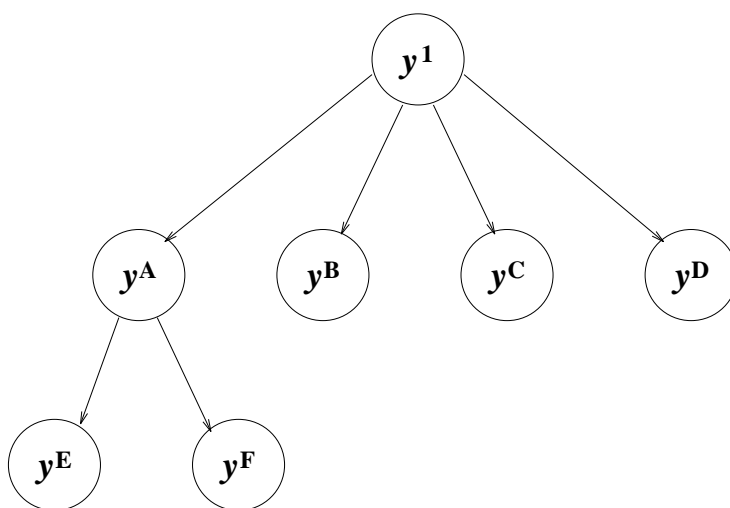

Figure 2b   Branch and bound tree for second iteration

can be seen that the *qualifying* constraints partition the y-space into the four regions. Each of the relaxed dual subproblems solved provides a valid underestimator for the corresponding region, as well as a solution point (denoted in the figure by y^A, y^B, y^C and y^D) in the region.

Figure 1b shows the corresponding branch-and-bound tree created by the solution of these four problems. The starting point y^1 is the root node, and it spawns four leaf nodes. The infimum of the four nodes provides the point for the next iteration, in this case, say y^A.

In the second iteration, the relaxed dual problem is equivalent to further partitioning the subregion that was selected for refinement. In each of these partitions, a *relaxed dual* subproblem is solved. Figure 2a shows the subregions created in the example, assuming that there was only one *connected* variable in this iteration. The two relaxed dual subproblems solved in this iteration give new solutions y^E and y^F and are possible candidates for entering at future iterations. Figure 2b shows the corresponding nodes in the branch-and-bound tree created by this iteration.

The preceding discussion illustrates the key features of a branch and bound framework for the algorithm. The framework is based upon the successive refinement of regions by partitioning on the basis of the *qualifying* constraints. In the next section, the key features of its implementation are discussed, based on which a formal statement of the algorithm is then presented.

4   A BRANCH-AND-BOUND FRAMEWORK FOR THE GOP ALGORITHM

The terminology used in this section is as follows. Given a node j in the branch and bound tree, P_j is its parent node, and I_j is the iteration at which node j is created. R_j is the set of constraints defining the region corresponding to node j. At any point, N denotes the total number of nodes in the tree, and C denotes the current node.

4.1   Root Node and Starting Region

At the beginning of the algorithm, there are no subdivisions in the y-space. Therefore, the root node in the branch and bound tree is simply the starting point for the algorithm, y^1. The region of application for this node (i.e., the *current* region) is the entire y-space.

4.2 Reduction Tests at Each node

At each node, the current region of application is divided into several subregions using the *qualifying* constraints of the current Lagrange function. It is possible to conduct simple tests on the basis of the signs of the *qualifying* constraints that can be used to reduce the number of *connected* variables. One such test, based upon the properties first presented in Visweswaran and Floudas (1993) is presented below:

Reduction Test:

Suppose a node j is to be partitioned in the kth iteration (i.e., $I_j = k$). Then,

(i) If $g_i^k(y) \geq 0 \quad \forall y \in R_j$, set $x_i = x_i^L$ in $L^k(x, y, \lambda^k, \mu^k)$ and remove i from the set of *connected* variables.

(ii) If $g_i^k(y) \leq 0 \quad \forall y \in R_j$, set $x_i = x_i^U$ in $L^k(x, y, \lambda^k, \mu^k)$ and remove i from the set of *connected* variables.

The proofs of the validity of these reductions can be easily obtained by considering that the term $x_i g_i^k(y)$ can be underestimated by $x_i^L g_i^k(y)$ for all positive $g_i^k(y)$ and $x_i^U g_i^k(y)$ for all negative $g_i^k(y)$. For more details, the reader is referred to Visweswaran and Floudas (1993).

4.3 Evaluation of bounds for the x variables

Often, the original problem contains linear and/or convex constraints in both x and y. When the relaxed dual problem is being solved at a given iteration, the region for the y variables is smaller than for the original problem. This can be exploited to provide tighter bounds on the x variables.

Consider, for example, a problem where there is a one-to-one correspondence between the x and y variable set in the feasible region of the problem (i.e., $y_i = x_i$). Then, consider the Kth iteration of the **GOP** algorithm, where the node j is being partitioned. R_j is the set of constraints defining the current region. Then, it is possible to obtain tighter bounds on the x variables by the following procedure:

1. Choose an $i \in I_c^K$.

2. Solve the following two problems:

$$\min_{x} \quad \pm x_i$$
$$x - y = 0$$
$$y \in R_j$$

Use the solutions of the two problems for the lower and upper bounds on x_i respectively. Note that the set R_j includes all linear and convex constraints from the original problem.

3. Repeat Step 1 and 2 for all $i \in I_c^K$.

Similarly, when there are other convex constraints in x and y, these constraints can be added to the above problem. This procedure can be very useful in obtaining the tightest bounds on the *connected* x variables at each iteration and consequently in obtaining the tightest underestimators for the relaxed dual subproblems. Note also that in the case of nonconvex constraints, we can incorporate their convex underestimators in the evaluation of the bounds problems.

4.4 Branch-and-Bound Algorithm

The major steps of the branch-and-bound version of the GOP algorithm are described in this section. The terminology is the same as described in section 4. In addition, F denotes the set of iterations with a feasible primal problem, while I denotes the set of iterations when the primal problem was infeasible.

STEP 0: Initialization

(a) Read in the data for the problem including tolerance for convergence, ϵ.

(b) Define initial upper and lower bounds (f^U, f^L) on the global optimum.

(c) Generate initial bounds for the x variables, x^L and x^U.

(d) Choose a starting point y^1 for the algorithm.

(e) Set $K = 1, C = P_C = 1, N = 1$.

STEP 1: Selection of Current Region

(a) If $K = 1$, set $R_C = \emptyset$ and goto Step 2.

(b) If $K \geq 2$, set $R_C = \emptyset$, $m = C$. Then:

    (i) Add the Lagrange function and *qualifying* constraints for node m to R_C.

    (ii) Set $m = P_m$. If $m = 1$, then goto Step 2.

    (iii) Repeat steps (i) and (ii).

STEP 2: Primal problem

(a) Solve the primal problem (3.2) to give $P^K(y^K)$.

    (i) If feasible, set $F = F \cup K$ and update $f^U = MIN[f^U, P^K(y^K)]$.

    (ii) If infeasible, solve a *relaxed* primal problem. Set $I = I \cup K$.

(b) Store y^K, λ^K and μ^K.

STEP 3: Determination of Current Partitions

(a) Generate the current Lagrange function $L^K(x, y, \lambda^K, \mu^K)$.

(b) Determine the set of *connected* variables I_c^K and the corresponding partial derivatives $g_i^K(y)$ $(i = 1, \ldots, I_c^K)$ of the current Lagrange function.

(c) For each *connected* variable, determine (if possible) tight lower and upper bounds x_i^L and x_i^U in the current region $y \in R_C$. Otherwise, use the original bounds.

(d) Evaluate lower and upper bounds on $g_i^K(y)$ in the region $y \in R_C$.

    (i) If $g_i^K(y) \geq 0$ $\forall y \in R_C$, set $x_i^B = x_i^L$ in the current Lagrange function, and remove i from the set I_c^K.

    (ii) If $g_i^K(y) \leq 0$ $\forall y \in R_C$, set $x_i^B = x_i^U$ in the current Lagrange function, and remove i from the set I_c^K.

STEP 4: Relaxed Dual Problem

(a) Select a combination of the bounds B_l of the *connected* variables, say $B_l = B_1$.

(b) Find the solution $(\mu_B^\star, y^\star)$ to the following *relaxed dual* subproblem:

$$\min_{y \in Y, \mu_B} \quad \mu_B$$

$$
\begin{aligned}
s.t. \quad \mu_B \;\geq\;& \left. L^K\left(x^{B_l}, y, \lambda^K, \mu^K\right)\right|_{x^K}^{lin} \\
g_i^K(y) \geq 0 \quad & if \; x_i^{B_l} = x_i^L \\
g_i^K(y) \leq 0 \quad & if \; x_i^{B_l} = x_i^U \\
(y, \mu_B) \;\in\;& R_C
\end{aligned}
$$

(i) If $\mu_B^\star < f^U - \epsilon$, set $j = N + 1, P(j) = C, N = N + 1$, and store the solution in μ_B^j, y^j.

(ii) If $\mu_B^\star \geq f^U - \epsilon$, fathom the solution.

(c) Select a new combination of bounds, say $B_l = B_2$, for the *connected* variables.

(d) Repeat steps (b) and (c) until all the combinations of bounds for the *connected* variables have been considered.

STEP 5: Selecting a new lower bound

Select the infimum of all μ_B^j, say μ_B^p. Set $C = p, y^{K+1} = y^p, f^L = \mu_B^p$.

STEP 6: Check for convergence

If $\left|\frac{f^U - f^L}{f^U}\right| < \epsilon$, STOP; otherwise, set $K = K + 1$ and return to step 1.

4.5   Illustration

Consider the application of the branch and bound algorithm to the following problem, taken from Al-Khayyal and Falk (1983):

$$
\begin{aligned}
\min_{x,y} \quad & -x + xy - y \\
-6x + 8y - 3 \;\leq\;& 0, \\
3x - y - 3 \;\leq\;& 0, \\
x, y \;\geq\;& 0
\end{aligned}
$$

Note that with these constraints, the bounds on both x and y are $(0, 1.5)$. Consider a starting point of $y^1 = 1$ for the algorithm.

Iteration 1

For $y^1 = 1$, the first primal problem has the solution of $x = 0$, $\mu_1^1 = \mu_2^1 = \mu_3^1 = 0$, with the objective value of -1. The upper bound on the problem is therefore -1. The Lagrange function is given by

$$L^1(x, y, \mu^1) = -x + xy - y = xg_1^1(y) - y$$

where $g_1^1(y) = y - 1$ is the first (and only) *qualifying* constraint. From the original problem, the bounds on y are $(0, 1.5)$. Therefore, $-1.0 \leq g_1^1(y) \leq 0.5$, implying that two relaxed dual subproblems need to be solved. These problems, solved for positive and negative $g_1^1(y)$ (with x set to 0 and 1.5 respectively), are shown below:

Node 1	Node 2

$$\begin{array}{ll} \min_{y, \mu_B} & \mu_B \\ \mu_B \geq & -y \\ g_1^1(y) = & y - 1 \geq 0 \\ 0 \leq & y \leq 1.5 \end{array} \qquad \begin{array}{ll} \min_{y, \mu_B} & \mu_B \\ \mu_B \geq & 0.5y - 1.5 \\ g_1^1(y) = & y - 1 \leq 0 \\ 0 \leq & y \leq 1.5 \end{array}$$

Solution: $y = 1.5, \mu_B = -1.5$ Solution: $y = 0.0, \mu_B = -1.5$

The solution of these two problems provides the first partition in the tree depicted in Figure 3. R is the root node corresponding to the starting point $y^1 = 1.0$. At this iteration, two nodes 1 and 2 are created by the solution of the two relaxed dual subproblems. Both nodes have the root node R as their *parent* node. Both problems have equal objective values, and are thus equal candidates for the best lower bound. Suppose that node 2 is selected for further exploration.

Iteration 2

From node 2, the current value of y is 0.0. For this value, the primal problem has the solution $x = 1.0$, $\mu_1^2 = 0$, $\mu_2^2 = \frac{1}{3}$, $\mu_3^2 = 0$, with the objective value of -1.0. The Lagrange function from this problem is

$$L^2(x, y, \mu^2) = -x + xy - y + \frac{1}{3}(3x - y - 3) = xg_1^2(y) - \frac{4}{3}y - 1$$

where $g_1^2(y) = y$ is the *qualifying* constraint for this Lagrange function.

For this iteration, the relaxed dual subproblems are solved in the region $0 \leq y \leq 1$. The tightest bounds on x for this region can be found by solving the following two
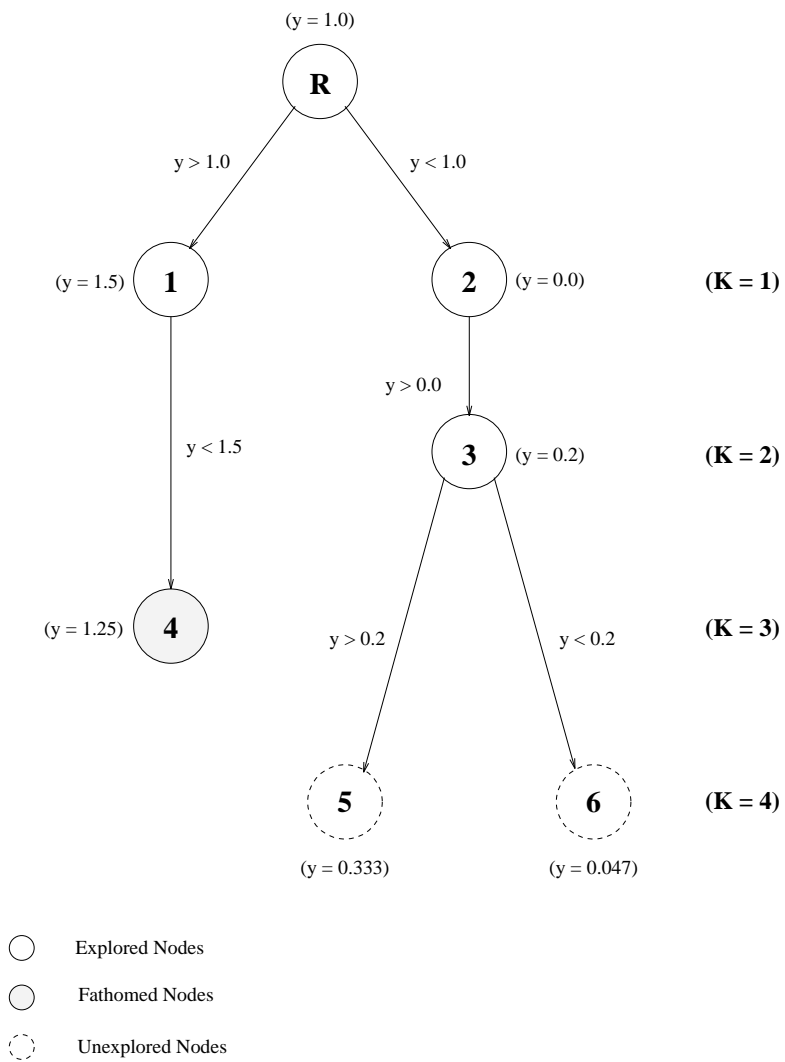
(y = 1.0)

R

y > 1.0   y < 1.0

(y = 1.5) 1   2 (y = 0.0)   **(K = 1)**

y > 0.0

y < 1.5

3 (y = 0.2)   **(K = 2)**

(y = 1.25) 4   **(K = 3)**

y > 0.2   y < 0.2

5   6   **(K = 4)**

(y = 0.333)   (y = 0.047)

○ Explored Nodes

○ Fathomed Nodes

○ Unexplored Nodes

Figure 3   Branch and Bound tree for Illustrating Example

problems:

$$\min_{x} \ \pm x$$
$$-6x + 8y - 3 \ \le \ 0,$$
$$3x - y - 3 \ \le \ 0,$$
$$0 \le y \ \le \ 1,$$
$$x \ \ge \ 0.$$

The solutions of these problems provide lower and upper bounds for x respectively. Thus, for the region $0 \le y \le 1$, this yields the bounds $0 \le x \le \frac{4}{3}$.

Since $y > 0$, it is obvious that $g_1^2(y)$ is positive for all y in the current region. Therefore, only one relaxed dual problem needs to be solved, with a valid underestimator to $L^2(x, y, \mu^2)$ being used by fixing x to its lower bound. Moreover, from the first iteration, the Lagrange function corresponding to node 2 is also a valid cut for this region. Note, however, that instead of using the original bounds on x in both these Lagrange functions, the improved bounds can be used. This yields the following relaxed dual problem:

<u>Node 3</u>

$$\min_{y, \mu_B} \quad \mu_B$$
$$\mu_B \ \ge \ \tfrac{1}{3}y - \tfrac{4}{3}$$
$$g_1^1(y) \ = \ y - 1 \le 0$$
$$\mu_B \ \ge \ -\tfrac{4}{3}y - 1$$
$$0 \ \le \ y \le 1.5$$

Solution: $y = 0.2, \mu_B = -1.26667.$

At the end of this iteration, there are two candidate regions for further partitioning: (i) the region $1 \le y \le 1.5$ corresponding to node 1, with a lower bound of -1.5, and (ii) the region $0 \le y \le 1$ corresponding to node 3, with the lower bound of -1.26667. Following the criterion of selecting the region with the best lower bound, node 1 is chosen for further exploration.

Iteration 3

From node 1, the current value of y is 1.5. For this value, the primal problem has the solution $x = 1.5$, $\mu_1^3 = \frac{1}{12}$, $\mu_2^3 = 0$, $\mu_3^3 = 0$, with the objective value of -0.75. The Lagrange function for this iteration is

$$L^3(x, y, \mu^3) = -x + xy - y + \frac{1}{12}(-6x + 8y - 3) = xg_1^3(y) - \frac{1}{3}y - \frac{1}{4}$$

where $g_1^3(y) = y - 1.5$ is the *qualifying* constraint for this Lagrange function.

For this iteration, the relaxed dual subproblems are solved in the region $1 \leq y \leq 1.5$, In this region, solving the bounds problems for x yields $\frac{5}{6} \leq x \leq 1.5$. Since $g_1^3(y) \leq 0$ for all y, only one relaxed dual problem needs to be solved, with x fixed to its upper bound. From the first iteration, the Lagrange function corresponding to node 1 is also a valid cut for this region. Using the improved bounds on x shown above yields the following relaxed dual problem:

Node 4

$$
\begin{aligned}
\min_{y, \mu_B} \quad & \mu_B \\
\mu_B \geq \quad & -\tfrac{1}{6}y - \tfrac{5}{6} \\
g_1^1(y) = \quad & y - 1 \geq 0 \\
\mu_B \geq \quad & \tfrac{7}{6}y - 2.5 \\
0 \leq \quad & y \leq 1.5
\end{aligned}
$$

Solution: $y = 1.25$, $\mu_B = -1.04167$.

Again, there is no partition of the region in this iteration, but the relaxed dual provides a tighter lower bound for this region than was originally available.

At the end of this iteration, there are two candidate regions for further partitioning: (i) the region $0 \leq y \leq 1$, corresponding to the node 3, with the lower bound of -1.26667, and (ii) the region $1 \leq y \leq 1.5$, corresponding to node 4, with the lower bound of -1.04167. Following the criterion of selecting the region with the best lower bound, node 3 is chosen for further exploration.

Iteration 4

From node 3, the current value of y is 0.2. For this value, the primal problem has the solution $x = 1.0667$, $\mu_1^4 = 0$, $\mu_2^4 = 0.2667$, $\mu_3^4 = 0$, with the objective value of -1.05333. Note that the solution of this problem has the immediate consequence that it provides an upper bound that is lower than the lower bound for node 4 (which is -1.04167). Therefore, node 4 can be immediately fathomed, i.e., removed from consideration for any further refinement or exploration.

The Lagrange function from the current primal problem is

$$L^4(x, y, \mu^4) = -x + xy - y + 0.2667(3x - y - 3) = xg_1^4(y) - 1.2667y - 0.8$$

where $g_1^4(y) = y - 0.2$ is the *qualifying* constraint for this Lagrange function.

For this iteration, the relaxed dual subproblems are solved in the region $0 \leq y \leq 1.0$, and try to provide refined lower bounds by partitioning the region further. The tightest bounds for x in this region are $0 \leq x \leq \frac{4}{3}$.

Unlike the previous two iterations, it is necessary to partition the current region since $-0.2 \leq g_1^4(y) \leq 1.3$ and the reduction tests of Section 4.2 do not provide any help. It is therefore necessary to solve two relaxed dual subproblems in the current iteration. For both these problems, the Lagrange functions from nodes 2 and 3 are valid underestimators. These two problems are shown below:

Node 5	Node 6

$$\begin{array}{ll}
\min\limits_{y,\mu_B} & \mu_B \\
\mu_B \geq & \frac{1}{3}y - \frac{4}{3} \\
g_1^1(y) = & y - 1 \leq 0 \\
\mu_B \geq & -\frac{4}{3}y - 1 \\
\mu_B \geq & -1.2667y - 0.8 \\
g_1^4(y) = & y - 0.2 \geq 0 \\
0 \leq & y \leq 1.5
\end{array}
\qquad
\begin{array}{ll}
\min\limits_{y,\mu_B} & \mu_B \\
\mu_B \geq & \frac{1}{3}y - \frac{4}{3} \\
g_1^1(y) = & y - 1 \leq 0 \\
\mu_B \geq & -\frac{4}{3}y - 1 \\
\mu_B \geq & 0.0667y - 1.0667 \\
g_1^4(y) = & y - 0.2 \leq 0 \\
0 \leq & y \leq 1.5
\end{array}$$

<div align="center">

Solution: $y = 0.333$,   Solution: $y = 0.04762$,

$\mu_B = -1.2222$. $\mu_B = -1.06349$.

</div>

Together, these two problems provide a tighter lower bound (-1.2222) for the region $0 \leq y \leq 1$ than before (-1.26667).

At the end of this iteration, there are two candidate regions for further partitioning -- the region $0 \leq y \leq 0.2$, corresponding to node 6, with the lower bound of -1.06349, and (ii) the region $0.2 \leq y \leq 1$, corresponding to node 5, with the lower bound of -1.2222. Therefore, node 5 is chosen for further refinement.

The algorithm continues in this fashion for 18 iterations, converging to the global solution of -1.0833 at $x = 1.1667, y = 0.5$ with a tolerance of 0.001 between the upper and lower bounds. It is interesting to note that the original GOP algorithm, which does not compute the tightest bounds on the x variables at each iteration, takes 76 iterations to converge with the same tolerance. This indicates the importance of having the tightest possible bounds on the *connected* variables at each iteration.

5   REFORMULATION OF THE RELAXED DUAL AS A SINGLE MILP PROBLEM

The solution of the relaxed dual subproblems at each node is the most time-consuming step in the algorithm outlined in Section 4. The reduction test mentioned in Section 4.2 can help to prune the branch-and-bound tree at each node; however, it is still necessary to solve a large number of subproblems at each iteration. It is very likely that the solution of most of these subproblems are useless as far as the succeeding iterations are concerned, that is, most of the nodes will be fathomed as soon as they are spawned. Naturally, this raises the question whether these subproblems can be solved implicitly. This section presents one possible approach for reformulation of the relaxed dual problem at each iteration so that the implicit enumeration of all the solutions can be achieved by solution of an MILP problem.

At the Kth iteration, the Lagrange function has the form given by (3.4). Consider the ith term in the summation. In each of the $2^{NI_C^k}$ relaxed dual subproblems, this term takes on either of two values:

$$x_i g_i^K(y) = \begin{cases} x_i^L g_i^K(y) & \text{if} \quad g_i^K(y) \geq 0 \\ x_i^U g_i^K(y) & \text{if} \quad g_i^K(y) \leq 0 \end{cases}$$

Now, x_i can be implicitly expressed as a combination of its lower and upper bounds:

$$x_i = \left(1 - \alpha_i^K\right) x_i^L + \alpha_i^K x_i^U \tag{3.5}$$

where $\alpha_i^K \in \{0, 1\}$.

This leads to the following formulation for the ith term in (3.4):

$$x_i g_i^K(y) \;=\; t_i + x_i^L g_i^K(y)$$

where

$$t_i \;\geq\; \alpha_i^K (x_i^U - x_i^L)\underline{g_i^K}$$
$$t_i \;\geq\; (x_i^U - x_i^L)(g_i^K(y) - (1 - \alpha_i^K)\overline{g_i^K})$$
$$\alpha_i^K \underline{g_i^K} \;\leq\; g_i^K(y) \leq (1 - \alpha_i^K)\overline{g_i^K}$$

where $\underline{g_i^K}$ and $\overline{g_i^K}$ are respectively the lower and upper bounds on the *qualifying* constraints. As the following property shows, this can be used to reformulate the relaxed dual problem as a mixed integer linear program (MILP):

Property 5.1 *Suppose that, at the Kth iteration, C denotes the current node to be partitioned, and R_C denotes the set of constraints defining the region associated with C. Then, the best solution from all the relaxed dual subproblems at this iteration can be obtained as the optimal solution of the following mixed-integer linear program.*

$$\min_{\substack{y \in Y, \mu_B \\ t, \alpha}} \; \mu_B \tag{3.6}$$

$$\text{s.t.} \quad \mu_B \;\geq\; \sum_{i=1}^{NI_c^K} t_i^K + \sum_{i=1}^{NI_c^K} x_i^L g_i^K(y) + L_0^K(y, \lambda^K, \mu^K) \tag{3.7}$$

$$t_i^K \;\geq\; \alpha_i^K (x_i^U - x_i^L)\underline{g_i^K} \tag{3.8}$$

$$t_i^K \;\geq\; (x_i^U - x_i^L)(g_i^K(y) - (1 - \alpha_i^K)\overline{g_i^K}) \tag{3.9}$$

$$\alpha_i^K \underline{g_i^K} \;\leq\; g_i^K(y) \leq (1 - \alpha_i^K)\overline{g_i^K} \tag{3.10}$$

$$t^K \;\in\; \Re^{NI_c^K}, \; \alpha^K \in \{0,1\}^{NI_c^K}, \; y \in Y \tag{3.11}$$

$$(y, \mu_B) \;\in\; R_C \tag{3.12}$$

where $\underline{g_i^K}$ and $\overline{g_i^K}$ are the lower and upper bounds on $g_i^K(y)$ over Y.

Proof. Since α_i^K is a binary variable, it can take on only two values in any solution, either 0 or 1. Consider these two possible cases for α_i^K:

Case I ($\alpha_i^K = 0$) :
 In this case, equations (3.8)-(3.10) reduce to

$$t_i^K \;\geq\; 0 \tag{3.13}$$

$$t_i^K \geq (x_i^U - x_i^L)(g_i^K(y) - \overline{g_i^K}) \tag{3.14}$$

$$0 \leq g_i^K(y) \leq \overline{g_i^K} \tag{3.15}$$

Since $g_i^K(y) \leq \overline{g_i^K}$ for all $y \in Y$, (3.14) is redundant. Similarly, the second inequality in (3.15) is also trivially satisfied. Therefore, if this set of constraints is active in any solution, then $t_i^K = 0$, the contribution from the ith components of the first two terms in (3.7) to μ_B is $x_i^L g_i^K(y)$, and in addition, we must also have $g_i^K(y) \geq 0$.

Case II ($\alpha_i^K = 1$) :

In this case, equations (3.8)-(3.10) reduce to

$$t_i^K \geq (x_i^U - x_i^L)\underline{g_i^K} \tag{3.16}$$

$$t_i^K \geq (x_i^U - x_i^L)g_i^K(y) \tag{3.17}$$

$$\underline{g_i^K} \leq g_i^K(y) \leq 0 \tag{3.18}$$

Since $g_i^K(y) \geq \underline{g_i^K}$ for all $y \in Y$, (3.16) is redundant. Similarly, the first inequality in (3.18) is trivially satisfied. Therefore, if this set of constraints is active in any solution, then $t_i^K = (x_i^U - x_i^L)g_i^K(y)$, the contribution from the ith components of the first two terms in (3.7) to μ_B is $x_i^U g_i^K(y)$, and in addition, $g_i^K(y) \leq 0$.

Thus, it can be seen that any solution of the relaxed dual problem in Step 4 of the algorithm in Section 4 is automatically embedded in the set of constraints described by (3.7)-(3.12). Therefore, (3.6)-(3.12) is a valid formulation for obtaining the solution of the relaxed dual problem.   □

Remark 5.1 If $L_0^K(y, \lambda^K, \mu^K)$ are convex functions in y, then (3.6)-(3.12) is a convex MINLP, and can be solved with the Generalized Benders Decomposition (Geoffrion, 1972; Floudas *et al.*, 1989) or the Outer Approximation algorithm (Duran and Grossmann, 1986).

It should be noted that the reduction tests of Section 4.2 can also be applied to the MILP formulation, as shown by the following property.

Property 5.2 *At the Kth iteration,*

(i) *If $g_i^K(y) \geq 0$ for all y (respectively $g_i^K(y) \leq 0$ for all y) then variable α_i^K can be fixed to 0 (respectively 1.)*

(ii) *If $g_i^K(y) = 0$ for all y then variable α_i^K vanishes from formulation (3.6)-(3.12).*

Proof. (i) Suppose that $g_i^K(y) \geq 0$ for all $y \in Y$. Then, to underestimate the Lagrange function from the Kth iteration, x_i^B must be set to x_i^L. By the definition of α_i^K,

$$x_i = (1 - \alpha_i^K)x_i^L + \alpha_i^K x_i^U$$

Hence, this leads to $\alpha_i^K = 0$. Conversely, if $g_i^K(y) \leq 0$ for all $y \in Y$, then α_i^K must be equal to 1.

(ii) If $g_i^K(y) = 0$ for all $y \in Y$, then this implies that

$$\underline{g_i^K} = g_i^K(y) = \overline{g_i^K} = 0$$

Therefore, in (3.6)-(3.12), t_i^K is always equal to zero, and the variable α_i^K vanishes from the formulation. □

Backtracking

With the MILP reformulation, it is possible to solve the relaxed dual subproblems implicitly for the best solution at each iteration. However, it is not sufficient to find the best solution; it must also be determined whether any of the other partitions can provide a useful solution for further refinement.

Consider the relaxed dual subproblems solved when node j is being partitioned. Suppose that this node was partitioned during iteration K. Then, there are NI_c^K binary variables, and $2^{NI_c^K}$ partitions to consider. Solving the problem (3.6)-(3.12) gives the best solution among these partitions. Suppose that this solution corresponds to the combination α^C. Suppose also that J_C is the set of binary variables that are equal to 1 in this combination, and that there are NJ_C of them. Consider now the following cut

$$\sum_{i \in J_C} \alpha_i - \sum_{i \notin J_C} \alpha_i \leq NJ_C - 1$$

If problem (3.6)-(3.12) is resolved with the above cut added to the problem, then the solution will have a value for α different from α^C, and will therefore correspond to a different subregion of the current problem. Note that the objective value of this problem represents the ''second'' best possible solution. The best solution, of course, is the one corresponding to the solution of the first MILP problem, with $\alpha = \alpha^C$. Therefore, this methodology is sufficient to go back to a partitioned node at any point.

Note that although the size of the MILP problems increases slightly at each iteration due to the accumulation of constraints from previous iterations, the number of binary variables present in these problems is equal to the number of *connected* variables for each iteration. In other words, the number of binary variables in the MILP problems is bounded by the number of x variables in the original problem.

5.1 The GOP/MILP Algorithm

The terminology is as described in Section 4. In addition, A_j denotes the set of integer cuts to be used when solving the MILP problem for the node j.

STEP 0: Initialization

This step is the same as in Section 4.4, with the addition of setting $A_1 = \emptyset$.

STEP 1 -- Step 3:

Same as in Section 4.4.

STEP 4: Current Relaxed Dual Problem

Solve the MILP problem (3.6)-(3.12).

(i) If $\mu_B^\star < f^U - \epsilon$, set $j = N + 1, P(j) = C, N = N + 1$, and store the solution in μ_B^j, y^j.

(ii) If $\mu_B^\star \geq f^U - \epsilon$, fathom the solution.

Let the solution for the binary variables in this problem be $\alpha = \alpha^C$. Let J_C be the set of variables which are 1 in this solution, and let NJ_C the number of such binary variables.

STEP 5: Selecting a new lower bound

Same as in Section 4.4.

STEP 6: Regenerating Solutions From Partitioned Nodes

Suppose that the solution selected in Step 5 corresponds to node C, and that this node was originally partitioned at iteration k. Then, add the cut

$$\sum_{i \in J_C} \alpha_i - \sum_{i \notin J_C} \alpha_i \leq N J_C - 1$$

to the set of binary cuts A_C. Solve the MILP problem (3.6)-(3.12) with the added set of binary cuts A_C. Suppose the solution of this problem is $\mu_B^\star$.

(i) If $\mu_B^\star < f^U - \epsilon$, then set $j = N + 1, P(j) = C, N = N + 1$, and store the solution in μ_B^j, y^j. Also set α^C to be the solution of the binary variables in this formulation.

(ii) If $\mu_B^\star \geq f^U - \epsilon$, fathom the node C.

STEP 7: Check for convergence

Same as in Section 4.4.

Remark 5.2 After the MILP problem has been solved in either Step 4 or Step 6, an integer cut is added to the corresponding formulation which ensures that that solution cannot be repeated. This implies that the same MILP formulation might be solved several times over the course of the iterations with small differences arising from the additional integer cuts. Subsequently, there is considerable potential for storing the tree information from these problems for use in future iterations.

Remark 5.3 At each iteration of the algorithm, there is a *single* MILP problem solved in Step 4 or Step 6 as compared to the original algorithm, which needs to solve $2^{NI_c^K}$ subproblems at the Kth iteration. This MILP problem contains NI_c^K binary variables in the case of Step 4, or NI_c^L variables in Step 6. In either case, the number of binary variables present in any MILP formulation during all the iterations is bounded by the maximum number of x variables. However, it is usually the case that the number of *connected* variables is a fraction of the total number of x variables, implying that the MILP problems are likely to have few binary variables.

Remark 5.4 The major advantage of the MILP problem appears when there are more than about 15 *connected* variables at any iteration. In such cases, the original algorithm would need to solve over 2 million problems at that iteration, the vast majority of which would never be considered as candidate solutions for further branching. In the case of the MILP algorithm, the implicit enumeration allows for far fewer problems to be solved. The maximum number of MILP problems solved is twice the number of iterations of the algorithm.

5.2 Illustration of the GOP/MILP Algorithm

Consider the example from Section 4.5, with a starting point of $y^1 = 1$ for the algorithm.

Iteration 1

For $y^1 = 1$, the first primal problem has the solution of $x = 0$, $\mu_1^1 = \mu_2^1 = \mu_3^1 = 0$, with the objective value of -1. The upper bound on the problem is therefore -1. The Lagrange function is given by

$$L^1(x, y, \mu^1) = -x + xy - y = xg_1^1(y) - y$$

where $g_1^1(y) = y - 1$ is the first (and only) *qualifying* constraint.

The following MILP problem is solved first in Step 4:

$$\min_{y, \mu_B} \quad \mu_B$$

$$\begin{aligned}
\mu_B &\geq t_1^1 - y \\
t_1^1 &\geq -1.5\alpha_1^1 \\
t_1^1 &\geq 1.5(g_1^1 - 0.5(1 - \alpha_1^1)) \\
\alpha_1^1 &\leq g_1^1 \leq 0.5(1 - \alpha_1^1) \\
g_1^1 &= y - 1 \\
0 &\leq y \leq 1.5
\end{aligned}$$

The solution of this problem is $y = 0.0$, $\mu_B = -1.5$, $\alpha_1^1 = 1$. Note that this corresponds to node 2 in the branch and bound tree in Figure 3. This solution is chosen to be the next candidate for branching. However, in order to ensure that the other regions are also considered for future reference, it is necessary to solve one more problem, with the cut

$$\alpha_1^1 \leq 0$$

added to the MILP. This problem has the solution $y = 0$, $\mu_B = -1.5$ and $\alpha_1^1 = 1$. It is stored for future reference.

Iteration 2

For $y = 0.0$, the primal problem has the solution $x = 1.0$, $\mu_1^3 = 0$, $\mu_2^3 = \frac{1}{3}$, $\mu_3^3 = 0$, with the objective value of -1.0. The Lagrange function from this problem is

$$L^3(x, y, \mu^3) = -x + xy - y + \frac{1}{3}(3x - y - 3) = xg_1^3(y) - \frac{4}{3}y - 1$$

where $g_1^3(y) = y$ is the *qualifying* constraint for this Lagrange function.

Since $0 \leq y \leq 1$, tight bounds on x can be obtained to be $0 \leq x \leq \frac{4}{3}$. Since $y > 0$, a valid underestimator to $L^2(x, y, \mu^2)$ for all y can be obtained by fixing x to its lower bound. Therefore, there are no binary variables, and consequently, the MILP formulation reduces to the same formulation as in Section 4.4. The solution of the resulting subproblem is $y = 0.2$, $\mu_B = -1.2667$.

At the end of this iteration, there are two candidate regions for further branching: (i) node 1 ($1 \leq y \leq 1.5$) with a lower bound of -1.5, and (ii) node 3 ($0 \leq y \leq 1$) with a lower bound of -1.2667. The former node is selected for further exploration.

Iteration 3

For $y = 1.5$, the primal problem has the solution $x = 1.5$, $\mu_1^2 = \frac{1}{12}$, $\mu_2^2 = 0$, $\mu_3^2 = 0$, with the objective value of -0.75. The Lagrange function from this problem is

$$L^2(x, y, \mu^2) = -x + xy - y + \frac{1}{12}(-6x + 8y - 3) = xg_1^2(y) - \frac{1}{3}y - \frac{1}{4}$$

where $g_1^2(y) = y - 1.5$ is the *qualifying* constraint for this Lagrange function.

For $y \leq 1.5$, the tightest bounds on x are $\frac{5}{6} \leq x \leq 1.5$. Again, only one relaxed dual problem needs to be solved, with a valid underestimator to $L^3(x, y, \mu^3)$ being used by fixing x to its upper bound. Therefore, the MILP is again identical to the original algorithm formulation, and has the solution $y = 1.25$, $\mu_B = -1.04167$.

At the end of this iteration, there are two candidate regions for further partitioning - (i) the region $0 \leq y \leq 1$, corresponding to node 3, with a lower bound of -1.26667, and (ii) the region $1 \leq y \leq 1.5$, corresponding to node 4, with the lower bound of

-1.04167. Following the criterion of selecting the region with the best lower bound, node 3 is chosen for further exploration.

Iteration 4

For $y = 0.2$, the primal problem has the solution $x = 1.0667$, $\mu_1^4 = 0$, $\mu_2^4 = 0.2667$, $\mu_3^4 = 0$, with the objective value of -1.05333. Note that the solution of this problem provides an upper bound that is lower than the lower bound for node 4 (which is -1.04167). Therefore, node 4 can be immediately fathomed, i.e., removed from consideration for any further refinement or exploration.

The Lagrange function from the current primal problem is

$$L^4(x, y, \mu^4) = -x + xy - y + 0.2667(3x - y - 3) = xg_1^4(y) - 1.2667y - 0.8$$

where $g_1^4(y) = y - 0.2$ is the *qualifying* constraint for this Lagrange function.

For this iteration, the relaxed dual subproblems are solved in the region $0 \leq y \leq 1.0$, and try to provide refined lower bounds by partitioning the region further. The tightest bounds for x in this region are $0 \leq x \leq \frac{4}{3}$.

Unlike the previous two iterations, it is necessary to partition the current region since $-0.2 \leq g_1^4(y) \leq 1.3$. Therefore, the MILP in this iteration takes the form:

$$
\begin{aligned}
\min_{y, \mu_B} \quad & \mu_B \\
\mu_B \geq \ & \frac{1}{3}y - \frac{4}{3} \\
g_1^1(y) = \ & y - 1 \leq 0 \\
\mu_B \geq \ & -\frac{4}{3}y - 1 \\
\mu_B \geq \ & t_1^4 - 1.2667y \\
t_1^4 \geq \ & -0.26667\alpha_1^4 \\
t_1^4 \geq \ & 1.3333(g_1^4 - (1 - \alpha_1^4) \cdot 1.0667) \\
-0.2\alpha_1^4 \leq \ & g_1^4 \leq (1 - \alpha_1^4) \cdot 0.8 \\
g_1^4 = \ & y - 0.2 \\
0 \leq \ & y \leq 1.5
\end{aligned}
$$

The solution of this problem is $y = 0.333$, $\mu_B = -1.222$, $\alpha_1^4 = 0$.

Thus, the MILP algorithm produces the *exact* sequence of solutions given by the original branch and bound algorithm. As in Section 4.5, this algorithm also takes 18 iterations to converge.

Remark 5.5 Note that in this example, there is no arguable advantage to using the MILP formulation, since it needs to be solved for both combinations of α_1 at each iteration. However, for problems with more than one *connected* variable, it is obvious that this formulation can offer a major advantage over the original formulation. This is because at each iteration, no more than 2 MILP problems need to be solved. Although these problems are bigger in size and more complex than the original relaxed dual subproblems, their structure is such that finding their solution is not really dependent on the presence of the binary variables, and a good MILP solver can be expected to solve them very efficiently. At the same time, they feature the key advantage of not having to solve the full set of subproblems at each iteration.

It should be noted, however, that the convenience of solving just one compact problem is achieved at the expense of problem size. Because all possible solutions of the relaxed dual problem have to be incorporated in the GOP/MILP formulation, the result is a much larger problem to solve. A number of constraints and variables need to be used to implicitly represent all the possible bound combinations. For large problems, this could cause difficulties, although the availability of increasingly fast MILP solvers makes this less of a drawback.

6   A LINEAR BRANCHING SCHEME FOR THE GOP ALGORITHM

In both the GOP and GOP/MILP algorithms, the *qualifying* constraints (i.e., the gradients of the Lagrange function) are used to partition the y-space. The reduction properties presented in Section 4 can provide a significant reduction in the number of *connected* variables and subsequently the number of partitions. However, in the worst case, the number of subproblems solved still increases exponentially with the number of *connected* variables. It is then natural to ask the following question: *Is it possible to develop a valid lower bound at each iteration using only a linearly increasing number of relaxed dual subproblems?* In this section, we present one branching scheme that achieves this goal. This scheme originates from the study of Barmish *et al* (1995a, 1995b) on the stability of polytopes of matrices of robust control systems.

6.1 Reformulation of Qualifying Constraints

Consider the relaxed dual problem at the kth iteration. This problem has the constraint

$$\mu_B \geq L_0^k(y, \lambda^k, \mu^k) + \sum_{i=1}^{NI_C^k} x_i g_i^k(y).$$

Suppose that all the x variables are bounded between -1 and 1. If this is not the case, it can be achieved by use of the following linear transformation. Suppose that $x^L \leq x \leq x^U$. Then, define x' such that $-1 \leq x' \leq 1$, and

$$x = a \cdot x' + b$$

The substitution of the lower and upper bounds gives

$$x^L = a \cdot (-1) + b, \quad \text{and} \quad x^U = a \cdot (1) + b$$

leading to

$$a = \frac{x^U - x^L}{2}, \quad \text{and} \quad b = \frac{x^U + x^L}{2}$$

The variables x' can then be substituted for x using the above transformation, leading to a Lagrange function in y and x'. We will continue the presentation in this section by considering the case $-1 \leq x \leq 1$.

The following observation is now made:

(a) If $g_i^k(y) \geq 0$,

$$x_i g_i^k(y) \geq x_i^L g_i^k(y) \quad \Rightarrow \quad x_i g_i^k(y) \geq -g_i^k(y)$$

(b) If $g_i^k(y) \leq 0$,

$$x_i g_i^k(y) \geq x_i^U g_i^k(y) \quad \Rightarrow \quad x_i g_i^k(y) \geq g_i^k(y)$$

Combining these two cases leads to the inequality

$$x_i g_i^k(y) \geq -|g_i^k(y)|$$

and

$$\mu_B \geq L_0^k(y, \lambda^k, \mu^k) - \sum_{i=1}^{NI_C^k} |g_i^k(y)| \tag{3.19}$$

The first term on the right hand side is convex, and can remain unaltered. Consider now the summation term. Using the concept of the infinity norm, (3.19) can be written as

$$\mu_B \geq L_0^k(y, \lambda^k, \mu^k) - NI_C^k \cdot \left(\max_i |g_i^k(y)|\right) \tag{3.20}$$

For any value of y, there is some $j \in 1, \ldots, NI_C^k$ such that

$$|g_j^k(y)| = \max_{i=1,\ldots,NI_C^k} |g_i^k(y)|$$

implying that

$$|g_j^k(y)| \geq |g_i^k(y)|, \quad i = 1, \ldots, NI_C^k \tag{3.21}$$

Consider the following two possibilities:

(a) If $g_j^k(y) \geq 0$, then $|g_j^k(y)| = g_j^k(y)$, and (3.21) reduces to the two inequalities

$$\begin{aligned} g_j^k(y) &\geq g_i^k(y) \\ g_j^k(y) &\geq -g_i^k(y) \end{aligned} \Bigg\} \quad i = 1, \ldots, NI_C^k, \, i \neq j \tag{3.22}$$

and (3.20) becomes

$$\mu_B \geq L_0^k(y, \lambda^k, \mu^k) - NI_C^k \cdot g_j^k(y)$$

(b) If $g_j^k(y) \leq 0$, then $|g_j^k(y)| = -g_j^k(y)$, and (3.21) reduces to the two inequalities

$$\begin{aligned} g_j^k(y) &\leq g_i^k(y) \\ g_j^k(y) &\leq -g_i^k(y) \end{aligned} \Bigg\} \quad i = 1, \ldots, NI_C^k, \, i \neq j \tag{3.23}$$

and (3.20) becomes

$$\mu_B \geq L_0^k(y, \lambda^k, \mu^k) + NI_C^k \cdot g_j^k(y)$$

The two cases presented above indicate how the summation in (3.19) can be replaced by a linear term when $g_j^k(y)$ represents the maximum of all the *qualifying* constraints at a given value of y. This concept can then be extended to cover the entire region for y. To do this, the above procedure needs to be repeated for all values of j, resulting in $2 \times NI_C^k$ subproblems that need to be solved in order to properly underestimate the Lagrange function at all values of y.

Remark 6.1 It should be noted that with the use of the linear branching scheme, the same space in y is now spanned by a linear number of underestimators (as opposed

to an exponential number in the original algorithm). Therefore, the tightness of these underestimators will be less than with the original algorithm. Therefore, at the end of each iteration, the lower bounds obtained from the dual problems with the linear branching scheme will be looser than those obtained with the original algorithm, resulting in an increase in the number of iterations required for convergence. At the same time, the number of subproblems solved at each iteration is vastly reduced. Therefore, the total computational effort required for the entire algorithm is likely to be much smaller with the linear branching scheme.

6.2   Illustration

Consider the following problem:

$$
\begin{aligned}
\min_{x,y} \quad & -x_1 y_1 - x_2 y_2 \\
s.t. \quad & x_1 - y_1 = 0 \\
& x_2 - y_2 = 0 \\
& -1 \le x, y \le 1
\end{aligned}
$$

Suppose that the GOP algorithm is applied to this problem, with the starting point of $y = 0$. The first primal problem has the solution $x = 0$, $\lambda_1^1 = 0$ and $\lambda_2^1 = 0$. This leads to the following constraint in the first relaxed dual problem:

$$
\begin{aligned}
\mu_B \ge \ & x_1(0 - y_1) + x_2(0 - y_2) \\
\ge \ & -|0 - y_1| - |0 - y_2|
\end{aligned}
$$

where $g_1^1(y) = 0 - y_1$ and $g_2^1(y) = 0 - y_2$ are the two *qualifying* constraints. The region in the y variables, as well as its division using these *qualifying* constraints as used by the original GOP and GOP/MILP algorithms, is shown in Figure 4(a). Note that the four regions A, B, C and D represent the four relaxed dual subproblems solved by the original algorithms.
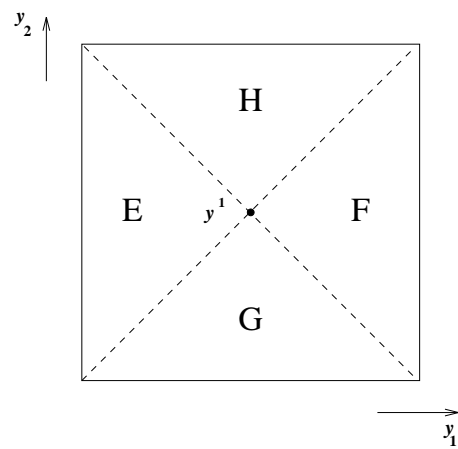
Suppose that $|g_1^1(y)| \ge |g_2^1(y)|$. There are two possibilities:

(a) $g_1^1(y) \ge 0$. Then, the use of (3.22) results in

$$
\begin{aligned}
y_1 - y_2 \ & \le \ 0 \\
y_1 + y_2 \ & \le \ 0 \\
y_1 \ & \le \ 0
\end{aligned}
$$

$g_1^1(y) = 0$

y_2

C

A

y^1

$g_2^1(y) = 0$

D

B

y_1
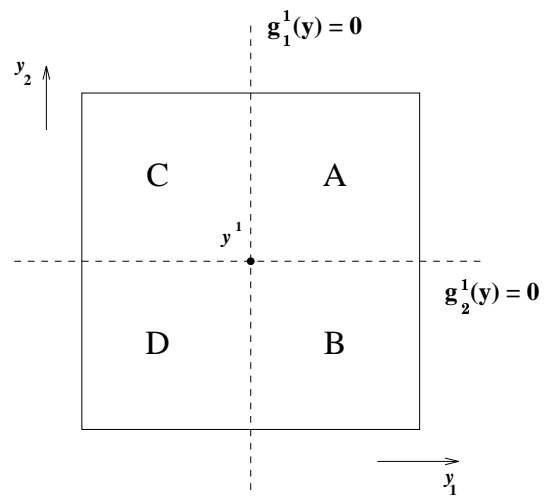
(a) Original *qualifying* constraints

y_2

H

E

y^1

F

G

y_1

(b) Transformed *qualifying* constraints

Figure 4 Transformed qualifying constraints

The region of y described by these constraints is shown as region E in Figure 4(b). The corresponding constraint for the relaxed dual problem is given by

$$\mu_B \geq -2g_1^1(y) = 2y_1$$

(b) $g_1^1(y) \leq 0$. Then, the use of (3.23) results in

$$y_1 - y_2 \geq 0$$
$$y_1 + y_2 \geq 0$$
$$y_1 \geq 0$$

These equations describe region F in Figure 4(b). The corresponding constraint for the relaxed dual problem is given by

$$\mu_B \geq 2g_1^1(y) = -2y_1$$

Similarly, when $|g_2^1(y)| \geq |g_1^1(y)|$, there are two possibilities:

(a) $g_2^1(y) \geq 0$. Then, the use of (3.22) results in

$$y_2 - y_1 \leq 0$$
$$y_2 + y_1 \leq 0$$
$$y_2 \leq 0$$

The region of y described by these constraints is shown as region G in Figure 4(b). The corresponding constraint for the relaxed dual problem is given by

$$\mu_B \geq -2g_2^1(y) = 2y_2$$

(b) $g_2^1(y) \leq 0$. Then, the use of (3.23) results in

$$y_2 - y_1 \geq 0$$
$$y_2 + y_1 \geq 0$$
$$y_2 \geq 0$$

These equations describe region H in Figure 4(b). The corresponding constraint for the relaxed dual problem is given by

$$\mu_B \geq 2g_1^1(y) = -2y_1$$

Thus, it can be seen that the use of the equations (3.22) and (3.23) result in a new set of partitions of the region in y. For this example, there are still 4 partitions, so there is no reduction in the number of subproblems to be solved. However, when the number of *connected* variables is more than 2, the use of these transformations will result in a linearly increasing (as opposed to exponentially increasing) number of subproblems at each iteration. For example, when there are 10 *connected* variables, the new partitioning scheme requires 20 relaxed dual subproblems as opposed to 1024 for the original GOP algorithm.

7   CONCLUSIONS

This paper has focussed on presenting the GOP Algorithm of Floudas and Visweswaran (1990, 1993) in a branch and bound framework. This framework is based upon branching on the gradients of the Lagrange function, and is considerably simpler than the original cutting plane algorithm. The primary advantage of the framework is in simplicity of implementation. In particular, the selection of previous Lagrange functions as cuts for current dual problems is considerably simplified. Moreover, the framework allows for the use of a mixed integer formulation that implicitly enumerates the solutions of all the dual subproblems. This paper has also considered the issue of reducing the number of subproblems at each iteration, and in Section 6, a new partitioning scheme was presented that requires only a linear number of subproblems. This is a significant reduction from the exponential number of subproblems required by the original algorithm.

The new algorithms have been implemented in a package **cGOP** (Visweswaran and Floudas, 1995a) and applied to a large number of problems. The results of these applications can be found in the companion paper (Visweswaran and Floudas, 1995b).

Acknowledgements

REFERENCES

[1] F. A. Al-Khayyal and J. E. Falk. Jointly constrained biconvex programming. *Math. of Oper. Res.*, 8(2):273, 1983.

[2] F. Archetti and F. Schoen. A Survey on the Global Optimization Problem: General Theory and Computational Approaches. *Annals of Operations Research*, 1:87, 1984.

[3] B. R. Barmish, C. A. Floudas, H. V. Hollot, and R. Tempo. A Global Linear Programming Solution to Some Open Robustness Problems Including Matrix Polytope Stability. *IEEE Transactions on Automatic Control*, 1995a. Submitted for Publication.

[4] B. R. Barmish, C. A. Floudas, H. V. Hollot, and R. Tempo. A Global Linear Programming Solution to Some Open Robustness Problems Including Matrix Polytope Stability. *Proceedings of the ACC 95, Seattle, June 21-23*, 1995b. To appear.

[5] L.C.W. Dixon and G.P. Szego. *Towards global optimisation*. North-Holland, Amsterdam, 1975.

[6] L.C.W. Dixon and G.P. Szego. *Towards global optimisation 2*. North-Holland, Amsterdam, 1978.

[7] M. A. Duran and I. E. Grossmann. An outer approximation algorithm for a class of mixed-integer nonlinear programs. *Mathematical Programming*, 36:307, 1986.

[8] C. A. Floudas, A. Aggarwal, and A. R. Ciric. Global optimum search for nonconvex NLP and MINLP problems. *Comp. & Chem. Eng.*, 13(10):1117, 1989.

[9] C. A. Floudas and P. M. Pardalos. *A Collection of Test Problems for Constrained Global Optimization Algorithms*, volume 455 of *Lecture Notes in Computer Science*. Springer-Verlag, Berlin, Germany, 1990.

[10] C. A. Floudas and P. M Pardalos. *Recent Advances in Global Optimization*. Princeton Series in Computer Science. Princeton University Press, Princeton, New Jersey, 1992.

[11] C. A. Floudas and V. Visweswaran. A global optimization algorithm (GOP) for certain classes of nonconvex NLPs: I. theory. *Comp. & Chem. Eng.*, 14:1397, 1990.

[12] C. A. Floudas and V. Visweswaran. A primal-relaxed dual global optimization approach. *J. Optim. Theory and Appl.*, 78(2):187, 1993.

[13] A. M. Geoffrion. Generalized Benders Decomposition. *J. Optim. Theory and Appl.*, 10(4):237, 1972.

[14] R. Horst and H. Tuy. *Global Optimization: Deterministic Approaches.* Springer-Verlag, Berlin, Germany, 1990.

[15] W. B. Liu and C. A. Floudas. A Remark on the GOP Algorithm for Global Optimization. *J. Global Optim.*, 3:519, 1993.

[16] W. B. Liu and C. A. Floudas. Convergence of the GOP Algorithm for a Large Class of Smooth Optimization Problems. *Journal of Global Optimization*, 6:207, 1995.

[17] J. Mockus. *Bayesian Approach to Global Optimization*. Kluwer Academic Publishers, Amsterdam, Holland, 1989.

[18] 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.

[19] P.M. Pardalos and J.B. Rosen. Methods for global concave minimization: A bibliographic survey. *SIAM Review*, 28(3):367, 1986.

[20] A. Torn and A. Zilinskas. *Global Optimization*, volume 350 of *Lecture Notes in Computer Science.* Springer-Verlag, Berlin, Germany, 1989.

[21] V. Visweswaran and C. A. Floudas. A Global optimization algorithm (GOP) for certain classes of nonconvex NLPs: II. Application of theory and test problems. *Comp. & Chem. Eng.*, 14:1419, 1990.

[22] V. Visweswaran and C. A. Floudas. New properties and computational improvement of the GOP algorithm for problems with quadratic objective function and onstraints. *J. Global Optim.*, 3(3):439, 1993.

[23] V. Visweswaran and C. A. Floudas. *cGOP: A User's Guide*. Princeton University, Princeton, New Jersey, 1995a.

[24] V. Visweswaran and C. A. Floudas. Computational Results For an Efficient Implementation of the GOP Algorithm and Its Variants. In *Global Optimization in Engineering Design*, (Ed.) I. E. Grossmann, Kluwer Book Series in Nonconvex Optimization and Its Applications, Chapter 4, 1995b.