# Convex Programming Methods for Global Optimization 

J. N. Hooker<br>GSIA, Carnegie Mellon University, Pittsburgh, USA, jh38@andrew.cmu.edu

November 2003


#### Abstract

We investigate some approaches to solving nonconvex global optimization problems by convex nonlinear programming methods. We assume that the problem becomes convex when selected variables are fixed. The selected variables must be discrete, or else discretized if they are continuous. We provide a survey of disjunctive programming with convex relaxations, logic-based outer approximation, and logic-based Benders decomposition. We then introduce a branch-and-bound method with convex quasi-relaxations (BBCQ) that can be effective when the discrete variables take a large number of real values. The BBCQ method generalizes work of Bollapragada, Ghattas and Hooker on structural design problems. It applies when the constraint functions are concave in the discrete variables and have a weak homogeneity property in the continuous variables.


We address global optimization problems that become convex when selected variables are fixed. If these variables are discrete, the constraints can be reformulated as logical disjunctions of convex constraints. If some of the selected variables are not discrete, we discretize them in order to obtain an approximate global solution.

The motivation for this approach is to take advantage of highly developed nonlinear programming methods for convex problems, as well as branch-andbound methods for discrete problems. A branch-and-bound method chooses the appropriate disjunct in each constraint. Nonlinear programming is applied to the convex subproblem that results when the disjuncts are chosen.

We present four variations of this general approach. Two of them are most practical when the discrete variables do not take a large number of possible values: (a) disjunctive programming with convex relaxations, and (b) logic-based outer approximation. The disjunctive programming model can also be solved as a mixed integer/nonlinear programming (MINLP) problem. When there are a large number of discrete values, as when some discrete variables represent discretized continuous variables, one can turn to methods that do not require explicit representation of the disjunctions: (c) logic-based Benders decomposition, and (d) branch and bound with convex quasi-relaxations (BBCQ). The
convergence rate of the Benders method depends heavily on the problem structure, however. BBCQ is intended for problems in which the discrete variables are real-valued. It does not rely on decomposition but requires that the constraint functions satisfy certain properties.

This paper begins with a summary of the first three methods, which are developed elsewhere. It then introduces the BBCQ method as a formalization and generalization of a technique applied by Bollapragada, Ghattas and Hooker to structural design problems [1]. This application is presented at the end of the paper as an illustration of disjunctive programming and BBCQ.

## 1 General Form of the Problem

We solve problems of the form

$$
\begin{array}{ll}
\min & x_{0} \\
\text { subject to } & g^{j}\left(x, y_{j}\right) \leq 0, \quad j \in J \\
& L(y)  \tag{1}\\
& x \in \mathbb{R}^{n}, y_{j} \in Y_{j}, j \in J
\end{array}
$$

where $g^{j}\left(x, y_{j}\right)$ is a vector of functions and $L(y)$ is a logical constraint on possible values of the discrete variables $y_{j}$. If some of the $y_{j}$ are continuous, we discretize them by converting $Y_{j}$ to a finite set. We assume that when each $y_{j}$ is fixed to some $\bar{y}_{j} \in Y_{j}$ we obtain the convex subproblem:

$$
\begin{array}{ll}
\min & x_{0} \\
\text { subject to } & g^{j}\left(x, \bar{y}_{j}\right) \leq 0, \quad j \in J  \tag{2}\\
& x \in \mathbb{R}^{n}
\end{array}
$$

It is convex in the sense that each $g^{j}\left(x, \bar{y}_{j}\right)$ is a vector of convex functions of $x$.
We assume without loss of generality that the objective function is a single variable $x_{0}$, since $x_{0}$ can be defined in the constraints. We also suppose that each constraint contains only one discrete variable $y_{j}$. Many problems naturally occur in this form. Problems that do not can in principle be put into this form by a change of variables. Thus a constraint $g^{j}\left(x, y_{1}, \ldots, y_{m}\right) \leq 0$ can be written $g^{j}\left(x, y^{j}\right) \leq 0$, where $y^{j}=\left(y_{1}^{j}, \ldots, y_{m}^{j}\right)$ is regarded as a single variable. The variables $y^{j}$ can now be related by the logical constraints $y^{j}=y^{1}$ for all $j \in J$. For instance, the constraints $x+y_{1}+y_{2} \geq b$ and $x+y_{2}+y_{3} \geq b$ can be rewritten $x+y_{1}^{1}+y_{2}^{1} \geq b$ and $x+y_{2}^{2}+y_{3}^{2} \geq b$ by adding the constraint $y_{2}^{1}=y_{2}^{2}$.

## 2 Structural Design Example

We use a simple structural design problem to illustrate all four solution methods. The two pillars of Fig. 1 support the two horizontal platforms shown. Pillar 1 bears a weight of 10 and pillar 2 a weight of 20 . The weight on pillar 1 causes


Fig. 1. A simple structural design problem.
it to compress an amount $x_{1}$ given by Hooke's law, $x_{1} y_{1}=10$, where $y_{1}$ is the thickness (cross-sectional area) of pillar 1, and similarly for pillar 2. Thus the top platform is displaced downward a distance $x_{1}+x_{2}$, and the bottom platform a distance $x_{2}$. The cost of steel in pillar $j$ is $300 y_{j}$, and we impose a penalty $\left(x_{1}+x_{2}\right)^{2}+x_{2}^{2}$ for displacement. The objective is to choose thicknesses for the two pillars so as to minimize the sum of the cost and the penalty:

$$
\begin{align*}
& \operatorname{minimize} \\
& \text { subject to } x_{1} y_{2}=10, x_{1}+300 y_{2}+\left(x_{1}+x_{2}\right)^{2}+x_{2}^{2}  \tag{3}\\
& \\
& \quad x_{j} \geq 0, y_{j} \in\{1,2\} \text { for } j=1,2
\end{align*}
$$

To simplify the example we consider two discrete thicknesses (1 and 2). Near the end of the paper we also consider the case in which $y_{1}$ and $y_{2}$ are continuous variables.

The model (3) is chosen for the sole purpose of illustrating the algorithms described below. It is not intended to be a realistic or proper engineering model for a structural design problem. In particular, the nonlinear penalty terms are inserted to show how they are treated in the algorithms, not because they necessarily represent the best way to limit displacement. A more realistic structural design model is presented in Section 8.

To verify that the model (3) has the general form (1), note that it can be written

$$
\begin{align*}
& \operatorname{minimize} z_{0} \\
& \text { subject to } z_{1}+z_{2}+\left(x_{1}+x_{2}\right)^{2}+x_{2}^{2}-x_{0} \leq 0 \\
& {\left[\begin{array}{c}
300 y_{1}-z_{1} \\
10-x_{1} y_{1}
\end{array}\right] \leq\left[\begin{array}{l}
0 \\
0
\end{array}\right] }  \tag{4}\\
& {\left[\begin{array}{c}
300 y_{2}-z_{2} \\
20-x_{2} y_{2}
\end{array}\right] \leq\left[\begin{array}{l}
0 \\
0
\end{array}\right] } \\
& x_{j} \geq 0, \quad y_{j} \in\{0,1\} \text { for } j=1,2
\end{align*}
$$

The model is clearly convex when the $y_{j}$ s are fixed.

## 3 Disjunctive Formulation

A straightforward but generally impractical way to solve (1) is by a branch-and-bound method that branches on the $y_{j}$ and solves a continuous relaxation of the problem at each node of the branching tree. The difficulty is that these continuous problems are in general nonconvex.

To obtain convex relaxations, we write (1) as a disjunctive programming problem by creating a disjunct for each possible value of $y_{j}$.

$$
\begin{array}{ll}
\min & x_{0} \\
\text { subject to } & \bigvee_{v \in Y_{j}}\left[\begin{array}{c}
y_{j}=v \\
g^{j}(x, v) \leq 0
\end{array}\right], j \in J  \tag{5}\\
& L(y) \\
& x \in \mathbb{R}^{n}
\end{array}
$$

The functions $g^{j}(x, v)$ are convex because the second argument is fixed. They may also simplify in form. In some cases singularities disappear, as for example when

$$
g^{j}\left(x, y_{j}\right)=\left[\begin{array}{c}
x_{1}-1 / y_{1} \\
x_{1}-x_{2} 0
\end{array}\right] \leq\left[\begin{array}{l}
0 \\
0
\end{array}\right]
$$

can be written simply $x_{1}-x_{2} \leq 0$ for $y_{j}=0$.

Example. The disjunctive formulation for the structural design example of the previous section is

$$
\begin{align*}
\operatorname{minimize} & x_{0} \\
\text { subject to } & z_{1}+z_{2}+\left(x_{1}+x_{2}\right)^{2}+x_{2}^{2}-x_{0} \leq 0 \\
& {\left[\begin{array}{c}
y_{1}=1 \\
300-z_{1} \leq 0 \\
10-x_{1} \leq 0
\end{array}\right] \vee\left[\begin{array}{c}
y_{1}=2 \\
600-z_{1} \leq 0 \\
10-2 x_{1} \leq 0
\end{array}\right] }  \tag{6}\\
& {\left[\begin{array}{c}
y_{2}=1 \\
300-z_{2} \leq 0 \\
20-x_{2} \leq 0
\end{array}\right] \vee\left[\begin{array}{c}
y_{2}=2 \\
600-z_{2} \leq 0 \\
20-2 x_{2} \leq 0
\end{array}\right] } \\
& x_{1}, x_{2} \geq 0
\end{align*}
$$

Note that the first disjunction contains only one disjunct, since no $y_{j}$ appears in the corresponding constraint. (For convenience we regard the first constraint function as $g^{0}\left(x, y_{0}\right)$, where $y_{0}$ does not actually appear.) All of the disjuncts are convex, and in fact all but the first are linear. The subproblem (2) becomes

$$
\begin{array}{cl}
\operatorname{minimize} & x_{0} \\
\text { subject to } & z_{1}+z_{2}+\left(x_{1}+x_{2}\right)^{2}+x_{2}^{2}-x_{0} \leq 0 \\
& 300 \bar{y}_{1}-z_{1} \leq 0 \\
& 10-x_{1} \bar{y}_{1} \leq 0  \tag{7}\\
& 300 \bar{y}_{2}-z_{2} \leq 0 \\
& 20-x_{2} \bar{y}_{2} \leq 0 \\
& x_{1}, x_{2} \geq 0
\end{array}
$$

## 4 Disjunctive Programming with Convex Relaxations

A branch-and-bound method can be practical for the disjunctive programming problem (5) when it is possible to devise a convex relaxation at each node of the search tree. Two such relaxations, based on big- $M$ and convex hull formulations, are presented here.

Branch and bound proceeds by branching on the alternatives in the disjunctions of (5). At each node of the search tree, some disjuncts have been selected by prior branching, and these are imposed as constraints. The disjunctions on which the algorithm has not yet branched are relaxed. A lower bound is obtained by solving a convex problem that minimizes $x_{0}$ subject to the imposed disjuncts and the relaxed disjunctions. The lower bound is used to prune the search as is normally done in branch-and-bound search (see $[8,10]$ for details).

A closely related approach is to apply an MINLP method to a 0-1 model of the disjunctive model (5), which results from imposing an integrality condition on either the big- $M$ or the convex hull relaxation of (5).

The big- $M$ relaxation introduces a variable $\beta_{j v}$ for each $v \in Y_{j}$, where $\beta_{j v}=1$ is interpreted as indicating $y_{j}=v$. It is assumed that there are bounds $x^{L} \leq$
$x \leq x^{U}$ on $x$. Let $L(\beta)$ be an inequality encoding of the logical constraints $L(y)$ [2]. The big- $M$ relaxation of (5) is:

$$
\begin{array}{ll}
\min & x_{0} \\
\text { subject to } & g^{j}(x, v) \leq M^{j v}\left(1-\beta_{j v}\right), \text { all } v \in Y_{j}, j \in J \\
& \sum_{v \in Y_{j}} \beta_{j v}=1, \beta_{j v} \geq 0, \text { all } v \in Y_{j}, j \in J  \tag{8}\\
& L(\beta), x^{L} \leq x \leq x^{U} \\
& 0 \leq \beta_{j v} \leq 1, \text { all } v \in Y_{j}, j \in J
\end{array}
$$

where $M^{j v}$ is a vector of valid upper bounds on the component functions of $g^{j}(x, v)$, given that $x^{L} \leq x \leq x^{U}$. This relaxation is clearly convex.

One can solve (5) by using relaxation (8) at each node, where $J$ in (8) corresponds to the set of disjunctions on which the algorithm has not yet branched. Alternatively, one can apply an MINLP algorithm to the $0-1$ model obtained by replacing $\beta_{j v} \in[0,1]$ in (8) with $\beta_{j v} \in\{0,1\}$, where $J$ corresponds to the original set of disjunctions.

The bounds $M^{j v}$ should be the tightest that can be practicably obtained. One valid bound is

$$
\begin{equation*}
M_{i}^{j v}=\max _{x^{L} \leq x \leq x^{U}}\left\{g_{i}^{j}(x, v)\right\} \tag{9}
\end{equation*}
$$

but the tightest bound is

$$
M_{i}^{j v}=\max _{v^{\prime} \in Y_{j} \backslash\{v\}}\left\{\max _{x^{L} \leq x \leq x^{U}}\left\{g_{i}^{j}(x, v) \mid g^{j}\left(x, v^{\prime}\right) \leq 0\right\}\right\}
$$

Example. Consider the disjunction

$$
\left[\begin{array}{c}
y_{j}=1  \tag{10}\\
x_{1}^{2}+x_{2}^{2}-1 \leq 0
\end{array}\right] \vee\left[\begin{array}{c}
y_{j}=2 \\
\left(x_{1}-2\right)^{2}+x_{2}^{2}-1 \leq 0
\end{array}\right]
$$

with $x_{1} \in[-1,3]$ and $x_{2} \in[-1,1]$. The feasible set for (10) is the union of the discs in Fig. 2. Setting $M^{j 1}=M^{j 2}=9$ as given by (9), we get the big- $M$ relaxation

$$
\begin{aligned}
& x_{1}^{2}+x_{2}^{2}-1 \leq 9(1-\beta) \\
& \left(x_{1}-2\right)^{2}+x_{2}^{2} \leq 9 \beta \\
& \beta \in[0,1]
\end{aligned}
$$

The elliptical area in Fig. 2 depicts the projection of the relaxation onto the $x$-space. The projection is described by $x_{1}^{2}+\left(x_{1}-2\right)^{2}+2 x_{2}^{2} \leq 11$.

A second convex relaxation for (5), based on convex hull descriptions of the disjunctions, was developed by Stubbs and Mehrotra [13] and Grossmann and Lee [6]. It is generally tighter than the big- $M$ relaxation but requires that we introduce for each disjunction $j$ a new continuous variable $x^{j v}$ for each $v \in Y_{j}$.


Fig. 2. A big- $M$ relaxation of a union of two discs.

The convex hull relaxation for a disjunction

$$
\begin{equation*}
\bigvee_{v \in Y_{j}} g^{j}(x, v) \leq 0 \tag{11}
\end{equation*}
$$

can be derived as follows. We assume that $x$ and $g^{j}$ are bounded; that is, $x \in$ $\left[x^{L}, x^{U}\right]$, and $g^{j}(x) \in[-L, L]$ for $x \in\left[x^{L}, x^{U}\right]$. We wish to characterize all points $x$ that can be written as a convex combination of points $\hat{x}^{j v}$ that respectively satisfy the disjuncts of (11). Thus we have

$$
\begin{aligned}
& x=\sum_{v \in Y_{j}} \beta_{j v} \hat{x}^{j v} \\
& g^{j}\left(\hat{x}^{j}, v\right) \leq 0, \text { all } v \in Y_{j} \\
& x^{L} \leq \hat{x}^{j} \leq x^{U} \\
& \sum_{v \in Y_{j}} \beta_{j v}=1, \quad \beta_{j v} \geq 0, \quad \text { all } v \in Y_{j}
\end{aligned}
$$

Using the change of variable $x^{j v}=\beta_{j v} \hat{x}^{j v}$, we obtain the relaxation

$$
\begin{align*}
& x=\sum_{v \in Y_{j}} x^{j v} \\
& g^{j}\left(\frac{x^{j v}}{\beta_{j v}}, v\right) \leq 0, \text { all } v \in Y_{j}  \tag{12}\\
& \beta_{j v} x^{L} \leq x^{j v} \leq \beta_{j v} x^{U}, \text { all } v \in Y_{j} \\
& \sum_{v \in Y_{j}} \beta_{j v}=1, \quad \beta_{j v} \geq 0, \text { all } v \in Y_{j}
\end{align*}
$$

The function $g^{j}\left(x^{j v} / \beta_{j v}, v\right)$ is in general nonconvex, but a classical result of convex analysis (e.g. [7]) implies that one can restore convexity by multiplying the second constraint of (12) by $\beta_{j v}$.

Theorem 1. Consider the set $S$ consisting of all $(x, \beta)$ with $\beta \in[0,1]$ and $x \in\left[\beta x^{L}, \beta x^{U}\right]$. If $g(x)$ is convex and bounded for $x \in\left[\beta x^{L}, \beta x^{U}\right]$, then

$$
h(x, \beta)= \begin{cases}\beta h(x / \beta) & \text { if } \beta>0 \\ 0 & \text { if } \beta=0\end{cases}
$$

is convex and bounded on $S$.

Proof. To show convexity of $h(x, \beta)$ we arbitrarily choose $\left(x^{1}, \beta_{1}\right),\left(x^{2}, \beta_{2}\right) \in$ $S$. Supposing first that $\beta_{1}, \beta_{2}>0$, we have convexity since

$$
\begin{aligned}
& h\left(\alpha x^{1}+(1-\alpha) x^{2}, \alpha \beta_{1}+(1-\alpha) \beta_{2}\right) \\
& =\left(\alpha \beta_{1}+(1-\alpha) \beta_{2}\right) g\left(\frac{\alpha x^{1}+(1-\alpha) x^{2}}{\alpha \beta_{1}+(1-\alpha) \beta_{2}}\right) \\
& =\left(\alpha \beta_{1}+(1-\alpha) \beta_{2}\right) g\left(\frac{\alpha \beta_{1}}{\alpha \beta_{1}+(1-\alpha) \beta_{2}} \frac{x^{1}}{\beta_{1}}+\frac{(1-\alpha) \beta_{1}}{\alpha \beta_{1}+(1-\alpha) \beta_{2}} \frac{x^{2}}{\beta_{2}}\right) \\
& \leq\left(\alpha \beta_{1}+(1-\alpha) \beta_{2}\right)\left[\frac{\alpha \beta_{1}}{\alpha \beta_{1}+(1-\alpha) \beta_{2}} g\left(\frac{x^{1}}{\beta_{1}}\right)+\frac{(1-\alpha) \beta_{1}}{\left.\alpha \beta_{1}+(1-\alpha) \beta_{2}\right)} g\left(\frac{x^{2}}{\beta_{2}}\right)\right] \\
& =\alpha h\left(x^{1}, \beta_{1}\right)+(1-\alpha) h\left(x^{2}, \beta_{2}\right)
\end{aligned}
$$

for any $\alpha \in[0,1]$, where the inequality is due to the convexity of $g(x)$. If $\beta_{1}=$ $\beta_{2}=0$, then
$h\left(\alpha x^{1}+(1-\alpha) x^{2}, \alpha \beta_{1}+(1-\alpha) \beta_{2}\right)=h(0,0)=\alpha h\left(x^{1}, \beta_{1}\right)+(1-\alpha) h\left(x^{2}, \beta_{2}\right)$
since $-\beta_{j} L \leq x^{j} \leq \beta_{j} L$ implies $x^{j}=0$. If $\beta_{1}=0$ and $\beta_{2}>0$, we have

$$
\begin{aligned}
& h\left(\alpha x^{1}+(1-\alpha) x^{2}, \alpha \beta_{1}+(1-\alpha) \beta_{2}\right) \\
& =h\left((1-\alpha) x^{2},(1-\alpha) \beta_{2}\right)=(1-\alpha) g\left(\frac{x^{2}}{\beta_{2}}\right) \\
& =\alpha h(0,0)+(1-\alpha) h\left(x^{2}, \beta_{2}\right)
\end{aligned}
$$

Finally, $h(x, \beta)=\beta g(x / \beta)$ is bounded because $\beta \in[0,1], x / \beta \in\left[x^{L}, x^{U}\right]$, and $g(x)$ is bounded for $x \in\left[x^{L}, x^{U}\right]$.

We now obtain the following convex relaxation for (5):

$$
\begin{array}{ll}
\min & x_{0} \\
\text { subject to } & x=\sum_{v \in Y_{j}} x^{j v}, \text { all } j \in J \\
& \beta_{j v} g^{j}\left(\frac{x^{j v}}{\beta_{j v}}, v\right) \leq 0, \text { all } v \in Y_{j}, j \in J  \tag{13}\\
& \beta_{j v} x^{L} \leq x^{j v} \leq \beta_{j v} x^{U}, \text { all } v \in Y_{j}, j \in J \\
& \sum_{v \in Y_{j}} \beta_{j v}=1, \beta_{j v} \geq 0, \text { all } v \in Y_{j}, j \in J \\
& L(\beta), \quad x, x^{j v} \in \mathbb{R}^{n}, \text { all } v \in Y_{j}, j \in J
\end{array}
$$

This is not a convex hull relaxation for (5) as a whole, but it provides a convex hull relaxation of each disjunction of (5).

Since $\beta_{j v}$ can vanish, it is common in practice to use the constraint

$$
\left(\beta_{j v}+\epsilon\right) g^{j}\left(\frac{x^{j v}}{\beta_{j v}+\epsilon}, v\right) \leq 0, \text { all } v \in Y_{j}, j \in J
$$

The introduction of $\epsilon$ preserves convexity. Grossmann and Lee [6] suggest using $\epsilon=10^{-4}$.

Example. The convex hull relaxation for the disjunction (10) is

$$
\begin{aligned}
& {\left[\begin{array}{l}
x_{1} \\
x_{2}
\end{array}\right]=\left[\begin{array}{l}
x_{11} \\
x_{21}
\end{array}\right]+\left[\begin{array}{l}
x_{12} \\
x_{22}
\end{array}\right]} \\
& \frac{x_{11}^{2}+x_{21}^{2}}{\beta+\epsilon} \leq \beta+\epsilon \\
& \frac{x_{12}^{2}+x_{22}^{2}}{1-\beta+\epsilon}-4 x_{12}+3(1-\beta+\epsilon) \leq 0 \\
& \beta\left[\begin{array}{l}
-1 \\
-1
\end{array}\right] \leq\left[\begin{array}{l}
x_{11} \\
x_{21}
\end{array}\right] \leq \beta\left[\begin{array}{l}
3 \\
1
\end{array}\right] \\
& (1-\beta)\left[\begin{array}{l}
-1 \\
-1
\end{array}\right] \leq\left[\begin{array}{l}
x_{12} \\
x_{22}
\end{array}\right] \leq(1-\beta)\left[\begin{array}{l}
3 \\
1
\end{array}\right] \\
& 0 \leq \beta \leq 1
\end{aligned}
$$

The bounds on $x^{j v}$ are redundant in this case and can be dropped. Figure 3 shows the projection of the feasible set of this relaxation onto the original $x$-space.


Fig. 3. The convex hull relaxation of a union of two discs.

Example. A big- $M$ relaxation for the structural design problem is

$$
\begin{array}{lll}
\operatorname{minimize} & x_{0} \\
\text { subject to } & z_{1}+z_{2}+\left(x_{1}+x_{2}\right)^{2}+x_{2}^{2}-x_{0} \leq 0 \\
& 300-z_{1} \leq 0 & 600-z_{1} \leq 300 \beta_{1} \\
10-x_{1} \leq 5\left(1-\beta_{1}\right) & 10-2 x_{1} \leq 0 \\
& 300-z_{2} \leq 0 & 600-z_{2} \leq 300 \beta_{2}  \tag{14}\\
20-x_{2} \leq 10\left(1-\beta_{2}\right) & 20-2 x_{2} \leq 0 \\
& 5 \leq x_{1} \leq 10 & 10 \leq x_{2} \leq 20 \\
& \beta_{1}, \beta_{2} \in[0,1] &
\end{array}
$$

The convex hull relaxation is

$$
\begin{align*}
& \text { minimize } x_{0} \\
& \text { subject to }\left[\begin{array}{l}
x_{1} \\
x_{2}
\end{array}\right]=\left[\begin{array}{l}
x_{11}+x_{12} \\
x_{21}+x_{22}
\end{array}\right]\left[\begin{array}{l}
z_{1} \\
z_{2}
\end{array}\right]=\left[\begin{array}{l}
z_{11}+z_{12} \\
z_{21}+z_{22}
\end{array}\right] \\
& z_{1}+z_{2}+\left(x_{1}+x_{2}\right)^{2}+x_{2}^{2}-x_{0} \leq 0 \\
& z_{11} \geq 300 \beta_{1} \quad z_{12} \geq 600\left(1-\beta_{1}\right)  \tag{15}\\
& x_{11} \geq 10 \beta_{1} \quad x_{12} \geq 10\left(1-\beta_{1}\right) \\
& z_{21} \geq 300 \beta_{2} \quad z_{22} \geq 600\left(1-\beta_{2}\right) \\
& x_{21} \geq 20 \beta_{2} \quad x_{22} \geq 20\left(1-\beta_{2}\right) \\
& \beta_{1}, \beta_{2} \in[0,1]
\end{align*}
$$

The disjunctive programming problems (14) and (15) can be solved directly by branch and bound, or by MINLP with $\beta_{j} \in\{0,1\}$ replacing $\beta_{j} \in[0,1]$. The optimal solution is $\left(y_{1}, y_{2}\right)=(1,2)$, meaning that the upper bar has size 1 and the lower bar size 2.

## 5 Logic-based Outer Approximation

One can use linear rather than convex nonlinear relaxations by modifying the outer approximation method for MINLP [3] to solve disjunctive programming problems, as shown by Türkay and Grossmann [14]. The drawback is that the linear relaxations must be updated and solved repeatedly.

Logic-based outer approximation solves a master problem containing firstorder approximations of the disjuncts of (5) to obtain a value $\bar{y}$ for $y$. It then solves the nonlinear but convex subproblem (2) to obtain a corresponding value for $x$. The first-order approximations are computed about the values of $x$ obtained in previous iterations. The process continues until optimal value of the master problem approximates the largest optimal subproblem value found so far.

Let $\left(x^{k}, y^{k}\right)$ for $k=1, \ldots, K$ be the solutions obtained by solving the master problem and subproblem in previous iterations. The master problem in iteration $K+1$ can be written

$$
\begin{array}{ll}
\min & x_{0} \\
\text { subject to } & y_{j}=v  \tag{16}\\
& \bigvee_{v \in Y_{j}}\left[\begin{array}{c}
g^{j}\left(x^{k}, v\right)+\nabla g^{j}\left(x^{k}, v\right)\left(x-x^{k}\right) \leq 0, \\
\text { all } k \in\{1, \ldots, K\} \text { with } y_{j}^{k}=v
\end{array}\right], \text { all } j \in J \\
& L(y), \quad x \in \mathbb{R}^{n}
\end{array}
$$

Since the disjuncts in (16) are linear, the relaxations (8) and (13) are likewise linear. One can therefore solve (16) by applying a mixed integer programming method to a $0-1$ formulation of (16). Again, either (8) or (13) can serve as a $0-1$ formulation if the variables $\beta_{j v}$ are treated as $0-1$ variables. The solution $y$ of (16) becomes $y^{K+1}$, and $x^{K+1}$ is an optimal solution of the subproblem (2) with $\bar{y}=y^{K+1}$.

In practice it is advantageous to obtain a warm start by solving the subproblem for several values of $\bar{y}$ before solving the first master problem.

Example. The master problem for the structural design problem in iteration $K+1$ is
minimize $x_{0}$
subject to $z_{1}+z_{2}+\left(x_{1}^{k}+x_{2}^{k}\right)^{2}+\left(x_{2}^{k}\right)^{2}+2\left(x_{1}^{k}+x_{2}^{k}\right)\left(x_{1}-x_{1}^{k}\right)$ $+2\left(x_{1}^{k}+2 x_{2}^{k}\right)\left(x_{2}-x_{2}^{k}\right)-x_{0} \leq 0, \quad k=1, \ldots K$

$$
\begin{align*}
& {\left[\begin{array}{c}
y_{1}=1 \\
300-z_{1} \leq 0 \\
10-x_{1} \leq 0
\end{array}\right] \vee\left[\begin{array}{c}
y_{1}=2 \\
600-z_{1} \leq 0 \\
10-2 x_{1} \leq 0
\end{array}\right]}  \tag{17}\\
& {\left[\begin{array}{c}
y_{2}=1 \\
300-z_{2} \leq 0 \\
20-x_{2} \leq 0
\end{array}\right] \vee\left[\begin{array}{c}
y_{2}=2 \\
600-z_{2} \leq 0 \\
20-2 x_{2} \leq 0
\end{array}\right]} \\
& x_{1}, x_{2} \geq 0
\end{align*}
$$

Since the disjuncts other than the first are already linear, no first order approximation is needed for them. We solve (17) by solving its big- $M$ formulation as an MILP:

$$
\begin{align*}
& \text { minimize } x_{0} \\
& \text { subject to } z_{1}+z_{2}+\left(x_{1}^{k}+x_{2}^{k}\right)^{2}+\left(x_{2}^{k}\right)^{2}+2\left(x_{1}^{k}+x_{2}^{k}\right)\left(x_{1}-x_{1}^{k}\right) \\
& +2\left(x_{1}^{k}+2 x_{2}^{k}\right)\left(x_{2}-x_{2}^{k}\right)-x_{0} \leq 0, \quad k=1, \ldots K \\
& 300-z_{1} \leq 0 \quad 600-z_{1} \leq 300 \beta_{1} \\
& 10-x_{1} \leq 5\left(1-\beta_{1}\right) \quad 10-2 x_{1} \leq 0  \tag{18}\\
& 300-z_{2} \leq 0 \quad 600-z_{2} \leq 300 \beta_{2} \\
& 20-x_{2} \leq 10\left(1-\beta_{2}\right) \quad 20-2 x_{2} \leq 0 \\
& 5 \leq x_{1} \leq 10 \quad 10 \leq x_{2} \leq 20 \\
& \beta_{1}, \beta_{2} \in\{0,1\}
\end{align*}
$$

To obtain a warm start, we solve the subproblem (7) with $\left(\bar{y}_{1}, \bar{y}_{2}\right)$ set to $y^{1}=$ $(1,1)$ and $y^{2}=(2,2)$, which yield $x^{1}=(20,20)$ and $x^{2}=(5,10)$. This results in the master problem

$$
\begin{align*}
& \text { minimize } x_{0} \\
& \text { subject to } z_{1}+z_{2}+2000+80\left(x_{1}-20\right)+120\left(x_{2}-20\right)-x_{0} \leq 0 \\
& z_{1}+z_{2}+325+30\left(x_{1}-5\right)+50\left(x_{2}-10\right)-x_{0} \leq 0 \\
& 300-z_{1} \leq 0 \quad 600-z_{1} \leq 300 \beta_{1} \\
& 10-x_{1} \leq 5\left(1-\beta_{1}\right) \quad 10-2 x_{1} \leq 0  \tag{19}\\
& 300-z_{2} \leq 0 \quad 600-z_{2} \leq 300 \beta_{2} \\
& 20-x_{2} \leq 10\left(1-\beta_{2}\right) \quad 20-2 x_{2} \leq 0 \\
& 5 \leq x_{1} \leq 10 \quad 10 \leq x_{2} \leq 20 \\
& \beta_{1}, \beta_{2} \in\{0,1\}
\end{align*}
$$

The master problem has solution $\beta=(1,0)$ and $x_{0}=1375$, which corresponds to $y^{3}=(1,2)$. This yields $x^{3}=(10,10)$ and $x_{0}=1400$ in the subproblem. The first order expansion of the nonlinear constraint about $x^{3}$ is

$$
z_{1}+z_{2}+500+40\left(x_{1}-10\right)+60\left(x_{2}-10\right)-x_{0} \leq 0
$$

This is added to the master problem, which now has solution $\beta=(1,0)$ corresponding to $y^{4}=(1,2)$, and $x_{0}=1400$. Since the master problem and subproblems converge, the algorithm terminates with $y=(1,2)$.

## 6 Logic-based Benders Decomposition

When a constraint in the disjunctive programming formulation contains many disjuncts, the number of variables in the relaxations (8) and (13) can become quite large. This can be avoided by applying logic-based Benders decomposition
to (5), which in effect uses a discrete relaxation of the problem and does not require an explicit formulation of the disjunctions $[8,11]$. However, the convergence rate is unpredictable.

In logic-based Benders, the master problem consists of Benders cuts that contain only the discrete variables $y_{j}$. At any point in the algorithm, the Benders cuts partially describe the projection of the original problem's feasible set onto the $y$-space.

In iteration $K$ the subproblem is (2) with $\bar{y}$ set to the solution $y^{K}$ of the current master problem. Let $\lambda^{K j}$ be the vector of Lagrange multipliers associated with constraint $j$ in the optimal solution of (2), and let $x_{0}^{K}$ be the optimal value of (2). Since constraints with vanishing Lagrange multipliers are inactive in the subproblem, we can state the following: whenever $\bar{y}_{j}$ is set to $y_{j}^{K}$ for all constraints $j$ with $\lambda^{K j} \neq 0$, the optimal value of the subproblem is still $x_{0}^{K}$. We generate a Benders cut that states this fact, and add it to the master problem for iteration $K+1$ :

$$
\begin{array}{ll}
\min & z \\
\text { subject to } & \bigwedge_{\substack{j \\
\lambda^{k j} \neq 0}}\left(y_{j}=y_{j}^{k}\right) \Longrightarrow\left(z \geq x_{0}^{k}\right), \quad k=1, \ldots, K  \tag{20}\\
& L(y)
\end{array}
$$

where $\Longrightarrow$ means "implies." For each $k$ the implication in (20) is the Benders cut generated in iteration $k$. The master problem is solved for $y^{K+1}$, and the process continues until the optimal value of (20) approximates the best subproblem value found so far.

The master problem can be solved by finite-domain constraint programming techniques or by converting it to an integer programming problem for an MILP solver.

In general, logic-based Benders cuts are obtained by solving the inference dual of the subproblem. This approach has been successfully applied to planning and scheduling problems in which the master problem is solved by integer programming and the subproblem by constraint programming [8, 9, 12]. There is little experience to date with continuous nonlinear subproblems, but decomposition is clearly more effective when most of the Lagrange multipliers vanish, since this results in stronger Benders cuts. When none of the multipliers vanish, the method reduces to exhaustive enumeration.

It is useful in practice to enhance the master problem with any known information about the $y_{j} \mathrm{~s}$, both valid constraints and "don't be stupid" constraints that exclude feasible but no optimal solutions. Such constraints can often be deduced from a practical understanding of the problem domain.

Example. The initial master problem for the structural design problem is

$$
\begin{align*}
& \operatorname{minimize} z \\
& \text { subject to } y_{1} \leq y_{2}  \tag{21}\\
& \qquad y_{1}, y_{2} \in\{1,2\}
\end{align*}
$$

Note the don't-be-stupid constraint $y_{1} \leq y_{2}$, which is based on the physical intuition that the top pillar will be no larger than the bottom pillar in any optimal solution. One optimal solution of (21) is $y^{1}=(1,1)$, with $z=-\infty$. Solving the subproblem (7) with $\bar{y}=(1,1)$, we obtain Lagrange multipliers $\lambda^{1}=(1,60)$ and $\lambda^{2}=(1,100)$, with $x_{0}^{1}=1900$. Since both multipliers are nonzero, the master problem becomes

$$
\begin{align*}
& \begin{array}{l}
\operatorname{minimize} \\
\text { subject to } \\
\\
\\
\\
\\
\\
\\
y_{1}, y_{2} \in(1,1) \Longrightarrow\{1,2\}
\end{array}
\end{align*}
$$

The next two iterations produce the additional Benders cuts

$$
\begin{aligned}
& y=(1,2) \Longrightarrow(z \geq 1400) \\
& y=(2,2) \Longrightarrow(z \geq 1525)
\end{aligned}
$$

When these are added to the master problem (22), we get $y^{4}=(1,2)$ with $z=$ 1400. The algorithm therefore terminates with $y=(1,2)$. In this small example the Benders approach is inefficient, since none of the Lagrange multiplers vanish, and each Benders cut excludes only one solution.

## 7 Branch and Bound with Convex Quasi-Relaxations

In the methods presented so far, the discrete variables need have no particular domain. However, in many applications the discrete variables are real-valued, as for example when they are discretized continuous variables. In such cases it may be advantageous to have a relaxation in both the $x$ and $y$ variables, so that one can branch on $y_{j}$ 's by splitting intervals. The solution of the relaxation would indicate where to split. Thus for example if $y_{j} \in\left[y_{j}^{L}, y_{j}^{U}\right]$ and the solution value of $y_{j}$ in the relaxation lies between discrete values $v, v^{\prime} \in Y_{j}$, one would split the interval into $\left[y_{j}^{L}, v\right]$ and $\left[v^{\prime}, y_{j}^{U}\right]$. If the solution value of $y_{j}$ lies at a bound $y_{j}^{L}$ or $y_{j}^{U}$, no more splitting is necessary. The relaxation may therefore accelerate the search not only by providing bounds, but by providing split points that lead more quickly to feasible solutions.

This strategy is practical, however, only when a convex relaxation involving the $y$ variables is available. Such a relaxation normally cannot be obtained by relaxing $y_{j}$ 's domain $Y_{j}$ to a continuous interval, since the resulting problem is in general nonconvex.

Even when a convex relaxation is unavailable, however, it may be possible to construct a convex quasi-relaxation that is equally useful for obtaining lower bounds. A quasi-relaxation of a problem $\min \{f(x) \mid x \in S\}$ is a problem $\min \left\{f^{\prime}(x) \mid x \in S^{\prime}\right\}$ with the property that for any $x \in S$, there exists an $x^{\prime} \in S^{\prime}$ for which $f\left(x^{\prime}\right) \leq f(x)$. It is clear that the optimal value of the quasi-relaxation,
if it exists, provides a valid lower bound on the optimal value of the original problem.

The following theorem provides conditions under which one may construct a convex quasi-relaxation for problem (1). Let function $g\left(x, y_{j}\right)$ be convex in $x$ when $g(x, v)$ is convex for any $v \in Y_{j}$. Also let $g\left(x, y_{j}\right)$ be semihomogeneous in $x$ if

$$
\begin{align*}
& g(\alpha x, v) \leq \alpha g(x, v) \text { for all } \alpha \in[0,1], x \in \mathbb{R}^{n}, v \in Y_{j}  \tag{a}\\
& g\left(0, y_{j}\right)=0 \text { for all } y_{j} \in Y_{j} \tag{b}
\end{align*}
$$

Theorem 2. Suppose each $g_{i}^{j}\left(x, y_{j}\right)$ in (1) is convex in $x$ and satisfies at least one of the following conditions:

1. $g_{i}^{j}\left(x, y_{j}\right)$ is convex.
2. $g_{i}^{j}\left(x, y_{j}\right)$ is semihomogeneous in $x$ and concave in $y_{j}$.

Let $(i, j)$ belong to $J_{1}$ when $g_{i}^{j}$ satisfies condition 1 and $J_{2}$ otherwise. Suppose also that $x^{L} \leq x \leq x^{U}$ and $y^{L} \leq y \leq y^{U}$. Then the following is a convex quasi-relaxation of (1):

$$
\begin{array}{cl}
\text { minimize } & x_{0} \\
\text { subject to } & \left.g_{i}^{j}\left(x, \alpha_{j} y_{j}^{L}+\left(1-\alpha_{j}\right) y_{j}^{U}\right)\right) \leq 0, \quad \text { all }(i, j) \in J_{1} \\
& g_{i}^{j}\left(x^{j 1}, y_{j}^{L}\right)+g_{j}^{i}\left(x^{j 2}, y_{j}^{U}\right) \leq 0, \quad \text { all }(i, j) \in J_{2} \\
& \alpha_{j} x^{L} \leq x^{j 1} \leq \alpha_{j} x^{U}, \text { all } j \in J \\
& \left(1-\alpha_{j}\right) x^{L} \leq x^{j 2} \leq\left(1-\alpha_{j}\right) x^{U} \quad \text { all } j \in J \\
& x=x^{j 1}+x^{j 2}, \quad \text { all } j \in J  \tag{e}\\
& x^{j 1}, x^{j 2} \in \mathbb{R}^{n}, \quad \alpha_{j} \in[0,1], \text { all } j \in J
\end{array}
$$

Furthermore, if each $\alpha_{j}$ is 0 or 1 in the optimal solution of (24), then (24) has the same optimal value as (1).

Proof. We first observe that (24) is convex. Constraint (a) is convex because $g_{i}^{j}\left(x, y_{j}\right)$ is convex for $(i, j) \in J_{1}$, and a convex function composed with an affine function is convex. Constraint (b) is convex because $g_{i}^{j}\left(x, y_{j}\right)$ is convex when $y_{j}$ is fixed. The remaining constraints are linear.

To show that (24) is a quasi-relaxation, take any feasible solution $(\bar{x}, \bar{y})$ of (1) and construct a feasible solution for (24) as follows. For each $j \in J$ choose $\alpha_{j} \in[0,1]$ so that $\bar{y}_{j}=\alpha_{j} y_{j}^{L}+\left(1-\alpha_{j}\right) y_{j}^{U}$. Set $x^{j 1}=\alpha_{j} \bar{x}_{j}, x^{j 2}=\left(1-\alpha_{j}\right) \bar{x}$, and $x=x^{j 1}+x^{j 2}$. To see that this produces a feasible solution of (24), note first that constraints (a) and (c)-(e) are satisfied by construction. Constraint (b) is also satisfied, since for $(i, j) \in J_{2}$ we have

$$
\begin{aligned}
& g_{i}^{j}\left(x^{j 1}, y_{j}^{L}\right)+g_{i}^{j}\left(x^{j 2}, y_{j}^{U}\right)=g_{i}^{j}\left(\alpha_{j} \bar{x}, y_{j}^{L}\right)+g_{i}^{j}\left(\left(1-\alpha_{j}\right) \bar{x}, y_{j}^{U}\right) \\
& \leq \alpha_{j} g_{i}^{j}\left(\bar{x}, y_{j}^{L}\right)+\left(1-\alpha_{j}\right) g_{i}^{j}\left(\bar{x}, y_{j}^{U}\right) \leq g_{i}^{j}\left(\bar{x}, \alpha_{j} y_{j}^{L}\right)+g_{i}^{j}\left(\bar{x},\left(1-\alpha_{j}\right) y_{j}^{U}\right) \\
& =g_{i}^{j}\left(\bar{x}, \bar{y}^{j}\right) \leq 0
\end{aligned}
$$

where the first inequality is due to the semihomogeneity of $g_{i}^{j}\left(x, y_{j}\right)$ in $x$, the second to the concavity of $g_{i}^{j}\left(x, y_{j}\right)$ in $y_{j}$, and the third to the feasibility of $\left(\bar{x}, \bar{y}_{j}\right)$ in (1). Also the objective function value of (24) is less than or equal to (in fact equal to) that of (1), since $x_{0}=\bar{x}_{0}$. Thus (24) is a convex quasi-relaxation of (1).

Finally, when $\alpha_{j}=1$ we have $x^{j 1}=x$ and $x^{j 2}=0$, and similarly if $\alpha_{j}=0$. It easy to verify, using the semihomogeneity of $g_{i}^{j}\left(x, y_{j}\right)$ in $x$, that (24) reduces to (1) when each $\alpha_{j} \in\{0,1\}$ and therefore has the same optimal value. This completes the proof.

Let $g\left(x, y_{j}\right)$ be homogeneous in $x$ when $g\left(\alpha x, y_{j}\right)=\alpha g\left(x, y_{j}\right)$ for all $\alpha \in$ $[0,1], y_{j} \in Y_{j}$.

Corollary 1. Theorem 2 holds in particular when each $g_{i}^{j}\left(x, y_{j}\right)$ is either (a) convex or (b) homogeneous in $x$ and concave in $y_{j}$.

If the global optimization problem (1) satisfies the conditions of Theorem 2, it can be solved by branch and bound as follows. Each node of the search tree is processed as in the algorithm below, where $z^{U}$ is the value of the best feasible solution found so far (initially $z^{U}=\infty$ ), and $\left[y_{j}^{L}, y_{j}^{U}\right]$ is the interval in which $y_{j}$ is currently contrained to lie (where $y_{j}^{L}, y_{j}^{U} \in Y_{j}$ ). Initially the only unprocessed node is the root node, which is processed first.

1. Compute an optimal solution $\bar{x}, \bar{x}^{j 1}, \bar{x}^{j 2}, \bar{\alpha}_{j}($ for $j \in J)$ of the convex quasirelaxation (24) at the current node. Set $\bar{y}_{j}=\bar{\alpha}_{j} y_{j}^{L}+\left(1-\bar{\alpha}_{j}\right) y_{j}^{U}$.
2. If $\bar{x}_{0} \geq z^{U}$, go to an unprocessed node and begin with step 1 .
3. If some $\bar{\alpha}_{j} \notin\{0,1\}$, let $v, v^{\prime}$ be the values in $Y_{j} \cap\left[y_{j}^{L}, y_{j}^{U}\right]$ on either side of $\bar{y}_{j}$ that are closest to $\bar{y}_{j}$. (Possibly $v$ or $v^{\prime}$ is identical to $\bar{y}_{j}$.) Branch on $y_{j}$ by creating an unprocessed node at which $y_{j} \in\left[y_{j}^{L}, v\right]$ and a second unprocessed node at which $y_{j} \in\left[v^{\prime}, y_{j}^{U}\right]$. Go to an unprocessed node and begin with step 1.
4. The solution $(\bar{x}, \bar{y})$ is feasible in (1). Set $z^{U}=\min \left\{\bar{x}_{0}, z^{U}\right\}$. Go to an unprocessed node and start with step 1.

The algorithm terminates when no unprocessed nodes remain. To ensure termination, one should fix $\alpha_{j}$ at 0 or 1 (either yields the same reuslt) whenever $y_{j}^{L}=y_{j}^{U}$.

Example. We return to the structural design problem (4) and solve it for continuous $y_{j}$ rather than over the finite set $\{1,2\}$. For illustrative purposes we discretize $y_{j}$ by setting each $Y_{j}=\{0,0.1,0.2, \ldots, 3.0\}$, although a finer resolution could be used.

We first check that (4) satisfies the conditions of Theorem 2. All the constraint functions are convex except $g_{2}^{1}\left(x, y_{1}\right)=10-x_{1} y_{1}$ and $g_{2}^{2}\left(x, y_{2}\right)=20-x_{2} y_{2}$. These are not semihomogeneous in $x$, since $g_{2}^{j}\left(0, y_{j}\right) \neq 0$, for example. However we can make them semihomogeneous (indeed, homogeneous) in $x$ by replacing the constant term with $s_{j}$ and adding the bounds $s_{1} \in[10,10]$ and $s_{2} \in[20,20]$.

We also use the bounds on $x_{j}$ introduced earlier. To simplify the problem we substitute $z_{j}=300 y_{j}$ into the objective function. The problem is now

$$
\begin{array}{ll}
\operatorname{minimize} & z_{0} \\
\text { subject to } & 300 y_{1}+300 y_{2}+\left(x_{1}+x_{2}\right)^{2}+x_{2}^{2}-x_{0} \leq 0 \\
& s_{1}-x_{1} y_{1} \leq 0 \\
& s_{2}-x_{2} y_{2} \leq 0  \tag{25}\\
& x_{1} \in[5,10], x_{2} \in[10,20] \\
& s_{1} \in[10,10], s_{2} \in[20,20] \\
& y_{1}, y_{2} \in\{0,0.1, \ldots, 3\}
\end{array}
$$

The convex quasi-relaxation (24) is

$$
\begin{array}{ll}
\operatorname{minimize} & z_{0} \\
\text { subject to } & 300 y_{1}+300 y_{2}+\left(x_{1}+x_{2}\right)^{2}+x_{2}^{2}-x_{0} \leq 0 \\
& \left(s_{11}-x_{11} y_{1}^{L}\right)+\left(s_{12}-x_{12} y_{1}^{U}\right) \leq 0 \\
& \left(s_{21}-x_{21} y_{2}^{L}\right)+\left(s_{22}-x_{22} y_{2}^{U}\right) \leq 0 \\
& 5 \alpha_{1} \leq x_{11} \leq 10 \alpha_{1}, \quad 5\left(1-\alpha_{1}\right) \leq x_{12} \leq 10\left(1-\alpha_{1}\right) \\
& 10 \alpha_{2} \leq x_{21} \leq 20 \alpha_{1}, \quad 10\left(1-\alpha_{2}\right) \leq x_{12} \leq 20\left(1-\alpha_{2}\right) \\
& 10 \alpha_{1} \leq s_{11} \leq 10 \alpha_{1}, \quad 10\left(1-\alpha_{1}\right) \leq s_{12} \leq 10\left(1-\alpha_{1}\right) \\
& 20 \alpha_{2} \leq s_{21} \leq 20 \alpha_{1}, \quad 20\left(1-\alpha_{2}\right) \leq s_{12} \leq 20\left(1-\alpha_{2}\right) \\
& {\left[\begin{array}{l}
x_{1} \\
x_{2}
\end{array}\right]=\left[\begin{array}{l}
x_{11}+x_{12} \\
x_{21}+x_{22}
\end{array}\right], \quad\left[\begin{array}{l}
s_{1} \\
s_{2}
\end{array}\right]=\left[\begin{array}{l}
s_{11}+s_{12} \\
s_{21}+s_{22}
\end{array}\right]} \\
& y_{j}=\alpha_{j} y_{j}^{L}+\left(1-\alpha_{j}\right) y_{j}^{U}, \alpha_{j} \in[0,1] j=1,2
\end{array}
$$

At this point we can reaggregate $s_{j}$, which allows it to be eliminated.

$$
\begin{array}{ll}
\operatorname{minimize} & z_{0} \\
\text { subject to } & 300 y_{1}+300 y_{2}+\left(x_{1}+x_{2}\right)^{2}+x_{2}^{2}-x_{0} \leq 0 \\
& 10-x_{11} y_{1}^{L}-x_{12} y_{1}^{U} \leq 0 \\
& 20-x_{21} y_{1}^{L}-x_{22} y_{1}^{U} \leq 0 \\
& 5 \alpha_{1} \leq x_{11} \leq 10 \alpha_{1}, \quad 5\left(1-\alpha_{1}\right) \leq x_{12} \leq 10\left(1-\alpha_{1}\right) \\
& 10 \alpha_{2} \leq x_{21} \leq 20 \alpha_{1}, \quad 10\left(1-\alpha_{2}\right) \leq x_{12} \leq 20\left(1-\alpha_{2}\right) \\
& {\left[\begin{array}{l}
x_{1} \\
x_{2}
\end{array}\right]=\left[\begin{array}{l}
x_{11}+x_{12} \\
x_{21}+x_{22}
\end{array}\right]} \\
& y_{j}=\alpha_{j} y_{j}^{L}+\left(1-\alpha_{j}\right) y_{j}^{U}, \alpha_{j} \in[0,1] j=1,2
\end{array}
$$

Figure 4 shows the first few nodes of the branching tree. A globally optimal solution $y=(1.1,2.0)$ with objective value $z_{0}=1394.5$ is found after processing 63 nodes, out of a possible $31 \times 31$ solutions. A more precise solution is $y=$ $(1.126,1.972)$ with $z_{0}=1394.1$.


Fig. 4. Beginning of a BBCQ branch-and-bound tree for the structural design example.

## 8 Truss Structure Design

We conclude with a more realistic truss structure design problem. The model presented here is a simplified version of that described in [1].

The notation is illustrated in Fig. 5. A truss structure consists of a number of bars $j$ joined at nodes, each bar having length $h_{j}$ and a cost of $c_{j}$ per unit volume. Each node can move in a specified number of directions. Thus if the problem is solved in three dimensions, there are at most three degrees of freedom at each node. Each degree of freedom $i$ is associated with a load $\ell_{i}$. The decision variables are the thickness (cross-sectional area) $y_{j}$ of the bars. Other variables are the elongation $s_{j}$ of bar $j$, the tension (pulling force) $f_{j}$ on bar $j$, and the displacement $x_{i}$ along degree of freedom $i$. The objective is to minimize the cost of the bars subject to bounds on elongation and displacement. Stress bounds


Fig. 5. Notation for a truss structure design problem.
also exist and are factored into the elongation bounds. The model is

$$
\begin{array}{lll}
\operatorname{minimize} & \sum_{j} c_{j} h_{j} y_{j} & \text { cost of bars } \\
\text { subject to } & \frac{E_{j}}{h_{j}} y_{j} s_{j}=f_{j}, \text { all } j & \text { Hooke's law } \\
& \sum_{j} f_{j} \cos \theta_{i j}=\ell_{i}, \text { all } i & \text { equilibrium equations }  \tag{26}\\
& \sum_{i} x_{i} \cos \theta_{i j}=s_{j}, \text { all } j & \text { compatibility equations } \\
& s_{j}^{L} \leq s_{j} \leq s_{j}^{U}, \text { all } j & \text { elongation bounds } \\
x_{j}^{L} \leq x_{j} \leq x_{j}^{U}, \text { all } j & \text { displacement bounds } \\
y_{j} \in Y_{j}, \text { all } j & \text { discrete thicknesses }
\end{array}
$$

where $E_{j}$ in Hooke's law is the modulus of elasticity for bar $j$. Since structural bars are generally available only in certain thicknesses, the variables $y_{j}$ can be regarded as discrete.

Since the problem becomes convex (in fact, linear) when variables $y_{j}$ are fixed, it is amenable to a logic-based method. We will apply disjunctive programming and BBCQ .

First we develop the disjunctive programming approach, using convex hull relaxations. A disjunctive representation of (26) is

$$
\begin{array}{lll}
\operatorname{minimize} & \sum_{j} z_{j} & \text { cost of bars } \\
\text { subject to } & \bigvee_{v \in Y_{j}}\left[\begin{array}{c}
y_{j}=v \\
z_{j} \geq c_{j} h_{j} v \\
\frac{E_{j}}{h_{j}} v s_{j}=f_{j}
\end{array}\right], \text { all } j & \text { cost, Hooke's law } \\
& \sum_{j} f_{j} \cos \theta_{i j}=\ell_{i}, \text { all } i & \text { equilibrium equations }  \tag{27}\\
& \sum_{i} x_{i} \cos \theta_{i j}=s_{j}, \text { all } j & \text { compatibility equations } \\
s_{j}^{L} \leq s_{j} \leq s_{j}^{U}, \text { all } j & \text { elongation bounds } \\
x_{j}^{L} \leq x_{j} \leq x_{j}^{U}, \text { all } j & \text { displacement bounds }
\end{array}
$$

Using convex hull relaxations of the disjunctions, we obtain the following convex relaxation of (27):

$$
\begin{align*}
& \operatorname{minimize} \sum_{j} z_{j} \\
& \text { subject to } z_{j}=\sum_{v \in Y_{j}} z_{j v}, \quad s_{j}=\sum_{v \in Y_{j}} s_{j v}, \quad f_{j}=\sum_{v \in Y_{j}} f_{j v}, \quad \text { all } j \\
& z_{j k} \geq c_{j} h_{j} v \beta_{j v}, \text { all } v \in Y_{j}, \text { all } j \\
& \frac{E_{j}}{h_{j}} v s_{j v}=f_{j v}, \text { all } v \in Y_{j}, \text { all } j \\
& \sum_{j} f_{j} \cos \theta_{i j}=\ell_{i}, \text { all } i  \tag{28}\\
& \sum_{i} x_{i} \cos \theta_{i j}=s_{j}, \quad \text { all } j \\
& \beta_{j v} s_{j}^{L} \leq s_{j v} \leq \beta_{j v} s_{j}^{U}, \quad \text { all } j \\
& x_{j}^{L} \leq x_{j} \leq x_{j}^{U}, \quad \text { all } j \\
& \sum_{v \in Y_{j}} \beta_{j v}=1, \quad \beta_{j v} \geq 0 \text { all } v \in Y_{j}, \text { all } j
\end{align*}
$$

The relaxation can be simplified, in part by summing each instance of Hooke's law over all $v \in Y_{j}$.

$$
\begin{align*}
\operatorname{minimize} & \sum_{j} \sum_{v \in Y_{j}} c_{j} h_{j} v \beta_{j v} \\
\text { subject to } & \frac{E_{j}}{h_{j}} \sum_{v \in Y_{j}} v s_{j v}=f_{j}, \text { all } j \\
& \sum_{j} f_{j} \cos \theta_{i j}=\ell_{i}, \text { all } i  \tag{29}\\
& \sum_{i} x_{i} \cos \theta_{i j}=s_{j}, \text { all } j \\
& \beta_{j v} s_{j}^{L} \leq s_{j v} \leq \beta_{j v} s_{j}^{U}, \text { all } j \\
& x_{j}^{L} \leq x_{j} \leq x_{j}^{U}, \quad \text { all } j \\
& \sum_{v \in Y_{j}} \beta_{j v}=1, \quad \beta_{j v} \geq 0 \text { all } v \in Y_{j}, \text { all } j
\end{align*}
$$

The disjunctive problem (27) can be solved as an MINLP by solving (29) with the integrality condition $\beta_{j v} \in\{0,1\}$. This MINLP model was in fact proposed by Ghattas, Voudouris and Grossmann [4, 5].

We now develop a BBCQ approach to solving (27). Note first that the model (26) satisfies the conditions of Theorem 2, since all of the constraints are convex (in fact, linear) except Hooke's law, which is convex (in fact, linear) when the $y_{j} \mathrm{~s}$ are fixed. In addition, the constraint function in Hooke's law is homogeneous in the continuous variables $s_{j}, f_{j}$ and concave (in fact, linear) in the discrete variable $y_{j}$. The convex quasi-relaxation (24) therefore becomes

$$
\begin{align*}
\text { minimize } & \sum_{j} c_{j} h_{j} y_{j} \\
\text { subject to } & \frac{E_{j}}{h_{j}}\left(y_{j}^{L} s_{j 1}+y_{j}^{U} s_{j 2}\right)=f_{j}, \text { all } j \\
& \sum_{j} f_{j} \cos \theta_{i j}=\ell_{i}, \text { all } i \\
& \sum_{i} x_{i} \cos \theta_{i j}=s_{j}, \text { all } j  \tag{30}\\
& \alpha_{j} s_{j}^{L} \leq s_{j 1} \leq \alpha_{j} s_{j}^{U}, \text { all } j \\
& \left(1-\alpha_{j}\right) s_{j}^{L} \leq s_{j 2} \leq\left(1-\alpha_{j}\right) s_{j}^{U}, \text { all } j \\
& x_{j}^{L} \leq x_{j} \leq x_{j}^{U}, \quad \text { all } j \\
& x_{j}=x_{j 1}+x_{j 2}, \text { all } j \\
& y_{j}=\alpha_{j} y_{j}^{L}+\left(1-\alpha_{j}\right) y_{j}^{U}, \text { all } j \\
& \alpha_{j} \in[0,1], \text { all } j
\end{align*}
$$

Bollapragada et al. [1] applied both the MILP and BBQC methods to the structural design problems illustrated in Fig. 6. Each structural bar had 11


Fig. 6. A 10-bar cantilever truss, 25-bar electrical treansmission tower, and 72-bar building.
possible thicknesses. Symmetries in the transmission tower and buildings were exploited to reduce the number of variables. Computational results are summarized in Table 1. MILP was implemented in CPLEX, and BBCQ in C with calls to the CPLEX linear programming solver. All problems were solved on a Sun Sparc Ultra work station.

These results suggest that BBCQ can carry a substantial advantage over a disjunctive approach when the constraint functions satisfy the conditions of Theorem 2.

## References

1. S. Bollapragada, O. Ghattas and J. N. Hooker, Optimal design of truss structures by mixed logical and linear programming, Operations Research 49 (2001) 42-51.
2. V. Chandru and J. N. Hooker, Optimization Methods for Logical Inference, John Wiley \& Sons (New York, 1999).
3. M. A. Duran and I. E. Grossmann, An outer-approximation algorithm for a class of mixed-integer nonlinear programs, Mathematical Programming 36 (1986) 307.
4. O. Ghattas and I. E. Grossmann, MINLP and MILP strategies for discrete sizing structural optimization problems, Proceeedings of the ASCE 10th Conference on Electronic Communication, Indianapolis (1991).
5. I. E. Grossmann, V. T. Voudouris, and O. Ghattas, Mixed-integer linear programming formulations of some nonlinear discrete design optimization problems, in C. A.

Table 1. Summary of solution times in seconds for MILP and BBCQ applied to truss structure design problems. When there two "loads" (i.e., two sets of loads applied to each degree of freedom), the structure is required to withstand each of the two loads, and a constraint set is written for each one. BBCQ was enhanced with some simple cutting planes when solving the cantilever and tower problems.

| Problem | Instance | MILP | BBCQ |
| :--- | :--- | ---: | ---: |
| 10-bar | l load | 1.3 | 0.3 |
| cantilever | 1 load, wider stress bounds | 1.6 | 0.3 |
| truss | 1 load, still wider stress bounds | 2.6 | 1.2 |
|  | 1 load, still wider stress bounds | 2.6 | 1.4 |
|  | 2 loads | 23.6 | 5.8 |
|  | 1 load, displacement bounds | 1089.4 | 67.5 |
|  | 2 loads, displacement bounds | 13743.9 | 1654.0 |
| 25-bar | 2 loads | 271.7 | 225.8 |
| transmission |  |  |  |
| tower |  | 12692.7 | 207.9 |
| Building | 72 bars, 2 loads | $*$ | 168.9 |
|  | 90 bars, 2 loads | $*$ | 329.4 |
|  | 108 bars, 2 loads |  |  |

*No solution after 20 hours ( 72,000 seconds).

Floudas and P. M. Pardalos, eds., Recent Advances in Global Optimization, Princeton University Press (1992).
6. I. E. Grossmann and S. Lee, Generalized disjunctive programming: Nonlinear convex hull relaxation, Carnegie Mellon University (2001) submitted.
7. J. Hiriart-Urruty and C. Lemaréchal, Convex Analysis and Minimization Algorithms, Vol. 1 (Springer-Verlag, 1993).
8. J. N. Hooker, Logic-Based Methods for Optimization: Combining Optimization and Constraint Satisfaction, John Wiley \& Sons (2000).
9. J. N. Hooker, Logic-based Benders decomposition for planning and scheduling, manuscript, GSIA, Carnegie Mellon University 2003).
10. J. N. Hooker and M. A. Osorio, Mixed logical/linear programming, Discrete Applied Mathematics 96-97 (1999) 395-442.
11. J. N. Hooker and G. Ottosson, Logic-based Benders decomposition, Mathematical Programming 96 (2003) 33-60.
12. Jain, V., and I. E. Grossmann, Algorithms for hybrid MILP/CP models for a class of optimization problems, INFORMS Jorunal on Computing 13 (2001) 258-276.
13. R. Stubbs and S. Mehrotra, A branch-and-cut method for 0-1 mixed convex programming, Mathematical Programming 86 (1999) 515-532.
14. M. Türkay and I. E. Grossmann, Logic-based outer-approximation algorithm for MINLP optimization of process flowsheets, Computers and Chemical Engineering 19 (1996) S131-S136.

