Obtaining Tighter Relaxations of Mathematical Programs with Complementarity Constraints

John E. Mitchell*, Jong-Shi Pang†, and Bin Yu‡

Original: February 19, 2011. Revised: October 11, 2011

Abstract

The class of mathematical programs with complementarity constraints (MPCCs) constitutes a powerful modeling paradigm. In an effort to find a global optimum, it is often useful to examine the relaxation obtained by omitting the complementarity constraints. We discuss various methods to tighten the relaxation by exploiting complementarity, with the aim of constructing better approximations to the convex hull of the set of feasible solutions to the MPCC, and hence better lower bounds on the optimal value of the MPCC. Better lower bounds can be useful in branching schemes to find a globally optimal solution. Different types of linear constraints are constructed, including cuts based on bounds on the variables and various types of disjunctive cuts. Novel convex quadratic constraints are introduced, with a derivation that is particularly useful when the number of design variables is not too large. A lifting process is specialized to MPCCs. Semidefinite programming constraints are also discussed. All these constraints are typically applicable to any convex program with complementarity constraints. Computational results for linear programs with complementarity constraints (LPCCs) are included, comparing the benefit of the various constraints on the value of the relaxation, and showing that the constraints can dramatically speed up the solution of the LPCC.

Keywords: MPCCs; linear programs with linear complementarity constraints; LPCCs; convex relaxation; semidefinite programming

*Department of Mathematical Sciences, Rensselaer Polytechnic Institute, Troy, New York 12180-1590, U.S.A. Email:mitchj@rpi.edu. The work of this author was supported by the National Science Foundation under grant DMS-0715446 and by the Air Force Office of Sponsored Research under grants FA9550-08-1-0081 and FA9550-11-1-0151.

†Department of Industrial and Enterprise Systems Engineering, University of Illinois, Urbana, Illinois 61801, U.S.A. Email: jspang@illinois.edu. The work of this author was supported by the National Science Foundation grant CMMI-0969600 and by the Air Force Office of Sponsored Research under grants FA9550-08-1-0061 and FA9550-11-1-0151.

‡Department of Industrial and Systems Engineering, Rensselaer Polytechnic Institute, Troy, New York 12180-1590, U.S.A. Email:yub@rpi.edu.

1 Introduction

Mathematical programs with complementarity constraints (MPCCs) arise in many settings. For example, Hobbs et al. [25] discuss applications in deregulated electricity markets; Pang et al. [38] discuss an application in maximum-likelihood-based target classification. The paper [37] shows how the MPCC provides a unifying framework for various modeling paradigms, including hierarchical and inverse optimization. Most recently, the MPCC is used as a tractable formulation for the estimation of pure characteristics models based on empirical market shares [39]. There has been a great deal of research on finding stationary points for MPCCs; see [37] for a list of references. In order to determine a globally optimal solution to an MPCC, it is necessary to find a lower bound on the problem, typically by relaxing the problem. Tightening the relaxation can lead to improved lower bounds, which can be exploited in, for example, branching and domain decomposition schemes. In this paper, we describe several methods for tightening relaxations of MPCCs. We focus on linear programs with complementarity constraints (LPCCs), a rich subclass of MPCCs. We have previously described logical Benders decomposition and branch and cut methods for finding globally optimal solutions to LPCCs [27, 26, 28]; these methods can be improved by the techniques presented in the current paper. The proposed tightening techniques are expected to be particularly useful for solving the class of convex programs with complementarity constraints, which is a subclass of MPCCs broader than the LPCCs; this extension is presently being investigated.

An LPCC is a linear program with additional complementarity constraints on certain pairs of variables. Because of the complementarity constraints, it is a nonconvex, nonlinear disjunctive program. These problems arise in many settings, with the complementarity constraints often used to model logical relations. For example, LPCCs can be used to model bilevel programs, inverse problems, quantile problems, indefinite quadratic programs, and piecewise linear programs; see [28] for a recent summary of such applications.

Given vectors and matrices: $c \in I\!R^n$, $d \in I\!R^m$, $e \in I\!R^m$, $b \in I\!R^k$, $A \in I\!R^{k\times n}$, $B \in I\!R^{k\times m}$, and $C \in I\!R^{k\times m}$, the LPCC is to find a triple $v := (x, y, w) \in I\!R^n \times I\!R^m \times I\!R^m$ in order to globally solve the optimization problem

$$\begin{array}{rlcccccccl} \Phi \triangleq & \underset{x,y,w}{\text{minimize}} & c^T x & + & d^T y & + & e^T w & & \\ & \text{subject to} & Ax & + & By & + & Cw & \geq & b \\ & & 0 & \leq & y & \perp & w & \geq & 0 \end{array} \qquad (1)$$

where the $\perp$ notation denotes the perpendicularity between two vectors, which in this context, pertains to the complementarity of these vectors. Thus, without the orthogonality condition: $y \perp w$, the LPCC is a linear program (LP). With this condition, the LPCC is equivalent to 2^m LPs. The variables x are sometimes called design variables.

Relaxing the complementarity condition leads to a linear programming problem. For some problems, this relaxation can be quite weak, so in this paper we consider methods for improving the relaxation. Bounds on the variables y and w can be used to construct linear constraints, as we show in §3. These cuts are well-known, and we test refinements of the cuts. Disjunctive cuts were developed in the 1970's [7] initially for integer programming. They have been studied extensively and can be generated to cut off points that violate the complementarity constraints, using the optimal simplex tableau to the LP relaxation and in other ways. We discuss the specialization of disjunctive cuts to LPCCs in §4. The nonconvex quadratic constraint $y^T w \leq 0$ is valid for (1); we consider novel convex quadratic relaxations of this constraint in §5. Constraints that are valid on part of the feasible region can be lifted to give constraints valid throughout the feasible region, a technique that can also be used to strengthen other constraints; we discuss lifting in §6. The products of variables can be linearized and a semidefinite constraint imposed to tighten the linearization, as discussed in §7. Each of the families of cuts can be strengthened by exploiting the other families. Thus, the overall strength of the relaxation depends on the order in which the cuts are derived, and it can be further strengthened by repeatedly generating constraints. Under certain conditions, the work of Balas [7] and of Kojima and Tunçel [30] shows that repeated generation of cuts leads to the convex hull of the feasible region of (1).

2 Problem generation and computational setup

We experimented with a randomly generated collection of linear programs with complementarity constraints. The parameters in (1) were generated as follows. The matrices A, B, and C are written as $A = [\bar{A}^T, -N^T, I]^T$, $B = [\bar{B}^T, -M^T, 0]^T$, and $C = [0, I, 0]^T$ where 0 denotes a matrix of zeroes of the appropriate dimension, $\bar{A}$ and $\bar{B}$ have $\bar{k} = k - m - n$ rows, and M and N have m rows. Similarly, the right hand side is written $b = [\bar{b}^T, q^T, 0]^T$ with $\bar{b} \in I\!R^{\bar{k}}$ and $q \in I\!R^m$. We set $e = 0$, so the problem is equivalent to

$$\begin{array}{llcll} \underset{x,y,w}{\text{minimize}} & c^T x & + & d^T y & \\ \text{subject to} & \bar{A}x & + & \bar{B}y & \geq \bar{b} \\ & x & & & \geq 0 \\ & \multicolumn{4}{l}{0 \leq y \perp q + Nx + My \geq 0,} \end{array}$$

a standard form in the LPCC literature [28].

The entries in c and d are uniformly distributed integers between 0 and 9, which ensures the problem is not unbounded. The entries in $\bar{A}$, $\bar{B}$, and N are uniformly generated integers between -5 and 5, with a proportion of the entries zeroed out. The matrix $\frac{1}{2}(M + M^T)$ is set equal to LL^T where L is an $m \times r$ matrix whose entries are uniformly generated integers between -5 and 5, with a proportion of the entries zeroed out. This construction ensures that $\frac{1}{2}(M + M^T)$ is positive semidefinite, which is necessary for the approach we develop in §5 and which occurs in some classes of practical instances [28]. The matrix M is then obtained from $\frac{1}{2}(M + M^T)$ by adjusting the nonzero off-diagonal entries by a uniformly distributed random integer between -2 and 2.

To ensure feasibility of (1), a solution $\bar{x}$, $\bar{y}$ is generated. The entries in $\bar{x}$ are integers uniformly distributed between 0 and 9. Two thirds of the entries in $\bar{y}$ are set equal to zero, and the remainder are integers uniformly distributed between 0 and 9. The entries in the right hand side $\bar{b}$ are chosen so that each slack with the generated solution is an integer uniformly distributed between 1 and 10, so the constraints are strictly satisfied by the generated solution. The third of the entries in q corresponding to the positive components of $\bar{y}$ are chosen so that complementarity is satisfied. Another third of the entries in q are chosen so that the corresponding components of $q + N\bar{x} + M\bar{y}$ are zero, so $\bar{y}_i = \bar{w}_i = 0$ for these entries. The final third of the entries of q are chosen so that the corresponding slack in $q + N\bar{x} + M\bar{y} \geq 0$ is an integer uniformly distributed between 1 and 10. The construction is designed so that it is unlikely that the generated solution $\bar{x}$, $\bar{y}$ is optimal.

The tests in §3, §4, and §5 were run on a single core of an AMD Phenom II X4 955@3.2GHZ with 4GB memory, using C++ with callable CPLEX 11.0. All times are reported in seconds. Problems with $m = 100$, 150, and 200 complementarities were solved, with $n = 2$ and $\bar{k} = 20$. The matrices $\bar{A}$, $\bar{B}$, N, and L were either 20% or 70% dense. The rank r of L was either 30 or 60 for $m = 100$, either 30 or 100 for $m = 150$, and either 30 or 120 for $m = 200$. Five problems were solved for each choice of m, sparsity, and rank, leading to a total of 60 problems.

3 Linear constraints based on bounds on the variables

3.1 Construction of the constraints

Given finite upper bounds y_i^u and w_i^u on y_i and w_i respectively, the constraint

$$w_i^u y_i + y_i^u w_i \leq y_i^u w_i^u, \tag{2}$$

which we term a *bound cut*, is valid for the LPCC (1), because of the complementarity restriction on y_i and w_i. The bounds y_i^u and w_i^u may not be readily available and can be calculated by solving linear programming problems. Before calculating bounds, it is useful to find a good feasible solution to the LPCC, using either a heuristic or a nonlinear programming solver such as KNITRO [20] or FILTER [23]. The value of this solution provides an upper bound Φ^{UB} on the optimal value Φ of (1) and so the constraint

$$c^T x + d^T y + e^T w \ \leq \ \Phi^{\text{UB}} \tag{3}$$

is valid for any optimal solution to (1). We let S denote the set of feasible solutions to (1) that satisfy (3). The convex hull of S is a polyhedron and it can be outer approximated by

$$\Xi \triangleq \{(x, y, w) \in I\!R^{n+2m} \ : \ Ax + By + Cz \geq b, \ c^T x + d^T y + e^T w \ \leq \ \Phi^{\text{UB}}, \ y \geq 0, \ w \geq 0\}. \tag{4}$$

If an inequality description of the convex hull of S was known then the LPCC could be determined by solving the linear program of minimizing the objective over the convex hull. The aim of constructing the inequalities in this paper is to obtain a tighter approximation to the convex hull than Ξ. Under certain conditions, this constraint defines a facet of the convex hull of feasible solutions to the LPCC, as shown by De Farias et al. [22] for the case of a knapsack LPCC. It is straightforward to prove the following proposition.

Proposition 1. Assume either (i) $y_i \geq 0$ defines a facet of S and $w_i \leq w_i^u$ defines a facet of $\{(x,y,s) \in S : y_i = 0\}$ and there is a point in S with $y_i = y_i^u$ or (ii) $w_i \geq 0$ defines a facet of S and $y_i \leq y_i^u$ defines a facet of $\{(x,y,s) \in S : w_i = 0\}$ and there is a point in S with $w_i = w_i^u$. Then (2) defines a facet of S.

Proof. We prove case (i). If $y_i^u = 0$ then the result is immediate. If $y_i^u > 0$ then the facet assumptions together with a point satisfying the constraint (2) at equality that is not in $\{(x,y,s) \in S : y_i = 0\}$ leads to the conclusion, from standard lifting arguments. □

We will give a generalization of this proposition later, in Proposition 3. In principle, an upper bound on y_i can be found by solving the linear program

$$\begin{array}{llcccccl} \underset{x,y,w}{\text{maximize}} & & & y_i & & & & \\ \text{subject to} & Ax & + & By & + & Cw & \geq & b \\ & c^T x & + & d^T y & + & e^T w & \leq & \Phi^{\text{UB}} \\ & & & & & w_i & = & 0 \\ & 0 & \leq & y, & & w & \geq & 0, \end{array} \tag{5}$$

and an upper bound on w_i can be constructed similarly. These bounds may be infinite; they can be tightened by further exploiting complementarity. For example, we experimented with the following procedure, choosing p equal to 2 or 3:

Bound tightening procedure

Step 0. Find initial upper bound y_i^u by solving (5). Let $(\hat{x}, \hat{y}, \hat{w})$ denote the optimal solution.

Step 1. Let $s_j = \hat{y}_j \hat{w}_j$ for $j = 1, \ldots, m$. Let J denote the indices of the largest p components of s.

Step 2. Solve the 2^p linear programs of the form (5) with the additional constraints that for each $j \in J$ either $y_j = 0$ or $w_j = 0$. Update y_i^u to be the largest of the optimal values of these 2^p linear programs.

A similar procedure is used to improve w_i^u.

It may be advisable computationally to limit the number of bounds calculated. One approach to do this is to first solve the LP relaxation of (1) and then only calculate bounds for variables where the complementarity constraint is violated. Similarly, the bound tightening procedure could be used only for constraints (2) that are tight at the solution to the LP relaxation of (1).

If upper bounds are available for all the variables y_i and w_i then (1) is equivalent to the

# splits	# refine	% gap closed			CPU time (secs)			sufficient
		100	150	200	100	150	200	
0	0	45.5	48.0	51.2	1.4	6.8	22.5	0
	4	69.8	75.9	74.9	16.3	76.5	257.7	10
	8	72.7	76.3	75.3	30.8	151.4	482.9	15
1	0	55.8	55.4	57.4	3.9	18.3	60.2	1
	4	76.9	84.1	76.6	42.1	194.3	660.5	15
	8	81.1	85.5	77.1	73.4	367.0	1171.1	17
2	0	58.3	59.0	60.0	6.0	29.0	93.1	1
	4	82.6	86.1	77.6	61.8	285.6	999.7	16
	8	86.6	87.8	79.5	104.7	542.2	1786.1	19
3	0	60.6	60.1	61.8	10.4	49.9	159.4	1
	4	85.6	88.9	80.0	102.7	505.1	1672.4	18
	8	91.3	90.6	80.6	171.6	901.6	3150.6	21

Table 1: Bound cuts for LPCCs.

following integer programming problem:

$$\begin{array}{lrcrcrcl} \underset{x,y,w}{\text{minimize}} & c^T x & + & d^T y & + & e^T w & & \\ \text{subject to} & Ax & + & By & + & Cw & \geq & b \\ & & & 0 & \leq & y & \leq & Y^u z \\ & & & 0 & \leq & w & \leq & W^u(\mathbf{1} - z) \\ & & & & & z & \in & \{0,1\}^m \end{array} \tag{6}$$

Here, $\mathbf{1}$ denotes the vector of all ones, and Y^u and W^u are diagonal matrices, with diagonal entries equal to the bounds y^u and w^u.

3.2 Computational results

An experiment was performed to test the ideas presented so far. The results were not significantly affected by sparsity or rank, so we aggregate into 20 problems each with 100, 150, or 200 complementarities. The results are contained in Table 1. The first column reports the choice of p in the bound tightening procedure. The second column gives the number of refinements of the bound tightening procedure: once bounds have been found, the bound finding LP (5) can be tightened. The third, fourth, and fifth columns give the relative improvement in the gap between the lower bound on (1) given by its LP relaxation and the optimal value of (1); each entry in these columns is averaged over 20 problems. The sixth, seventh, and eighth columns give the run time, with each entry averaged over 20 instances. The final column notes the number of problems (out of 60) for which the bound cuts were sufficient to prove global optimality.

The bound cuts are effective at closing a large proportion of the duality gap. However, they are quite expensive, especially with additional refinements and splits. It is noticeable that the smaller problems benefit slightly more than the larger problems from additional refinements and splits. The bound cuts are surprisingly effective at proving optimality for these problems, with over one-third of the problems solved to optimality with the most extensive version of the cuts. Refining the bound cuts 8 times closes between 47% and 78% of the gap remaining after the addition of just the initial bound cuts.

We also used CPLEX 11 to try to solve these problems to optimality, on the same computer hardware. The callable library version of CPLEX was used, which allows the representation of disjunctive constraints using indicator constraints, so it is possible to work directly with formulation (1) together with constraints (2) and (3). An initial feasible solution was found with a heuristic. A time limit of two hours was placed on each run. The results are contained in Table 2. The columns indicate the number of problems that were solved in the two hour time limit, and the average time for each set of 20 instances. Both the time to solve the problems to optimality after adding the cuts and the total time including the cut-generation time are included in the table. All total times that are within 50% of the minimum are highlighted. The cuts lead to dramatic improvements in the ability of CPLEX to solve the instances and in the total time required. It is not worthwhile to refine the cuts for the smaller problems with $m = 100$ complementarities. For larger problems, the time invested in generating the cuts and refining them can lead to strong overall performance, with one good option being to use four refinements along with one split. Additional refinements or splits aid the algorithm in finding a solution within the two hour limit; only one problem cannot be solved within two hours with the more extensive cut generation choices. Even better performance could probably be obtained by generating and adding the bound cuts selectively, based on the solution to the LP relaxation; in this paper, we are examining the strength of the class of cuts as a whole.

4 Linear constraints based on disjunctive programming

4.1 Disjunctive cuts

Valid constraints can be constructed from any point in Ξ that is not in the convex hull of feasible solutions to (1), using a disjunctive programming approach. If $y_i w_i > 0$ in an extreme point optimal solution to a relaxation of (1) then it is not in the convex hull, so valid constraints are constructed that are satisfied by all points in Ξ with $y_i = 0$ and by

# splits	# refine	successful			solve time (secs)			total time (secs)		
		100	150	200	100	150	200	100	150	200
no cuts		20	11	5	98	3862	5852	98	3862	5852
0	0	20	19	12	13	632	3265	**14**	639	3290
	4	20	20	18	6	102	1558	22	**179**	**1816**
	8	20	20	18	5	66	1382	36	**217**	**1871**
1	0	20	19	12	12	843	2992	**16**	861	3052
	4	20	20	19	4	64	906	46	**258**	**1567**
	8	20	20	19	3	78	942	76	445	**2113**
2	0	20	20	13	10	496	2962	**16**	525	3055
	4	20	20	19	4	67	930	66	811	**1930**
	8	20	20	19	2	60	977	107	602	2763
3	0	20	19	13	8	536	2763	23	586	2922
	4	20	20	19	3	60	795	106	565	2467
	8	20	20	19	2	52	841	174	954	3992

Table 2: Solving to optimality with bound cuts for LPCCs.

all points in Ξ with $w_i = 0$. A cut generation linear program can be formulated to find such valid constraints. Balas [7, 8] developed many of the results regarding disjunctive cuts for integer programming. Many of the approaches used in integer programming are also useful in more general disjunctive programs. For example, Audet et al. [4] consider disjunctive cuts for bilevel linear programs. For good recent surveys of methods of generating disjunctive cuts see [11, 16, 40, 44]. It was shown empirically in the 1990's that general cuts such as disjunctive cuts [9] and Gomory cuts [10] could be very effective for general integer programs. Theoretically, the convex hull of an LPCC can be obtained using the lift-and-project procedure, since the disjunctions are facial [21]. Also of interest is recent work showing that disjunctive cuts can be effective for mixed integer nonlinear programming problems [14, 45, 46, 49]. Judice et al. [29] investigated disjunctive cuts for problems with complementarity constraints.

Let $v = (x, y, w) \in I\!R^{n+2m}$ and let $\hat{v}$ be the optimal solution to the LP relaxation. A general disjunctive cut for the union of a family of polyhedra is an inequality that is valid for each polyhedron in the family. It can be obtained by solving a cut generation LP which ensures the cut is dominated by a nonnegative linear combination of the valid constraints for each polyhedron. This cut generation LP can be large, so methods have been developed to find cuts without solving the full cut generation LP. The optimal simplex tableau for the linear programming relaxation can be used directly to generate constraints that cut off $\hat{v}$ if it violates the complementarity restrictions. In particular, if $y_i w_i > 0$ then the two rows of the simplex tableau corresponding to the basic variables y_i and w_i can be written as follows, where R denotes the set of nonbasic variables:

$$\begin{array}{rcl} y_i \qquad + \sum_{j\in R} \hat{a}_j^{yi} v_j &=& \hat{y}_i \\ w_i + \sum_{j\in R} \hat{a}_j^{wi} v_j &=& \hat{w}_i. \end{array} \tag{7}$$

The disjunction $y_i = 0 \vee w_i = 0$ is equivalent to the disjunction

$$\sum_{j\in R} \frac{\hat{a}_j^{yi}}{\hat{y}_i} v_j \geq 1 \quad \vee \quad \sum_{j\in R} \frac{\hat{a}_j^{wi}}{\hat{w}_i} v_j \geq 1$$

since y_i and w_i are nonnegative variables. Let $\alpha_j = \max \left\{ \frac{\hat{a}_j^{yi}}{\hat{y}_i}, \frac{\hat{a}_j^{wi}}{\hat{w}_i} \right\}$ for $j \in R$. We can construct the following valid constraint for (1):

$$\sum_{j\in R} \alpha_j v_j \geq 1. \tag{8}$$

This constraint is violated by $\hat{v}$ since $\hat{v}_j = 0$ for $j \in R$. It is valid because either $y_i = 0$ or $w_i = 0$ in any feasible solution, so the sum of the nonbasic variables in (7) must be equal to the right hand side for at least one of the constraints, and the sum of the nonbasic variables (scaled by the right hand side) is overestimated by the sum given in (8). This is called a *simple cut* by Audet et al. [5], and is based on intersection cuts for 0-1 programming [6] and has also been investigated by Balas and Perregaard [12].

If the complementarity restrictions for components i and k are both violated by $\hat{v}$ then the corresponding 4 rows of the simplex tableau can be combined to obtain valid constraints for

$$\Xi^{ik} \triangleq \Xi \cap \{v \mid y_i w_i = 0\} \cap \{v \mid y_k w_k = 0\}.$$

In particular, we can set up the following cut generation LP which generates a constraint that is valid for each of the four pieces of Ξ^{ik} corresponding to each assignment of the i and k complementarity relationships. Any feasible solution to this LP gives a valid constraint of the form (8) that cuts off $\hat{v}$:

$$\begin{array}{llcll}
\underset{\alpha,u}{\text{minimize}} & \sum_{j\in R} \alpha_j & & & \\
\text{subject to} & \alpha_j & \geq & u_{1i}\dfrac{\hat{a}_j^{yi}}{\hat{y}_i} + u_{1k}\dfrac{\hat{a}_j^{yk}}{\hat{y}_k} & \forall j \in R \\
& \alpha_j & \geq & u_{2i}\dfrac{\hat{a}_j^{yi}}{\hat{y}_i} + u_{2k}\dfrac{\hat{a}_j^{wk}}{\hat{w}_k} & \forall j \in R \\
& \alpha_j & \geq & u_{3i}\dfrac{\hat{a}_j^{wi}}{\hat{w}_i} + u_{3k}\dfrac{\hat{a}_j^{yk}}{\hat{y}_k} & \forall j \in R \\
& \alpha_j & \geq & u_{4i}\dfrac{\hat{a}_j^{wi}}{\hat{w}_i} + u_{4k}\dfrac{\hat{a}_j^{wk}}{\hat{w}_k} & \forall j \in R \\
& 1 & \leq & u_{pi} + u_{pk} & \text{for } p = 1,\ldots,4 \\
& u_{pq} & \geq & 0 & \text{for } p = 1,\ldots,4,\ q = i,k.
\end{array} \tag{9}$$

The first four constraints correspond to different pieces of Ξ^{ik} and ensure that constraint (8) is dominated by a nonnegative combination of the constraints for that piece. For example, the first constraint corresponds to the piece with $y_i = y_k = 0$, and ensures the constraint is dominated by a combination of the corresponding nonbasic parts of (7). The objective function together with the constraints $1 \leq u_{pi} + u_{pk}$ act to normalize the constraint generation LP; other normalizations could be used instead. This linear program is far smaller than the standard disjunctive cut generation LP; it in effect constrains many variables from the standard LP to be equal to zero. Balas and Perregaard [11] discuss similar methods for making the cut generation LP easier to solve. It should be noted that the standard simple cut is an optimal solution to a constrained version of (9), obtained by adding the constraints $u_{pi} = 1$ and $u_{pk} = 0$ for $p = 1,\ldots,4$.

cut type	# refine	% gap closed			CPU time (secs)			sufficient
		100	150	200	100	150	200	
disjunctive	$m/8$	65.3	65.7	61.9	18.4	162.9	708.5	0
	$m/2$	75.9	75.1	70.8	78.6	605.2	2547.6	1
simple	$m/8$	32.7	25.3	21.4	0.1	0.8	3.4	0
	$m/2$	34.6	26.0	21.9	2.6	23.8	118.1	0

Table 3: Computational results with disjunctive cuts.

4.2 Computational results

The disjunctive cuts and simple cuts were tested for the same problems as in §2, in the same computational environment. Computational results are contained in Table 3. The cuts were refined successively, with the number of refinements proportional to the number of complementarities and given in the second column of the table. Columns 3 to 9 of the table have the same meanings as in Table 1.

The general disjunctive cuts are far more effective than the simple cuts, but they are considerably more expensive. Additional refinement is quite useful for the disjunctive cuts, but far less so for the simple cuts. Audet et al. [5] have experimented with disjunctive cuts for LPCCs arising from bilevel programs, with encouraging results.

5 Convex quadratic constraints

5.1 Construction of the constraints

The complementarity constraint

$$0 \leq y \perp w \geq 0$$

is equivalent to the nonnegativity constraints $y, w \geq 0$ together with the nonconvex quadratic constraint

$$y^T w \leq 0. \tag{10}$$

In this section, we consider convex quadratic relaxations of (10). We assume w can be written as a linear function of x and y, so

$$w = q + Nx + My \tag{11}$$

where q, N, and M are dimensioned appropriately. We express the complementarity restriction in terms of x and y, so the number of constraints depends on the dimension n of x rather than on the number m of complementarity constraints. We have

$$y^T w = q^T y + y^T N x + \tfrac{1}{2} y^T \tilde{M} y$$

where $\tilde{M} = M + M^T$, so we look for convex relaxations of the quadratic constraint

$$q^T y + y^T N x + \tfrac{1}{2} y^T \tilde{M} y \;\leq\; 0. \tag{12}$$

Let p denote the number of nonnegative eigenvalues of $\tilde{M}$ and construct an eigendecomposition of $\tilde{M}$ as

$$\tilde{M} = V \Lambda V^T$$

where V is an orthogonal matrix with columns denoted v_i, Λ is a diagonal matrix, and the diagonal entries λ_i of Λ are arranged in decreasing order. Let k denote the rank of N and construct a factorization $N = \Gamma^T \Psi$, where Γ is a $k \times m$ matrix and Ψ is a $k \times n$ matrix. With the definition of k-dimensional variables $\tilde{y}$ and $\tilde{x}$, and the addition of the constraints

$$\tilde{y} \;=\; \Gamma y \tag{13}$$

$$\tilde{x} \;=\; \Psi x, \tag{14}$$

constraint (12) is equivalent to the constraint

$$q^T y + \sum_{j=1}^{k} \tilde{y}_j \tilde{x}_j + \tfrac{1}{2} \sum_{i=1}^{p} \lambda_i (v_i^T y)^2 \;\leq\; \tfrac{1}{2} \sum_{i=p+1}^{m} |\lambda_i| (v_i^T y)^2$$

or equivalently the convex quadratic constraint

$$q^T y + \sum_{j=1}^{k} \sigma_k + \tfrac{1}{2} \sum_{i=1}^{p} \lambda_i (v_i^T y)^2 \;\leq\; \tfrac{1}{2} \sum_{i=p+1}^{m} |\lambda_i| \pi_i \tag{15}$$

with the additional nonconvex constraints

$$\tilde{y}_j \tilde{x}_j \;=\; \sigma_j \qquad j = 1, \ldots, k \tag{16}$$

$$(v_i^T y)^2 \;\geq\; \pi_i \qquad i = p+1, \ldots, m \tag{17}$$

When the feasible region for $\tilde{y}_j$ and $\tilde{x}_j$ is given by a rectangle, it was shown by Al-Khayyal and Falk [2] that the lower convex envelope and upper concave envelope of (16) are given by the following *McCormick cuts* [34]:

$$\begin{array}{rcll} \tilde{x}_j^l \tilde{y}_j + \tilde{y}_j^l \tilde{x}_j & \leq & \sigma_j + \tilde{x}_j^l \tilde{y}_j^l & j = 1, \ldots, k \\ \tilde{x}_j^u \tilde{y}_j + \tilde{y}_j^u \tilde{x}_j & \leq & \sigma_j + \tilde{x}_j^u \tilde{y}_j^u & j = 1, \ldots, k \\ \tilde{x}_j^l \tilde{y}_j + \tilde{y}_j^u \tilde{x}_j & \geq & \sigma_j + \tilde{x}_j^l \tilde{y}_j^u & j = 1, \ldots, k \\ \tilde{x}_j^u \tilde{y}_j + \tilde{y}_j^l \tilde{x}_j & \geq & \sigma_j + \tilde{x}_j^u \tilde{y}_j^l & j = 1, \ldots, k, \end{array} \tag{18}$$

where $\tilde{y}_j^l$, $\tilde{y}_j^u$, $\tilde{x}_j^l$ and $\tilde{x}_j^u$ denote the bounds on $\tilde{y}_j$ and $\tilde{x}_j$. These constraints are exploited in packages for nonconvex optimization, including BARON [43, 50], αBB [1], and

COUENNE [15]. Tightenings of these inequalities combining together terms for several indices j have been recently investigated by Bao et al. [13].

We look for convex quadratic relaxations of (16) and linear relaxations of (17) that exploit the structure of the other linear constraints on $\tilde{y}_j$ and $\tilde{x}_j$. If $v_i^l \le v_i^T y \le v_i^u$ for all valid choices of y then the concave envelope of (17) is

$$\pi_i \le (v_i^l + v_i^u) v_i^T y - v_i^l v_i^u. \tag{19}$$

For any scalar $\alpha > 0$, (16) is equivalent to the constraint

$$\frac{1}{4\alpha}(\tilde{y}_j + \alpha\tilde{x}_j)^2 \le \sigma_j + \frac{1}{4\alpha}(\tilde{y}_j - \alpha\tilde{x}_j)^2$$

which can be relaxed to the convex quadratic constraint

$$\frac{1}{4\alpha}(\tilde{y}_j + \alpha\tilde{x}_j)^2 \le \sigma_j + \frac{1}{4\alpha}((\alpha^l + \alpha^u)(\tilde{y}_j - \alpha\tilde{x}_j) - \alpha^l\alpha^u) \tag{20}$$

where α^l and α^u denote lower and upper bounds respectively on $\tilde{y}_j - \alpha\tilde{x}_j$. Methods for choosing α are discussed in [35]. It is also shown in this reference that (20) can define part of the envelope of (16). Further, for certain configurations of the feasible $(\tilde{y}_j, \tilde{x}_j)$ region, inequalities (20) together with (18) define the lower convex envelope and upper concave envelope of (16). Thus, we propose to relax (12) using the linear constraints (13), (14), (18), and (19), and the convex quadratic constraints (15) and (20). We have the following proposition regarding the strength of (20).

Proposition 2. [35] Let $\mathcal{P}^j$ denote the projection of a polyhedral relaxation of (1) onto the $(\tilde{x}_j, \tilde{y}_j)$ plane. Let

$$\bar{\mathcal{P}}^j \triangleq \{(\tilde{x}_j, \tilde{y}_j, \sigma_j) \mid (\tilde{x}_j, \tilde{y}_j) \in \mathcal{P}^j,\ \sigma_j = \tilde{x}_j\tilde{y}_j\}.$$

(a) Assume $\mathcal{P}^j$ has the form

$$\tilde{x}_j^L \le \tilde{x}_j \le \tilde{x}_j^U,\ \tilde{y}_j^L \le \tilde{y}_j \le \tilde{y}_j^U,\ L^j \le \tilde{x}_j - \bar{\alpha}\tilde{y}_j \le U^j$$

for parameters $\bar{\alpha}$, $\tilde{x}_j^L$, $\tilde{x}_j^U$, $\tilde{y}_j^L$, $\tilde{y}_j^U$, L^j and U^j.

1. If $\bar{\alpha} > 0$, $\tilde{x}_j^U - \tilde{x}_j^L = \bar{\alpha}(\tilde{y}_j^U - \tilde{y}_j^L)$, and $L^j + U^j = \tilde{x}_j^L + \tilde{x}_j^U - \bar{\alpha}(\tilde{y}_j^L + \tilde{y}_j^U)$ then the lower convex underestimator of σ_j over $\mathcal{P}^j$ is given by (18) together with (20) with $\alpha = \bar{\alpha}$.
2. If $\bar{\alpha} < 0$, $\tilde{x}_j^U - \tilde{x}_j^L = -\bar{\alpha}(\tilde{y}_j^U - \tilde{y}_j^L)$, and $L^j + U^j = \tilde{x}_j^L + \tilde{x}_j^U - \bar{\alpha}(\tilde{y}_j^L + \tilde{y}_j^U)$ then the upper concave overestimator of σ_j over $\mathcal{P}^j$ is given by (18) together with (20) with $\alpha = \bar{\alpha}$.

(b) Let $\mathcal{P}^j$ have the form

$$L_1^j \le \tilde{x}_j - \bar{\alpha}\tilde{y}_j \le U_1^j,\ L_2^j \le \tilde{x}_j + \bar{\alpha}\tilde{y}_j \le U_2^j$$

for some $\bar{\alpha} > 0$. If $\mathcal{P}^j$ is nonempty then the convex envelope of $\bar{\mathcal{P}}^j$ is given by (20) with $\alpha = \pm\bar{\alpha}$. □

	% gap closed			CPU time (secs)			sufficient
	100	150	200	100	150	200	
McCormick	44.3	49.1	47.9	0.2	0.5	1.2	0
+ 8 refines	83.5	91.5	90.7	9.7	29.4	67.6	6
+ quadratic	83.5	91.5	90.8	10.0	30.3	69.3	8

Table 4: McCormick cuts and convex quadratic cuts.

5.2 Computational results

The test problems and computational environment were the same as in section 2. The convex quadratic program solver CPLEX reported that the matrix $M + M^T$ was not positive semidefinite for 5 of the 60 instances, due to numerical errors, and so these problems were not solved. Consequently, each entry in the "gap" columns and the "time" columns represents a mean of 20, 19, or 16 instances. The performance of refining the McCormick cuts procedure is compared with refining the bound generation procedure in Figure 1.

Using convex relaxations of the constraint $y^T w \leq 0$ is very effective for this class of problems, giving better bounds than from either the bound cuts of section 3.2 or the disjunctive cuts of section 4.2, in far less time. The quadratic constraint (20) is only marginally helpful for these problems; simply iteratively tightening the McCormick bounds (18) works very well.

We also attempted to solve these problems to optimality, using a combination of bound cuts and McCormick cuts. We used the bound cut procedure with no splits and 4 refinements, since these bounds can be found in a moderate amount of time (Table 1) and the cuts are quite effective at solving the problems (Table 2). After adding these refined bound cuts, the McCormick cut generation procedure was used, with 8 refinements, as in Table 4. For these runs, CPLEX reported that the matrix $M + M^T$ was indefinite for 4 of the 60 instances, and it was unable to solve an additional three instances in the two hour time limit. This left 20, 19, and 14 instances with 100, 150, and 200 complementarities, respectively. These 53 instances were also all solved in the two hour limit using just the bound cuts. The runtimes with just the bound cuts and also with the bound cuts together with the McCormick cuts are contained in Table 5. Five additional problems were solved at the root node through the addition of the McCormick cuts. The additional time required to generate the McCormick cuts is worthwhile for the larger instances.

When using just the bound cuts, 2 of the 200 complementarity instances cannot be solved in 2 hours, and 5 other instances require at least 20 minutes. The two instances requiring at least 2 hours still cannot be solved within that time limit with the addition of McCormick cuts, but the duality gap at the end of the time limit has been noticeably reduced. CPLEX reported that $M + M^T$ was indefinite for one of the remaining hard instances, and it was unable to solve one of them within the 2 hour limit. The McCormick cuts reduced the run

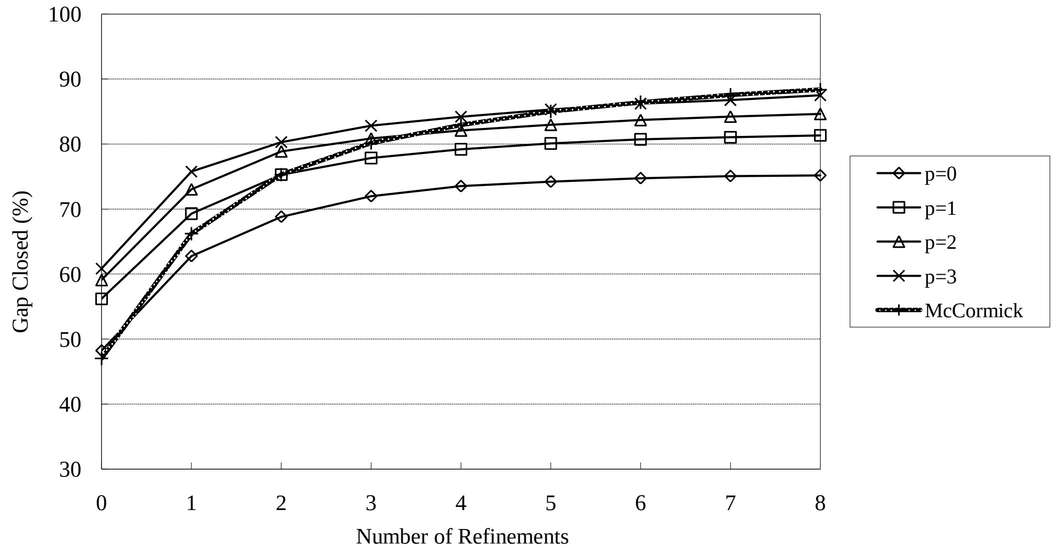

Figure 1: Comparison of average gap closed by refinements of bound cuts and McCormick cuts.

	solve time (secs)			total time (secs)		
	100	150	200	100	150	200
Bound cuts	6	74	626	48	268	1286
Bound + McCormick	6	37	368	58	260	1095

Table 5: Solving to optimality with McCormick cuts.

time on each of the other 3 hard instances, with the average solve time dropping from 2681 seconds to 1598 seconds.

6 Lifting constraints

Lifting is a methodology for modifying a constraint that is valid on one part of the feasible region so that it is valid throughout the feasible region. Let $\mathcal{P} \subseteq \mathbb{R}^n_+$ denote a polyhedron and let $\mathcal{P}^0 \triangleq \{x \in \mathcal{P} \,:\, x_i = 0\}$ for a fixed component i. Given a constraint $a^T x \geq \beta$ that is valid on $\mathcal{P}^0$, a lifting procedure can be used to extend this constraint so that it is valid throughout $\mathcal{P}$. This idea is widely employed in integer programming [36].

De Farias et al. [22] describe a lifting procedure for LPCCs with $k = 1$ and all coefficients negative. They show that (2) defines a facet of the feasible region under some mild conditions on the coefficients of the problem. They also show that if an inequality is facet defining when $w_i = 0$ then it can be lifted to a facet-defining inequality for the whole feasible region by solving a parametric linear program, again under certain mild assumptions. Further, they describe various families of facet-defining inequalities. Richard and Tawarmalani [42] generalize lifting to nonlinear programs. Given an affine minorant of a function $f(x, y) : \mathbb{R}^{m+n} \to \mathbb{R}$ that is valid for a particular choice of y, they show how the affine minorant can be extended to be a minorant for all y. Of particular interest is the case when $f(x, y)$ is a membership indicator function for a set S, which is zero for points in the set and infinite otherwise.

The general lifting framework of [42] can be specialized to LPCCs by using a membership indicator function for the set S of feasible points to the LPCC (1) that satisfy the objective function bound constraint (3). Given disjoint subsets I_1, $I_2 \subseteq \{1, \ldots, m\}$, let

$$S^{I_1,I_2} \triangleq \{v \mid (x, y, w) \in S \,:\, y_i = 0 \ \forall i \in I_1,\, w_i = 0 \ \forall i \in I_2\}.$$

Let

$$\alpha^T v \leq \beta$$

be a valid constraint for S^{I_1,I_2} and let $i \in I_1$. We want to extend the constraint so that it is valid for $S^{I_1 \setminus i,\, I_2}$, constructing a constraint of the form

$$\alpha^T v + \nu_i y_i \leq \beta$$

for some constant ν_i. As shown in [42], it suffices to choose ν_i so that $\nu_i y_i$ underestimates

$$g(\xi) \triangleq \inf \left\{ \beta - \alpha^T v \mid v \in S^{I_1 \setminus i,\, I_2},\, y_i = \xi \right\}.$$

If $S^{I_1 \setminus i,\, I_2}$ is compact, the parameter ν_i can be determined by solving a fractional program:

$$\nu_i \triangleq \inf \left\{ \frac{\beta - \alpha^T v}{y_i} \mid v \in S^{I_1 \setminus i,\, I_2},\, y_i > 0 \right\},$$

which can be solved as a parametric LPCC. For example, (2) can be derived by lifting the inequality $w_i \leq w_i^{\mathrm{UB}}$ that is valid on $S^{i,\emptyset}$, when we obtain $\nu_i = w_i^{\mathrm{UB}} / y_i^{\mathrm{UB}}$. The lifting

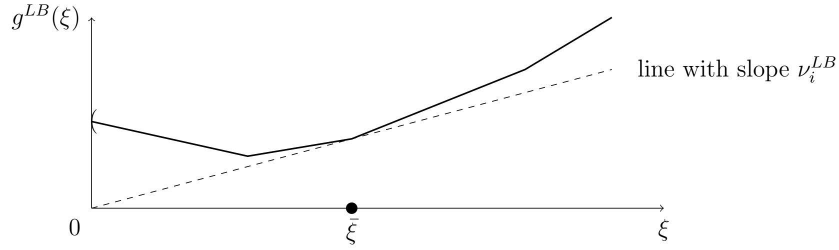

Figure 2: Illustration of the function $g^{\text{LB}}(\xi)$ for $0 < \xi \leq y_i^{\text{UB}}$.

procedure can be used to obtain facets of conv(S) using the following proposition from [42] (see also [22, 36]) specialized to the case of the LPCC:

Proposition 3. If $\alpha^T v \leq \beta$ defines a facet of conv(S^{I_1,I_2}), if the dimension of $S^{I_1\backslash i,I_2}$ is one more that the dimension of S^{I_1,I_2}, and if the constraint $\alpha^T v + \nu_i y_i \leq \beta$ is valid for $S^{I_1\backslash i,I_2}$ and satisfied at equality by at least one point in $S^{I_1\backslash i,I_2} \setminus S^{I_1,I_2}$, then $\alpha^T v + \nu_i y_i \leq \beta$ defines a facet of $S^{I_1\backslash i,I_2}$. □

Determining the optimal choice for ν_i is itself a hard problem, so a relaxation can be used in order to obtain a lower bound ν_i^{LB}. Any lower bound will provide a constraint

$$\alpha^T v \, + \, \nu_i^{\text{LB}} y_i \, \leq \, \beta$$

that is valid throughout $S^{I_1\backslash i,I_2}$. For example, a parametric linear programming problem can be solved to find a lower bound $g^{\text{LB}}(\xi)$. The function $g^{\text{LB}}(\xi)$ is then a piecewise linear convex function in $0 < \xi \leq y_i^{\text{UB}}$, as illustrated in Figure 2. In order to construct a lower bound using parametric linear programming, it is necessary to have polyhedral outer approximations $\bar{S}^{I_1\backslash i,I_2} \supseteq S^{I_1\backslash i,I_2}$ and $\bar{S}^{I_1,I_2} \supseteq S^{I_1,I_2}$. We have $g^{\text{LB}}(0) \geq 0$, since we can assume the constraint $\alpha^T v \leq \beta$ is included in the description of $\bar{S}^{I_1,I_2}$. The left hand limit of $g^{LB}(\xi)$ as $\xi \to 0+$ is found by solving the linear program

$$g^{\text{LB}}(0+) \; = \; \min \{ \beta - \alpha^T v \; : \; v \in \bar{S}^{I_1\backslash i,I_2}, \, w_i = 0, \, y_i = 0 \}.$$

If $g^{\text{LB}}(0+) < 0$ then it is not possible to lift the constraint using the relaxation, since the resulting bound on ν_i is $-\infty$. Using a parametric LP approach, the lower bound ν_i^{LB} is chosen to equal

$$\nu_i^{\text{LB}} \; = \; \inf \left\{ \frac{\beta - \alpha^T v}{\xi} \; \mid \; v \in \bar{S}^{I_1\backslash i,I_2}, \, w_i = 0, \, 0 < \xi \leq y_i^{\text{UB}} \right\},$$

illustrated in the figure, with $\bar{\xi}$ equal to the arginf. It is the slope of the greatest homogeneous affine minorant of $g^{\text{LB}}(\xi)$, and may well be negative. Note that if $g^{\text{LB}}(0+) = 0$ then ν_i^{LB} is the slope of the first line segment of $g^{\text{LB}}(\xi)$.

Example 1. Consider the following LPCC feasible region:

$$\begin{aligned} 2x_1 - y_1 &\leq 4 & (21)\\ 2x_1 + y_1 &\leq 6 & (22)\\ x_1 + 2y_1 &\leq 6 & (23)\\ y_1 - y_2 &\leq 2 & (24)\\ x_1,\, x_2 &\geq 0 & (25)\\ 0 \leq y_1 \perp w_1 \triangleq 3x_1 - 2y_1 + 2 &\geq 0 & (26)\\ 0 \leq y_2 \perp w_2 \triangleq 3x_1 + x_2 + 6y_1 - 14 &\geq 0. & (27) \end{aligned}$$

When $y_1 = 0$, it follows from (21) that $x_1 \leq 2$. We lift this constraint so that it is valid when $y_1 > 0$, giving a constraint of the form

$$x_1 + \nu_1 y_1 \leq 2.$$

By complementarity (26), if $y_1 > 0$ then $w_1 = 0$. We can calculate the function $g^{\text{LB}}(\xi)$ for $\xi > 0$ using the following LP:

$$\begin{aligned} g^{\text{LB}}(\xi) \;=\; \text{minimize} \quad & 2 - x_1 \\ \text{subject to} \quad & 2x_1 \leq 4 + \xi \\ & 2x_1 \leq 6 - \xi \\ & x_1 \leq 6 - 2\xi \\ & -y_2 \leq 2 - \xi \\ & 3x_1 = 2\xi - 2 \\ & 3x_1 + x_2 \geq 14 - 6\xi \\ & x_1,\, x_2,\, y_2 \geq 0. \end{aligned}$$

This gives

$$g^{\text{LB}}(\xi) = \begin{cases} +\infty & \text{if } 0 < \xi < 1 \\ \dfrac{8 - 2\xi}{3} & \text{if } 1 \leq \xi \leq 2.5 \\ +\infty & \text{if } 2.5 < \xi \end{cases}$$

as illustrated in Figure 3(a). The greatest slope for a homogeneous affine minorant of $g^{\text{LB}}(\xi)$ is $\nu_1^{\text{LB}} = 0.4$, leading to the lifted constraint

$$x_1 + 0.4y_1 \leq 2. \qquad (28)$$

The projection of the LP relaxation onto the (x_1, y_1) space is illustrated in Figure 3(b). Note that if the complementarity condition in (27) is imposed when calculating $g(\xi)$ then either

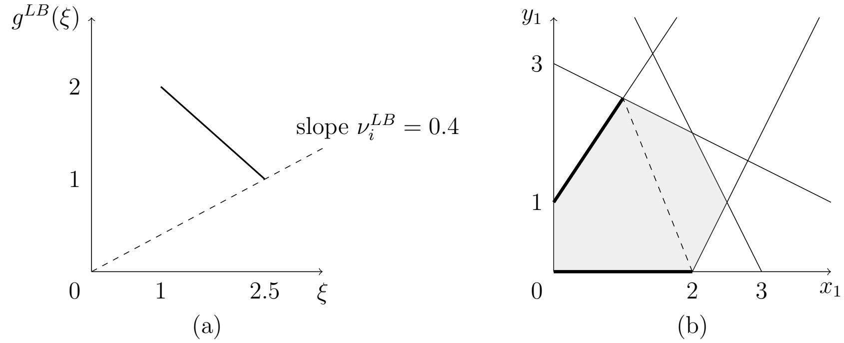

Figure 3: (a) Illustration of the function $g^{\text{LB}}(\xi)$ for $0 < \xi$ in Example 1. (b) Projection of example on (x_1, y_1) space, with the LP relaxation feasible region shaded. The thick line segments indicate points satisfying $y_1 w_1 = 0$. The dashed line is the lifted constraint (28).

$y_1 \le 2$ or $3x_1 + 6y_1 \le 14$. This leads to a slightly larger coefficient $\nu = 2/3$, resulting in a somewhat stronger lifted constraint. Thus, $g^{\text{LB}}(\xi)$ is a strict minorant of $g(\xi)$ for this example. □

A constraint that is valid for S^{I_1,I_2} can be successively lifted in all the variables $y_i, i \in I_1$ and $w_i, i \in I_2$, leading to a valid constraint on S. The order of lifting can affect the resulting constraint. Finding sequence-independent liftings is a topic of active research in integer programming. For more details see [42].

7 Semidefinite constraints

Let $\xi = (x, y, w) \in \mathbb{R}^{n+2m}$. By taking products of the constraints defining (1), we can obtain nonconvex quadratic constraints on the elements of ξ. For example, the complementarity relationships imply $y_i w_i = 0$ for each i. These constraints can be relaxed to linear constraints by introducing a matrix Υ to represent

$$\Upsilon = \xi\xi^T,$$

replacing all quadratic terms by the corresponding entries in Υ and then relaxing the equality to the semidefinite inequality

$$\Upsilon \succeq \xi\xi^T.$$

This leads to a semidefinite programming relaxation of (1) that is tighter than the LP relaxation. This approach is well-known for quadratically constrained quadratic programs, and there has been recent research on trying to improve it; see Luo et al. [33] and its references, for example.

Tightened linear equalities for (1) can be obtained by projecting down from the (Υ, ξ) space onto the ξ space. This convex relaxation procedure has been extensively analyzed for $0 - 1$ programming, and it has been shown by Lovasz and Schrijver [32] that repeated application leads to the convex hull of the feasible region. Kojima and Tunçel [30] explore

			Percentage of gap closed	
n	m	k	SOCP	SDP
20	50	50	15.2	75.0
30	40	60	24.3	63.4
40	30	60	26.7	69.3
40	40	10	22.6	54.0
50	30	20	54.2	99.9
60	20	30	76.9	100.0

Table 6: SDP results.

semidefinite programming (SDP) relaxations of quadratic constraints in detail and characterize the results of successive convex relaxation. They extend the work of [32] to finding the convex hull of a general nonconvex region, not necessarily one that arises from a 0-1 integer programming problem. Variants of their procedure converge to the convex hull in a finite number of steps. Anstreicher [3] showed that an SDP approach can be complementary to using the reformulation-linearization technique. Recent work on semidefinite relaxation approaches to mixed integer nonlinear programming problems includes [19, 41].

We experimented with using CSDP [17] for solving SDP relaxations of LPCCs generated in the same way as those in §2. We were unable to solve instances as large as those in §2 in reasonable computational times, so we report results on smaller instances. We used a positive semidefinite matrix of the form

$$\bar{\Upsilon} \triangleq \begin{bmatrix} 1 \\ x \\ y \end{bmatrix} \begin{bmatrix} 1 & x^T & y^T \end{bmatrix} = \begin{bmatrix} 1 & x^T & y^T \\ x & xx^T & xy^T \\ y & yx^T & yy^T \end{bmatrix},$$

with the equality constraint relaxed. The model included constraints that all entries in Υ be nonnegative, that the entries corresponding to $y_i w_i$ be zero, that the entries in the first row and column of $\bar{\Upsilon}$ satisfy the appropriate linear constraints on x and y in (1), that the linear combinations of entries in $\bar{\Upsilon}$ corresponding to the products $y_i(\bar{A}x + \bar{B}y + \bar{C}w - b)_j$, $(\bar{A}x + \bar{B}y + \bar{C}w - b)_i(\bar{A}x + \bar{B}y + \bar{C}w - b)_j$ and $x_i(\bar{A}x + \bar{B}y + \bar{C}w - b)_j$ be nonnegative, and that the entries in $\bar{\Upsilon}$ corresponding to the terms x_i, x_i^2, y_j, and y_j^2 satisfy the convex quadratic constraints

$$x_i^2 \leq x_i^{\text{UB}} x_i \quad \text{and} \quad y_j^2 \leq y_j^{\text{UB}} y_j.$$

The results are contained in Table 6. Also contained in the table are results for the second-order cone programming (SOCP) relaxation which includes all the constraints of the SDP, except for the requirement that Υ be positive semidefinite. A lower bound was obtained by solving the relaxations of these problems containing the bound cuts (2) and the McCormick constraints (18), and the integer program (6) was solved in order to obtain the optimal value. The table gives the proportion of the gap between the optimal value and the lower bound from the McCormick cuts that is closed using the SOCP and SDP relaxations.

It is clear from the table that the SDP relaxation can be very strong. However, the computational time for this approach is not competitive with an LP-based integer programming approach, at least for these problems and when solving the SDPs to optimality with a primal-dual method. It may be helpful to use alternative techniques to solve the SDP problems, such as those in, for example, [18, 24, 31, 47, 48], techniques that can also be used to solve the SDPs approximately. Approximate solutions may be appropriate when the solver is incorporated into a branching scheme. It may also be more effective to add the linear constraints on Υ selectively as cutting planes.

It is also possible to construct constraints that the linear combinations of entries in $\bar{\Upsilon}$ corresponding to the products $y_i w_k$, $w_i w_k$, $w_i(\bar{A}x + \bar{B}y + \bar{C}w - b)_j$, and $x_i w_k$ be nonnegative, but we found these constraints resulted in SDPs that were too large for our solver. In principle, these constraints could be added as cutting planes, as could the earlier ones.

8 Conclusions

The cuts described in this paper can dramatically reduce the gap between the global optimal value and the lower bound provided by a simple linear programming relaxation. We investigated adding whole families of cuts and quantifying how the lower bound is improved. The cuts can often be greatly improved by refining them, that is, by applying them and then recalculating them. For example, with bound cuts, a lower bound can be calculated using a single application of the bound-calculating procedure. The gap between the optimal value and this lower bound is reduced by 47% or more in all our test cases when the bounds are refined 8 times, and the overall reduction in the duality gap is on the order of 75% to 90% when compared with the initial LP relaxation. Further, using refinements can often result in a lower overall runtime to solve the LPCC to optimality, even when taking the time for the refinements into account. The bound cuts appear to be more effective than the disjunctive cuts, which are expensive to calculate in their full form. Methods to speed up the calculation of disjunctive cuts certainly save time, but they appear to give cuts that are noticeably weaker than using the full disjunctive cut generation LP. When the dimension of the x variables is not too large, we have shown how to construct linear relaxations of the nonconvex quadratic constraint $y^T w \leq 0$ by expressing the constraint in terms of just the x and y variables. These McCormick constraints can be very effective, especially if they are refined. The refined cuts can typically reduce the gap by about 90% on our test problems. Novel quadratic constraints can be used to improve the McCormick cuts, but the improvement is not great for our test instances. Tighter relaxations can be obtained by using semidefinite relaxations, but these are currently expensive computationally to solve to optimality. Methods to approximately solve the SDP relaxations could be useful, as could methods using the SDP as a cut-generation mechanism in an LP approach as in [31, 41, 48].

We are currently investigating the use of these classes of constraints in a branch-and-cut algorithm for finding the global optimum to the LPCC. In such an algorithm, the cuts are added more selectively, rather than adding whole families of cuts.

Acknowledgements

We are grateful to two anonymous referees for their careful reading of the manuscript and constructive comments.

References

[1] C. S. Adjiman, I. P. Androulakis, and C. A. Floudas. A global optimization method, αBB, for general twice-differentiable constrained NLPs — I. Theoretical advances. *Computers and Chemical Engineering*, 22(9):1137–1158, 1998.

[2] F. A. Al-Khayyal and J. E. Falk. Jointly constrained biconvex programming. *Mathematics of Operations Research*, 8(2):273–286, 1983.

[3] K. M. Anstreicher. Semidefinite programming versus the reformulation-linearization technique for nonconvex quadratically constrained quadratic programming. *Journal of Global Optimization*, 43(2–3):471–484, 2009.

[4] C. Audet, J. Haddad, and G. Savard. Disjunctive cuts for continuous linear bilevel programming. *Optimization Letters*, 1(3):259–267, 2007.

[5] C. Audet, G. Savard, and W. Zghal. New branch-and-cut algorithm for bilevel linear programming. *Journal of Optimization Theory and Applications*, 38(2):353–370, 2007.

[6] E. Balas. Intersection cuts — a new type of cutting planes for integer programming. *Operations Research*, 19:19–39, 1971.

[7] E. Balas. Disjunctive programming. *Annals of Discrete Mathematics*, 5:3–51, 1979.

[8] E. Balas. Disjunctive programming: Properties of the convex hull of feasible points. *Discrete Applied Mathematics*, 89(1–3):3–44, 1998.

[9] E. Balas, S. Ceria, and G. Cornuéjols. Mixed 0–1 programming by lift-and-project in a branch-and-cut framework. *Management Science*, 42(9):1229–1246, 1996.

[10] E. Balas, S. Ceria, G. Cornuéjols, and N. Natraj. Gomory cuts revisited. *Operations Research Letters*, 19:1–9, 1996.

[11] E. Balas and M. Perregaard. Lift-and-project for mixed 0-1 programming: recent progress. *Discrete Applied Mathematics*, 123(1-3):129–154, 2002.

[12] E. Balas and M. Perregaard. A precise correspondence between lift-and-project cuts, simple disjunctive cuts, and mixed integer Gomory cuts for 0-1 programming. *Mathematical Programming*, 94(2–3):221–245, 2003.

[13] X. Bao, N. V. Sahinidis, and M. Tawarmalani. Multiterm polyhedral relaxations for nonconvex, quadratically-constrained quadratic programs. *Optimization Methods and Software*, 24(4–5):485–504, 2009.

[14] P. Belotti. Disjunctive cuts for nonconvex MINLPs. Technical report, Lehigh University, Bethlehem, PA, 2009.

[15] P. Belotti, J. Lee, L. Liberti, F. Margot, and A. Wachter. Branching and bounds tightening techniques for non-convex MINLP. *Optimization Methods and Software*, 24(4–5):597–634, 2009.

[16] P. Belotti, L. Liberti, A. Lodi, G. Nannicini, and A. Tramontani. Disjunctive inequalities: Applications and extensions. In J. J. Cochran, editor, *Encyclopedia of Operations Research and Management Science*. Wiley, 2010 (accepted for publication).

[17] B. Borchers. CSDP, a C library for semidefinite programming. *Optimization Methods and Software*, 11:613–623, 1999.

[18] S. Burer and R. D. C. Monteiro. A nonlinear programming algorithm for solving semidefinite programs via low-rank factorization. *Mathematical Programming*, 95(2):329–357, 2003.

[19] S. Burer and A. Saxena. Old wine in a new bottle: The MILP road to MIQCP. Technical report, Department of Management Sciences, University of Iowa, Iowa City, IA 52242, 2009.

[20] R. Byrd, J. Nocedal, and R. Waltz. KNITRO: An integrated package for nonlinear optimization. In G. di Pillo and M. Roma, editors, *Large-Scale Nonlinear Optimization*, pages 35–59. Springer-Verlag, 2006.

[21] S. Ceria and G. Pataki. Solving integer and disjunctive programs by lift-and-project. In *Proceedings of the Sixth IPCO Conference*, volume 1412 of *Lecture Notes in Computer Science*, pages 271–283, Berlin, 1998. Springer Verlag.

[22] I. R. de Farias Jr., E. L. Johnson, and G. L. Nemhauser. Facets of the complementarity knapsack polytope. *Mathematics of Operations Research*, 27(1):210–226, 2002.

[23] R. Fletcher and S. Leyffer. Solving mathematical programs with complementarity constraints as nonlinear programs. *Optimization Methods and Software*, 18(1):15–40, 2004.

[24] C. Helmberg. Numerical evaluation of SBmethod. *Mathematical Programming*, 95(2):381–406, 2003.

[25] B. F. Hobbs, C. B. Metzler, and J.S. Pang. Strategic gaming analysis for electric power systems: an MPEC approach. *IEEE Transactions on Power Systems*, 15(2):638–645, 2000.

[26] J. Hu, J. E. Mitchell, and J.S. Pang. An LPCC approach to nonconvex quadratic programs. *Mathematical Programming*, online first, 2011.

[27] J. Hu, J. E. Mitchell, J.S. Pang, K. P. Bennett, and G. Kunapuli. On the global solution of linear programs with linear complementarity constraints. *SIAM Journal on Optimization*, 19(1):445–471, 2008.

[28] J. Hu, J. E. Mitchell, J.S. Pang, and B. Yu. On linear programs with linear complementarity constraints. Technical report, Department of Mathematical Sciences, Rensselaer Polytechnic Institute, Troy, NY 12180, September 2009. Accepted for publication in *Journal of Global Optimization*.

[29] J. J. Júdice, H. D. Sherali, I. M. Ribeiro, and A. M. Faustino. A complementarity-based partitioning and disjunctive cut algorithm for mathematical programming problems with equilibrium constraints. *Journal of Global Optimization*, 36:89–114, 2006.

[30] M. Kojima and L. Tuncel. Cones of matrices and successive convex relaxations of nonconvex sets. *SIAM Journal on Optimization*, 10(3):750–778, 2000.

[31] K. Krishnan and J. E. Mitchell. A unifying framework for several cutting plane methods for semidefinite programming. *Optimization Methods and Software*, 21(1):57–74, 2006.

[32] L. Lovász and A. Schrijver. Cones of matrices and set-functions and 0-1 optimization. *SIAM Journal on Optimization*, 1(2):166–190, 1991.

[33] Z.-Q. Luo, W.-K. Ma, A. M.-C. So, Y. Ye, and S. Zhang. Semidefinite relaxation of quadratic optimization problems. *IEEE Signal Processing Magazine*, 27(3):20–34, 2010.

[34] G. P. McCormick. Computability of global solutions to factorable nonconvex programs: part I — convex underestimating problems. *Mathematical Programming*, 10:147–175, 1976.

[35] J. E. Mitchell, J.S. Pang, and B. Yu. Convex quadratic relaxations of nonconvex quadratically constrained quadratic programs. Technical report, Mathematical Sciences, Rensselaer Polytechnic Institute, Troy, NY 12180, March 2011.

[36] G. L. Nemhauser and L. A. Wolsey. *Integer and Combinatorial Optimization*. John Wiley, New York, 1988.

[37] J.S. Pang. Three modeling paradigms in mathematical programming. *Mathematical Programming*, 125(2):297–323, 2010.

[38] J.S. Pang, T. Olson, and C. Priebe. A likelihodd-MPEC approach to target classification. *Mathematical Programming*, 96(1):1–31, 2003.

[39] J.S. Pang and C. L. Su. On estimating pure characteristics models. Technical report, Industrial and Enterprise Systems Engineering, University of Illinois, Urbana-Champaign, IL, in preparation.

[40] M. Perregaard. *Generating disjunctive cuts for mixed integer programs.* PhD thesis, Carnegie Mellon University, Graduate School of Industrial Administration, Pittsburgh, PA, September 2003.

[41] A. Qualizza, P. Belotti, and F. Margot. Linear programming relaxations of quadratically constrained quadratic programs. In *IMA Volume Series*. Springer, 2010. Accepted. Tepper Working Paper 2009-E20 (revised 4/2010).

[42] J.-P. P. Richard and M. Tawarmalani. Lifted inequalities: a framework for generating strong cuts for nonlinear progams. *Mathematical Programming*, 121(1):61–104, 2010.

[43] N. Sahinidis. BARON: A general purpose global optimization software package. *Journal of Global Optimization*, 8:201–205, 1996.

[44] A. Saxena. *Integer Programming, a Technology.* PhD thesis, Tepper School of Business, Carnegie Mellon University, 2007.

[45] A. Saxena, P. Bonami, and J. Lee. Convex relaxations of non-convex mixed integer quadratically constrained programs: extended formulations. *Mathematical Programming*, 124(1–2):383–411, 2010.

[46] A. Saxena, P. Bonami, and J. Lee. Convex relaxations of non-convex mixed integer quadratically constrained programs: projected formulations. *Mathematical Programming*, 3 March:online first, 2010.

[47] K. K. Sivaramakrishnan. A parallel interior point decomposition algorithm for block angular semidefinite programs. *Computational Optimization and Applications*, 46(1):1–29, 2010.

[48] K. K. Sivaramakrishnan and J. E. Mitchell. Properties of a cutting plane method for semidefinite programming. Technical report, Mathematical Sciences, Rensselaer Polytechnic Institute, Troy, NY 12180, September 2011.

[49] R. A. Stubbs and S. Mehrotra. A branch-and-cut method for 0-1 mixed convex programming. *Mathematical Programming*, 86:515–532, 1999.

[50] M. Tawarmalani and N. Sahinidis. *Convexification and Global Optimization in Continuous and Mixed-Integer Nonlinear Programming: Theory, Algorithms, Software, and Applications.* Kluwer, Dordrecht, The Netherlands, 2002.