ON THE GLOBAL SOLUTION OF LINEAR PROGRAMS WITH LINEAR COMPLEMENTARITY CONSTRAINTS*

JING HU, JOHN E. MITCHELL, JONG-SHI PANG, KRISTIN P. BENNETT, AND GAUTAM KUNAPULI

Abstract. This paper presents a parameter-free integer-programming based algorithm for the global resolution of a linear program with linear complementarity constraints (LPCC). The cornerstone of the algorithm is a minimax integer program formulation that characterizes and provides certificates for the three outcomes—infeasibility, unboundedness, or solvability—of an LPCC. An extreme point/ray generation scheme in the spirit of Benders decomposition is developed, from which valid inequalities in the form of satisfiability constraints are obtained. The feasibility problem of these inequalities and the carefully guided linear programming relaxations of the LPCC are the workhorse of the algorithm, which also employs a specialized procedure for the sparsification of the satifiability cuts. We establish the finite termination of the algorithm and report computational results using the algorithm for solving randomly generated LPCCs of reasonable sizes. The results establish that the algorithm can handle infeasible, unbounded, and solvable LPCCs effectively.

1. Introduction. Forming a subclass of the class of mathematical programs with equilibrium/complementarity constraints (MPECs/MPCCs) [37, 39, 12], linear programs with linear complementarity constraints (LPCCs) are disjunctive linear optimization problems that contain a set of complementarity conditions. In turn, a large subclass of LPCCs are bilevel linear/quadratic programs [11] that provide a broad modeling framework for parameter identification in convex quadratic programming; an example of such an application was proposed recently for the cross validation of a host of machine learning problems [6, 33, 32]. While there have been significant recent advances on nonlinear programming (NLP) based computational methods for solving MPECs and the closely related MPCCs, [1, 2, 3, 8, 15, 16, 19, 20, 29, 30, 25, 35, 36, 44, 45], much of which have nevertheless focused on obtaining stationary solutions [13, 14, 37, 39, 38, 40, 44, 50, 49, 48], the global solution of an LPCC remains elusive. Particularly impressive among these advances is the suite of NLP solvers publicly available on the NEOS system at `http://www-neos.mcs.anl.gov/neos/solvers/index.html`; many of them, such as FILTER and KNITRO, are capable of producing a solution of some sort to an LPCC very efficiently. Yet, they are incapable of ascertaining the quality of the computed solution. This is the major deficiency of these numerical solvers. Continuing our foray into the subject of computing global solutions of LPCCs, which begins with the recent article [42] that pertains to a special problem arising from the optimization of the value-at-risk, the present paper proposes a parameter-free integer-programming based cutting-plane algorithm for globally resolving a general LPCC.

As a disjunctive linear optimization problem, the global solution of an LPCC has been the subject of sustained, but not particularly focused investigation since the early work of Ibaraki [26, 27] and Jeroslow [28], who pioneered some cutting-plane methods for solving a "complementary program", which is a historical and not widely used name for an LPCC. Over the years, various integer programming based methods [4, 5, 21] and global optimization based methods [17, 18, 46, 47] have been developed that are applicable to an LPCC. In this paper, we present a new cutting-plane method that will successfully resolve a general LPCC in finite time; i.e., the method will terminate with one of the following three mutually exclusive conclusions: the LPCC is infeasible, the LPCC is feasible but has an unbounded objective, or the LPCC attains a finite optimal solution. We also leverage the advances of the NLP solvers and use two of them to benchmark our algorithm. In addition, we propose a simple linear programming based pre-processor whose effectiveness will be demonstrated via computational results.

The proposed method begins with an equivalent formulation of an LPCC as a 0-1 integer program (IP) involving a conceptually very large parameter, whose existence is not guaranteed unless a certain boundedness condition holds. Via dualization of the linear programming relaxation of the IP, we obtain a minimax 0-1 integer program, which yields a certificate for the three states of the LPCC, without any *a priori* boundedness assumption. The original 0-1 IP with the conceptual parameter provides the formulation for the application of Benders decomposition [34], which we show can be implemented without involving the parameter in any way. Thus, the resulting algorithm is reminiscent of the well-known Phase I implementation of the "big-M" method for solving linear programs, wherein the big-M formulation is only conceptual whose practical solution does not require the knowledge of the scalar M.

The implementation of our parameter-free algorithm is accomplished by solving integer subprograms defined solely by *satisfiability constraints* [7, 31]; in turn, each such constraint corresponds to a "piece" of the LPCC. Using this interpretation, the overall algorithm can be considered as solving the LPCC by searching on its (finitely

*This work was supported in part by the Office of Naval Research under grant no. N00014-06-1-0014. The work of Mitchell was supported by the National Science Foundation under grant DMS-0317323. All authors, except Pang, are affiliated with the Department of Mathematical Sciences, Rensselaer Polytechnic Institute, Troy, New York 12180-1590, U.S.A. Their email addresses are: huj, mitchj, bennek, kunapg@rpi.edu, respectively. Pang was affiliated with the same department when the original manuscript was written; he is presently affiliated with the Department of Industrial and Enterprise Systems Engineering, University of Illinois at Urbana-Champaign, Urbana, Illinois 61801, U.S.A. Pang's email address is: jspang@uiuc.edu.

many) linear programming pieces, with the search guided by solving the satisfiability IPs. The implementation of the algorithm is aided by valid upper bounds on the LPCC optimal objective value that are being updated as the algorithm progresses, which also serve to provide the desired certificates at the termination of the algorithm.

Hooker [22, 23] and Hooker and Ottosson [24] have presented a general Benders decomposition framework for integer programming problems, where the subproblems and the master problem may be solved by various techniques, for example constraint programming. The constraints returned by the subproblems may well be satisfiability constraints similar to those we derive from solving a linear programming subproblem, leading to a satisfiability master problem. Hooker develops a broad framework for integrating different solution methodologies, with the solution approaches driven by the types of constraints. Different decomposition methods are available depending on the particular mix of types of constraints. Codato and Fischetti [9] have specialized Hooker's approach to solve integer programs arising from linear programs with conditional constraints of the kind "if then", of which the complementarity condition is a special case. Such constraints are modeled with the introduction of integer variables together with a big-M coefficients. The cited article presents computational results for feasible bounded problems where only the binary variables associated with the big-M coefficients appear in the objective function. In our problem (2.3), the objective function is a linear function of the continuous variables; the objective function could be regarded as a nonlinear function of the binary variables (see $\varphi(z)$ defined in (2.9)). Our algorithmic approach can successfully characterize infeasible and unbounded LPCC problems as well as solve problems with finite optimal value. Hooker [23] states that "the success of a Benders method often rests on finding strong Benders cuts that rule out as many infeasible solutions as possible". The sparsification methodology we present in Section 4 is an approach to generate strong Benders cuts.

The organization of the rest of the paper is as follows. Section 2 presents the formal statement of the LPCC, summarizes the three states of the LPCC, and introduces the new minmax IP formulation. Section 3 reformulates the minmax IP formulation in terms of the extreme points and rays of the key polyhedron Ξ (see (2.6)) and established the theoretical foundation for the cutting-plane algorithm to be presented in Section 5. The key steps of the algorithm, which involve solving linear programs (LPs) to sparsify the satisfiability constraints, are explained in Section 4. The sixth and last section reports the computational results and completes the paper with some concluding remarks.

2. Preliminary Discussion.

Let $c \in \Re^n$, $d \in \Re^m$, $f \in \Re^k$, $q \in \Re^m$, $A \in \Re^{k\times n}$, $B \in \Re^{k\times m}$, $M \in \Re^{m\times m}$, and $N \in \Re^{m\times n}$ be given. Consider the linear program with linear complementarity constraints (LPCC) [41] of finding $(x, y) \in \Re^n \times \Re^m$ in order to

$$\begin{array}{ll} \underset{(x,y)}{\text{minimize}} & c^T x + d^T y \\ \text{subject to} & Ax + By \geq f \\ \text{and} & 0 \leq y \perp q + Nx + My \geq 0, \end{array} \tag{2.1}$$

where $a \perp b$ means that the two vectors are orthogonal; i.e., $a^T b = 0$. It is well-known that the LPCC is equivalent to the minimization of a large number of linear programs, each defined on one *piece* of the feasible region of the LPCC. That is, for each subset α of $\{1, \cdots, m\}$ with complement $\bar{\alpha}$, we may consider the LP(α):

$$\begin{array}{ll} \underset{(x,y)}{\text{minimize}} & c^T x + d^T y \\ \text{subject to} & Ax + By \geq f \\ & (\,q + Nx + My\,)_\alpha \geq 0 = y_\alpha \\ \text{and} & (\,q + Nx + My\,)_{\bar{\alpha}} = 0 \leq y_{\bar{\alpha}}. \end{array} \tag{2.2}$$

The following facts are consequences of the disjunctive property of the complementarity condition:

(a) the LPCC (2.1) is infeasible if and only if the LP(α) is infeasible for *all* $\alpha \subseteq \{1, \cdots, m\}$;

(b) the LPCC (2.1) is feasible and has an unbounded objective if and only if the LP(α) is feasible and has an unbounded objective for *some* $\alpha \subseteq \{1, \cdots, m\}$;

(c) the LPCC (2.1) is feasible and attains a finite optimal objective value if and only if (i) a subset α of $\{1, \cdots, m\}$ exists such that the LP(α) is feasible, and (b) every such feasible LP(α) has a finite optimal objective value; in this case, the optimal objective value of the LPCC (2.1), denoted $\text{LPCC}_{\min}$, is the minimum of the optimal objective values of all such feasible LPs.

The first step in our development of an IP-based algorithm for solving the LPCC (2.1) without any *a priori* assumption is to derive results parallel to the above three facts in terms of some parameter-free integer problems.

For this purpose, we recall the standard approach of solving (2.1) as an IP containing a large parameter. This approach is based on the following "equivalent" IP formulation of (2.1) wherein the complementarity constraint is reformulated in terms of the binary vector $z \in \{0,1\}^m$ via a conceptually very large scalar $\theta > 0$:

$$\begin{array}{lrcl}
\underset{(x,y,z)}{\text{minimize}} & c^T x + d^T y & & \\
\text{subject to} & Ax + By & \geq & f \\
& \theta\, z \,\geq\, q + Nx + My & \geq & 0 \\
& \theta(\, \mathbf{1} - z\,) \,\geq\, y & \geq & 0 \\
\text{and} & z & \in & \{\,0,1\,\}^m,
\end{array} \tag{2.3}$$

where $\mathbf{1}$ is the m-vector of all ones. In the standard approach, we first derive a valid value on θ by solving LPs to obtain bounds on all the variables and constraints of (2.1). We then solve the fixed IP (2.3) using the so-obtained θ by, for example, the Benders approach. There are two drawbacks of such an approach: one is the limitation of the approach to problems with bounded feasible regions; the other drawback is the nontrivial computation to derive the required bounds even if they are known to exist implicitly. In contrast, our new approach removes such a theoretical restriction and eliminates the front-end computation of bounds. The price of the new approach is that it solves a (finite) family of IPs of a special type, each defined solely by constraints of the *satisfiability* type. The following discussion sets the stage for the approach. [A referee suggests that "another drawback of attacking the integer program (2.3) is the probably (very) weak LP relaxations (which will affect the convergence of branch and cut methods as well as approaches based on Benders decomposition)".]

For a given binary vector z and a positive scalar θ, we associate with (2.3) the linear program below, which we denote $\text{LP}(\theta; z)$:

$$\begin{array}{lrcll}
\underset{(x,y)}{\text{minimize}} & c^T x + d^T y & & & \\
\text{subject to} & Ax + By & \geq & f & (\,\lambda\,) \\
& Nx + My & \geq & -q & (\,u^-\,) \\
& -Nx - My & \geq & q - \theta\, z & (\,u^+\,) \\
& -y & \geq & -\theta\,(\,\mathbf{1} - z\,) & (\,v\,) \\
\text{and} & y & \geq & 0, &
\end{array} \tag{2.4}$$

where the dual variables of the respective constraints are given in the parentheses. The dual of (2.4), which we denote $\text{DLP}(\theta, z)$, is:

$$\begin{array}{ll}
\underset{(\lambda, u^\pm, v)}{\text{maximize}} & f^T\lambda + q^T(\,u^+ - u^-\,) - \theta\,\left[\,z^T u^+ + (\,\mathbf{1} - z\,)^T v\,\right] \\
\text{subject to} & A^T\lambda - N^T(\,u^+ - u^-\,) \,=\, c \\
& B^T\lambda - M^T(\,u^+ - u^-\,) - v \,\leq\, d \\
\text{and} & (\,\lambda, u^\pm, v\,) \,\geq\, 0.
\end{array} \tag{2.5}$$

Let $\Xi \subseteq \Re^{k+3m}$ be the feasible region of the $\text{DLP}(\theta, z)$; i.e.,

$$\Xi \equiv \left\{ \begin{array}{ll} (\,\lambda, u^\pm, v\,) \,\geq\, 0 : & A^T\lambda - N^T(\,u^+ - u^-\,) \,=\, c \\ & B^T\lambda - M^T(\,u^+ - u^-\,) - v \,\leq\, d \end{array} \right\}. \tag{2.6}$$

Note that Ξ is a fixed polyhedron independent of the pair (θ, z); Ξ has at least one extreme point if it is nonempty. Let $\text{LP}_{\min}(\theta; z)$ and $d(\theta; z)$ denote the optimal objective value of (2.4) and (2.5), respectively. Throughout, we adopt the standard convention that the optimal objective value of an infeasible maximization (minimization) problem is defined to be $-\infty$ (∞, respectively). We summarize some basic relations between the above programs in the following result.

Proposition 2.1. The following three statements hold.

(a) Any feasible solution (x^0, y^0) of (2.1) induces a pair (θ_0, z^0), where $\theta_0 > 0$ and $z^0 \in \{0,1\}^m$, such that the tuple (x^0, y^0, z^0) is feasible to (2.3) for all $\theta \geq \theta_0$; such a z^0 has the property that

$$\begin{array}{rcl} (\, q + Nx^0 + My^0 \,)_i \, > \, 0 & \Rightarrow & z_i^0 \, = \, 1 \\ (\, y^0 \,)_i \, > \, 0 & \Rightarrow & z_i^0 \, = \, 0. \end{array} \tag{2.7}$$

(b) Conversely, if (x^0, y^0, z^0) is feasible to (2.3) for some $\theta \geq 0$, then (x^0, y^0) is feasible to (2.1).

(c) If (x^0, y^0) is an optimal solution to (2.1), then it is optimal to the LP(θ, z^0) for all pairs (θ, z^0) such that $\theta \geq \theta_0$ and z^0 satisfies (2.7); moreover, for each $\theta > \theta_0$, any optimal solution $(\widehat{\lambda}, \widehat{u}^{\pm}, \widehat{v})$ of the DLP(θ, z^0) satisfies $(z^0)^T \widehat{u}^+ + (\,\mathbf{1} - z^0\,)^T \widehat{v} = 0$.

Proof. Only (c) requires a proof. Suppose (x^0, y^0) is optimal to (2.1). Let (θ, z^0) such that $\theta \geq \theta_0$ and $z^0 \in \{0,1\}^m$ satisfies (2.7). Then (x^0, y^0) is feasible to the LP(θ, z^0); hence

$$c^T x^0 + d^T y^0 \, \geq \, \mathrm{LP}_{\min}(\theta, z^0). \tag{2.8}$$

But the reverse inequality must hold because of (b) and the optimality of (x^0, y^0) to (2.1). Consequently, equality holds in (2.8). For $\theta > \theta_0$, if i is such that $z_i^0 > 0$, then

$$(\, q + Nx^0 + My^0 \,) \, \leq \, \theta_0 \, z_i^0 \, < \theta \, z_i^0,$$

and complementary slackness implies $(\widehat{u}^+)_i = 0$. Similarly, we can show that $z_i^0 = 0 \Rightarrow v_i = 0$. Hence (c) follows. $\square$

2.1. The parameter-free dual programs. Property (c) of Proposition 2.1 suggests that the inequality constraint $z^T u^+ + (\mathbf{1} - z)^T v \leq 0$, or equivalently, the equality constraint $z^T u^+ + (\mathbf{1} - z)^T v = 0$ (because all variables are nonnegative and $z \in \{0,1\}^m$), should have an important role to play in an IP approach to the LPCC. This motivates us to define two value functions on the binary vectors. Specifically, for any $z \in \{0,1\}^m$, define

$$\begin{array}{rll} \Re \cup \{\, \pm\infty \,\} \ni \varphi(z) \equiv & \underset{(\lambda, u^{\pm}, v)}{\text{maximum}} & f^T \lambda + q^T (\, u^+ - u^- \,) \\ & \text{subject to} & A^T \lambda - N^T (\, u^+ - u^- \,) \, = \, c \\ & & B^T \lambda - M^T (\, u^+ - u^- \,) - v \, \leq \, d \\ & & (\, \lambda, u^{\pm}, v \,) \, \geq \, 0 \\ & \text{and} & z^T u^+ + (\mathbf{1} - z)^T v \, \leq \, 0 \end{array} \tag{2.9}$$

and its homogenization:

$$\begin{array}{rll} \{\, 0, \infty \,\} \ni \varphi_0(z) \equiv & \underset{(\lambda, u^{\pm}, v)}{\text{maximum}} & f^T \lambda + q^T (\, u^+ - u^- \,) \\ & \text{subject to} & A^T \lambda - N^T (\, u^+ - u^- \,) \, = \, 0 \\ & & B^T \lambda - M^T (\, u^+ - u^- \,) - v \, \leq \, 0 \\ & & (\, \lambda, u^{\pm}, v \,) \, \geq \, 0 \\ & \text{and} & z^T u^+ + (\, \mathbf{1} - z \,)^T v \, \leq \, 0. \end{array} \tag{2.10}$$

Clearly, (2.10) is always feasible and $\varphi_0(z)$ takes on the values 0 or ∞ only. Unlike (2.10) which is independent of the pair (c, d), (2.9) depends on (c, d) and is not guaranteed to be feasible; thus $\varphi(z) \in \Re \cup \{\pm\infty\}$. For any pair (c, d) for which (2.9) is feasible, we have

$$\varphi(z) \, < \, \infty \, \Leftrightarrow \, \varphi_0(z) \, = \, 0.$$

To this equivalence we add the following proposition that describes a one-to-one correspondence between (2.10) and the feasible pieces of the LPCC. The *support* of a vector z, denoted supp(z) is the index set of the nonzero components of z.

PROPOSITION 2.2. For any $z \in \{0,1\}^m$, $\varphi_0(z) = 0$ if and only if the LP(α) is feasible, where $\alpha \equiv \text{supp}(z)$.

Proof. The dual of (2.10) is

$$\begin{array}{llll} \underset{(x,y)}{\text{minimize}} & 0^T x + 0^T y & & \\ \text{subject to} & Ax + By & \geq & f \\ & \theta\, z \,\geq\, q + Nx + My & \geq & 0 \\ \text{and} & \theta\,(\,\mathbf{1} - z\,) \,\geq\, y & \geq & 0. \end{array} \tag{2.11}$$

By LP duality, it follows that if $\varphi_0(z) = 0$, then (2.11) is feasible for any $\theta > 0$; conversely, if (2.11) is feasible for some $\theta > 0$, then $\varphi_0(z) = 0$. In turn, (2.11) is feasible for some $\theta > 0$ if and only if the LP(α) is feasible for $\alpha \equiv \text{supp}(z)$. $\square$

For subsequent purposes, it would be useful to record the following equivalence between the extreme points/rays of the feasible region of (2.9) and those of the feasible set Ξ.

PROPOSITION 2.3. For any $z \in [0,1]^m$, a feasible solution $(\lambda^p, u^{\pm,p}, v^p)$ of (2.9) is an extreme point in this region if and only if it is extreme in Ξ; a feasible ray $(\lambda^r, u^{\pm,r}, v^r)$ of (2.9) is extreme in this region if and only if it is extreme in Ξ.

Proof. We prove only the first assertion; that for the second is similar. The sufficiency holds because the feasible region of (2.9) is a subset of Ξ. To prove the converse, suppose that $(\lambda^p, u^{\pm,p}, v^p)$ is an extreme solution of (2.9). Then this triple must be an element of Ξ. If it lies on the line segment of two other feasible solutions of Ξ, then the latter two solutions must satisfy the additional constraint $z^T u^+ + (\mathbf{1} - z)^T v \leq 0$. Therefore, $(\lambda^p, u^{\pm,p}, v^p)$ is also extreme in Ξ. $\square$

2.2. The set $\mathcal{Z}$ and a minimax formulation. We now define the key set of binary vectors:

$$\mathcal{Z} \equiv \{\, z \in \{\,0,1\,\}^m : \varphi_0(z) = 0\,\},$$

which, by Proposition 2.2, is the feasibility descriptor of the feasible region of the LPCC (2.1). Note that $\mathcal{Z}$ is a finite set. We also define the minimax integer program:

$$\underset{z \in \mathcal{Z}}{\text{minimize}}\ \varphi(z) \equiv \left[\begin{array}{ll} \underset{(\lambda, u^\pm, v)}{\text{maximum}} & f^T\lambda + q^T(\,u^+ - u^-\,) \\ \text{subject to} & A^T\lambda - N^T(\,u^+ - u^-\,) = c \\ & B^T\lambda - M^T(\,u^+ - u^-\,) - v \leq d \\ & (\,\lambda, u^\pm, v\,) \geq 0 \\ \text{and} & z^T u^+ + (\mathbf{1} - z)^T v \leq 0. \end{array} \right] \tag{2.12}$$

Since $\mathcal{Z}$ is a finite set, and since $\varphi(z) \in \Re \cup \{-\infty\}$ for $z \in \mathcal{Z}$, it follows that $\underset{z \in \mathcal{Z}}{\text{argmin}}\,\varphi(z) \neq \emptyset$ if and only if $\mathcal{Z} \neq \emptyset$. The following result rephrases the three basic facts connecting the LPCC (2.1) and its LP pieces in terms of the IP (2.12).

THEOREM 2.4. The following three statements hold:

(a) the LPCC (2.1) is infeasible if and only if $\min\limits_{z \in \mathcal{Z}} \varphi(z) = \infty$ (i.e., $\mathcal{Z} = \emptyset$);

(b) the LPCC (2.1) is feasible and has an unbounded objective value if and only if $\min\limits_{z \in \mathcal{Z}} \varphi(z) = -\infty$ (i.e., $z \in \mathcal{Z}$ exists such that $\varphi(z) = -\infty$);

(c) the LPCC (2.1) attains a finite optimal objective value if and only if $-\infty < \min\limits_{z \in \mathcal{Z}} \varphi(z) < \infty$.

In all cases, $\text{LPCC}_{\min} = \min\limits_{z \in \mathcal{Z}} \varphi(z)$; moreover, for any $z \in \{0,1\}^m$ for which $\varphi(z) > -\infty$, $\text{LPCC}_{\min} \leq \varphi(z)$.

Proof. Statement (a) is an immediate consequence of Proposition 2.2. Statement (b) is equivalent to saying that the LPCC (2.1) is feasible and has an unbounded objective if and only if $z \in \{0,1\}^m$ exists such that $\varphi_0(z) = 0$ and $\varphi(z) = -\infty$. Suppose that the LPCC (2.1) is feasible and unbounded. Then an index set $\alpha \subseteq \{1, \cdots, m\}$ exists such that the LP(α) is feasible and unbounded. Letting $z \in \{0,1\}^m$ be such that $\text{supp}(z) = \alpha$ and $\bar{\alpha}$ be the complement of α in $\{1, \cdots, m\}$, we have $\varphi_0(z) = 0$. Moreover, the dual of the (unbounded) LP(α) is

$$\begin{array}{lll} \underset{(\lambda, u_{\bar{\alpha}}, u_{\alpha}^-)}{\text{maximize}} & f^T\lambda + (\, q_{\bar{\alpha}}\,)^T u_{\bar{\alpha}} - (\, q_{\alpha}\,)^T u_{\alpha}^- & \\ \text{subject to} & A^T\lambda - (\, N_{\bar{\alpha}\bullet}\,)^T u_{\bar{\alpha}} + (\, N_{\alpha\bullet}\,)^T u_{\alpha}^- \;=\; c & \\ & (\, B_{\bullet\bar{\alpha}}\,)^T\lambda - (\, M_{\bar{\alpha}\bar{\alpha}}\,)^T u_{\bar{\alpha}} + (\, M_{\alpha\bar{\alpha}}\,)^T u_{\alpha}^- \;\leq\; d_{\bar{\alpha}} & \\ \text{and} & (\, \lambda, u_{\alpha}^-\,) \;\geq\; 0, & \end{array} \tag{2.13}$$

which is equivalent to the problem (2.9) corresponding to the binary vector z defined here. (Note, the $\bullet$ in the subscripts is the standard notation in linear programming, denoting rows/columns of matrices.) Therefore, since (2.13) is infeasible, it follows that $\varphi(z) = -\infty$ by convention. Conversely, suppose that $z \in \{0,1\}^m$ exists such that $\varphi_0(z) = 0$ and $\varphi(z) = -\infty$. Let $\alpha \equiv \text{supp}(z)$ and $\bar{\alpha} \equiv$ complement of α in $\{1, \cdots, m\}$. It then follows that (2.11), and thus the LP(α), is feasible. Moreover, since $\varphi(z) = -\infty$, it follows that (2.13), being equivalent to (2.9), is infeasible; thus the LP(α) is unbounded. Statement (c) follows readily from (a) and (b). The equality between $\text{LPCC}_{\min}$ and $\min\limits_{z \in \mathcal{Z}} \varphi(z)$ is due to the fact that the maximizing LP defining $\varphi(z)$ is essentially the dual of the piece LP(α). To prove the last assertion of the theorem, let $z \in \{0,1\}^m$ be such that $\varphi(z) > -\infty$. Without loss of generality, we may assume that $\varphi(z) < \infty$. Thus the LP (2.9) attains a finite maximum; hence $\varphi_0(z) = 0$. Therefore $z \in \mathcal{Z}$ and the bound $\text{LPCC}_{\min} \leq \varphi(z)$ holds readily. □

3. The Benders Approach. In essence, our strategy for solving the LPCC (2.1) is to apply a Benders approach to the minimax IP (2.12). For this purpose, we let $\left\{ \left(\lambda^{p,i}, u^{\pm,p,i}, v^{p,i} \right) \right\}_{i=1}^{K}$ and $\left\{ \left(\lambda^{r,j}, u^{\pm,r,j}, v^{r,j} \right) \right\}_{j=1}^{L}$ be the finite set of extreme points and extreme rays of the polyhedron Ξ. Note that $K \geq 1$ if and only if $\Xi \neq \emptyset$. (These extreme points and rays will be generated as needed. For the discussion in this section, we take them as available.) In what follows, we derive a restatement of Theorem 2.4 in terms of these extreme points and rays.

The IP (2.12) can be written as:

$$\underset{z \in \mathcal{Z}}{\text{minimize}} \left[\begin{array}{ll} \underset{(\rho^p, \rho^r) \geq 0}{\text{maximum}} & \displaystyle\sum_{i=1}^{K} \rho_i^p \left[f^T\lambda^{p,i} + q^T(u^{+,p,i} - u^{-,p,i}) \right] + \sum_{j=1}^{L} \rho_j^r \left[f^T\lambda^{r,j} + q^T(u^{+,r,j} - u^{-,r,j}) \right] \\ \text{subject to} & \displaystyle\sum_{i=1}^{K} \rho_i^p \left[z^T u^{+,p,i} + (\mathbf{1} - z)^T v^{p,i} \right] + \sum_{j=1}^{L} \rho_j^r \left[z^T u^{+,r,j} + (\mathbf{1} - z)^T v^{r,j} \right] \;\leq\; 0 \\ \text{and} & \displaystyle\sum_{i=1}^{K} \rho_i^p \;=\; 1, \end{array} \right] \tag{3.1}$$

which is the *master IP*. It turns out that the set $\mathcal{Z}$ can be completely described in terms of certain *ray cuts*, whose definition requires the index set:

$$\mathcal{L} \;\equiv\; \left\{ j \,\in\, \{1, \cdots, L\} \;:\; f^T\lambda^{r,j} + q^T(u^{+,r,j} - u^{-,r,j}) \;>\; 0 \right\}.$$

The following proposition shows that the set $\mathcal{Z}$ can be described in terms of satisfiability inequalities using the extreme rays in $\mathcal{L}$.

PROPOSITION 3.1. $\mathcal{Z} = \left\{ z \in \{0,1\}^m \;:\; \displaystyle\sum_{\ell : u_\ell^{+,r,j} > 0} z_\ell + \sum_{\ell : v_\ell^{r,j} > 0} (\, 1 - z_\ell\,) \;\geq\; 1, \;\; \forall\, j \,\in\, \mathcal{L} \right\}.$

Proof. Since a tuple $(\lambda, u^{\pm}, v)$ is feasible to (2.10) if and only if it is a nonnegative combination of the extreme rays of (2.9), which are necessarily extreme rays of Ξ by Proposition 2.3, it follows that a tuple $(\lambda, u^{\pm}, v)$ is feasible to (2.10) if and only if there exist nonnegative coefficients $\{\rho_j^r\}_{j=1}^{L}$ such that

$$(\, \lambda, u^{\pm}, v\,) \;=\; \sum_{j=1}^{L} \rho_j^r \,(\, \lambda^{r,j}, u^{\pm,r,j}, v^{r,j}\,)$$

and $\sum_{j=1}^{L} \rho_j^r \left[z^T u^{+,r,j} + (\mathbf{1} - z)^T v^{r,j} \right] \leq 0$. Therefore, $\varphi_0(z)$ is equal to

$$\begin{array}{ll} \underset{\rho^r \geq 0}{\text{maximize}} & \sum_{j=1}^{L} \rho_j^r \left[f^T \lambda^{r,j} + q^T (u^{+,r,j} - u^{-,r,j}) \right] \\ \text{subject to} & \sum_{j=1}^{L} \rho_j^r \left[z^T u^{+,r,j} + (\mathbf{1} - z)^T v^{r,j} \right] \leq 0 \end{array}$$

and the latter maximization problem has a finite optimal solution if and only if

$$\begin{aligned} f^T \lambda^{r,j} + q^T (u^{+,r,j} - u^{-,r,j}) > 0 \quad &\Longrightarrow \quad z^T u^{+,r,j} + (\mathbf{1} - z)^T v^{r,j} > 0 \\ &\Longleftrightarrow \quad \sum_{\ell : u_\ell^{+,r,j} > 0} z_\ell + \sum_{\ell : v_\ell^{r,j} > 0} (1 - z_\ell) \geq 1. \end{aligned}$$

Therefore, the equality between $\mathcal{Z}$ and the right-hand set is immediate. $\square$

An immediate corollary of Proposition 3.1 is that it provides a certificate of infeasibility for the LPCC.

Corollary 3.2. If $\mathcal{R} \subseteq \mathcal{L}$ exists such that

$$\left\{ z \in \{0,1\}^m : \sum_{\ell : u_\ell^{+,r,j} > 0} z_\ell + \sum_{\ell : v_\ell^{r,j} > 0} (1 - z_\ell) \geq 1, \ \forall \, j \in \mathcal{R} \right\} = \emptyset,$$

then the LPCC (2.1) is infeasible.

Proof. The assumption implies that $\mathcal{Z} = \emptyset$. Thus the infeasibility of the LPCC follows from Theorem 2.4(a). $\square$

In view of Proposition 3.1, (3.1) is equivalent to:

$$\underset{z \in \mathcal{Z}}{\text{minimize}} \left[\begin{array}{ll} \underset{\rho^p \geq 0}{\text{maximum}} & \sum_{i=1}^{K} \rho_i^p \left[f^T \lambda^{p,i} + q^T (u^{+,p,i} - u^{-,p,i}) \right] \\ \text{subject to} & \sum_{i=1}^{K} \rho_i^p \left[z^T u^{+,p,i} + (\mathbf{1} - z)^T v^{p,i} \right] \leq 0 \\ \text{and} & \sum_{i=1}^{K} \rho_i^p = 1, \end{array} \right]. \tag{3.2}$$

Note that the $\text{LPCC}_{\min}$ is equal to the minimum objective value of (3.2). Similar to the inequality:

$$\sum_{\ell : u_\ell^{+,r,j} > 0} z_\ell + \sum_{\ell : v_\ell^{r,j} > 0} (1 - z_\ell) \geq 1,$$

which we call a *ray cut* (because it is induced by an extreme ray), we will make use of a *point cut*:

$$\sum_{\ell : u_\ell^{+,p,i} > 0} z_\ell + \sum_{\ell : v_\ell^{p,i} > 0} (1 - z_\ell) \geq 1,$$

that is induced by an extreme point $(\lambda^{p,i}, u^{\pm,p,i}, v^{p,i})$ of Ξ chosen from the following collection:

$$\mathcal{K} \equiv \left\{ i \in \{ 1, \cdots, K \} : f^T \lambda^{p,i} + q^T (u^{+,p,i} - u^{-,p,i}) = \varphi(z) \text{ for some } z \in \mathcal{Z} \right\}.$$

Note that $\mathcal{K} \neq \emptyset \Rightarrow \mathcal{Z} \neq \emptyset$, which in turn implies that the LPCC (2.1) is feasible. Moreover,

$$\min_{i \in \mathcal{K}} \left[f^T \lambda^{p,i} + q^T (u^{+,p,i} - u^{-,p,i}) \right] \geq \text{LPCC}_{\min}.$$

For a given pair of subsets $\mathcal{P} \times \mathcal{R} \subseteq \mathcal{K} \times \mathcal{L}$, let

$$\mathcal{Z}(\mathcal{P},\mathcal{R}) \equiv \left\{ \begin{array}{ll} z \in \{0,1\}^m : & \displaystyle\sum_{\ell : u_\ell^{+,r,j} > 0} z_\ell + \sum_{\ell : v_\ell^{r,j} > 0} (1 - z_\ell) \geq 1, \ \forall j \in \mathcal{R} \\ & \displaystyle\sum_{\ell : u_\ell^{+,p,i} > 0} z_\ell + \sum_{\ell : v_\ell^{p,i} > 0} (1 - z_\ell) \geq 1, \ \forall i \in \mathcal{P} \end{array} \right\}.$$

We have the following result.

Proposition 3.3. If there exists $\mathcal{P} \times \mathcal{R} \subseteq \mathcal{K} \times \mathcal{L}$ such that

$$\min_{i \in \mathcal{P}} \left[f^T \lambda^{p,i} + q^T (u^{+,p,i} - u^{-,p,i}) \right] > \text{LPCC}_{\min},$$

then $\operatorname*{argmin}_{z \in \mathcal{Z}} \varphi(z) \subseteq \mathcal{Z}(\mathcal{P},\mathcal{R})$.

Proof. Let $\widetilde{z} \in \mathcal{Z}$ be a minimizer of $\varphi(z)$ on $\mathcal{Z}$. (The proposition is clearly valid if no such minimizer exists.) If $\widetilde{z} \notin \mathcal{Z}(\mathcal{P},\mathcal{R})$, then there exists $i \in \mathcal{P}$ such that

$$\sum_{\ell : u_\ell^{+,p,i} > 0} \widetilde{z}_\ell + \sum_{\ell : v_\ell^{p,i} > 0} (1 - \widetilde{z}_\ell) = 0.$$

Hence, $(\lambda^{p,i}, u^{\pm,p,i}, v^{p,i})$ is feasible to the LP (2.9) corresponding to $\varphi(\widetilde{z})$; thus

$$\text{LPCC}_{\min} = \varphi(\widetilde{z}) \geq f^T \lambda^{p,i} + q^T (u^{+,p,i} - u^{-,p,i}) > \text{LPCC}_{\min},$$

which is a contradiction. □

Analogous to Corollary 3.2, we have the following corollary of Proposition 3.3.

Corollary 3.4. If there exists $\mathcal{P} \times \mathcal{R} \subseteq \mathcal{K} \times \mathcal{L}$ with $\mathcal{P} \neq \emptyset$ such that $\mathcal{Z}(\mathcal{P},\mathcal{R}) = \emptyset$, then

$$\text{LPCC}_{\min} = \min_{i \in \mathcal{P}} \left[f^T \lambda^{p,i} + q^T (u^{+,p,i} - u^{-,p,i}) \right] \in (-\infty, \infty). \tag{3.3}$$

Proof. Indeed, if the claimed equality does not hold, then $\operatorname*{argmin}_{z \in \mathcal{Z}} \varphi(z) = \emptyset$. But this implies $\mathcal{Z} = \emptyset$, which contradicts the assumption that $\mathcal{P} \neq \emptyset$. □

Combining Corollaries 3.2 and 3.4, we obtain the desired restatement of Theorem 2.4 in terms of the extreme points and rays of Ξ.

Theorem 3.5. The following three statements hold:

(a) the LPCC (2.1) is infeasible if and only if a subset $\mathcal{R} \subseteq \mathcal{L}$ exists such that $\mathcal{Z}(\emptyset,\mathcal{R}) = \emptyset$;

(b) the LPCC (2.1) is feasible and has an unbounded objective if and only if $\mathcal{Z}(\mathcal{K},\mathcal{L}) \neq \emptyset$;

(c) the LPCC (2.1) attains a finite optimal objective value if and only if a pair $\mathcal{P} \times \mathcal{R} \subseteq \mathcal{K} \times \mathcal{L}$ exists with $\mathcal{P} \neq \emptyset$ such that $\mathcal{Z}(\mathcal{P},\mathcal{R}) = \emptyset$.

Proof. Statement (a) follows from Corollary 3.2 by noting that a subset $\mathcal{R} \subseteq \mathcal{L}$ exists such that $\mathcal{Z}(\emptyset,\mathcal{R}) = \emptyset$ if and only if $\mathcal{Z} = \mathcal{Z}(\emptyset,\mathcal{L}) = \emptyset$. To prove (b), suppose first $\mathcal{Z}(\mathcal{K},\mathcal{L}) \neq \emptyset$. Let $\widehat{z} \in \mathcal{Z}(\mathcal{K},\mathcal{L})$. Then $\widehat{z} \in \mathcal{Z}$. We claim that $\varphi(\widehat{z}) = -\infty$; i.e., the LP (2.9) corresponding to $\widehat{z}$ is infeasible. Assume otherwise, then since $\varphi_0(\widehat{z}) = 0$, it follows that $\varphi(\widehat{z})$ is finite. Hence there exists an extreme point $(\lambda^{p,i}, u^{\pm,p,i}, v^{p,i})$ of the LP (2.9) corresponding to $\widehat{z}$ such that $f^T \lambda^{p,i} + q^T (u^{+,p,i} - u^{-,p,i}) = \varphi(\widehat{z})$; thus the index $i \in \mathcal{K}$, which implies

$$\sum_{\ell : u_\ell^{+,p,i} > 0} \widehat{z}_\ell + \sum_{\ell : v_\ell^{p,i} > 0} (1 - \widehat{z}_\ell) \geq 1,$$

because $\widehat{z} \in \mathcal{Z}(\mathcal{K},\mathcal{L})$. But this contradicts the feasibility of $(\lambda^{p,i}, u^{\pm,p,i}, v^{p,i})$ to the LP (2.9) corresponding to $\widehat{z}$. Therefore, the LPCC (2.1) is feasible and has an unbounded objective value; thus, the "if" statement in (b) holds. Conversely, suppose $\text{LPCC}_{\min} = -\infty$. By Theorem 2.4, it follows that $\widehat{z} \in \mathcal{Z}$ exists such that $\varphi(\widehat{z}) = -\infty$; i.e., the LP (2.9) corresponding to $\widehat{z}$ is infeasible. In turn, this means that

$$\widehat{z}^T u^{+,p,i} + (\mathbf{1} - \widehat{z})^T v^{p,i} > 0$$

for all $i = 1, \cdots, K$; or equivalently,

$$\sum_{\ell : u_\ell^{+,p,i} > 0} \widehat{z}_\ell + \sum_{\ell : v_\ell^{p,i} > 0} (1 - \widehat{z}_\ell) \geq 1,$$

for all $i = 1, \cdots, K$. Consequently, $\widehat{z} \in \mathcal{Z}(\mathcal{K}, \mathcal{L})$. Hence, statement (b) holds. Finally, the "if" statement in (c) follows from Corollary 3.4. Conversely, if the LPCC (2.1) has a finite optimal solution, then by (b), it follows that $\mathcal{Z}(\mathcal{K}, \mathcal{L}) = \emptyset$. Since the LPCC (2.1) is feasible, $\mathcal{K} \neq \emptyset$ by (a), establishing the "only if" statement in (c). □

Theorem 3.5 constitutes the theoretical basis for the algorithm to be presented in Section 5 for resolving the LPCC. Through the successive generation of extreme points and rays of Ξ, the algorithm searches for a pair of subsets $\mathcal{P} \times \mathcal{R}$ such that $\mathcal{Z}(\mathcal{P}, \mathcal{R}) = \emptyset$. If such a pair can be successfully identified, then the LPCC is either infeasible ($\mathcal{P} = \emptyset$) or attains a finite optimal solution ($\mathcal{P} \neq \emptyset$). If no such pair is found, then the LPCC is unbounded. In the algorithm, the last case is identified with a binary vector $z \in \mathcal{Z}$ with $\varphi(z) = -\infty$, i.e., the LP (2.9) is infeasible. Based on the value function $\varphi(z)$ and the point/ray cuts, the algorithm will be shown to terminate in finite time.

4. Simple Cuts and Sparsification. In this section, we explain several key steps in the main algorithm to be presented in the next section. The first idea is a version of the well-known Gomory cut in integer programming specialized to the LPCC and which has previously been employed for bilevel LPs; see [5]; the second idea aims at "sparsifying" the ray/point cuts to facilitate the computation of elements of the working sets $\mathcal{Z}(\mathcal{P}, \mathcal{R})$. Specifically, a satisfiability constraint:

$$\sum_{i \in \mathcal{I}'} z_i + \sum_{j \in \mathcal{J}'} (1 - z_j) \geq 1 \quad \text{is } \textit{sparser} \text{ than} \quad \sum_{i \in \mathcal{I}} z_i + \sum_{j \in \mathcal{J}} (1 - z_j) \geq 1$$

if $\mathcal{I}' \subseteq \mathcal{I}$ and $\mathcal{J}' \subseteq \mathcal{J}$. In general, a satisfiability inequality cuts off certain LP pieces of the LPCC; the sparser the inequality is the more pieces it cuts off. Thus, it is desirable to sparsify a cut as much as possible. Nevertheless, sparsification requires the solution of linear subprograms; thus one needs to balance the work required with the benefit of the process.

4.1. Simple cuts. The following discussion is a minor variant of that presented in [5] for bilevel LPs. Consider the LP relaxation of the LPCC (2.1):

$$\begin{array}{ll} \underset{(x,y,w)}{\text{minimize}} & c^T x + d^T y \\ \text{subject to} & Ax + By \geq f \\ \text{and} & 0 \leq y, \quad w \equiv q + Nx + My \geq 0, \end{array} \tag{4.1}$$

where the orthogonal condition $y^T w = 0$ is dropped. Assume that by solving this LP, an optimal solution is obtained that fails the latter orthogonality condition, say $y_i w_i > 0$ in this solution. Thus, y_i and w_i must be basic variables in a basic optimal solution of the LP; in such a solution, w_i and y_i can be expressed in terms of the nonbasic variables, which we denote by the generic variables s_j, as follows: for some constants a_j and b_j,

$$w_i = w_{i0} - \sum_{s_j : \text{nonbasic}} a_j \, s_j \quad \text{and} \quad y_i = y_{i0} - \sum_{s_j : \text{nonbasic}} b_j \, s_j$$

where w_{i0} and y_{i0} are the current values of the variables w_i and y_i, respectively, with $\min(w_{i0}, y_{i0}) > 0$. It is not difficult to show that the following inequality must be satisfied by all feasible solutions of the LPCC (2.1)

$$\sum_{\substack{s_j \,:\, \text{nonbasic} \\ \max(a_j, b_j) > 0}} \max\left(\frac{a_j}{w_{i0}}, \frac{b_j}{y_{i0}} \right) s_j \geq 1 \tag{4.2}$$

Note that if $a_j \leq 0$ for all nonbasic j, then $w_i > 0 = y_i$ for every feasible solution of the LPCC (2.1). A similar remark can be made if $b_j \leq 0$ for all nonbasic j.

Following the terminology in [5], we call the inequality (4.2) a *simple cut*. Multiple such cuts can be added to the constraint $Ax + By \geq f$, resulting in a modified inequality $\widetilde{A}x + \widetilde{B}y \geq \widetilde{f}$. We can generate and add even more simple cuts by repeating the above step. This strategy turns out to be a very effective pre-processor for the algorithm to be described in the next section. At the end of this pre-processor, we obtain an optimal solution $(\bar{x}, \bar{y}, \bar{w})$ of (4.1) that remains infeasible to the LPCC (otherwise, this solution would be optimal for the LPCC); the optimal objective value $c^T \bar{x} + d^T \bar{y}$ provides a valid lower bound for $\text{LPCC}_{\min}$. (Note: if (4.1) is unbounded, then the pre-processor does not produce any cuts or a finite lower bound.)

LPCC feasibility recovery. Occurring in many applications of the LPCC, the special case $B = 0$ deserves a bit more discussion. First note that in this case, the modified matrix $\widetilde{B}$ is not necessarily zero. Nevertheless, the solution $(\bar{x}, \bar{y}, \bar{w})$ obtained from the simple-cut pre-processor can be used to produce a feasible solution to the LPCC (2.1) by simply solving the linear complementarity problem (LCP): $0 \leq y \perp q + N\bar{x} + My \geq 0$ (assuming that the matrix M has favorable properties so that this step is effective). Letting $\bar{y}'$ be a solution to the latter LCP, the objective value $c^T \bar{x} + d^T \bar{y}'$ yields a valid upper bound to $\text{LPCC}_{\min}$. This recovery procedure of an LPCC feasible solution can be extended to the case where $B \neq 0$. (Incidentally, this class of LPCCs is generally "more difficult" than the class where $B = 0$, where the difficulty is determined by our empirical experience from the computational tests.) Indeed, from any feasible solution $(\bar{x}, \bar{y}, \bar{w})$ to the LP relaxation of the LPCC (2.1) but not to the LPCC itself, we could attempt to recover a feasible solution to the LPCC along with an element in $\mathcal{Z}$ by either solving the LP(α), where $\alpha \equiv \{i : \bar{y}_i \leq \bar{w}_i\}$, or by solving $\varphi(z)$, where $z_\alpha = 1$ and $z_{\bar{\alpha}} = 0$. A feasible solution to this LP piece yields a feasible solution to the LPCC and a finite upper bound. In general, there is no guarantee that this procedure will always be successful; nevertheless, it is very effective when it works.

4.2. Cut management. A key step in our algorithm involves the selection of elements in the sets $\mathcal{Z}(\mathcal{P}, \mathcal{R})$ for various index pairs $(\mathcal{P}, \mathcal{R})$. Generally speaking, this involves solving integer subprograms. Recognizing that the constraints in each $\mathcal{Z}(\mathcal{P}, \mathcal{R})$ are of the satisfiability type, we could in principle employ special algorithms for implementing this step (see [7, 31] and the references therein for some such algorithms). To facilitate such selection, we have developed a special heuristic that utilizes a valid upper bound of $\text{LPCC}_{\min}$ to sparsify the terms in the ray/point cuts in a working set. In what follows, we describe how the algorithm manages these cuts.

There are three pools of cuts, labeled $\mathcal{Z}_{\text{work}}$–the working pool, $\mathcal{Z}_{\text{wait}}$–the wait pool, and $\mathcal{Z}_{\text{cand}}$–the candidate pool. Inequalities in $\mathcal{Z}_{\text{work}}$ are valid sparsifications of those in $\mathcal{Z}(\mathcal{P}, \mathcal{R})$ corresponding to a current pair $(\mathcal{P}, \mathcal{R})$. Thus, the set of binary vectors satisfying the inequalities in $\mathcal{Z}_{\text{work}}$, which we denote $\widehat{\mathcal{Z}}_{\text{work}}$, is a subset of $\mathcal{Z}(\mathcal{P}, \mathcal{R})$. Inequalities in $\mathcal{Z}_{\text{cand}}$ are candidates for sparsification; the sparsification procedure described below always ends with this set empty. The decision of whether or not to sparsify a valid inequality is made according to a current LPCC upper bound and a small scalar $\delta > 0$. In essence, the sparsification is an effective way to facilitate the search for a feasible element in $\widehat{\mathcal{Z}}_{\text{work}}$. At one extreme, a sparsest inequality with only one term in it automatically fixes one complementarity; e.g., $z_1 \geq 1$ fixes $w_1 = 0$; at another extreme, it is computationally more difficult to find feasible points satisfying many dense inequalities.

We sparsify an inequality

$$\sum_{i \in \mathcal{I}} z_i + \sum_{j \in \mathcal{J}} (1 - z_j) \geq 1 \tag{4.3}$$

in the following way. Let $\mathcal{I} = \mathcal{I}_1 \cup \mathcal{I}_2$ be a partition of $\mathcal{I}$ into two disjoint subsets $\mathcal{I}_1$ and $\mathcal{I}_2$; similarly, let $\mathcal{J} = \mathcal{J}_1 \cup \mathcal{J}_2$. We split (4.3), which we call the *parent*, into two sub-inequalities:

$$\sum_{i \in \mathcal{I}_1} z_i + \sum_{j \in \mathcal{J}_1} (1 - z_j) \geq 1 \quad \text{and} \quad \sum_{i \in \mathcal{I}_2} z_i + \sum_{j \in \mathcal{J}_2} (1 - z_j) \geq 1; \tag{4.4}$$

and test both to see if they are valid for the LPCC. To test the left-hand inequality, we consider the LP relaxation (4.1) of the LPCC (2.1) with the additional constraints $w_i = (q + Nx + My)_i = 0$ for $i \in \mathcal{I}_1$ and $y_i = 0$ for $i \in \mathcal{J}_1$, which we call a *relaxed LP with restriction*. If this LP has an objective value greater than the current LPCC_{ub}, then we have successfully sparsified the inequality (4.3) into the sparser inequality:

$$\sum_{i \in \mathcal{I}_1} z_i + \sum_{j \in \mathcal{J}_1} (1 - z_j) \geq 1, \tag{4.5}$$

which must be valid for the LPCC. (In this situation, any dual solution to the relaxed LP with restriction is feasible in the dual LP (2.9) for any binary vector z that violates (4.5). Hence, the value $\varphi(z)$ of the LPCC on this piece must be at least LPCC_{ub}, implying that such a piece cannot contain an optimal solution of the LPCC.) Otherwise, using the feasible solution to the relaxed LP, we employ the LPCC feasibility recovery procedure to compute an LPCC feasible solution along with a binary $z \in \mathcal{Z}$. If successful, one of two cases happen: if $\varphi(z) \geq \text{LPCC}_{\text{ub}}$, then a new cut can be generated; otherwise, we have reduced the LPCC upper bound. Either case, we obtain positive progress in the algorithm. If no LPCC feasible solution is recovered, then we save the cut (4.5) in the wait pool $\mathcal{Z}_{\text{wait}}$ for later consideration. In essence, cuts in the wait pool are not yet proven to be valid for the LPCC; they will be revisited when there is a reduction in LPCC_{ub}. Note that every inequality in $\mathcal{Z}_{\text{wait}}$ has an LP optimal objective value associated with it that is less than the current LPCC upper bound.

In our experiment, we randomly divide the sets $\mathcal{I}$ and $\mathcal{J}$ roughly into two equal halves each and adopt a strategy that attempts to sparsify the *root inequality* (4.3) as much as possible via a random branching rule. The following illustrates one such division:

$$\begin{array}{ccc} & z_1 + z_3 + z_4 + (1 - z_2) + (1 - z_6) \geq 1 & \\ \swarrow & & \searrow \\ z_1 + z_3 + (1 - z_2) \geq 1 & & z_4 + (1 - z_6) \geq 1. \end{array}$$

We use a small scalar $\delta > 0$ to help decide on the subsequent branching. In essence, we only branch if the inequality appears strong. Solving LPs, the procedure below sparsifies a given valid inequality for the LPCC, called the *root* of the procedure.

Sparsification procedure. Let (4.3) be the root inequality to be sparsified, LPCC_{ub} be the current LPCC upper bound, and $\delta > 0$ be a given scalar. Branch (4.3) into two sub-inequalities (4.4), both of which we put in the set $\mathcal{Z}_{\text{cand}}$.

Main step. If $\mathcal{Z}_{\text{cand}}$ is empty, terminate. Otherwise pick a *candidate* inequality in $\mathcal{Z}_{\text{cand}}$, say the left one in (4.4) with the corresponding pair of index sets $(\mathcal{I}_1, \mathcal{J}_1)$. Solve the LP relaxation (4.1) of the LPCC (2.1) with the additional constraints $w_i = (q + Nx + My)_i = 0$ for $i \in \mathcal{I}_1$ and $y_i = 0$ for $i \in \mathcal{J}_1$, obtaining an LP optimal objective value, say $\text{LP}_{\text{rlx}} \in \Re \cup \{\pm\infty\}$. We have the following three cases.

• If $\text{LP}_{\text{rlx}} \in [\,\text{LPCC}_{\text{ub}}, \text{LPCC}_{\text{ub}} + \delta\,]$, move the candidate inequality from $\mathcal{Z}_{\text{cand}}$ into $\mathcal{Z}_{\text{work}}$ and remove its parent; return to the main step.

• If $\text{LP}_{\text{rlx}} < \text{LPCC}_{\text{ub}}$, apply the LPCC feasibility recovery procedure to the feasible solution at termination of the current relaxed LP with restriction. If the procedure is successful, return to the main step with either a new cut or a reduced LPCC_{ub}. Otherwise, move the incumbent candidate inequality from $\mathcal{Z}_{\text{cand}}$ into $\mathcal{Z}_{\text{wait}}$; return to the main step.

• If $\delta + \text{LPCC}_{\text{ub}} < \text{LP}_{\text{rlx}}$, move the candidate inequality from $\mathcal{Z}_{\text{cand}}$ into $\mathcal{Z}_{\text{work}}$ and remove its parent; further branch the candidate inequality into two sub-inequalities, both of which we put into the candidate pool $\mathcal{Z}_{\text{cand}}$; return to the main step.

During the procedure, the set $\mathcal{Z}_{\text{cand}}$ may grow from the initial size of 2 inequalities when the root of the procedure is first split. Nevertheless, by solving finitely many LPs, this set will eventually shrink to empty; when that happens, either we have successfully sparsified the root inequality and placed multiple sparser cuts into $\mathcal{Z}_{\text{work}}$, or some sparser cuts are added to the pool $\mathcal{Z}_{\text{wait}}$, waiting to be proven valid for the LPCC in subsequent iterations. Note that associated with each inequality in $\mathcal{Z}_{\text{wait}}$ is the value LP_{rlx}.

5. The IP Algorithm. We are now ready to present the parameter-free IP-based algorithm for resolving an arbitrary LPCC (2.1). Subsequently, we will establish that the algorithm will successfully terminate in a finite number of *iterations* with a definitive resolution of the LPCC in one of its three states. Referring to a return to Step 1, each iteration consists of solving one feasibility IP of the satisfiability kind, a couple LPs to compute $\varphi(\widehat{z})$ and possibly $\varphi_0(\widehat{z})$ corresponding to a binary vector $\widehat{z}$ obtained from the IP, and multiple LPs within the sparsification procedure associated with an induced point/ray cut.

The algorithm

Step 0. (Preprocessing and initialization) Generate multiple simple cuts to tighten the complementarity constraints. If any of the LPs encountered in this step is infeasible, then so is the LPCC (2.1). In general, let LPCC_{lb} ($-\infty$ allowed) and LPCC_{ub} (∞ allowed) be valid lower and upper bounds of $\text{LPCC}_{\min}$, respectively. Let $\delta > 0$ be a small scalar. [A finite optimal solution to a relaxed LP provides a finite lower bound, and a feasible solution to the LPCC, which could be obtained by the LPCC feasibility recovery procedure, provides a finite upper bound.] Set $\mathcal{P} = \mathcal{R} = \emptyset$ and $\mathcal{Z}_{\text{work}} = \mathcal{Z}_{\text{wait}} = \emptyset$. (Thus, $\widehat{\mathcal{Z}}_{\text{work}} = \{0,1\}^m$.)

Step 1. (Solving a satisfiability IP) Determine a vector $\widehat{z} \in \widehat{\mathcal{Z}}_{\text{work}}$. If this set is empty, go to Step 2. Otherwise go to Step 3.

Step 2. (Termination: infeasibility or finite solvability) If $\mathcal{P} = \emptyset$, we have obtained a certificate of infeasibility for the LPCC (2.1); stop. If $\mathcal{P} \neq \emptyset$, we have obtained a certificate of global optimality for the LPCC (2.1) with $\text{LPCC}_{\min}$ given by (3.3); stop.

Step 3. (Solving dual LP) Compute $\varphi(\widehat{z})$ by solving the LP (2.9). If $\varphi(\widehat{z}) \in (-\infty, \infty)$, go to Step 4a. If $\varphi(\widehat{z}) = \infty$, proceed to Step 4b. If $\varphi(\widehat{z}) = -\infty$, proceed to Step 4c.

Step 4a. (Adding an extreme point) Let $(\lambda^{p,i}, u^{\pm,p,i}, v^{p,i}) \in \mathcal{K}$ be an optimal extreme point of Ξ. There are 3 cases.

• If $\varphi(\widehat{z}) \in [\ \text{LPCC}_{\text{ub}},\ \text{LPCC}_{\text{ub}} + \delta]$, let $\mathcal{P} \leftarrow \mathcal{P} \cup \{i\}$ and add the corresponding point cut to $\mathcal{Z}_{\text{work}}$; return to Step 1.

• If $\varphi(\widehat{z}) > \text{LPCC}_{\text{ub}} + \delta$, let $\mathcal{P} \leftarrow \mathcal{P} \cup \{i\}$ and add the corresponding point cut to $\mathcal{Z}_{\text{work}}$. Apply the sparsification procedure to the new point cut, obtaining an updated $\mathcal{Z}_{\text{work}}$ and $\mathcal{Z}_{\text{wait}}$, and possibly a reduced LPCC_{ub}. If the LPCC upper bound is reduced during the sparsification procedure, go to Step 5 to activate some of the cuts in the wait pool; otherwise, return to Step 1.

• If $\varphi(\widehat{z}) < \text{LPCC}_{\text{ub}}$, let $\text{LPCC}_{\text{ub}} \leftarrow \varphi(\widehat{z})$ and go to Step 5.

Step 4b. (Adding an extreme ray) Let $(\lambda^{r,j}, u^{\pm,r,j}, v^{r,j}) \in \mathcal{L}$ be an extreme ray of Ξ. Set $\mathcal{R} \longleftarrow \mathcal{R} \cup \{j\}$ and add the corresponding ray cut to $\mathcal{Z}_{\text{work}}$. Apply the sparsification procedure to the new ray cut, obtaining an updated $\mathcal{Z}_{\text{work}}$ and $\mathcal{Z}_{\text{wait}}$, and possibly a reduced LPCC_{ub}. If the LPCC upper bound is reduced during the sparsification procedure, go to Step 5 to activate some of the cuts in the wait pool; otherwise, return to Step 1.

Step 4c. (Determining LPCC unboundedness) Solve the LP (2.10) to determine $\varphi_0(\widehat{z})$. If $\varphi_0(\widehat{z}) = 0$, then the vector $\widehat{z}$ and its support provide a certificate of unboundedness for the LPCC (2.1). Stop. If $\varphi_0(\widehat{z}) = \infty$, go to Step 4b.

Step 5. (LPCC_{ub} is reduced) Move all inequalities in $\mathcal{Z}_{\text{wait}}$ with values LP_{rlx} greater than (the just reduced) LPCC_{ub} into $\mathcal{Z}_{\text{work}}$. Apply the sparsification procedure to each newly moved inequality with $\text{LP}_{\text{rlx}} >\ \text{LPCC}_{\text{ub}} + \delta$. Re-apply this step to the cuts in $\mathcal{Z}_{\text{wait}}$ each time the LPCC upper bound is reduced from the sparsification procedure. Return to Step 1 when no more cuts in $\mathcal{Z}_{\text{wait}}$ are eligible for sparsification.

We have the following finiteness result.

THEOREM 5.1. The algorithm terminates in a finite number of iterations.

Proof. The finiteness is due to several observations: (a) the set of m-dimensional binary vectors is finite, (b) each iteration of the algorithm generates a new binary vector that is distinct from all those previously generated, and (c) there are only finitely many cuts, sparsified or not. In turn, (a) and (c) are obvious; and (b) follows from the operation of the algorithm: whenever $\varphi(\widehat{z}) \geq\ \text{LPCC}_{\text{ub}}$, the new point cut or ray cut will cut off all binary vectors generated so far, including $\widehat{z}$; if $\varphi(\widehat{z}) <\ \text{LPCC}_{\text{ub}}$, then $\widehat{z}$ cannot be one of previously generated binary vectors because its φ-value is smaller than those of the other vectors. $\square$

5.1. A numerical example. We use the following simple example to illustrate the algorithm:

$$
\begin{array}{ll}
\underset{(x,y)}{\text{minimize}} & x_1 + 2\,y_1 - y_3 \\
\text{subject to} & x_1 + x_2 \geq 5 \\
& x_1,\, x_2 \geq 0 \\
& \;0 \leq y_1 \;\perp\; x_1 - y_3 + 1 \qquad\quad \geq 0 \\
& \;0 \leq y_2 \;\perp\; x_2 + y_1 + y_2 \qquad\;\; \geq 0 \\
& \;0 \leq y_3 \;\perp\; x_1 + x_2 - y_2 + 2 \;\geq 0.
\end{array}
\tag{5.1}
$$

Note that the LCP in the variable y is not derived from a convex quadratic program; in fact the matrix

$$
M \equiv \begin{bmatrix} 0 & 0 & -1 \\ 1 & 1 & 0 \\ 0 & -1 & 0 \end{bmatrix}
$$

has all principal minors nonnegative but the LCPs defined by this matrix may have zero or unbounded solutions.

Initialization: Set the upper bound as infinity: $\text{LPCC}_{\text{ub}} = \infty$. Set the working set $\mathcal{Z}_{\text{work}}$ and the waiting set $\mathcal{Z}_{\text{wait}}$ both equal to empty.

Iteration 1: Since $\widehat{\mathcal{Z}}_{\text{work}} = \{0,1\}^3$, we can pick an arbitrary binary vector z. We choose $z = (0,0,0)$ and solve the dual LP (2.9):

$$
\begin{array}{lrcl}
\underset{(\lambda,u^{\pm},v)}{\text{maximize}} & \multicolumn{3}{l}{5\,\lambda + u_1^+ + 2\,u_3^+ - u_1^- - 2\,u_3^-} \\
\text{subject to} & \lambda - u_1^+ + u_1^- - u_3^+ + u_3^- & \leq & 1 \\
& \lambda - u_2^+ + u_2^- - u_3^+ + u_3^- & \leq & 0 \\
& -u_2^+ + u_2^- - v_1 & \leq & 2 \\
& -u_2^+ + u_2^- + u_3^+ - u_3^- - v_2 & \leq & 0 \\
& u_1^+ - u_1^- - v_3 & \leq & -1 \\
& v_1 + v_2 + v_3 & \leq & 0 \\
& (\,\lambda, u^{\pm}, v\,) & \geq & 0,
\end{array}
\tag{5.2}
$$

which is unbounded, yielding an extreme ray with $u^+ = (0, 10/7, 10/7)$ and $v = (0,0,0)$ and a corresponding ray cut: $z_2 + z_3 \geq 1$. (Briefly, this cut is valid since $z_2 = z_3 = 0$ implies both $x_2 + y_1 + y_2 = 0$ and $x_1 + x_2 - y_2 + 2 = 0$, which can't both hold for nonnegative x and y.) Add this cut to $\mathcal{Z}_{\text{work}}$ and initiate the sparsification procedure. This inequality $z_2 + z_3 \geq 1$ can be branched into: $z_2 \geq 1$ or $z_3 \geq 1$. To test if $z_2 \geq 1$ is a valid cut, we form the following relaxed LP of (5.1) by restricting $x_2 + y_1 + y_2 = 0$:

$$
\begin{array}{lrcl}
\underset{(x,y)}{\text{minimize}} & \multicolumn{3}{l}{x_1 + 2\,y_1 - y_3} \\
\text{subject to} & x_1 + x_2 & \geq & 5 \\
& x_1 - y_3 + 1 & \geq & 0 \\
& x_2 + y_1 + y_2 & = & 0 \\
& x_1 + x_2 - y_2 + 2 & \geq & 0 \\
& x,\, y \geq 0. & &
\end{array}
\tag{5.3}
$$

An optimal solution of the LP (5.3) is $(x_1, x_2, y_1, y_2, y_3) = (5,0,0,0,6)$ with the optimal objective value $\text{LP}_{\text{rlx}} = -1$. This is not a feasible solution of the LPCC (5.1) because the third complementarity is violated. The inequality $z_2 \geq 1$ is therefore placed in the waiting set $\mathcal{Z}_{\text{wait}}$. We then use $(x_1, x_2) = (5,0)$ to recover an LPCC feasible solution by solving the LCP in the variable y. This yields $y = (0,0,0)$ and $w = (6,0,7)$, and hence a corresponding vector $z = (1,0,1)$. Using this z in (2.9), we get another dual problem:

$$\begin{array}{lrcl}
\underset{(\lambda,u^{\pm},v)}{\text{maximize}} & 5\,\lambda + u_1^+ + 2\,u_3^+ - u_1^- - 2\,u_3^- & & \\
\text{subject to} & \lambda - u_1^+ + u_1^- - u_3^+ + u_3^- & \leq & 1 \\
& \lambda - u_2^+ + u_2^- - u_3^+ + u_3^- & \leq & 0 \\
& -u_2^+ + u_2^- - v_1 & \leq & 2 \\
& -u_2^+ + u_2^- + u_3^+ - u_3^- - v_2 & \leq & 0 \\
& u_1^+ - u_1^- - v_3 & \leq & -1 \\
& u_1^+ + v_2 + u_3^+ & \leq & 0 \\
& (\,\lambda, u^{\pm}, v\,) \geq 0, & &
\end{array} \tag{5.4}$$

which has an optimal value 5 that is smaller than the current upper bound $\mathrm{LPCC_{ub}}$. So we update the upper bound as $\mathrm{LPCC_{ub}} = 5$. Note that this update occurs during the sparsification step. A corresponding optimal solution to (5.4) is $u^+ = (0,1,0)$ and $v = (0,0,1)$. Hence we can add the point cut: $z_2 + (1 - z_3) \geq 1$ to $\mathcal{Z}_{\mathrm{work}}$.

When we next proceed to the other branch: $z_3 \geq 1$, we have a relaxed LP:

$$\begin{array}{lrcl}
\underset{(x,y)}{\text{minimize}} & x_1 + 2\,y_1 - y_3 & & \\
\text{subject to} & x_1 + x_2 & \geq & 5 \\
& x_1 - y_3 + 1 & \geq & 0 \\
& x_2 + y_1 + y_2 & \geq & 0 \\
& x_1 + x_2 - y_2 + 2 & = & 0 \\
& x,\, y \geq 0 & &
\end{array} \tag{5.5}$$

Solving (5.5) gives an optimal value $\mathrm{LP_{rlx}} = -1$, which is smaller than $\mathrm{LPCC_{ub}}$, and a violated complementarity with $w_2 = 12$ and $y_2 = 7$. Adding $z_3 \geq 1$ to $\mathcal{Z}_{\mathrm{wait}}$, we apply the LPCC feasibility recovering procedure to $x = (0,5)$, and get a new LPCC feasible piece with $z = (1,1,1)$. Substituting z into (2.9), we get another LP:

$$\begin{array}{lrcl}
\underset{(\lambda,u^{\pm},v)}{\text{maximize}} & 5\,\lambda + u_1^+ + 2\,u_3^+ - u_1^- - 2\,u_3^- & & \\
\text{subject to} & \lambda - u_1^+ + u_1^- - u_3^+ + u_3^- & \leq & 1 \\
& \lambda - u_2^+ + u_2^- - u_3^+ + u_3^- & \leq & 0 \\
& -u_2^+ + u_2^- - v_1 & \leq & 2 \\
& -u_2^+ + u_2^- + u_3^+ - u_3^- - v_2 & \leq & 0 \\
& u_1^+ - u_1^- - v_3 & \leq & -1 \\
& u_1^+ + u_2^+ + u_3^+ & \leq & 0 \\
& (\,\lambda, u^{\pm}, v\,) \geq 0 & &
\end{array} \tag{5.6}$$

which has an optimal objective value 0. So a better upper bound is found; thus $\mathrm{LPCC_{ub}} = 0$. A point cut: $1 - z_3 \geq 1$ is derived from an optimal solution of (5.6). This cut obviously implies the previous cut: $z_2 + (1 - z_3) \geq 1$. In order to reduce the work load of the IP solver, we can delete $z_2 + (1 - z_3) \geq 1$ from $\mathcal{Z}_{\mathrm{work}}$ and add in $1 - z_3 \geq 1$ instead. So far, we have the updated upper bound: $\mathrm{LPCC_{ub}} = 0$ and the working set $\mathcal{Z}_{\mathrm{work}}$ defined by the two inequalities:

$$z_2 + z_3 \geq 1 \quad \text{and} \quad 1 - z_3 \geq 1. \tag{5.7}$$

This completes iteration 1. During this one iteration, we have solved 5 LPs, the $\mathrm{LPCC_{ub}}$ has improved twice, and we have obtained 2 valid cuts.

Iteration 2: Solving a satisfiability IP yields a $z = (0,1,0) \in \widehat{\mathcal{Z}}_{\mathrm{work}}$. Indeed, any element in $\widehat{\mathcal{Z}}_{\mathrm{work}}$, which is defined by the two inequalities in (5.7), must have $z_2 = 1$ and $z_3 = 0$; thus it remains to determine z_1. As it turns out, z_1 is irrelevant. To see this, we substitute $z = (0,1,0)$ into (2.9), obtaining

$$
\begin{array}{lrcl}
\underset{(\lambda,u^{\pm},v)}{\text{maximize}} & 5\,\lambda + u_1^+ + 2\,u_3^+ - u_1^- - 2\,u_3^- & & \\
\text{subject to} & \lambda - u_1^+ + u_1^- - u_3^+ + u_3^- & \leq & 1 \\
 & \lambda - u_2^+ + u_2^- - u_3^+ + u_3^- & \leq & 0 \\
 & -u_2^+ + u_2^- - v_1 & \leq & 2 \\
 & -u_2^+ + u_2^- + u_3^+ - u_3^- - v_2 & \leq & 0 \\
 & u_1^+ - u_1^- - v_3 & \leq & -1 \\
 & u_2^+ + v_1 + v_3 & \leq & 0 \\
 & (\lambda, u^{\pm}, v) \geq 0. & &
\end{array}
\tag{5.8}
$$

The LP (5.8) is unbounded and has an extreme ray where $u^+ = (0,0,10/7)$ and $v = (0,10/7,0)$. So we can add a valid ray cut: $(1 - z_2) + z_3 \geq 1$ to $\mathcal{Z}_{\text{work}}$.

Termination: The updated working set $\mathcal{Z}_{\text{work}}$ consists of 3 inequalities:

$$
\left\{
\begin{array}{rcl}
z_2 + z_3 & \geq & 1 \\
1 - z_3 & \geq & 1 \\
(1 - z_2) + z_3 & \geq & 1
\end{array}
\right\},
$$

which can be seen to be inconsistent. Hence we get a certificate of termination. Since there is one point cut in $\mathcal{Z}_{\text{work}}$, the LPCC (5.1) has an optimal objective value 0, which happens on the piece $z = (1,1,1)$. (This termination can be expected from the fact that $z_2 = 1$ and $z_3 = 0$ for elements in the set $\widehat{\mathcal{Z}}_{\text{work}}$ prior to the last ray cut; these values of z imply that $y_2 = w_3 = 0$, which are not consistent with the nonnegativity of x. This inconsistency is detected by the algorithm through the generation of a ray cut that leaves $\widehat{\mathcal{Z}}_{\text{work}}$ empty.) □

6. Computational Results. To test the effectiveness of the algorithm, we have implemented and compared it with benchmark algorithms from NEOS, which for the purpose here were chosen to be the FILTER solver and the KNITRO solver. As expected, these two solvers consistently produce high-quality LPCC feasible solutions. For the test problems we used, both often found solutions that turned out to be globally optimal, as was proved by our algorithm. (The details can be seen in Table 1, 2 and 3). We coded our algorithm in MATLAB and used CPLEX 9.1 to solve the LPs and the satisfiability IPs. The experiments were run on a DELL desktop computer with 1.40GHz Pentium 4 processor and 1.00GB of RAM.

Our goal in this computational study is threefold: (A) to test the practical ability of the algorithm to provide a certificate of global optimality for LPCCs with finite optimal solutions; (B) to determine the quality of the solutions obtained using the simple-cut pre-processor; and (C) to demonstrate that the algorithm is capable of detecting infeasibility and unboundedness for LPCCs of these kinds. All problems are randomly generated. One at a time, a total of $\lfloor m/3 \rfloor$ simple cuts are generated in the pre-processing step for each problem. To test (A) and (B), the problems are generated to have optimal solutions; for (C), the problems are generated to be either infeasible or have unbounded objective values. The algorithm does not make use of such information in any way; instead, it is up to the algorithm to verify the prescribed problem status. In all the experiments, optimality of the LPCC is declared if the difference between the lower and upper bound is less than or equal to 1e-6; this tolerance is also employed to determine the LPCC feasibility of the relaxed LP solutions. The parameter δ for the sparsification step is selected to be 0.2.

All problems have the nonnegativity constraint $x \geq 0$. The computational results for the problems with finite optima are reported in Figures 1, 2, and 3 and Tables 1, 2, and 3. Each figure contains one set of ten randomly generated problems with the same characteristics. Figures 1, 2, and 3 correspond to problems with $[n,m,k] = [100,100,90]$, $[300,300,200]$ and $[50,50,55]$, respectively. These sizes and the data density are dictated by the limitations of MATLAB that is the environment where our experiments were performed. All data are randomly generated with uniform distributions. The objectives vectors c and d are generated from the intervals [0 1] and [1 3], respectively. For Figures 1 and 2, the matrix $B = 0$, and the matrix M is generated with up to 2,000 nonzero entries and of the form:

$$
M \equiv \begin{bmatrix} D_1 & E^T \\ -E & D_2 \end{bmatrix}, \tag{6.1}
$$

where D_1 and D_2 are positive diagonal matrices of random order and with elements chosen from [0 2], and E is arbitrary with elements in [-1 1]. The vector q is randomly generated with elements in the interval [-20 -10]. Note that M is positive definite, albeit not symmetric. This property of M and the choice of $B = 0$ ensure LPCC feasibility, and thus optimality (because c and d are nonnegative and the variables are nonnegative). For Figure 3, $B \neq 0$ and the matrix M has no special structure but has only 10% density. The rest of the data A, f, q, and N are generated to ensure LPCC feasibility, and thus optimality. Details of the data generation and the resulting data can be found on the webpage `http://www.rpi.edu/~mitchj/generators/lpcc/`.

Figures 1, 2 and 3 detail the progress of the runs, showing in particular how LPCC_{ub} decreases with the number of iterations. The vertical axis refers to the LPCC objective values and the horizontal axis labels the number of iterations as defined in the opening paragraph of Section 5. The top value on the vertical axis is the LPCC objective value obtained at termination of the pre-processor with the LPCC feasibility recovery step. The bottom value is verifiably LPEC_{min}. The vertical axis is scaled differently in each run with respect to the difference between the top and the bottom values. As comparison, the objective values obtained from FILTER (marked by the red square) and KNITRO (marked by the blue diamond) are also shown on the vertical axis; if the difference between the FILTER and KNITRO values in a run is within 1e-3, we only mark the KNITRO result (the exact values from these two solvers can be found in Tables 1, 2 and 3). The upper limit of the horizontal axis indicates the number of IPs needed to be solved in each run. Note that in some runs, a globally optimal solution might have been obtained in an earlier iteration without certification, and the algorithm needs more subsequent iterations to verify its global optimality. For example, in the fourth run of the right-hand column in Figure 1, a globally optimal solution is first obtained at iteration 2, but the certificate is established only after 23 more iterations. Other details about the figures are summarized in the remarks below the figures.

Corresponding to the problems in Figures 1, 2 and 3 respectively, Tables 1, 2 and 3 report more details about the runs, which are indexed by counting first row-wise and then column-wise in the figures (for example, the fourth run in Table 1 is the second row on the right column in Figure 1). In addition to the objective values obtained in our algorithm and from the NEOS solvers, these tables also report the numbers of IPs and LPs (excluding the $\lfloor m/3 \rfloor$ relaxed LPs solved in the pre-processor), solved in the solution process. These numbers, which are independent of the computational platform and machine, provide a good indicator of the efforts required by the algorithm in processing the LPCCs. We did not report computational times for two reasons: (i) the MATLAB results are computer dependent and the runs involve interfaces between MATLAB and CPLEX, and (ii) our runs are experimental and our coding is at an amateur level.

The computational results for the infeasible and unbounded LPCCs are reported in Table 4, which contains 3 sub-tables (a), (b), and (c). The first two sub-tables (a) and (b) pertain to feasible but unbounded LPCCs. For the unbounded problems, we set $B = 0$, q is arbitrary, and we generate A with a nonnegative column, M given by (6.1) and f such that $\{x \geq 0 : Ax \geq f\}$ is feasible. Problems in (a) and (b) have the same parameters except for the objective vector c and d and matrix A. For the problems in (a), we simply maximize one single x-variable whose A column is nonnegative. For the problems in (b), the objective vectors c and d are both negative; and the matrix A is the same as it is in group (a) except that a small number 0.005 is added to its nonnegative column (see the discussion in the first conclusion below for why this is done). The third sub-table (c) pertains to a class of infeasible LPCCs generated as follows: q, N, and M are all positive so that the only solution to the LCP: $0 \leq y \perp q + Nx + My \geq 0$ for $x \geq 0$ is $y = 0$; $Ax + By \geq f$ is feasible for some $(x, y) \geq 0$ with $y \neq 0$ but $Ax \geq f$ has no solution in $x \geq 0$.

To illustrate the effectiveness of the sparsification step, we generated some LPCCs with $n = m = k = 25$ and the same characteristics as the problems in Figure 3. Table 5 reports the numbers of LPs and IPs that are needed to be solved in both runs with or without this step.

The main conclusions from the experiments are summarized below.

• The algorithm successfully terminates with the correct status of all the LPCCs reported. In fact, we have tested many more problems than those reported and obtained similar success. There are, nevertheless, a few instances where the LPCC are apparently unbounded but the algorithm fails to terminate after 6,000 iterations without the definitive conclusion, even though the LPCC objective is noticeably tending to $-\infty$. We cannot explain these exceptional cases which we suspect are due to round-off errors in the computations. This suspicion led us to add the small 0.005 in the unbounded set of runs reported above; with this small adjustment, the algorithm successfully terminated with the desired certificate of unboundedness.

• For the special LPCCs with $B = 0$, the results from the two NEOS algorithms, FILTER and KNITRO, are proved to be suboptimal in 2 out of the 20 runs (the first and fourth runs on the left column in Figure 1). In the other 18 runs, our algorithm is able to obtain an optimal solution with little computational effort (within 5 iterations), but requires significant additional computations to produce the desired certificate of global optimality. For the general LPCCs with $B \neq 0$, the objective values obtained from FILTER and KNITRO are suboptimal in 6 out of 10 runs. In the other 4 runs, only 5 iterations are needed to derive either a globally optimal solution or an LPCC feasible solution whose objective value is within 3% of the optimal value. These results confirm that the verification of global optimality is generally much more demanding than the computation of the solution without proof of optimality.

• Except for one problem (problem 7 in Table 3), the solutions obtained by the simple-cut pre-processor for all LPCCs with finite optima are within 5% of the globally optimal solutions. In fact, some of the solutions obtained from the pre-processing are immediately verified to be optimal. This suggests that very high-quality LPCC feasible solutions can be produced efficiently by solving a reasonable number of LPs.

• The sparsification procedure is quite effective; so is the LPCC feasibility recovery step. Indeed without the latter, there is a significant percentage of problems where the algorithm fails to make progress after 3,000 iterations. With this step installed, all problems are resolved satisfactorily.

• While the numbers of IPs solved are quite reasonable in most cases, there are several runs where the numbers of relaxed LPs solved are unusually large, especially when the problem size increases. This suggests that stronger cuts are needed for both general LPCCs and for specialized problems arising from large-scale applications. The implementation of a dedicated solver for satisfiability problems, such as those described in [7, 31], could considerably improve the overall solution times of the LPCC algorithm. These refinements of the algorithm are presently being investigated.

Concluding remarks. In this paper, we have presented a parameter-free IP based algorithm for the global resolution of an LPCC and reported computational results with the application of the algorithm for solving a set of randomly generated LPCCs of moderate sizes. Continued research on refining the algorithm and applying it to realistic classes of LPCCs, such as the bilevel machine learning problems described in [6, 32, 33] and other applied problems, is currently underway.

Acknowledgements. The authors are grateful to 2 referees for their constructive comments that have significantly improve the presentation of the paper. In particular, one of them alerted the authors to the reference [9] that discusses Benders techniques for conditional linear constraints without requiring the big-M.

REFERENCES

[1] M. Anitescu. On using the elastic mode in nonlinear programming approaches to mathematical programs with complementarity constraints. *SIAM Journal on Optimization* 15 (2005) 1203–1236.

[2] M. Anitescu. Global convergence of an elastic mode approach for a class of mathematical programs with complementarity constraints. *SIAM Journal on Optimization* 16 (2005) 120–145.

[3] M. Anitescu, P. Tseng, and S.J. Wright. Elastic-mode algorithms for mathematical programs with equilibrium constraints: Global convergence and stationarity properties. *Mathematical Programming* 110 (2007) 337–371.

[4] C. Audet, P. Hansen, B. Jammard, and G. Savard. Links between linear bilevel and mixed 0–1 programming problems. *Journal of Optimization Theory and Applications* 93 (1997) 273–300.

[5] C. Audet, G. Savard, and W. Zghal. New branch-and-cut algorithm for bilevel linear programming, *Journal of Optimization Theory and Applications* 134 (2007) 353–370.

[6] K. Bennett, X. Ji, J. Hu, G. Kunapuli, and J.S. Pang. Model selection via bilevel programming, *Proceedings of the International Joint Conference on Neural Networks (IJCNN'06) Vancouver, B.C. Canada, July 16–21, 2006* pp.

[7] B. Borchers and J. Furman. A two-phase exact algorithm for MAX-SAT and weighted MAX-SAT Problems. *Journal of Combinatorial Optimization* 2 (1998) 299–306.

[8] L. Chen and D. Goldfarb. An active-set method for mathematical programs with linear complementarity constraints. Manuscript, Department of Industrial Engineering and Operations Research, Columbia University (January 2007).

[9] G. Codato and M. Fischetti. Combinatorial Benders' Cuts for mixed-integer linear programming. *Operations Research* 54 (2006) 758–766.

[10] R.W. Cottle, J.S. Pang, and R.E. Stone. *The Linear Complementarity Problem*, Academic Press, Boston (1992)

[11] S. Dempe. *Foundations of bilevel programming*. Kluwer Academic Publishers (Dordrecht, The Netherlands 2002).

[12] S. Dempe. Annotated bibliography on bilevel programming and mathematical programs with equilibrium constraints. *Optimization* 52 (2003) 333–359.

[13] S. Dempe, V.V. Kalashnikov, and N. Kalashnykova. Optimality conditions for bilevel programming problems. In S. Dempe and N. Kalashnykova, editors, *Optimization with Multivalued Mappings: Theory, Applications and Algorithms*. Springer-Verlag New York Inc. (New York 2006) pp. 11–36.

[14] M.L. Flegel and C. Kanzow. On the Guignard constraint qualification for mathematical programs with equilibrium constraints. *Optimization* 54 (2005) 517–534.

[15] R. Fletcher and S. Leyffer. Solving mathematical program with complementarity constraints as nonlinear programs. *Optimization Methods and Software* 19 (2004) 15-40.

[16] R. Fletcher, S. Leyffer, D. Ralph, and S. Scholtes. Local convergence of SQP methods for mathematical programs with equilibrium constraints. *SIAM Journal on Optimization* 17 (2006) 259-286.

[17] C.A. Floudas and V. Visweswaran. A global optimization algorithm (GOP) for certain classes of nonconvex NLPs-I. Theory. *Computers and Chemical Engineering* 14 (1990) 1397–1417.

[18] C.A. Floudas and V. Visweswaran. Primal-relaxed dual global optimization approach. *Journal of Optimization Theory and Applications* 78 (1993) 187–225.

[19] M. Fukushima and J.S. Pang. Convergence of a smoothing continuation method for mathematical programs with complementarity constraints. In M. Thera and R. Tichatschke, editors, *Ill-Posed Variational Problems and Regularization Techniques*. Lecture Notes in Economics and Mathematical Systems, Vol. 477, Springer-Verlag, (Berlin/Heidelberg 1999) pp. 99–110.

[20] M. Fukushima and P. Tseng. An implementable active-set algorithm for computing a B-stationary point of the mathematical program with linear complementarity constraints. *SIAM Journal on Optimization* 12 (2002) 724–739. [With erratum].

[21] P. Hansen, B. Jammard, and G. Savard. New branch-and-bound rules for linear bilevel programming. *SIAM Journal on Scientific and Statistical Computing* 13 (1992) 1194–1217.

[22] J.N. Hooker. *Logic-Based Methods for Optimization*. Wiley (2000).

[23] J.N. Hooker. *Integrated Methods for Optimization*. Springer (New York 2006).

[24] J.N. Hooker and G. Ottosson. Logic-based Benders decomposition. *Mathematical Programming* 96 (2003) 33–60.

[25] X.M. Hu and D. Ralph. Convergence of a penalty method for mathematical programming with complementarity constraints. *Journal of Optimization Theory and Applications* 123 (2004) 365–390.

[26] T. Ibaraki. Complementary programming. *Operations Research* 19 (1971) 1523–1529.

[27] T. Ibaraki. The use of cuts in complementary programming. *Operations Research* 21 (1973) 353–359.

[28] R.G. Jeroslow. Cutting planes for complementarity constraints. *SIAM Journal on Control and Optimization* 16 (1978) 56–62.

[29] H. Jiang and D. Ralph. Smooth SQP methods for mathematical programs with nonlinear complementarity constraints. *SIAM Journal of Optimization* 10 (2000) 779–808.

[30] H. Jiang and D. Ralph. Extension of quasi-Newton methods to mathematical programs with complementarity constraints. *Computational Optimization and Applications* 25 (2003) 123–150.

[31] S. Joy, J.E. Mitchell, and B. Borchers. A branch-and-cut algorithm for MAX-SAT and weighted MAX-SAT. In D. Du, J. Gu, and P.M. Pardalos, editors, *Satisfiability Problem: Theory and Applications*. [DIMACS Series in Discrete Mathematics and Theoretical Computer Science, volume 35], American Mathematical Society (1997) pp. 519–536.

[32] G. Kunapuli, K. Bennett, J. Hu, and J.S. Pang. Classification model selection via bilevel programming, *Optimization Methods and Software*, forthcoming.

[33] G. Kunapuli, K. Bennett, J. Hu, and J.S. Pang. Bilevel model selection for support vector machines. Manuscript, Department of Mathematical Sciences, Rensselaer Polytechnic Institute (March 2007).

[34] L. Lasdon. *Optimization Theory of Large Systems*. Dover Publications (2002).

[35] S. Leyffer. Complementarity constraints as nonlinear equations: Theory and numerical experience. In S. Dempe and V. Kalashnikov, editors, *Optimization and Multivalued Mappings*. Springer (2006) 169-208.

[36] S. Leyffer, G. Lopéz-Calva, and J. Nocedal. Interior point methods for mathematical programs with complementarity constraints. *SIAM Journal on Optimization* 17 (2006) 52-77.

[37] Z.Q. Luo, J.S. Pang, and D. Ralph. *Mathematical Programs with Equilibrium Constraints*. Cambridge University Press (Cambridge 1996).

[38] J.V. Outrata. Optimality conditions for a class of mathematical programs with equilibrium constraints. *Mathematics of Operations Research* 24 (1999) 627–644.

[39] J. Outrata, M. Koçvara, and J. Zowe. *Nonsmooth Approach to Optimization Problems with Equilibrium Constraints*. Kluwer Academic Publishers (Dordrecht 1998).

[40] J.S. Pang and M. Fukushima. Complementarity constraint qualifications and simplified B-stationarity conditions for mathematical programs with quilibrium constraints, *Computational Optimization and Applications* 13 (1999) 111–136.

[41] J.S. Pang and M. Fukushima. Some feasibility issues in mathematical programs with equilibrium constraints, *SIAM Journal of Optimization* 8 (1998) 673–681.

[42] J.S. Pang and S. Leyffer. On the global minimization of the value-at-risk, *Optimization Methods and Software* 19 (2004) 611–631.

[43] D. Ralph and S.J. Wright. Some properties of regularization and penalization schemes for MPECs. *Optimization Methods and Software* 19 (2004) 527–556.

[44] H. Scheel and S. Scholtes. Mathematical programs with complementarity constraints: Stationarity, optimality and sensitivity. *Mathematics of Operations Research* 25 (2000) 1–22.

[45] S. Scholtes. Convergence properties of a regularization scheme for mathematical programs with complementarity constraints. *SIAM Journal on Optimization* 11 (2001) 918–936.

[46] M. Tawarmalani and N.V. Sahinidis. *Convexification and Global Optimization in Continuous and Mixed Integer Nonlinear Programming*. Kluwer Academic Publishers (Dordrecht 2002).

[47] V. Visweswaran, C.A. Floudas, M.G. Ierapetritou, and E.N. Pistikopoulos. A decomposition based global optimization approach for solving bilevel linear and nonlinear quadratic programs. In C.A. Floudas and P.M. Pardalos, editors. *State of the Art in Global Optimization : Computational Methods and Applications*. [Book Series on Nonconvex Optimization and Its Applications, Vol. 7] Kluwer Academic Publishers (Dordrecht 1996) pp. 139–162.

[48] J.J. Ye. Optimality conditions for optimization problems with complementarity constraints. *SIAM Journal on Optimization* 9 (1999) 374–387.

[49] J.J. Ye. Constraint qualifications and necessary optimality conditions for optimization problems with variational inequality constraints. *SIAM Journal on Optimization* 10 (2000) 943–962.

[50] J.J. Ye. Necessary and sufficient optimality conditions for mathematical programs with equilibrium constraints. *Journal of Mathematical Analysis and Applications* 30 (2005) 350–369.

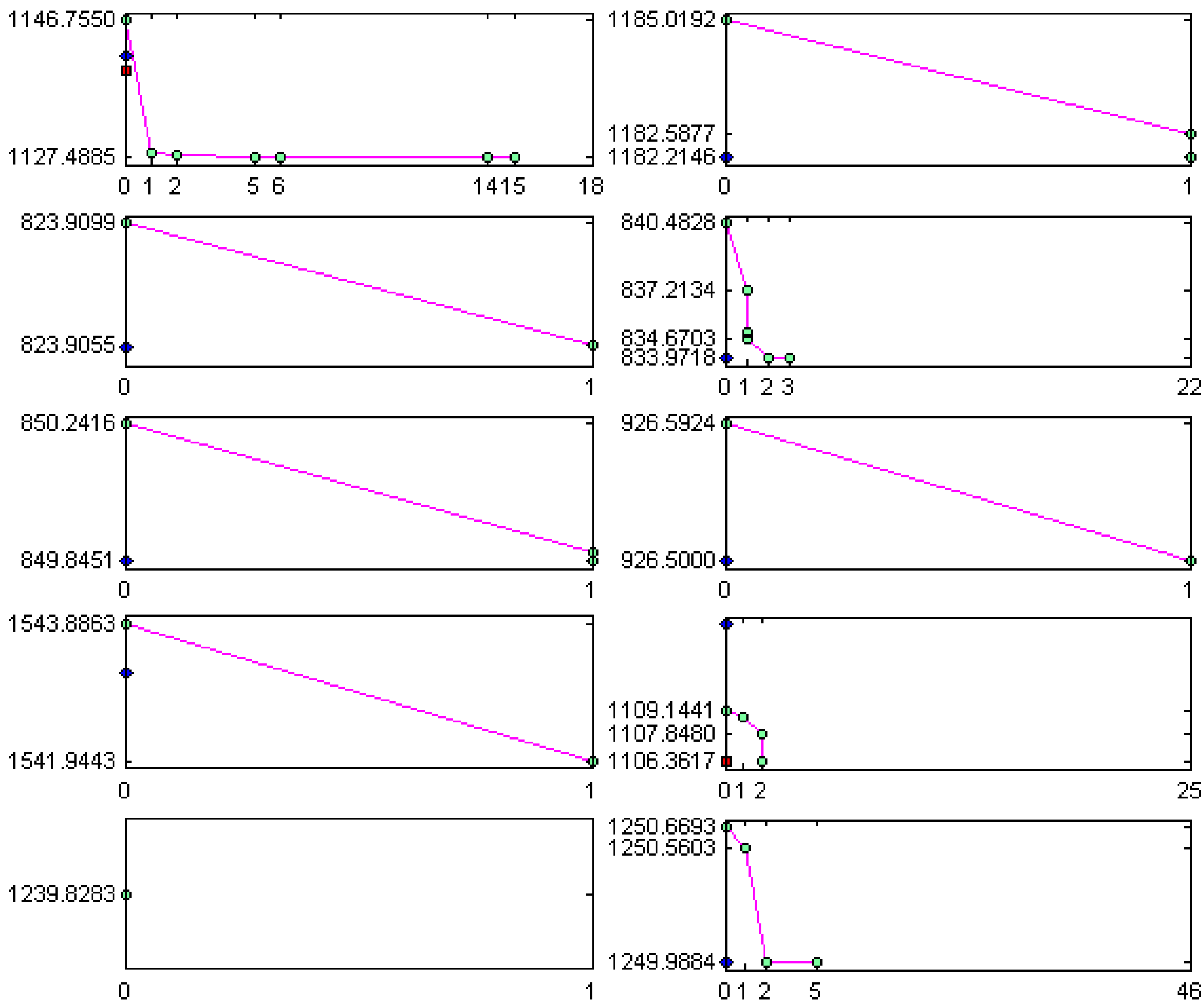

Fig. 6.1. Special LPCCs with $B = 0$, $A \in \Re^{90\times 100}$, and 100 complementarities.

Remark: Each circle signifies that a better feasible LPCC solution is found. The circle's horizontal coordinate indicates the iteration where LPCC_{ub} is updated; its vertical coordinate gives the value of updated LPCC_{ub}, (we omitted some values if they are not significantly improved). Note that it is possible for LPCC_{ub} to improve within one iteration by the sparsification step; see the example in Subsection 5.1 and also the top run in the right column. In the fifth run in the left column, both of the FILTER and KNITRO results coincide with LPCC_{min}, which is obtained after pre-processing and verified to be optimal after 1 iteration.

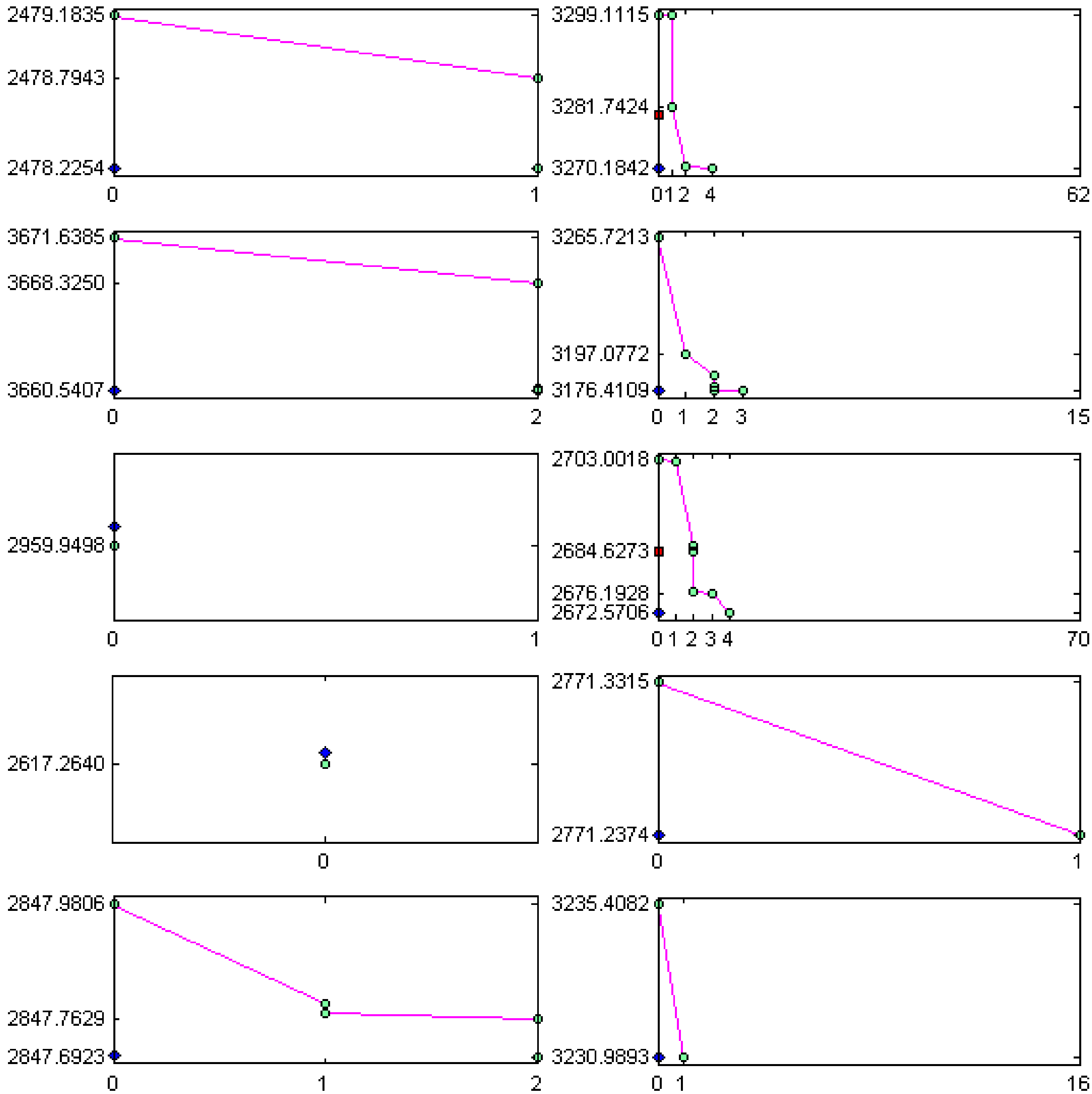

Fig. 6.2. Special LPCCs with $B = 0$, $A \in \Re^{200\times300}$, and 300 complementarities.

Remark: The explanation for the figure is similar to that of Figure 1. Note that in the third and fourth runs in the left column, LPCC_{ub} is obtained right after preprocessing. In the third run, the solution's global optimality is verified after 1 iteration; while in the fourth run, the solution is immediately verified to be globally optimal (the difference between the upper and lower bound of the LPEC is within 1e-6).

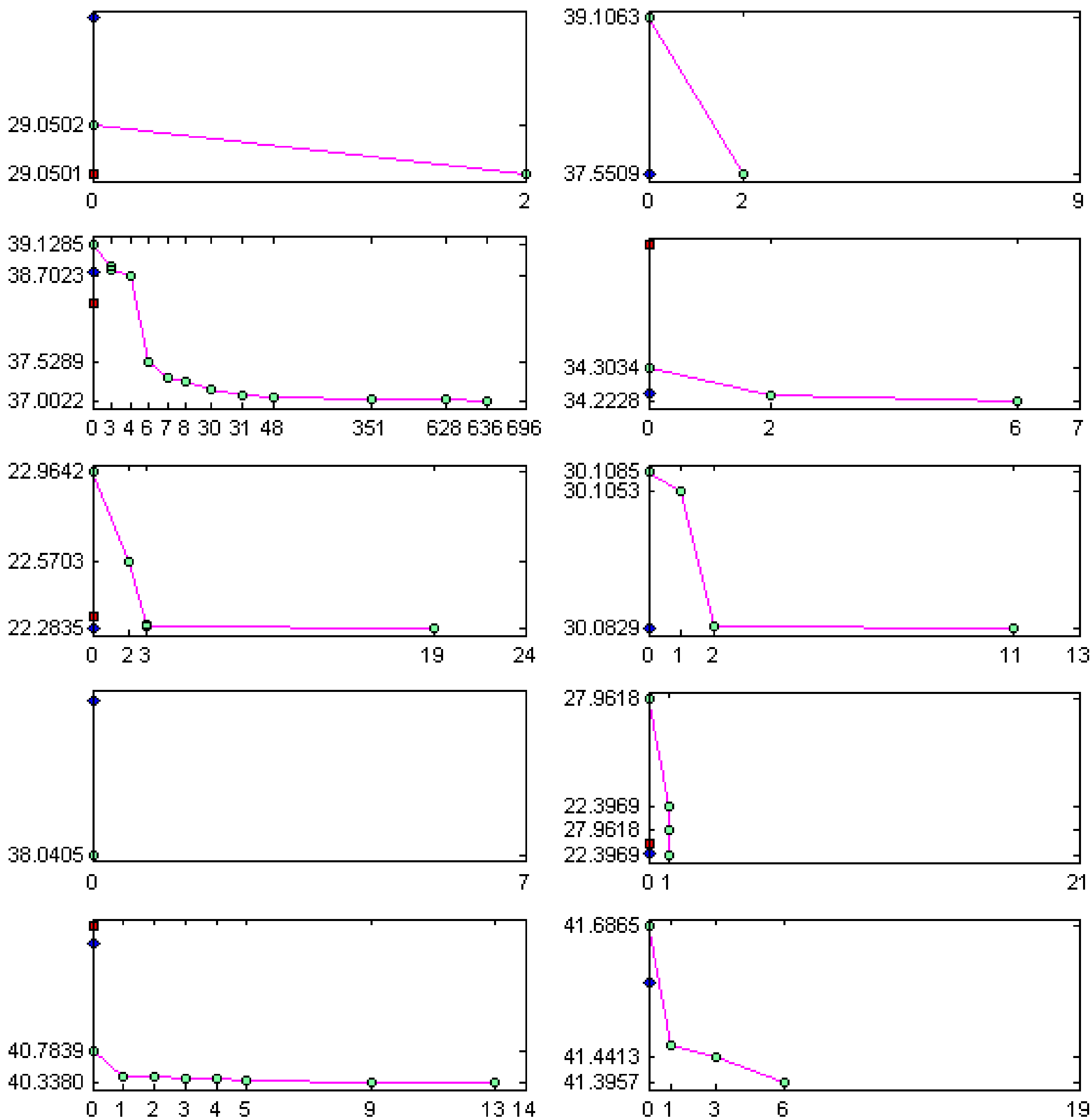

Fig. 6.3. General LPECs with $B \neq 0$, $A \in \Re^{55\times 50}$, and 50 complementarities.

Prob	LPCC_{lb}		LPCC_{ub}		FILTER	KNITRO	LPs	IPs
	relaxed LP	preprocess	preprocess	$\text{LPCC}_{\min}$				
1	$1094.6041_{+12.2255}$	1106.8297	$1146.7550_{-19.2665}$	1127.4885	1140.5614_{5}	1141.6696_{52}	396	18
2	$1172.1830_{+4.5063}$	1176.6893	$1185.0192_{-2.8046}$	1182.2146	1182.2145_{5}	1182.2147_{47}	57	1
3	$820.2584_{+3.6328}$	823.8912	$823.9099_{-0.0044}$	823.9055	823.9055_{10}	823.9058_{55}	14	1
4	$796.9560_{+17.0192}$	813.9752	$840.4828_{-6.5110}$	833.9718	833.9717_{6}	833.9718_{41}	611	22
5	$841.1786_{+7.8336}$	849.0122	$850.2416_{-0.3965}$	849.8451	849.8451_{5}	849.8452_{44}	66	1
6	$924.7529_{+1.3500}$	926.1028	$926.5924_{-0.0923}$	926.5000	926.5000_{5}	926.5000_{56}	21	1
7	$1536.1748_{+5.2715}$	1541.4464	$1543.8863_{-1.9419}$	1541.9443	1543.1950_{6}	1543.1951_{55}	35	1
8	$1076.8760_{+13.3395}$	1090.2155	$1109.1441_{-2.7824}$	1106.3617	1106.3616_{5}	1113.8938_{70}	363	25
9	$1232.7912_{+6.9243}$	1239.7156	1239.8283	1239.8283	1239.8284_{7}	1239.8285_{62}	10	1
10	$1217.1191_{+12.1543}$	1229.2734	$1250.6693_{-0.6808}$	1249.9884	1249.9884_{8}	1249.9886_{67}	832	46

Table 6.1
Special LPECs with $B = 0$, $A \in \Re^{90\times 100}$, and 100 complementarities.

Remark: The first column "Prob" is the problem counter; the second column "LPCC_{lb}" contains the objective values of LP relaxations before and after the pre-processing. The subscript at each number in the left subcolumn indicates the difference between the objective value before and after the $\lfloor m/3 \rfloor$ simple cuts. The column "LPCC_{ub}" reports the objective values of the LPCC feasible solutions. The right subcolumn contains the verifiably optimal $\text{LPCC}_{\min}$. The left subcolumn contains the values obtained after pre-processing with the LPCC feasibility recovery procedure; the footnotes give the difference between the left and right subcolumns. The objective values obtained from FILTER and KNITRO are reported in the next columns. (Note that these values are very comparable and practically optimal in all problems except #1 and 7 for both and 8 for KNITRO.) The total number of LPs solved (excluding the $\lfloor m/3 \rfloor$ relaxed LPs in the pre-processing step), and the number of IPs solved in the run are reported in the last two columns. At the suggestion of a referee, we also reported the number of "major iterations" in the two NEOS solvers; these are placed as subscripts in the objective values of the respective solvers. It should be noted that such iterations refer to different procedures in the two solvers.

Prob	LPCC_{lb}		LPCC_{ub}		FILTER	KNITRO	LPs	IPs
	relaxed LP	preprocess	preprocess	$\text{LPCC}_{\min}$				
1	$2469.4400_{+4.8766}$	2474.3166	$2479.1835_{-0.9581}$	2478.2254	2478.2256_{19}	2478.2264_{66}	125	1
2	$3213.7176_{+15.4754}$	3229.1930	$3299.1115_{-28.9273}$	3270.1842	3280.1865_{8}	3270.1844_{72}	4071	62
3	$3639.4490_{+12.1224}$	3651.5714	$3671.6385_{-11.0978}$	3660.5407	3660.5412_{42}	3660.5412_{79}	350	2
4	$3127.3708_{+12.9412}$	3140.3119	$3265.7213_{-89.3103}$	3176.4109	3176.4108_{11}	3176.4115_{69}	1249	15
5	$2958.9147_{+1.0234}$	2959.9381	2959.9498	2959.9498	2959.9495_{6}	2959.9529_{66}	5	1
6	$2630.3282_{+15.3489}$	2645.6771	$2703.0018_{-30.4312}$	2672.5706	2684.5288_{30}	2672.5710_{60}	4511	70
7	$2616.9852_{+0.2788}$	2617.2640	2617.2640	2617.2640	2617.2638_{14}	2617.2673_{65}	0	0
8	$2766.9544_{+3.1966}$	2770.1510	$2771.3315_{-0.0940}$	2771.2374	2771.2372_{27}	2771.2379_{70}	26	1
9	$2842.4480_{+4.2693}$	2846.7174	$2847.9806_{-0.2882}$	2847.6923	2847.6926_{7}	2847.6929_{48}	319	2
10	$3207.6861_{+12.6949}$	3220.3810	$3235.4082_{-4.4189}$	3230.9893	3230.9896_{6}	3230.9897_{66}	1569	16

Table 6.2
Special LPECs with $B = 0$, $A \in \Re^{200\times 300}$, and 300 complementarities.

Remark: The explanation of this table is the same as Table 1. Note that in the problem 7, the solution obtained after the pre-processing step is immediately verified to be globally optimal. For these runs, the KNITRO solutions are practically optimal in all cases; but the FILTER solution in problem #2 is noticeably suboptimal.

Prob	LPCC_{lb}		LPEC_{ub}		FILTER	KNITRO	LPs	IPs
	relaxed LP	preprocess	preprocess	$\text{LPCC}_{\min}$				
1	$28.7739_{+0.2580}$	29.0318	$29.0502_{-0.0001}$	29.0501	29.0501_{8}	30.0155_{32}	21	2
2	$36.1885_{+0.6373}$	36.8258	$39.1063_{-1.5554}$	37.5509	37.5509_{7}	37.5510_{25}	229	9
3	$33.8630_{+0.6357}$	34.4988	$39.1285_{-2.1263}$	37.0022	38.3216_{7}	38.7521_{28}	4842	696
4	$33.7618_{+0.3861}$	34.1479	$34.3034_{-0.0806}$	34.2228	34.6057_{5}	34.2398_{54}	102	7
5	$21.4187_{+0.5059}$	21.9246	$22.9642_{-0.6806}$	22.2835	22.2945_{5}	22.2837_{35}	209	24
6	$29.8919_{+0.0762}$	29.9681	$30.1085_{-0.0255}$	30.0829	30.0829_{6}	30.0830_{26}	108	13
7	$37.6712_{+0.3261}$	37.9972	38.0405	38.0405	38.0419_{6}	38.0419_{29}	92	7
8	$20.8210_{+0.6375}$	21.4586	$27.9618_{-5.5649}$	22.3969	22.7453_{7}	22.4164_{37}	187	21
9	$39.0227_{+0.4565}$	39.4792	$40.7839_{-0.4459}$	40.3380	44.7872_{8}	44.3173_{26}	321	14
10	$40.0135_{+0.7859}$	40.7994	$41.6865_{-0.2908}$	41.3957	41.5810_{5}	41.5810_{37}	190	19

TABLE 6.3
General LPCCs with $B \neq 0$, $A \in \Re^{55\times 50}$, and 50 complementarities.

Remark: For these runs, there are more instances where the two NEOS solutions are noticeably suboptimal.

Prob	# iters (a)	# cuts (a)	# LPs (a)	# iters (b)	# cuts (b)	# LPs (b)	# iters (c)	# cuts (c)	# LPs (c)
1	50	47	195	4	3	9	14	14	28
2	6	4	14	5	4	12	2	2	4
3	1081	828	2604	2	1	5	38	38	76
4	166	144	424	42	39	120	7	7	14
5	436	305	991	735	621	1860	47	49	100
6	18	17	54	498	379	1125	48	48	96
7	3	4	11	352	127	663	20	20	40
8	426	356	1191	489	373	1158	13	13	26
9	9	9	26	5	4	12	50	50	100
10	4	3	11	9	7	22	6	6	12

TABLE 6.4
Infeasible and unbounded LPECs with 50 complementarities.

iters = number of returns to Step 1 = number of IPs solved
cuts = number of satisfiability constraints in $\mathcal{Z}_{\text{work}}$ at termination
LPs = number of LPs solved, excluding the $\lfloor m/3 \rfloor$ relaxed LPs in the pre-processing step

Prob	A		B	
	# LPs	# IPs	# LPs	# IPs
1	122	122	61	18
2	17	17	33	11
3	7	7	51	9
4	16	16	41	12
5	280	280	70	21
6	598	598	85	23
7	195	195	90	26
8	65	65	43	10
9	9	9	41	9
10	8	8	33	8

TABLE 6.5
General LPCCs with $B \neq 0$, $A \in \Re^{25\times 25}$, and 25 complementarities.

column A = number of IPs or LPs solved in the run without sparsification
column B = number of IPs or LPs solved in the run with sparsification

Remark: In column A, the number of IPs solved in the run is equal to the number of solved LPs. Except for the problems 3, 9 and 10, the B approach (with sparsification step implemented) is doing much better than the A approach. Especially for problems 1, 5–8, the numbers of IPs and LPs are dramatically reduced. For the remaining problems, the computational effort with sparsification is at least comparable to, if not better than, the approach without sparsification. With the number of complementarities in the LPCCs grows, we expect more computational savings with the sparsification step implemented.