ARGONNE NATIONAL LABORATORY
9700 South Cass Avenue
Argonne, Illinois 60439

Regularizing Bilevel Nonlinear Programs by Lifting

Kathrin Hatz, Sven Leyffer, Johannes P. Schlöder, and Hans Georg Bock

Mathematics and Computer Science Division

Preprint ANL/MCS-P4076-0613

June 3, 2013

This work was supported by the Office of Advanced Scientific Computing Research, Office of Science, U.S. Department of Energy, under Contract DE-AC02-06CH11357.

REGULARIZING BILEVEL NONLINEAR PROGRAMS BY LIFTING

KATHRIN HATZ *, SVEN LEYFFER†, JOHANNES P. SCHLÖDER‡, AND HANS GEORG BOCK§

Abstract. This paper considers a bilevel nonlinear program (NLP) whose lower-level problem satisfies a linear independence constraint qualification (LICQ) and a strong second-order condition (SSOC). One would expect the resulting mathematical program with complementarity constraints (MPCC), whose constraints are the first-order optimality conditions of the lower-level NLP, to satisfy an MPEC-LICQ. We provide an example which demonstrates that this is not the case. A lifting technique is presented to remedy this problem. A componentwise lifting of the inequality constraints of the lower-level problem implies that the resulting MPCC satisfies an MPCC-LICQ which leads to a faster convergence. We generalize the lifting approach to general MPCCs. Convergence results and numerical experiments are provided that show the promise of our approach.

Key words. nonlinear programming, bilevel optimization, lifting bilevel programs, mathematical programs with equilibrium constraints, mathematical programs with complementarity constraints.

AMS subject classifications. 90C30, 90C33, 90C46, 65K10

1. Introduction. We are concerned with a special class of optimization problems with equilibrium constraints that arise from bilevel nonlinear programs of the form

$$\begin{array}{lll} \underset{x}{\text{minimize}} & F(x,y) & \\ \text{subject to} & \underset{y}{\text{minimize}} & f(x,y) \\ & \text{subject to} & h(x,y) \geq 0, \end{array} \tag{1.1}$$

where $x \in \mathbb{R}^n$ and $y \in \mathbb{R}^m$. The inequality constraints are denoted as $h(x,y) \in \mathbb{R}^k$. We use upper case letters for the objective function of the upper level problem, $F(x,y)$ which is assumed to be twice continuously differentiable, and lower case letters for the objective $f(x)$ and $h(x)$ of the lower level problem, which are also assumed to be twice continuously differentiable.

For the sake of simplifying the presentation, we have omitted upper level constraints and equality constraints in the lower-level problem from (1.1). Equality constraints in the lower level problem are readily included, and the same is true for upper-level equality or inequality constraints $G(x) \in \mathbb{R}^l$. We can readily extend the proposed lifting technique provided that the Jacobian of the additional upper-level constraints $\nabla_x G_{\mathcal{A}}(x)$ with $\mathcal{A} := \{i \in \{1, \cdots, l\} \mid G_i(x) = 0\}$ has full rank with $l \leq n$.

We can formulate (1.1) as an MPCC by replacing the lower level NLP by its necessary optimality conditions:

$$\begin{array}{lll} \underset{x,y,z}{\text{minimize}} & F(x,y) & \\ \text{subject to} & 0 & = \nabla_y \mathcal{L}(x,y,z) \\ & 0 & \leq z \perp h(x,y) \geq \ \ 0, \end{array} \tag{1.2}$$

*Interdisciplinary Center for Scientific Computing, University of Heidelberg, Germany, kathrin.hatz@iwr.uni-heidelberg.de

†Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, IL 60439, USA, leyffer@mcs.anl.gov

‡Interdisciplinary Center for Scientific Computing, University of Heidelberg, Germany, Schloeder@iwr.uni-heidelberg.de

§Interdisciplinary Center for Scientific Computing, University of Heidelberg, Germany, SciCom@iwr.uni-heidelberg.de

with the gradient of the Lagrangian given by $\nabla_y \mathcal{L}(x, y, z) := \nabla_y f(x, y) - \nabla_y h(x, y)^T z$ and the Lagrange multipliers $z \in \mathbb{R}^k$. We note, that in the absence of convexity of the lower-level problem, the MPCC, (1.2), is in general not equivalent to the bilevel NLP, (1.1). We are not concerned with this issue here. Instead, we investigate practical algorithms for solving (1.2).

In general, MPCCs violate the Mangasarian-Fromowitz constraint qualification (MFCQ) at every feasible point; see [5, 22]. This may cause problems for nonlinear solvers. The violation of MFCQ implies that the linear independence constraint qualification (LICQ) is also violated at every feasible point. Common approaches to tackle problem (1.2) are based on branch-and-bound methods [4], nonsmooth optimization [19], interior-point methods [14, 21], piecewise sequential quadratic programming (SQP) methods [17], or penalization and relaxation techniques [7, 12, 16, 23, 25]. Most of these approaches require a computational effort that is significantly larger than the one of an SQP method applied to an MPCC [10, 11]. Furthermore, the numerical experience with SQP methods applied to MPCCs is quite promising [10]. Hence, we follow the latter approach. Fletcher et al. [11] provide convergence results for SQP methods applied to MPCCs under relatively mild assumptions. One important assumption is that the MPCC satisfies an MPCC-LICQ. In this paper, we consider the bilevel NLP (1.1) and assume that its lower level satisfies LICQ and a strong second-order condition (SSOC). One would expect that (1.2) satisfies an MPCC-LICQ (inherited from LICQ and SSOC of the lower-level NLP), but this is not the case, and we provide a counterexample. We then propose a componentwise lifting of the lower-level inequality constraints that can be shown to ensure an MPCC-LICQ for (1.2). Furthermore, we generalize our approach to lift general MPCCs.

An overview and a general discussion of bilevel programs is provided in [6] and [1]. An idea related to the method described in this paper can be found in [2, 3, 24], and the references therein. In [2, 3] the authors consider general MPCCs, and introduce an additional scalar variable to relax the MPCC's standard equality and inequality constraint and the complementarity conditions is treated with an exact penalty term. Global convergence properties based on this formulation are discussed. However, the reformulated MPCC cannot be shown to guarantee an MPCC-LICQ. In [24], a smoothing approach for mathematical programs with complementarity constraints is presented, based on the projection of a suitable set in $\mathbb{R}^3$. The author introduces a regularization approach involving a new concept of tilting stability, discusses the regularity of the feasible set and presents preliminary numerical results. In contrast to [2, 3], our approach is able to guarantee MPEC-LICQ at any feasible point. Furthermore, in contrast to [24], our approach does not require a multiphase procedure, we just solve one MPCC which is the lifted version of the original problem.

This paper is organized as follows. We start with a motivating example showing that LICQ and SSOC on the lower level do not imply an MPCC-LICQ of the resulting MPCC. This is followed by a detailed description of the new lifting technique. Finally, a detailed convergence analysis and numerical results are presented.

2. Lifting for MPCC-LICQ. We start by reviewing a standard NLP constraint qualification and second-order conditions for an NLP of the following form

$$\begin{array}{ll} \underset{y}{\text{minimize}} & f(x, y) \\ \text{subject to} & h(x, y) \geq 0. \end{array} \tag{2.1}$$

The NLP (2.1) coincides with the lower-level problem of (1.1). We note that x is not a variable in (2.1). For a given point $y^* \in \mathbb{R}^m$, the active set of (2.1) is denoted by

$\mathcal{A}(x, y^*) := \{i \in \{1, \cdots, k\} \mid h_i(x, y^*) = 0\}$.

DEFINITION 2.1. *The linear independence constraints qualification (LICQ) is said to hold at* y^* *if the set of active constraint gradients* $\{\nabla_y h_i(x, y^*),\ i \in \mathcal{A}(x, y^*)\}$ *is linearly independent.*

The set $\mathcal{F}(x, y) := \{y \in \mathbb{R}^m \mid h(x, y) \geq 0\}$ denotes the feasible set, $\mathcal{L}(x, y, z) := f(x, y) - h(x, y)^T z$ is the Lagrangian of (2.1) with Lagrange multipliers $z \in \mathbb{R}^k$ and $\mathcal{T}(x, y) := \{p \in \mathbb{R}^m \mid \nabla_y h_i(x, y)^T p = 0,\ \forall i \in \mathcal{A}(x, y) \text{ with } z_i > 0\ \}$ is the tangent cone.

DEFINITION 2.2. *If*

$$p^T \nabla_y^2 \mathcal{L}(x, y^*, z^*) p > 0,\ \forall p \in \mathcal{T}(y^*) \backslash \{0\}, \tag{2.2}$$

then (2.1) *is said to satisfy the strong second order condition (SSOC) at* (x, y^*, z^*).

Next, we briefly review MPCC-LICQ following the notation in [11]. In particular, we consider two index sets $\mathcal{Z}_1$ and $\mathcal{Z}_2$ with $\mathcal{Z}_1, \mathcal{Z}_2 \subset \{1, \cdots, k\}$ and denote their respective complement in $\{1, \cdots, k\}$ by $\mathcal{Z}_1^\perp$ and $\mathcal{Z}_2^\perp$. For any pair $\mathcal{Z}_1, \mathcal{Z}_2$ we define the relaxed NLP corresponding to the MPCC (1.2) as

$$\begin{array}{lll} \underset{x,y,z}{\text{minimize}} & F(x, y) & \\ \text{subject to} & \nabla_y \mathcal{L}(x, y, z) & = 0 \\ & h_i(x, y) & = 0 \quad \forall i \in \mathcal{Z}_1^\perp \\ & z_i & = 0 \quad \forall i \in \mathcal{Z}_2^\perp \\ & h_i(x, y) & \geq 0 \quad \forall i \in \mathcal{Z}_1 \\ & z_i & \geq 0 \quad \forall i \in \mathcal{Z}_2. \end{array} \tag{2.3}$$

The term relaxed NLP stems from the fact that if (x^*, y^*, z^*) is a solution of (2.3) and, in addition, $h(x^*, y^*)^T z^* = 0$ is satisfied, then it is a solution of the original problem (1.2). We further define the set of biactive (or second-level degenerate) components by $\mathcal{D} := \mathcal{Z}_1 \cap \mathcal{Z}_2$. This implies that $(\mathcal{Z}_1^\perp, \mathcal{Z}_2^\perp, \mathcal{D})$ is a partition of $\{1, \cdots, k\}$. A solution (x^*, y^*) is said to be second-level nondegenerate if $\mathcal{D} = \emptyset$. We now restate the extension of LICQ to MPCCs.

DEFINITION 2.3. *Let* $z \geq 0$ *and* x, y *be such that* $h(x, y) \geq 0$, *and define index sets*

$$\mathcal{Z}_1 = \{i \in \{1, \cdots, k\} \mid z_i = 0\} \quad \text{and} \quad \mathcal{Z}_2 = \{i \in \{1, \cdots, k\} \mid h_i(x, y) = 0\}. \tag{2.4}$$

The MPCC (1.2) *is said to satisfy* MPCC-LICQ *at* (x^*, y^*) *if the corresponding relaxed NLP* (2.3) *satisfies LICQ at* (x^*, y^*).

In the remainder of this paper we refer to the problem

$$\begin{array}{lll} \underset{x,y,z}{\text{minimize}} & F(x, y) & \\ \text{subject to} & \nabla_y \mathcal{L}(x, y, z) & = 0 \\ & h(x, y) & \geq 0 \\ & z & \geq 0 \\ & h(x, y)^T z & \leq 0, \end{array} \tag{2.5}$$

as the NLP formulation corresponding to MPCC (1.2). Problem (2.5) is formulated without slacks to keep the notation simple. A reformulation of (2.5) using slacks does not change the results derived in this paper. Please note that when solving an MPCC of type (2.5), slacks should be used to achieve a better convergence behavior and to maintain linear feasibility [11].

2.1. A Motivating Example. The following example demonstrates that the MPCC (1.2) in general does not satisfy MPCC-LICQ even if the lower-level problem of (1.1) satisfies LICQ and SSOC. Consider the following example:

$$\begin{array}{lll} \underset{x}{\text{minimize}} & -x + 2y_1 + y_2 & \\ \text{subject to} & \underset{y:=(y_1,y_2)}{\text{minimize}} & (x-y_1)^2 + y_2^2 \\ & \text{subject to} & y_1, y_2 \geq 0, \end{array} \tag{2.6}$$

with the solution $(x^*, y_1^*, y_2^*) = (0, 0, 0)$. LICQ and SSOC are satisfied for the lower level problem. The gradient of the Lagrangian of the lower level problem is given by

$$\nabla_y \mathcal{L}(x, y, z) = \begin{pmatrix} -2x + 2y_1 - z_1 \\ 2y_2 - z_2 \end{pmatrix}, \tag{2.7}$$

and the MPCC corresponding to (2.6) is given by

$$\begin{array}{lll} \underset{x,y,z}{\text{minimize}} & -x + 2y_1 + y_2 & \\ \text{subject to} & 0 & = \begin{pmatrix} -2x + 2y_1 - z_1 \\ 2y_2 - z_2 \end{pmatrix} \\ & 0 & \leq z \perp y \geq \;\; 0. \end{array} \tag{2.8}$$

In the solution, we have $z_1 = z_2 = 0$. This means that all constraints are weakly active at the solution and the relaxed NLP of (2.8) is

$$\begin{array}{lll} \underset{x,y,z}{\text{minimize}} & -x + 2y_1 + y_2 & \\ \text{subject to} & 0 & = \begin{pmatrix} -2x + 2y_1 - z_1 \\ 2y_2 - z_2 \end{pmatrix} \\ & 0 & = y \\ & 0 & = z. \end{array} \tag{2.9}$$

Clearly, (2.9) does not satisfy LICQ because the Jacobian of the constraints (the columns are the constraints normals) given by

$$\begin{pmatrix} 2 & 0 & 1 & 0 & 0 & 0 \\ 0 & 2 & 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & 0 & 0 & 1 \\ -2 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}, \tag{2.10}$$

has rank 5. This fact implies that the MPCC (1.2) does not satisfy MPCC-LICQ. The degeneracy problem is caused by the weakly active constraints. However, if the lower level NLP satisfies strict complementarity, LICQ and SSOC in the solution, then it can be shown that the resulting MPCC satisfies MPCC-LICQ (see [20], or Remark 2.1).

2.2. Lifting Bilevel NLPs. In the previous section, we have shown that the bilevel NLP (1.1) does not inherit MPCC-LICQ even if its lower level satisfies LICQ and SSOC. To fix this problem, we introduce new variables $w \in \mathbb{R}^k$ to define the following lifting of (1.1):

$$\begin{array}{lll} \underset{x,y,w}{\text{minimize}} & F(x,y) + \pi\, p(w) & \\ \text{subject to} & \underset{y}{\text{minimize}} & f(x,y) \\ & \text{subject to} & h(x,y) \geq w, \end{array} \tag{2.11}$$

where $\pi \in \mathbb{R}$ is a constant parameter and $p : \mathbb{R}^k \to \mathbb{R}$ is a function that drives w to zero and ensures convergence to a solution of the original unlifted problem. The function $p(w)$ will be discussed in detail in Section 2.4. However, two possible choices are $p(w) = \|w\|_1$ or $p(w) = \|w\|_2^2$. The inequality constraints of the lower level problem are lifted componentwise. This leads to the MPCC

$$\begin{array}{lll} \underset{x,y,z,w}{\text{minimize}} & F(x,y) + \pi\, p(w) & \\ \text{subject to} & 0 \;\; = \nabla_y \mathcal{L}(x,y,z) & \\ & 0 \;\; \le z \;\perp (h(x,y) - w) \;\ge\; 0. & \end{array} \tag{2.12}$$

Example (2.6) shows that in general, SSOC and LICQ of the lower-level problem are not sufficient to ensure that the MPCC (1.2) satisfies MPCC-LICQ. In addition, we require a second-level nondegeneracy assumption which is not typically satisfied in practice. If there are degenerate lower-level components, then we need $|\mathcal{D}|$ (which is at most equal to k) variables to regularize the Jacobian of the active constraint normals. This observation forms the basis of the following theorem, which also shows that the lifted problem (2.12) satisfies an MPCC-LICQ.

THEOREM 2.1. *If the lower level problem of* (2.11) *satisfies LICQ and SSOC at* (x^*, y^*, z^*, w^*)*, then the MPCC* (2.12) *satisfies MPCC-LICQ at* (x^*, y^*, z^*, w^*).

Proof. We need to show that the relaxed NLP formulation of (2.12) given by

$$\begin{array}{llll} \underset{x,y,z,w}{\text{minimize}} & F(x,y) + \pi\, p(w) & & \\ \text{subject to} & \nabla_y \mathcal{L}(x,y,z) & = 0 & \\ & h_i(x,y) - w_i & = 0 & \forall i \in \mathcal{Z}_1^\perp \\ & z_i & = 0 & \forall i \in \mathcal{Z}_2^\perp \\ & h_i(x,y) - w_i & \ge 0 & \forall i \in \mathcal{Z}_1 \\ & z_i & \ge 0 & \forall i \in \mathcal{Z}_2, \end{array} \tag{2.13}$$

satisfies LICQ (according to Definition 2.3 with $\mathcal{Z}_1 = \{i \mid z_i = 0\}$ and $\mathcal{Z}_2 = \{i \mid h_i(x,y) - w_i = 0\}$). The notation $\tilde{h}_{\mathcal{I}} := (h_i(x,y) - w_i \mid i \in \mathcal{I})$ denotes the subvector of $h(x,y) - w$ whose indices belong to $\mathcal{I}$. We now consider the gradients of the active constraints of (2.13) with respect to all optimization variables, i.e. with respect to y, z, w and x (columns are constraint normals):

$$J := \left(\begin{array}{cc|ccc} \nabla_{yy}\mathcal{L}(x,y,z) & \nabla_y h_{\mathcal{Z}_1^\perp} & & & \nabla_y h_{\mathcal{D}} \\ -\nabla_y h_{\mathcal{Z}_1^\perp}^T & & & & \\ \hline -\nabla_y h_{\mathcal{Z}_2^\perp}^T & & I & & \\ -\nabla_y h_{\mathcal{D}}^T & & & I & \\ & & & & -I \\ \hline & -I & & & \\ & & & & \\ \nabla_{yx}\mathcal{L}(x,y,z) & \nabla_x h_{\mathcal{Z}_1^\perp} & & & \nabla_x h_{\mathcal{D}} \end{array}\right) \tag{2.14}$$

$$=: \left(\begin{array}{c|c} A & B \\ \hline C & D \\ \hline E & F \end{array}\right), \tag{2.15}$$

where we have skipped zero entries and assumed that the identity matrices I are of appropriate size. The columns of (2.14) correspond to the constraints

$$\left(\nabla_y \mathcal{L}(x,y,z), \quad \tilde{h}_{\mathcal{Z}_1^\perp}, \quad z_{\mathcal{Z}_2^\perp}, \quad z_{\mathcal{D}}, \quad \tilde{h}_{\mathcal{D}}\right) \tag{2.16}$$

and the rows of (2.14) correspond to the variables

$$\big(y, \quad z_{\mathcal{Z}_1^\perp}, \quad z_{\mathcal{Z}_2^\perp}, \quad z_{\mathcal{D}}, \quad w_{\mathcal{D}}, \quad w_{\mathcal{Z}_1^\perp}, \quad w_{\mathcal{Z}_2^\perp}, \quad x\big). \tag{2.17}$$

Let $\bar{J}$ be the submatrix of J that consists of blocks A, B, C and D. We want to show that the constraint normals of active constraints of (2.13) (columns of (2.14)) are linearly independent. This statement is equivalent to showing that $\bar{J}$ has full rank, which means that $\bar{J}$ is of rank $(m+k+|\mathcal{D}|)$ (J is of size $(m+k+|\mathcal{D}|+|\mathcal{Z}_1^\perp|+|\mathcal{Z}_2^\perp|+n)\times(m+k+|\mathcal{D}|)$). As shown in e.g. [18], Matrix A has full rank since SSOC and LICQ are assumed to hold for the lower level problem of (2.11). Block D has full rank since it is a diagonal matrix with nonzeros on the diagonal. The submatrix $\bar{J}$ of J has full rank if

$$\left[\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right]\left[\begin{array}{c} q \\ r \end{array}\right] = \left[\begin{array}{c} 0 \\ 0 \end{array}\right] \Longleftrightarrow \left[\begin{array}{c} q \\ r \end{array}\right] = \left[\begin{array}{c} 0 \\ 0 \end{array}\right]. \tag{2.18}$$

We note that we assume that 0 is a scalar, vector or matrix of zeros of appropriate size. Using the Schur complement to perform a block Gaussian elimination by multiplying $\bar{J}$ from the right with the block matrix

$$\left[\begin{array}{c|c} I & 0 \\ \hline -D^{-1}C & D^{-1} \end{array}\right] \tag{2.19}$$

leads to

$$\left[\begin{array}{c|c} S_D & BD^{-1} \\ \hline 0 & D^{-1} \end{array}\right]\left[\begin{array}{c} q \\ r \end{array}\right] = \left[\begin{array}{c} 0 \\ 0 \end{array}\right], \tag{2.20}$$

where $S_D = A - BD^{-1}C$ is the Schur complement of D. Equation (2.20) implies that $r = 0$. It remains to show that $S_D q = 0 \Leftrightarrow q = 0$ which is true if $BD^{-1}C = 0$. The structure of $BD^{-1}C$ is as follows:

$$BDC^{-1} = \left[\begin{array}{c|c|c|c} & & & \nabla_y h_{\mathcal{D}} \\ \hline & & & \end{array}\right]\left[\begin{array}{c|c|c} I & & \\ \hline & I & \\ \hline & & -I \end{array}\right]\left[\begin{array}{c|c} -\nabla_y h^T_{\mathcal{Z}_2^\perp} & \\ \hline -\nabla_y h^T_{\mathcal{D}} & \\ \hline & \end{array}\right] \tag{2.21}$$

$$= \left[\begin{array}{c|c|c|c} & & & -\nabla_y h_{\mathcal{D}} \\ \hline & & & \end{array}\right]\left[\begin{array}{c|c} -\nabla_y h^T_{\mathcal{Z}_2^\perp} & \\ \hline -\nabla_y h^T_{\mathcal{D}} & \\ \hline & \end{array}\right] \tag{2.22}$$

$$= 0, \tag{2.23}$$

which means that the Schur complement of D has full rank, because A has full rank. Thus, (2.18) holds and problem (2.11) satisfies MPCC-LICQ. □

The proof of Theorem 2.1 points out that not even SSOC and LICQ on the lower level of (1.1) are sufficient to show that (1.2) satisfies MPCC-LICQ. The matrix of the active constraint normals (2.14) of the relaxed NLP of (1.2) is of size $(n_y+n_g+k+n_x)\times(n_y+n_g+k+|\mathcal{D}|)$, and for MPCC-LICQ we need this matrix to have rank $(n_y+n_g+k+|\mathcal{D}|)$. This means MPCC-LICQ cannot be achieved without either having assumptions on the number of upper-level variables n_x (which is rather restrictive), or introducing at least $|\mathcal{D}|$ new variables (which lifts the MPCC to a higher dimension).

Remark 2.1. *MPCC-LICQ is inherited without any modifications only if the lower level of* (1.1) *satisfies LICQ and SSOC, and if* $|\mathcal{D}| = \emptyset$, *which means that there*

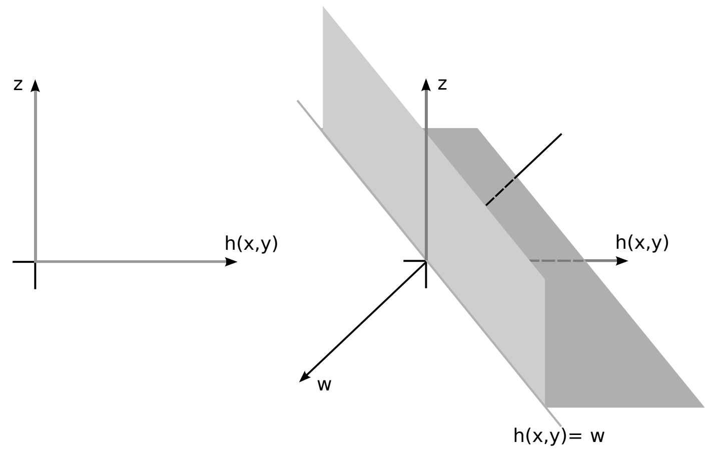

FIGURE 2.1. *The feasible set (in gray) of a complementarity constraint before (left) and after (right) lifting.*

are no degenerate indices. This follows directly from the proof of Theorem 2.1, or can be found in e.g. [20].

Figure 2.1 illustrates how the lifting changes the feasible set of the complementarity constraint (CC). The feasible set of $0 \le z \perp h(x,y) \ge 0$ for $k = 1$ is the nonnegative part of both axes. The feasible set of the lifted CC, $0 \le x \perp (h(x,y) - w) \ge 0$, for $k = 1$ is shown on the right of Figure 2.1. The original feasible set is now extended to a third dimension.

2.3. Lifting the Motivating Example. Let us consider the proposed lifting for Example (2.6):

$$\begin{array}{lll} \underset{x,w:=(w_1,w_2)}{\text{minimize}} & -x + 2y_1 + y_2 + \pi p(w) & \\ \text{subject to} & \underset{y}{\text{minimize}} & (x - y_1)^2 + y_2^2 \\ & \text{subject to} & y_1 \ge w_1 \\ & & y_2 \ge w_2, \end{array} \tag{2.24}$$

where $w := (w_1, w_2) \in \mathbb{R}^2$ are the additional lifting variables. The corresponding MPCC is then given by

$$\begin{array}{lll} \underset{x,y,w,z}{\text{minimize}} & -x + 2y_1 + y_2 + \pi p(w) & \\ \text{subject to} & 0 & = \nabla_y \mathcal{L}(x,y,z) \\ & 0 & \le z \perp y \ge \quad w. \end{array} \tag{2.25}$$

The Jacobian of active constraints of the NLP formulation of (2.25) has now two additional rows for variables w_1 and w_2:

$$\left(\begin{array}{cccccc} 2 & 0 & 1 & 0 & 0 & 0 \\ 0 & 2 & 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & 0 & 0 & 1 \\ -2 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 \end{array} \right), \tag{2.26}$$

which extends (2.10) by two rows that ensure the Jacobian has full rank. After lifting Example (2.6), the active constraint gradients of the NLP formulation are linearly independent and MPCC-LICQ is satisfied for (2.25) everywhere. It remains to discuss how to drive w to zero in order to solve the original unlifted problem.

2.4. Driving w to Zero. In order to solve the original unlifted problem (1.2), we have to ensure that $w_i = 0$, $\forall i$ at the solution. There are several ways to do that. In the following, we discuss an exact and an inexact penalty function for driving w to zero, namely

$$\text{(a)} \quad p(w) = \|w\|_1, \qquad \text{and} \tag{2.27a}$$

$$\text{(b)} \quad p(w) = \|w\|_2^2 \tag{2.27b}$$

The ℓ_1-norm penalty (2.27a) is an exact penalty function. It ensures that we solve the original unlifted problem when choosing π larger than a threshold $\bar{\pi}$ (the exact convergence behavior is discussed in the next section). However, the exact ℓ_1-norm penalty is a nonsmooth function. Instead of using a numerical method which is able to handle the nonsmoothness of the function (as proposed in, e.g., [8]), we add the additional constraint $w \geq 0$ which ensures the componentwise nonnegativity of w and the smoothness of the penalty $\|w\|_1$. For the lifting of the complementarity constraint this means that the we cut the two planes in Figure 2.1 at $w = 0$ and neglect the part with $w < 0$. The proof of Theorem 2.1 and the geometry of the feasible set directly imply that we keep MPCC-LICQ as long as the additional constraint $w_i \geq 0$, $i \in \mathbb{R}^k$ is not active at a point where $h_i(x, y)$ and z_i are zero for $i \in \{1, \cdots, k\}$. Our numerical experiments clearly indicate that having MPCC-LICQ everywhere but points where $h_i(x, y) = z_i = w_i = 0$ still stabilizes the MPCC and leads to a faster convergence. Furthermore, this approach allows us to use standard NLP solvers.

The squared ℓ_2-norm penalty (2.27b) is an analytic function, which is a nice property since most numerical NLP algorithms assume smoothness of the problem. However, with an inexact penalty function we have to solve a sequence of problems where $\pi \to \infty$ in order to ensure to converge to a solution of the original unlifted problem (1.2). This is an undesirable property. However, we decided to also investigate the inexact penalty because our numerical experience clearly shows, that having MPCC-LICQ everywhere decreases the number of iterations. Furthermore, to avoid an ill-conditioned Hessian, reformulations such as the one described in [18, Chapter 17] could be used. Both penalty functions are tested in Section 4, and the convergence properties of penalty (2.27a) are analyzed in the next section.

2.5. Local Convergence Properties. Fletcher et al. [11] provide convergence results for SQP methods applied to MPCCs under relatively mild conditions. One important assumption for the convergence to strongly stationary points is that the MPCC satisfies MPCC-LICQ. This condition keeps the multipliers bounded and ensures a fast local convergence. We now derive an assumption which is weaker than MPCC-LICQ for problems of type (1.2) and for penalty (2.27a). The analysis in [11] centers around the relaxed NLP

$$\begin{array}{lll} \underset{x,y,z}{\text{minimize}} & F(x, y) & \\ \text{subject to} & \nabla_y \mathcal{L}(x, y, z) & = 0 \\ & z_i & = 0 \quad \forall i \in \mathcal{Z}_2^\perp \\ & h_i(x, y) & \geq 0 \quad \forall i \in \mathcal{Z}_1 \\ & z_i & \geq 0 \quad \forall i \in \mathcal{Z}_2. \end{array} \tag{2.28}$$

Without loss of generality and to keep the notation simple, we assume $\mathcal{Z}_1^\perp = \emptyset$. The Lagrangian of (2.28) is given by

$$L(x,y,z,\lambda,\mu,\nu) := F(x,y) - \nabla_y \mathcal{L}(x,y,z)^T\lambda - z^T\mu - h(x,y)^T\nu, \tag{2.29}$$

with Lagrange multipliers $\lambda \in \mathbb{R}^m, \mu =: (\mu_1,\mu_2) \in \mathbb{R}^{|\mathcal{Z}_2^\perp|+|\mathcal{Z}_2|}$ with $\mu_2 \geq 0$ and $\nu \in \mathbb{R}^{|\mathcal{Z}_1|}$ with $\nu \geq 0$. Dual feasibility of (2.28) is given as

$$\begin{aligned}
\nabla_x L(x,y,z,\lambda,\mu,\nu) &= \nabla_x F(x,y) - \nabla_{yx}\mathcal{L}(x,y,z)^T\lambda - \nabla_x h(x,y)^T\nu &= 0\\
\nabla_y L(x,y,z,\lambda,\mu,\nu) &= \nabla_y F(x,y) - \nabla_{yy}\mathcal{L}(x,y,z)^T\lambda - \nabla_y h(x,y)^T\nu &= 0\\
\nabla_z L(x,y,z,\lambda,\mu,\nu) &= -\nabla_y h(x,y)^T\lambda - \mu &= 0.
\end{aligned} \tag{2.30}$$

Combining conditions (2.30), feasibility of (2.28) and complementary slackness

$$h_i(x,y)\nu_i = 0 \;\forall\, i \in \mathcal{Z}_1 \quad \text{and} \quad \mu_{2i} z_i = 0 \;\forall\, i \in \mathcal{Z}_2, \tag{2.31}$$

gives us the KKT conditions of (2.28). Now consider the relaxed lifted NLP

$$\begin{array}{lll}
\underset{x,y,z,w}{\text{minimize}} & F(x,y) + \pi\|w\|_1 & \\
\text{subject to} & \nabla_y\mathcal{L}(x,y,z) = 0 & \\
& z_i = 0 & \forall i \in \mathcal{Z}_2^\perp\\
& h_i(x,y) - w_i \geq 0 & \forall i \in \mathcal{Z}_1\\
& z_i \geq 0 & \forall i \in \mathcal{Z}_2.
\end{array} \tag{2.32}$$

To simplify the analysis, we will replace w by $w = w^+ - w^-$ with $w^+, w^- \geq 0$ and $\|w\|_1 = w^+ + w^-$, giving rise to the following conditions for dual feasibility of (2.32):

$$\begin{aligned}
\nabla_x L(\cdot) &= \nabla_x F(x,y) - \nabla_{yx}\mathcal{L}(x,y,z)^T\lambda - \nabla_x h(x,y)^T\nu &= 0\\
\nabla_y L(\cdot) &= \nabla_y F(x,y) - \nabla_{yy}\mathcal{L}(x,y,z)^T\lambda - \nabla_y h(x,y)^T\nu &= 0\\
\nabla_z L(\cdot) &= -\nabla_y h(x,y)^T\lambda - \mu &= 0\\
\nabla_{w^+} L(\cdot) &= \pi\, e + \nu - \xi^+ &= 0\\
\nabla_{w^-} L(\cdot) &= \pi\, e - \nu - \xi^- &= 0,
\end{aligned} \tag{2.33}$$

where the dependencies of L are skipped for a compact presentation, e denotes a vector of ones of appropriate size, and ξ^+, ξ^- are the Lagrange multipliers for $w^+, w^- \geq 0$. We note, that in order to keep the notation intuitive, L in (2.30) denotes the Lagrangian of (2.28), and L in (2.33) denotes the Lagrangian of (2.32). In the following, (x,y,z) is called a KKT point of (2.28) if the KKT conditions (as described in, e.g., [11]) are satisfied at (x,y,z). We can now state the following two propositions.

Proposition 2.1. *Let (x^*,y^*,z^*) be a KKT point of* (2.28) *and assume that $\pi > \|\nu^*\|_\infty$. Then it follows that (x^*,y^*,z^*,w^*) with $w^* = 0$ is a KKT point of* (2.32).

Proof. Clearly, the first three equations of (2.33) are satisfied at (x^*,y^*,z^*). We also see that $\xi^+ = \pi e + \nu^* \geq 0$ since $\pi, \nu \geq 0$. With $w^* = 0$ we get complementarity for the constraints $w^+, w^- \geq 0$. It remains for us to show that $\xi^- = \pi e - \nu^* \geq 0$, but we choose $\pi > \|\nu^*\|_\infty = \max_i \nu_i^*$, hence $\xi^- \geq 0$. □

Proposition 2.2. *Let (x^*,y^*,z^*,w^*) be a KKT point of* (2.32)*. If $w^* = 0$, then it follows that (x^*,y^*,z^*) is a KKT point of* (2.28).

Proof. Dual feasibility of (2.28) (equation (2.30)) directly follows from (2.33). The same holds true for the nonnegativity of μ_2^* and ν^* in the solution of (2.28). Given that $w^* = 0$, we have primal feasibility and complementarity for the unlifted problem (2.28). □

This result means that if we solve (2.32) with the solution (x^*, y^*, z^*, w^*) and $w^* = 0$, the point (x^*, y^*, z^*) is also a KKT point of (2.28), and we can now apply the convergence analysis from [11]. Furthermore, this means that instead of requiring MPCC-LICQ in the convergence proof of [11], it suffices for MPCCs arising from bilevel NLPs like problem (1.2) to require LICQ and SSOC for the lower-level NLP. LICQ and SSOC are reasonable assumptions which are satisfied for most well-posed problems. Hence, in order to have bounded multipliers and a faster convergence in practice for bilevel NLPs with LICQ and SOSC on the lower level, is suffices to lift the complementarity constraint as described in (2.12) and to choose a π that is sufficiently large. Without lifting the complementarity constraint, we would have to require strict complementarity of the lower-level problem ($|\mathcal{D}| = 0$) in order to guarantee MPCC-LICQ. This is a rather restrictive conditions which is already violated in simple examples like the one described in(2.6).

As discussed in the last section, the ℓ_1-norm is a nonsmooth function, but we wish to use standard SQP methods which usually require the objective and the constraints to be at least twice continuously differentiable. If we now assume that F, g and h are twice continuously differentiable functions, and if we add the additional constraint $w \geq 0$ to (2.32), we obtain the smooth problem

$$\begin{array}{lll} \underset{x,y,z}{\text{minimize}} & F(x,y) + \pi \sum_{i=1}^{k} w_i & \\ \text{subject to} & \nabla_y \mathcal{L}(x,y,z) & = 0 \\ & z_i & = 0 \quad \forall i \in \mathcal{Z}_2^{\perp} \\ & h_i(x,y) - w_i & \geq 0 \quad \forall i \in \mathcal{Z}_1 \\ & z_i & \geq 0 \quad \forall i \in \mathcal{Z}_2 \\ & w & \geq 0 \end{array} \tag{2.34}$$

which is sufficiently smooth for standard SQP methods. The disadvantage of the additional constraint $w \geq 0$ is that we cannot guarantee MPCC-LICQ anymore at points (x, y, z, w) with $h_i(x, y) = z_i = w_i = 0$ for at least one $i \in \{1, \cdots, k\}$. However, in the next section we show that our lifting technique still stabilizes the MPCC and leads to a faster convergence. We now briefly investigate how the convergence results from Propositions 2.1 and 2.2 change for (2.34). The only difference in the proof will be in the dual feasibility conditions (2.33), where the last two equations have to be replaced by

$$\nabla_w L(\cdot) = \pi\, e + \nu - \xi = 0, \tag{2.35}$$

where ξ is the Lagrange multiplier for $w \geq 0$. We have to ensure that $\xi = \pi e + \nu \geq 0$, which is true since $\pi, \nu \geq 0$. This implies that Proposition 2.1 and Proposition 2.2 also hold for problem (2.34). Moreover, the additional assumption on π is not needed anymore.

COROLLARY 2.1. *Let (x^*, y^*, z^*) be a KKT point of* (2.28)*. Then (x^*, y^*, z^*, w^*) is also a KKT point of* (2.34) *with $w^* = 0$. And vice versa, if (x^*, y^*, z^*, w^*) is a KKT point of* (2.34) *and if $w^* = 0$, then (x^*, y^*, z^*) is also a KKT point of* (2.28).

3. Lifting General MPCCs. In this section, we discuss the generalization of our lifting approach to general MPCCs (which do not originate from a bilevel program). In particular, we consider

$$\begin{array}{lll} \underset{x:=(x_1,x_2),y}{\text{minimize}} & F(x,y) & \\ \text{subject to} & 0 & \leq h(x,y) \\ & 0 & \leq x_1 \perp x_2 \geq \ 0, \end{array} \tag{3.1}$$

with $x_1, x_2 \in \mathbb{R}^n$, $y \in \mathbb{R}^m$ and $h(x,y) \in \mathbb{R}^k$. The relaxed NLP of (3.1) is given by

$$
\begin{array}{lll}
\underset{x,y}{\text{minimize}} & F(x,y) & \\
\text{subject to} & 0 & \leq h(x,y) \\
 & 0 & = x_1 \quad \forall i \in \mathcal{Z}_1^{\perp} \\
 & 0 & = x_2 \quad \forall i \in \mathcal{Z}_2^{\perp} \\
 & 0 & \leq x_1 \quad \forall i \in \mathcal{Z}_1 \\
 & 0 & \leq x_2 \quad \forall i \in \mathcal{Z}_2,
\end{array}
\tag{3.2}
$$

with $\mathcal{Z}_1 = \{i \in \{1, \cdots, n\} \mid x_{1i} = 0\}$ and $\mathcal{Z}_2 = \{i \in \{1, \cdots, n\} \mid x_{2i} = 0\}$. Lifting general MPCCs of the form (3.2) leads to the following problem:

$$
\begin{array}{lll}
\underset{w, x := (x_1, x_2), y}{\text{minimize}} & F(x,y) + \pi p(w) & \\
\text{subject to} & 0 & \leq h(x,y) - w \\
 & 0 & \leq x_1 \perp x_2 \geq \; 0.
\end{array}
\tag{3.3}
$$

The relaxed NLP for this MPCC is given by:

$$
\begin{array}{lll}
\underset{x,y,w}{\text{minimize}} & F(x,y) + \pi p(w) & \\
\text{subject to} & 0 & \leq h(x,y) - w \\
 & 0 & = x_1 \quad \forall i \in \mathcal{Z}_1^{\perp} \\
 & 0 & = x_2 \quad \forall i \in \mathcal{Z}_2^{\perp} \\
 & 0 & \leq x_1 \quad \forall i \in \mathcal{Z}_1 \\
 & 0 & \leq x_2 \quad \forall i \in \mathcal{Z}_2,
\end{array}
\tag{3.4}
$$

where the constraints $h(x,y)$ are lifted with variables $w \in \mathbb{R}^k$. The lifting of general MPCCs is not as elegant as in the bilevel case because of the lack of information about the structure of the constraints $h(x,y) \geq 0$. However, lifting still guarantees linearly independent constraint normals. To see this, we consider the Jacobian of active constraints of the lifted relaxed NLP (3.4):

$$
J := \left(
\begin{array}{c|c|c|c|c}
-I & & & & \\ \hline
\nabla_{x_{1\mathcal{Z}_1^{\perp}}} h & I & & & \\ \hline
\nabla_{x_{1\mathcal{D}}} h & & I & & \\ \hline
\nabla_{x_{2\mathcal{Z}_2^{\perp}}} h & & & I & \\ \hline
\nabla_{x_{2\mathcal{D}}} h & & & & I \\ \hline
\nabla_{x_{1\mathcal{Z}_2^{\perp}}} h & & & & \\ \hline
\nabla_{x_{2\mathcal{Z}_1^{\perp}}} h & & & & \\ \hline
\nabla_{y} h & & & &
\end{array}
\right)
$$

where the notation $x_{j\mathcal{I}} := (x_{ji} \mid i \in \mathcal{I})$ denotes the subvector of x_j whose indices belong to $\mathcal{I}$ for $j = 1, 2$. The column of J correspond to the constraints

$$
(h - w, x_{1\mathcal{Z}_1^{\perp}}, x_{1\mathcal{D}}, x_{2\mathcal{Z}_2^{\perp}}, x_{2\mathcal{D}})
\tag{3.5}
$$

and the rows of J correspond to the variables

$$
(w, x_{1\mathcal{Z}_1^{\perp}}, x_{1\mathcal{D}}, x_{2\mathcal{Z}_2^{\perp}}, x_{2\mathcal{D}}, x_{1\mathcal{Z}_2^{\perp}}, x_{2\mathcal{Z}_1^{\perp}}, y).
\tag{3.6}
$$

Zero entries are skipped. Clearly, the Jacobian J has full column rank without any restrictions on the structure of the constraint function $h(x, y)$. More general complementarity constraints of the form

$$0 \leq \ G(x) \ \perp \ H(x) \ \geq 0 \tag{3.7}$$

can easily be treated by introducing slack variables. No further restrictions on the functions G, H are needed, since G and H are then lifted in the same way as the inequalities $h(x, y) \geq 0$. This result is in contrast to [11] where we assumed that parts of $\nabla G, \nabla H$ have full rank. It is straightforward to show that the convergence results for the exact penalty from Section 2.5 remain true for the lifted general MPCC (3.4) with $p(w) = \|w\|_1$. Numerical test are promising and show that lifted MPCCs behave in the same way as lifted bilevel NLPs. The computational results are discussed in detail in the next section.

4. Numerical Results. We test the lifting approach for bilevel NLPs and for general MPCCs described in Sections 2 and 3. We compare the performance without lifting (formulation (2.5)), with the lifted ℓ_1-penalty approach (2.27a) including the additional constraint $w \geq 0$, with the lifted ℓ_2-penalty (2.27b). As test problems we use the `MacMPEC` collection [13] and Example (2.6) described in Section 2.1. We first compare all MPCCs arising from bilevel problems – 47 problems in total, excluding problems that are unchanged by the lifting such as `kth1`.

The performance of `filterSQP` for the 47 bilevel test problems without lifting, and with ℓ_1-lifting using the penalty functions (2.27a) and ℓ_2-lifting (2.27b) is shown in Figure 4.1. The plots can be interpreted as a probability distribution that a solver outperforms all other solvers. The performance profiles are generated as described in [9, 15]. The performance measure is the number of iterations, which is a consistent measure given that all problems have been solved with `filterSQP` in `AMPL`. For each problem $p \in \mathcal{P} := \{1, \cdots, 47\}$ and solver $s \in \mathcal{S}$ with $\mathcal{S} := \{$filterSQP, filterSQP_lift_ℓ_1, filterSQP_lift_$\ell_2\}$, we define

$$t_{p,s} := \text{iterations required to solve problem p by solver s.} \tag{4.1}$$

The performance ratio is defined by

$$r_{p,s} := \frac{t_{p,s}}{\min\{t_{p,s'} : s' \in \mathcal{S}\}}, \tag{4.2}$$

the probability for solver $s \in \mathcal{S}$ that a performance ratio $r_{p,s}$ is within a factor $\tau \in \mathbb{R}$ of the best possible ratio is denoted as

$$\rho_s(\tau) := \frac{1}{|\mathcal{P}|} |\{p \in \mathcal{P} : r_{p,s} \leq \tau\}|.$$

The profile in Figure 4.1 shows τ versus $\rho_s(\tau)$.

Table 4.1 shows the number of iterations, the objective value in the solution and the penalty parameter π for each problem and each approach. The theoretical results from Section 2.2 are confirmed for our motivating Example (2.6). Without lifting, 12 iterations are needed to solve the problem. Lifting with penalty (2.27b) ensures MPCC-LICQ for all $x \in \mathbb{R}, y \in \mathbb{R}$ and $w \in \mathbb{R}^2$ and the number of iterations decreases from 12 to 7 iteration. Using penalty (2.27a) and the additional constraints $w \geq 0$ leads to 8 iterations. The decrease is even large for `ex9.2.2` from 22 to 9 iterations for penalty (2.27b).

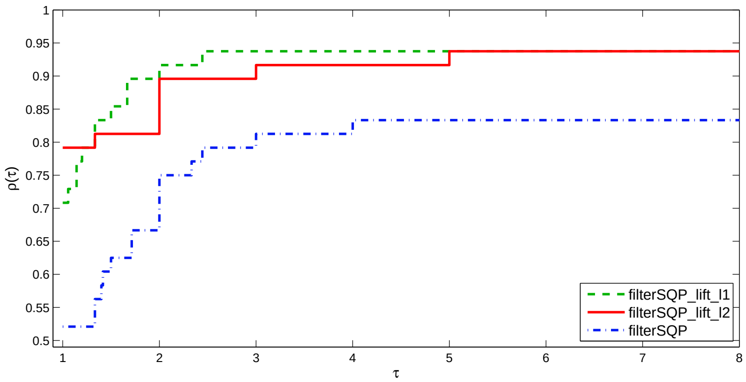

FIGURE 4.1. *Performance profile of filterSQP for bilevel problems from MacMPEC.*

Using the exact penalty (2.27a), for 28 of 48 problems we use $\pi = 1$. The largest π in this setting is $\pi = 20$. For both settings (penalty (2.27a) and penalty (2.27b)), the lifting is not sensitive with respect to the initial value for w. It some cases it helps to choose the initial for w such that $h(x, y) \geq w$ or even $h(x, y) = w$ is satisfied at the initial point. For penalty (2.27b), the largest value of π for driving w to zero is $1E6$. However, this is only the case for 2 of 48 problems. For 12 problems, $\pi = 1$ is sufficient. For most of the remaining problems we use $\pi = 1E4$ or $\pi = 1E5$. Practical experience shows that for lifting with the inexact penalty, it is reasonable to choose π to be in the same order of magnitude as $\max(F(x_0, y_0), \mathrm{NVAR}, 1E4)$, where x_0, y_0 are the initial values for x, y and NVAR is the total number of variables of the problem. For lifting with the exact penalty, $\pi = 1$ is a reasonable choice. If the initial penalty parameter π is not sufficiently large to drive w to zero in the solution, π has to be increased. In Table 4.1, (I) means that the problem is locally infeasible and (ERR) stands for an IEEE error in the AMPL function evaluations. For problem `design-cent2, design-cent3` and `design-cent21`, infeasibility has been detected or an error occurred with and without lifting. For six other problems without lifting the problem is locally infeasible, but after lifting the problem (independent of the penalty we use), `filterSQP` converges to a solution. There are three problems that could not be solved, neither the unlifted nor the lifted problem.

In total, for all bilevel test problems, the number of iterations with lifting has decreased from 273 to 222 iterations for penalty (2.27a), and to 205 for penalty (2.27b) as shown in Figure 4.1 and Table 4.1. Even though we loose MPCC-LICQ when using the exact penalty with $w \geq 0$ for points where $h_i(x, y) = z_i = w_i = 0$ for at least one $i \in \{1, \cdots, k\}$, Figure 4.1 shows that the performance of `filterSQP_lift_l1` is the best. The performance of `filterSQP_lift_l2` is worse than the ℓ_1-lifting, but still better than `filterSQP` without lifting.

We now consider general MPCCs from the `MacMPEC` collection [13] which do not arise from bilevel problem – 118 problems in total. Again, note that if lifting does not change the problem formulation, the problem is skipped. Table 4.2 shows the number of iterations, the objective value in the solution and the penalty parameter π for each problem and each approach, and the performance profile is illustrated in Figure 4.2. The total number of iterations decreases from 1288 for the unlifted problem to 871 for

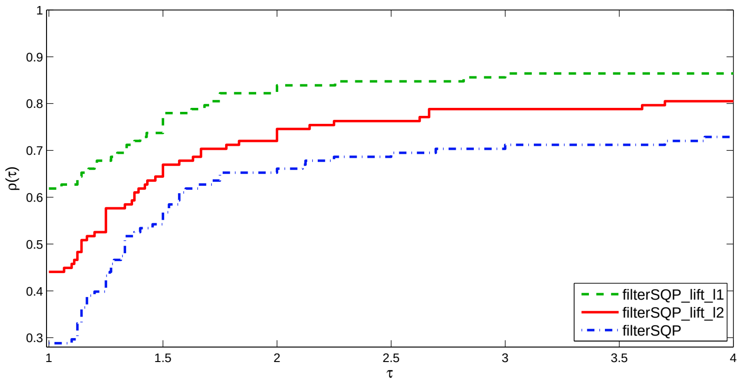

FIGURE 4.2. *Performance profile of filterSQP for general MPCCs from MacMPEC.*

the $\ell1$-lifted problem, and to 1090 for the $\ell2$-lifted problem. The performance profile in Figure 4.2 clearly shows that `filterSQP_lift_l1` performs best. Ten problems could not be solved (neither lifted nor unlifted) because of infeasibility. In Table 4.2, (no conv) means that the trust region becomes too small, ($w \neq 0$) means that we were not able to drive w to zero, and (fail QP) means that the QP solver exits with an error. For fourteen problems, `filterSQP` detected infeasibility in the unlifted problem, but after lifting (independent of the penalty), `filterSQP` converges to a solution. There is one problem of special interest, which is problem `scholtes4`, known for a solution which is a B-stationary point that is not strongly stationary. Without lifting, `filterSQP` detects infeasibility. After lifting, the problem can be solved and `filterSQP` converges to the reported solution. There are two problems, where `filterSQP` without lifting converges, but we cannot get a solution for the lifted problem (independent of the penalty we use).

For a better understanding of the convergence behavior of `filterSQP` applied to MPCCs with and without lifting, we take a closer look at problem `incidset1-8`. Figure 4.3 shows the convergence behavior without lifting and with ℓ_1-lifting, where the constraint violation on the y-axis is plotted on a logarithmic scale. Except for the last iteration, Figure 4.3 indicates a linear rate of convergence for the unlifted problem, whereas with lifting, we obtain a superlinear rate of convergence.

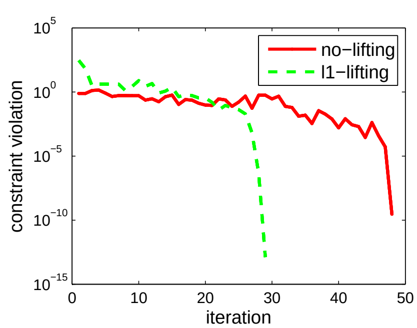

FIGURE 4.3. *Convergence behavior of* `incidset1-8`.

In summary, lifting the complementarity constraint (independent of the penalty function) leads to a more stable method with a better convergence behavior, for bilevel programs, but also for general MPCCs. Practical experience and the performance profiles 4.1 and 4.2 show, that lifting with the exact penalty performs best, and if we converge to a KKT-point of the ℓ_1-lifted problem with $w^* = 0$, we can guarantee that this is also a KKT-point of the original unlifted problem. Furthermore, the lifting proposed in this paper does not require a tailored solver, standard SQP-solver can be used after adapting the problem formulation.

	filterSQP_lift_ℓ_1			filterSQP_lift_ℓ_2			filterSQP	
	iter.	pen.	objf.	iter.	pen.	objf.	iter.	objf.
bard1m	3	3	17	3	1.50E+04	17	3	17
bard2m	3	2	-6598	3	1.00E+04	-6598	3	-6598
bard3m	4	2	-12.68	4	1.00E+04	-12.68	4	-12.68
design-cent-31	94	2	0	89	1.00E+00	0	126	0
desilva	2	1	-1	2	1.00E+00	-1	2	-1
hs044-i	4	1	17.09	4	1.00E+05	17.09	4	17.09
portfl1	6	1	1.50E-005	3	1.00E+04	1.50E-005	6	1.50E-005
portfl2	5	1	1.46E-005	3	1.00E+04	1.46E-005	6	1.46E-005
portfl3	4	1	6.27E-006	4	1.00E+00	6.27E-006	4	6.27E-006
portfl4	4	1	2.18E-006	4	1.00E+02	2.18E-006	4	2.18E-006
portfl6	4	1	2.36E-006	3	4.00E+00	2.36E-006	4	2.36E-006
hs044-i	4	1	17.09	4	1.00E+05	17.09	4	17.09
liswet1-050	1	1	0.16	2	1.00E+05	0.14	1	0.16
liswet1-100	1	1	0.16	3	1.00E+04	0.14	1	0.16
liswet1-200	1	1	0.17	5	1.00E+04	0.15	1	0.17
sl1	1	1	0	1	1.00E+00	0	1	0
bard1	3	3	17	3	1.00E+05	17	3	17
bard2	1	1	6598	1	1.00E+06	6598	1	6598
bard3	2	1	-12.68	2	1.00E+04	-12.68	4	-12.68
bilevel1	3	3	0	3	1.00E+05	0	4	-60
bilevel2	2	1	-6600	2	1.00E+06	-6600	6	-6600
bilevel3	6	5	-20	5	1.00E+05	-20	7	-15.82
ex9.1.1	1	1	-13	1	1.00E+00	-13	1	-13
ex9.1.3	2	20	-29.2	2	1.00E+05	-29.2	2	-29.2
ex9.1.4	1	1	-37	2	1.00E+04	-37	4	-37
ex9.1.5	3	1	-1	2	1.00E+00	-1	3	-1
ex9.1.6	2	3	-21	2	1.00E+05	-21	2	-21
ex9.1.7	5	15	-23	6	1.00E+05	-23	3	-23
ex9.1.8	2	1	-3.25	2	1.00E+04	-3.25	2	-3.25
ex9.1.9	2	2	3.11	2	1.00E+04	3.11	2	3.11
ex9.1.10	1	1	-3.25	2	1.00E+04	-3.25	2	-3.25
ex9.2.1	1	1	2	1	1.00E+05	2	1	2
ex9.2.2	22	7	100	9	1.00E+05	100	22	100
ex9.2.4	3	1	0.5	3	1.00E+00	0.5	3	0.5
ex9.2.5	3	10	9	3	1.00E+00	9	7	9
ex9.2.6	3	1	-1	3	1.00E+00	-1	3	-1
ex9.2.7	2	10	17	2	1.00E+05	17	2	17

ex9.2.8	3	1	-1.5	3	1.00E+00	-1.5	3	-1.5
Example (2.6)	8	1	0	7	1.00E+03	0	12	0
ex9.2.3	3	1	5	3	1.00E+05	5	(I)	
ex9.2.9	1	1	2	1	1.00E+00	2	(I)	
bilin	3	8	5.6	4	1.00E+04	5.6	(I)	
design-cent-1	5	2	1.86	5	1.00E+04	1.86	(I)	
design-cent-4	1	1	0	1	1.00E+00	0	(I)	
design-cent-2	(I)			(I)			(I)	
design-cent-3	(ERR)			(ERR)			(ERR)	
design-cent-21	(I)			(I)			(I)	

Table 4.1: Numerical results for bilevel problems from MacMPEC.

	filterSQP_lift_ℓ_1			filterSQP_lift_ℓ_2			filterSQP	
bar-truss	13	1000	10166.6	37	1.00E+06	10166.6	10	10166.6
dempe	39	1	28.25	8	1.00E+05	49	58	28.25
df-1	2	1	0	2	1.00E+00	0	2	0
flp2	1	1	0	1	1.00E+00	0	1	0
flp4-1	3	1	0	3	1.00E+00	0	3	0
flp4-2	3	1	0	3	1.00E+00	0	3	0
flp4-3	3	1	0	3	1.00E+00	0	3	0
flp4-4	3	1	0	3	1.00E+00	0	3	0
gnash10	7	1	-230.82	7	1.00E+05	-230.82	8	-230.82
gnash11	7	1	-129.91	7	1.00E+05	-129.91	7	-129.91
gnash12	8	1	-36.93	7	1.00E+05	-36.93	8	-36.93
gnash13	8	1	-7.06	8	1.00E+05	-7.06	12	-7.06
gnash14	14	1	-0.18	8	1.00E+04	-0.18	24	-0.18
gnash15	9	100	-354.7	8	1.00E+05	-354.7	17	-354.7
gnash16	12	1	-241.44	7	1.00E+05	-241.44	14	-241.44
gnash17	8	1	-90.75	9	1.00E+05	-90.75	12	-90.75
gnash18	14	100	-25.7	8	1.00E+05	-25.7	14	-25.7
gnash19	7	1	-6.12	8	1.00E+05	-6.12	9	-6.12
gnash10m	7	1	-230.82	8	1.00E+05	-230.82	11	-230.82
gnash11m	7	1	-129.91	7	1.00E+05	-129.91	12	-129.91
gnash12m	8	1	-36.93	9	1.00E+05	-36.93	10	-36.93
gnash13m	8	1	-7.06	8	1.00E+04	-7.06	11	-7.06
gnash14m	8	1	-0.18	8	1.00E+04	-0.18	31	-0.18
hakonsen	10	1	24.37	10	1.00E+04	24.37	10	24.37
incid-set1-8	19	1	0.23	27	1.00E+00	0	29	0.23
incid-set1-16	40	1	0.17	39	1.00E+00	0	28	0.17
incid-set1c-8	23	1	0.23	19	9.00E+00	0	29	0.23
incid-set1c-16	37	1	0.17	22	1.00E+00	0	28	0.17
incid-set2-32	1	1	0	1	1.00E+00	0	169	0.02
monteiro	8	1000	37.53	10	1.00E+06	38.25	9	37.53
monteiroB	8	1000	827.86	10	1.00E+06	828.04	9	827.86

nash1	1	1	0	1	1.00E+00	0	5	0
outrata31	7	3	3.21	7	1.00E+04	3.21	8	3.21
outrata32	7	2	3.45	7	1.00E+04	3.45	8	3.45
outrata33	6	2	4.6	6	1.00E+04	4.6	7	4.6
outrata34	6	1	6.59	6	1.00E+04	6.59	6	6.59
pack-comb1-8	5	1	0.6	6	1.00E+03	0.6	8	0.6
pack-comb1-16	9	110	0.62	10	1.00E+03	0.6	19	0.62
pack-comb1c-8	6	1	0.6	8	1.00E+03	0.6	8	0.6
pack-comb1c-16	7	27	0.62	10	1.00E+03	0.82	5	0.62
pack-comb2-8	8	14	0.67	10	1.00E+05	0.67	8	0.67
pack-comb2-16	10	30	0.73	36	1.00E+05	0.72	37	0.73
pack-comb2c-8	9	15	0.66	9	1.00E+04	0.67	6	0.67
pack-comb2c-16	10	30	0.73	10	1.00E+04	0.7	15	0.73
pack-rig1-4	8	2	0.72	10	1.00E+05	0.72	8	0.72
pack-rig1-8	14	3	0.79	11	1.00E+05	0.79	14	0.79
pack-rig1-16	27	3	0.83	23	1.00E+05	0.82	62	0.83
pack-rig1c-4	5	10	0.72	5	1.00E+05	0.72	6	0.72
pack-rig1c-8	8	10	0.79	8	1.00E+05	0.79	9	0.79
pack-rig1c-16	7	1	0.83	11	1.00E+05	0.82	11	0.83
pack-rig1p-4	6	1	0.6	7	1.00E+04	0.6	7	0.6
pack-rig1p-8	16	3	35.94	22	1.00E+06	0.73	12	35.94
pack-rig1p-16	27	7	264.48	27	1.00E+06	0.72	18	264.5
pack-rig2-4	8	2	0.69	11	1.00E+05	0.69	9	0.69
pack-rig2-8	9	6	0.77	16	1.00E+05	0.78	9	0.78
pack-rig2c-4	4	2	0.71	8	1.00E+05	0.71	7	0.71
pack-rig2c-8	7	8	0.8	8	5.00E+05		7	0.8
pack-rig2p-4	6	1	0.6	6	1.00E+04	0.6	6	0.6
pack-rig2p-8	15	6	46.68	22	1.00E+07	0.76	19	46.68
pack-rig2p-16	19	400	625.94	31	1.00E+06	-14.44	25	625.93
qpec-100-1	6	2	0.1	5	1.00E+04	0.1	7	0.1
qpec-100-2	6	3	-6.59	9	1.00E+05	-3.84	7	-6.26
qpec-100-3	4	2	-5.48	5	1.00E+05	-5.38	5	-5.48
qpec-100-4	4	5	-4.06	4	1.00E+05	-1.44	5	-3.6
qpec-200-1	4	5	-1.93	9	1.00E+05	-1.93	10	-1.94
qpec-200-2	11	3	-23.88	15	1.00E+05	-22.69	11	-24.04
qpec-200-3	10	3	-1.92	11	1.00E+06	-1.93	11	-1.95
qpec-200-4	4	3	-6.04	5	1.00E+06	-6.04	5	-6.22
ralphmod	59	1	-683.03	44	1.00E+01	-683.03	64	-683.03
scholtes1	3	1	2	3	1.00E+04	2	4	2
scholtes2	2	4	15	2	1.00E+07	15	2	15
stackelberg1	4	1	-3266.67	4	1.00E+06	-3266.67	4	-3266.67
tap-09	8	1	109.15	11	1.00E+04	109.15	8	109.13
TraficSignalCycle-1	3	1	56.73	8	1.00E+05	54.96	4	56.73
TraficSignalCycle-2	3	1	54.34	8	1.00E+04	52.57	4	54.34
TraficSignalCycle-3	2	1	88.84	16	1.00E+04	87.07	1	88.84
TraficSignalCycle-4	4	1	80.81	31	1.00E+05	79.04	9	80.84
TraficSignalCycle-5	3	1	103.24	25	1.00E+05	101.47	1	103.24

TraficSignalCycle-6	3	1	103.3	26	1.00E+04	101.53	3	103.3
TraficSignalCycle-9	3	1	54.98	6	1.00E+04	53.2	3	54.98
TraficSignalCycle-10	3	1	56.57	15	1.00E+04	54.8	3	56.57
TraficSignalCycle-11	2	1	103.34	20	1.00E+04	101.57	1	103.34
TraficSignalCycle-13	3	1	88.17	5	1.00E+04	86.4	4	88.17
water-net	95	5	974.39	136	1.00E+07	918.43	149	918.36
b-pn2	33	1	0.09	55	1.00E+06	1020.93	(I)	
gauvin	3	7	20	3	1.00E+04	20	(I)	
gnash15m	12	1	-354.7	8	1.00E+04	-354.7	(I)	
gnash16m	9	1	-241.44	8	1.00E+05	-241.44	(I)	
gnash17m	12	1	-90.75	8	1.00E+05	-90.75	(I)	
gnash18m	13	1	-25.7	8	1.00E+04	-25.7	(I)	
gnash19m	11	1	-6.12	8	1.00E+04	-6.12	(I)	
scholtes4	19	1	0	18	1.00E+04	0	(I)	
taxmcp	15	11	0.82	16	1.00E+04	0.82	(I)	
incid-set1-32	115	1	0.15	51	1.00E+00	0	(I)	
incid-set1c-32	56	1	0.15	147	1.00E+00	0	(I)	
incid-set2-8	1	1	0	1	1.00E+00	0	(I)	
incid-set2-16	1	1	0	1	1.00E+00	0	(I)	
incid-set2c-8	21	1	0.02	45	1.00E+04	0.03	(I)	
incid-set2c-16	131	100	0	(I)			(I)	
incid-set2c-32	(I)			(no conv)			(I)	
TraficSignalCycle-7	(I)			(w $\neq$ 0)			(I)	
TraficSignalCycle-8	(I)			(w $\neq$ 0)			(I)	
TraficSignalCycle-12	(I)			(w $\neq$ 0)			(I)	
pack-comb1-32	(no conv)			(I)			(I)	
pack-comb2-32	(fail QP)			(fail QP)			(I)	
pack-comb2c-32	(fail QP)			(I)			(I)	
pack-comb1c-32	(no conv)			(no conv)			(I)	
pack-rig2-16	(I)			28	1.00E+05	0.83	(I)	
pack-rig2-32	(I)			145	1.00E+04	0.74	(I)	
pack-rig2c-16	(I)			13	5.00E+05	0.92	(I)	
pack-rig2c-32	(I)			(I)			(I)	
tap-15	(I)			(I)			(I)	
pack-rig1-32	(no conv)			(no conv)			40	0.85
pack-rig1c-32	18	10	0.85	(I)			12	0.85
pack-rig1p-32	61	100	2242.07	(I)			101	2242.06
pack-rig2p-32	(I)			(I)			23	871.75
tollmpec	31	100	208.26	(w $\neq$ 0)			11	208.26
tollmpec1	19	100	979.39	(w $\neq$ 0)			(I)	

Table 4.2: Numerical results for general MPCCs from MacMPEC.

5. Conclusion. We have shown that bilevel nonlinear programs whose lower level problem satisfies LICQ and SSOC do not in general inherit MPCC-LICQ to the MPCC that results from replacing the lower level NLP by its first order optimality conditions. MPCC-LICQ is inherited if strict complementarity is satisfied for the lower level problem. However, this is a rather restrictive condition which is not satisfied for many practical problems. A lifting technique has been presented to fix this lack of MPCC-LICQ. Componentwise lifting of the lower level inequality constraints leads to a well-behaved feasible set for which MPCC-LICQ can be guaranteed everywhere. Two penalty approaches for driving the additional lifting variables to zero in the solution have been discussed. Furthermore, convergence results for the exact penalty approach and a generalization of the lifting technique for general MPCCs have been provided. The proposed method has been tested for all problems from the `MacMPEC` test set using `filterSQP`. The numerical results have clearly shown that lifting leads to a better convergence behavior for both penalty approaches.

Acknowledgments. The first author gratefully acknowledges the financial support provided by the Heidelberg Graduate School of Mathematical and Computational Methods for the Sciences at the IWR Heidelberg, funded by DFG (Deutsche Forschungsgemeinschaft). The first and second author were partially supported by the Office of Advanced Scientific Computing Research, Office of Science, U.S. Department of Energy, under Contract DE-AC02-06CH11357 and by the U.S. Department of Energy through grant DE-FG02-05ER25694.

REFERENCES

[1] G.A. Allende and G. Still, *Solving bilevel programs with the KKT-approach*, Mathematical Programming, 138, 1-2, 309 – 332, 2013.

[2] M. Anitescu, *On Using the Elastic Mode in Nonlinear Programming Approaches to Mathematical Programs with Complementarity Constraints*, SIAM Journal on Optimization, 15, 4, 1203 – 1236, 2005.

[3] M. Anitescu, P. Tseng and S. Wright, *Elastic-mode algorithms for mathematical programs with equilibrium constraints: global convergence and stationarity properties*, Mathematical Programming, Springer Berlin / Heidelberg, 110, 2, 337–371, 2007.

[4] J.F. Bard, *Convex two-level optimization*, Mathematical Programming, 40, 15 – 27, 1988.

[5] Y. Chen and M. Florian, *The nonlinear bilevel programming problem: Formulations, regularity and optimality conditions*, Optimization, 32, 193 – 209, 1995.

[6] S. Dempe and A.B. Zemkoho, *The bilevel programming problem: reformulations, constraint qualifications and optimality conditions*, Mathematical Programming, 138, 1-2, 447 – 473, 2013.

[7] A.-V. DeMiguel, M. P. Friedlander, F. J. Nogales and S. Scholtes, *A two-sided relaxation scheme for mathematical programs with equilibrium constraints*, SIAM Journal on Optimization, 16, 587-609, 2005.

[8] A.F. Izmailov, A.L. Pogosyan and M.V. Solodov, *Semismooth SQP method for equality-constrained optimization problems with an application to the lifted reformulation of mathematical programs with complementarity constraints*, Optimization Methods and Software, 26, 847-872, 2011.

[9] Elizabeth D. Dolan and Jorge J. Moré *Benchmarking Optimization Software with Performance Profiles* Mathematical Programming, 91, 201–213, 2002.

[10] R. Fletcher, S. Leyffer, D. Ralph and S. Scholtes, *Numerical experience with solving MPECs as NLPs*, Report NA/210, University of Dundee, 2002.

[11] 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, 259 – 286, 2006.

[12] T. Hoheisel, C. Kanzow, and A. Schwartz, *Theoretical and numerical comparison of relaxation methods for mathematical programs with complementarity constraints*, Preprint 299, University of Würzburg, 2010.

[13] S. Leyffer, *MacMPEC: AMPL collection of MPECs*, available online from www.mcs.anl.gov/leyffer/MacMPEC/, 2000.

[14] S. Leyffer, G. López-Calva and J. Nocedal, *Interior Methods for Mathematical Programs with Complementarity Constraints*, SIAM Journal on Optimization, 17, 52–77, 2006.

[15] Ashutosh Mahajan, Sven Leyffer, and Christian Kirches *Solving Mixed-Integer Nonlinear Programs by QP-Diving* Preprint ANL/MCS-2071-0312, Argonne National Laboratory, Mathematics and Computer Science Division, March 2012.

[16] G. H. Lin, M. Fukushima, *New relaxation method for mathematical programs with complementarity constraints*, Journal of Optimization Theory and Applications, 118, 81—116, 2003.

[17] Z.-Q. Luo, J.-S. Pang, and D. Ralph, *Piecewise sequential quadratic programming for mathematical programs with nonlinear complementarity constraints*, in Multilevel Optimization: Algorithms, Complexity and Applications, A. Migdalas, P. Pardalos, and P. Värbrand, eds., Kluwer Academic Publishers, 209-–229, 1998.

[18] J. Nocedal and S.J. Wright, *Numerical Optimization*, Springer Verlag, 2006.

[19] J. Outrata, M. Kocvara and J. Zowe, *Nonsmooth Approach to Optimization Problems with Equilibrium Constraints: Theory, Applications and Numerical Results*, Kluwer Academic Publishers, Dordrecht, 1998.

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

[21] A.U. Raghunathan, L.T. Biegler, *Interior point methods for Mathematical Programs with Complementarity Constraints (MPCCs)*, SIAM Journal on Optimization, 15, 720–750, 2005.

[22] H. Scheel and S. Scholtes, *Mathematical program with complementarity constraints: Stationarity, optimality and sensitivity*, Math. Oper. Res., 25, 1 – 22, 2000.

[23] S. Scholtes, *Convergence properties of regularization schemes for mathematical programs with complementarity constraints*, SIAM Journal on Optimization, 11, 918 – 936, 2001.

[24] O. Stein, *Lifting mathematical programs with complementarity constraints* Mathematical Programming, Springer Berlin / Heidelberg, 131, 1, 71 – 94, 2012.

[25] S. Steffensen and M. Ulbrich, *A New Relaxation Scheme for Mathematical Programs with Equilibrium Constraints*, SIAM Journal on Optimization, 20, 2504–2539, 2010.