Computational Optimization and Applications, 30, 211–227, 2005

A Trust-Region Method for Nonlinear Bilevel Programming: Algorithm and Computational Experience

BENOÎT COLSON benoit.colson@samcef.com
SAMTECH S.A., B-4031 Liège, Belgium (Formerly: Dept. of Mathematics, FUNDP, B-5000, Namur, Belgium)

PATRICE MARCOTTE marcotte@iro.umontreal.ca
CRT and DIRO, Université de Montréal, CP 6128, Succursale Centre-Ville, Montréal, Québec, Canada H3C 3J7

GILLES SAVARD gilles@crt.umontreal.ca
Département de Mathématiques et de génie industriel and GERAD, Ecole Polytechnique de Montréal, CP 6079, Succursale Centre-Ville, Montréal, Québec, Canada H3C 3A7

Received July 23, 2002; Revised April 28, 2004; Accepted April 28, 2004

Abstract. We consider the approximation of nonlinear bilevel mathematical programs by solvable programs of the same type, i.e., bilevel programs involving linear approximations of the upper-level objective and all constraint-defining functions, as well as a quadratic approximation of the lower-level objective. We describe the main features of the algorithm and the resulting software. Numerical experiments tend to confirm the promising behavior of the method.

Keywords: bilevel programming, nonlinear programming, trust-region methods, approximation, numerical results

1. Introduction

Bilevel programs, which are closely related to leader-follower (Stackelberg) games in economics, describe hierarchical structures that occur naturally in real-world situations: management (network facility location, environmental regulation, credit allocation, energy policy), economic planning (social and agricultural policies, electric power pricing, oil production), engineering design (optimal structure and shape) and optimal control, to cite but a few (see Section 2 in [50], Section 4 in [73] and Part III in [8]). Another important area of application is transportation planning and modeling (see [57] for a review). Mathematically, the general problem we address can be stated as follows:

$$\min_{x_1,x_2} \quad F(x_1,x_2) \tag{1a}$$

$$\text{s.t.} \quad G(x_1,x_2) \le 0 \tag{1b}$$

$$x_2 \in \arg\min_{x_2'} \; f(x_1,x_2') \tag{1c}$$

$$\text{s.t.} \quad g(x_1,x_2') \le 0. \tag{1d}$$

Our algorithm is designed for the case where G depends solely on the upper-level variable x_1 and where F: $\mathbb{R}^{n_1 \times n_2} \to \mathbb{R}$, f: $\mathbb{R}^{n_1 \times n_2} \to \mathbb{R}$, G: $\mathbb{R}^{n_1 \times n_2} \to \mathbb{R}^{m_1}$, g: $\mathbb{R}^{n_1 \times n_2} \to \mathbb{R}^{m_2}$ are twice continuously differentiable functions. We assume that the constraint sets defined by the functions G and g are regular, i.e., some constraint qualification is satisfied. We denote by Ω the set of feasible points for the bilevel program:

$$\Omega = \{(x_1, x_2)\colon G(x_1, x_2) \leq 0 \quad \text{and} \quad g(x_1, x_2) \leq 0\}.$$

The main feature of problem (1) is that it involves two mathematical programs, the solution of one being part of the constraints of the other. Due to the implicit dependence of the lower-level solution on the upper-level decision vector, bilevel programs are intrinsically hard, and the early contributions in the literature are limited to particular subclasses possessing properties such as linearity or bilinearity, which allow the development of efficient approaches.

Also note that there are two possible approaches when considering bilevel programming problems, namely the *optimistic* (or *weak*) approach and the *pessimistic* (or *strong*) one. The former assumes that, whenever the *reaction set*

$$R(\bar{x}_1) = \{x_2 \in \mathbb{R}^{n_2}\colon x_2 \in \operatorname{argmin}\{f(\bar{x}_1, \hat{x}_2)\colon \hat{x}_2 \in \Omega(\bar{x}_1)\}\},$$

is not a singleton, the leader selects an element in the lower-level feasible set that suits him best. Conversely, the latter approach deals with the situation where cooperation of the leader and the follower is not allowed or impossible, and as a result the leader is not able to influence the follower anymore; all he can do is to bound the damage resulting from an undesirable selection of the follower (for a more complete discussion on these issues, we refer the reader to Loridan and Morgan [55] and Chapter 5, Section 5.1 in Dempe [29]).

The algorithm presented in this paper is based on the optimistic approach. More precisely, our approach consists in approximating a nonlinear bilevel program by a solvable problem of the same type, namely, a bilevel program of the linear-quadratic type. Although dissimilar on the surface, our technique is somewhat related to the trust-region approach developed by Scholtes and Stöhr [64] for solving mathematical programs with equilibrium constraints. However, unlike their algorithm, ours preserves the original non-convex bilevel structure of the original program. One key feature of the method is that it combines two ways of thought in mathematical optimization, namely combinatorial optimization (mixed-integer programming) and nonlinear programming (trust regions, descent). The convergence process is monitored by a trust-region mechanism which ensures that any progress obtained while minimizing the linear-quadratic model is compared to that achieved for the original problem. This allows, as new approximations are computed, efficient control and refinement of the region under consideration.

This paper is organized as follows. First, we provide a short survey of existing work related to bilevel programming and more specifically to its nonlinear instances. This is the subject of Section 2 and is based on previous work by the first author [25]. In Section 3 we recall the basic trust-region framework before we describe our algorithm for solving nonlinear bilevel programming problems. Implementation details are provided in Section 4 together with an analysis of the numerical results. We conclude by discussing extensions of our methodology to mathematical programs with equilibrium constraints.

2. Bilevel programming: A short bibliographical review

The usefulness of bilevel programming in modeling hierarchical decision processes and engineering design problems prompted researchers, in the nineteen eighties, to pay close attention to bilevel programs. Their formulation can be traced back to the 1973 article by Bracken and McGill [16], while Candler and Norton [22] were the first to use the terms bilevel and multilevel programming. A bibliographical review on the subject was written by Kolstad [50] as early as 1985, while the most recent surveys are those of Vicente and Calamai [73] and Dempe [30]. Other reference works can be found in the papers of Anandalingam and Friesz [4] and the books of Shimizu et al. [66], Bard [8] and Dempe [29].

Bilevel programming problems are difficult to solve. Even the simplest instance—in which all functions are linear—is strongly NP-hard, as was shown by Hansen et al. [42]. More recently, Vicente et al. [75] showed that checking local optimality for a linear bilevel program is also NP-hard. It is therefore no surprise that most algorithmic research to date has focused on the simplest cases of bilevel programs, that is problems having nice properties such as linear, quadratic or convex objective and/or constraint functions (see Bard [7] for the study of the convex case). In particular, *linear bilevel problems* have received much attention for the last two decades. Linear bilevel programs have the property (see Savard [61]) that the solution set, whenever it is nonempty, contains at least one vertex of the polyhedron.

$$\Omega = \{(x_1, x_2)\colon G(x_1, x_2) \leq 0 \quad \text{and} \quad g(x_1, x_2) \leq 0\}.$$

A wide class of methods is based on vertex enumeration (see e.g. Candler and Townsley [23], Bialas and Karwan [14], Tuy et al. [72]). Other approaches are based on replacing the lower-level problem by its primal-dual optimality conditions. This operation reduces the original problem to a single-level program involving disjunctive constraints. Other branch-and-bound algorithms have been developed by Bard and Falk [9] and Fortuny-Amat and McCarl [39]. Combining branch-and-bound, monotonicity principles and penalization, Hansen et al. [42] have developed a code capable of solving medium-size instances of linear bilevel programs. Alternatively, Bialas et al. [15] and Júdice and Faustino [48, 49] based their solution approach on complementary pivoting theory. Comprehensive reviews related to that particular class of problems may be found in Hsu and Wen [43, 76] and Ben-Ayed [11].

Several algorithms have been proposed for *nonlinear bilevel programming*, under various assumptions. Roughly speaking, methods for solving the general nonlinear case may belong to one of three categories: enumeration (branch-and-bound), descent and penalty methods.

The abovementioned *branch-and-bound* approach can be extended to cover the situation where the lower-level objective is quadratic in x_2 and the lower-level constraints are linear in x_2. In this case, one replaces the lower-level problem by the equivalent Karush-Kuhn-Tucker system, whose equalities and inequalities are linear, with the exception of the complementarity constraint. The latter being a difficult type of constraint to deal with, branch-and-bound methods may be used to defer the introduction of such constraints in the solution process for as long as possible. This is the basic idea underlying the approaches of Edmunds and Bard [32], Al-Khayyal et al. [3] and more recently Thoai, Yamamoto and Yoshise [71].

Some authors have proposed *descent methods* for solving bilevel programs. Assuming that for any x_1 the optimal solution x_2^* of the lower-level problem is unique and defines x_2 as an implicit function of x_1, problem (1) may be viewed solely in terms of the upper-level variables $x_1 \in \mathbb{R}^{n_1}$. Given a feasible point x_1, an attempt is made to find a direction $d \in \mathbb{R}^{n_1}$ along which the upper-level objective decreases. A new point $x_1 + \alpha d$ ($\alpha > 0$) is computed so as to ensure a reasonable decrease in F while maintaining feasibility for the bilevel problem. However, a major issue is the availability of the gradient (or a generalized gradient) of the upper-level objective at a feasible point. While Kolstad and Lasdon [51] have proposed a method for approximating this term, a slightly different approach is that of Savard and Gauvin [62], for problems where no upper-level constraints of the type (1b) are present and where the lower-level constraints are rewritten in the following way:

$$\begin{aligned} g_i(x_1, x_2) &\leq 0, \quad i \in I, \\ g_j(x_1, x_2) &= 0, \quad j \in J. \end{aligned}$$

They first show that an upper-level descent direction at a given point x_1 is a vector $d \in \mathbb{R}^{n_1}$ such that

$$\nabla_{x_1} F(x_1, x_2^*)\, d + \nabla_{x_2} F(x_1, x_2^*)\, w(x_1, d) < 0, \tag{2}$$

where $w \in \mathbb{R}^{n_2}$ is a solution of the program

$$\begin{array}{ll} \min\limits_{w} & (d^T, w^T)\, \nabla^2_{x_1 x_2} \mathcal{L}(x_1, x_2^*, \lambda)\, (d, w) \\ \text{s.t.} & \nabla_{x_2} g_i(x_1, x_2^*)\, w \leq -\nabla_{x_1} g_i(x_1, x_2^*)\, d, \quad i \in I\,(x_1), \\ & \nabla_{x_2} g_j(x_1, x_2^*)\, w = -\nabla_{x_1} g_j(x_1, x_2^*)\, d, \quad j \in J, \\ & \nabla_{x_2} f(x_1, x_2^*) w = -\nabla_{x_1} f(x_1, x_2^*)\, d + \nabla_{x_1} \mathcal{L}(x_1, x_2^*, \lambda)\, d, \end{array} \tag{3}$$

where $I(x_1) = \{i \in I : g_i(x_1, x_2^*) = 0\}$ and

$$\mathcal{L}(x_1, x_2, \lambda) = f(x_1, x_2) + \sum_{i \in I(x_1) \cup J} \lambda_i g_i(x_1, x_2)$$

is the Lagrangian of the lower-level problem with respect to the active constraints. The steepest descent direction among those defined by (2) may be shown to coincide with the optimal solution of the *linear-quadratic bilevel program*

$$\begin{array}{ll} \min\limits_{d} & \nabla_{x_1} F(x_1, x_2^*)\, d + \nabla_{x_2} F(x_1, x_2^*) w(x_1, d) \\ \text{s.t.} & \|d\| \leq 1, \\ & w(x_1, d) \text{ solves the quadratic program (3),} \end{array}$$

for which exact algorithms exist (see e.g. Bard and Moore [10], Jaumard et al. [46]). We also mention the work of Falk and Liu [34] who present a bundle method where the decrease of the upper-level objective is adapted according to subdifferential information obtained from the lower-level problem. They call the resulting setup a *leader predominate algorithm* in view of the role of the leader in the sequential decision making process. Note that all these descent methods only intend to find local optima.

Finally, *penalty methods* constitute a third category of algorithms. Because of the inherent difficulty of the nonlinear case, these methods are generally limited to computing stationary points and local minima. An initial step in this direction was achieved by Aiyoshi and Shimizu [1, 2, 65]. However the main drawback of their method is that, for a fixed value of the penalty parameter, it must compute the solution of the unconstrained lower-level problem at every update of the upper-level variables. Ishizuka and Aiyoshi [45] proposed a double penalty method in which both objective functions (1a) and (1c) are penalized. They still use the augmented lower-level objective and the penalty function of Aiyoshi and Shimizu but replace the lower-level problem by its stationarity condition, thus transforming (1) into a single-level program. For a given value of the penalty parameter, this single-level problem is solved using a second penalty function applied to the upper-level objective. A more recent contribution is proposed in Case's Ph.D. thesis [24] and follows up on ideas of Bi et al. [13], who themselves extend a technique they proposed earlier for linear bilevel programs [12]. Their approach focuses on a bilevel program for which the lower-level problem has been replaced by its Karush-Kuhn-Tucker conditions. More precisely, Case [24] builds a penalty function $\nu(x_1, x_2, \lambda)$ with respect to the l_1 norm, defined as the sum of the terms associated with each constraint of the KKT reformulation of the original problem. In view of the complex structure of their penalty function, the authors use a trust-region method, i.e., they minimize a simplified version of p whose local behavior is expected to model that of p. The trust-region model adopted by [24] corresponds to replacing each component function of the penalty by its second-order Taylor expansion around the current iterate.

We conclude this section by mentioning the relationship between bilevel programming problems and another important class of problems, namely *Mathematical Programs with Equilibrium Constraints*, or MPECs. The word *equilibrium* frequently refers to the formulation of the lower-level problem as a finite-dimensional variational inequality. This allows the mathematical representation of phenomena which may be encountered in engineering, physics, chemistry or economics. For a given function $F\colon \mathbb{R}^{n_1+n_2} \to \mathbb{R}$, the general form of an MPEC is as follows

$$\begin{array}{ll} \min\limits_{x_1,x_2} & F(x_1, x_2) \\ \text{s.t.} & (x_1, x_2) \in Z \quad \text{and} \quad x_2 \in \mathcal{S}(x_1), \end{array} \tag{4}$$

where $Z \subseteq \mathbb{R}^{n_1+n_2}$ is a nonempty closed set and $\mathcal{S}(x_1)$ is the solution set of a variational inequality (or VI) parameterized in x_1, that is,

$$\begin{array}{r} x_2 \in \mathcal{S}(x_1) \Leftrightarrow x_2 \in C(x_1) \quad \text{and} \\ (v - x_2)^T \phi(x_1, x_2) \ \geq 0 \quad \forall v \in C(x_1), \end{array} \tag{5}$$

where $C(x_1)$ is a closed convex subset of $\mathbb{R}^{n_2}$ and $\phi\colon \mathbb{R}^{n_1+n_2} \to \mathbb{R}^{n_2}$. As for bilevel problems, the terms upper-level and lower-level variables are also used to designate x_1 and x_2 respectively. If the mapping $\phi(x_1, \cdot)$ is the partial gradient map of a real-valued C^1 function f: $\mathbb{R}^{n_1+n_2} \to \mathbb{R}$, that is $\phi(x_1, x_2) = \nabla_{x_2} f(x_1, x_2)$, then, for any fixed x_1, the VI (5) defines the set of stationarity conditions of the optimization problem

$$\begin{array}{ll} \min\limits_{x_2} & f(x_1, x_2) \\ \text{s.t.} & x_2 \in C(x_1). \end{array} \tag{6}$$

Moreover, if $C(x_1)$ takes the form

$$C(x_1) = \{x_2\colon g(x_1, x_2) \leq 0\},$$

then problem (6) is nothing but the lower-level problem (1c)–(1d). This shows that MPECs subsume bilevel programs with convex lower-level problems. Conversely, an MPEC can be formulated as a bilevel program by reformulated the lower-level VI as an optimization problem. The optimization problem can, for instance, be constructed around the gap function (see e.g. Auslender [6] or Fukushima [40]), or by simply minimizing the complementarity slackness term.

While many active set approaches have been proposed for solving MPECs (see Bard [8]), more recent methods deal with constraint regularization (e.g. Facchinei et al. [33], Fukushima and Pang [41], or Scholtes [63]), implicit programming techniques (Outrata et al. [59]), or nonlinear programming techniques (filter methods: Fletcher and Leyffer [35, 36], sequential quadratic programming: see Jiang and Ralph [47], Ralph [60] and Fletcher et al. [37]). We refer to the monographs of Luo et al. [56] and Outrata et al. [59] for a comprehensive analysis of MPECs.

3. An approximation algorithm for bilevel programming problems

In this section, we describe our trust-region algorithm for solving nonlinear bilevel programming problems in the absence of joint upper-level constraints, i.e., constraints unrelated to the lower level and involving both upper and lower-level decision variables. We briefly recall that trust-region methods are iterative methods whose underlying idea is both intuitive and simple: at each iteration, a *model* of the considered problem is built around the incumbent solution and this model is then minimized within some prescribed region where the approximation is thought to be a sufficiently accurate representation of the original problem; this gives a new point and the next step consists in verifying that this point yields a sufficient improvement of the true objective function: if this is the case, the model is good and the trust region may be enlarged while if the objective function is not improved the region is shrinked and a new model is computed (for more details and an in-depth study of trust-region methods, we refer the reader to the books of Dennis and Schnabel [31] and Conn et al. [27]).

Note that our algorithm is not the first attempt to solve bilevel programming problems by means of a trust-region method since such an approach was presented by Liu et al.

[54]. Their algorithm is also based on a linear-quadratic approximation of the original problem, but its application is restricted to bilevel programs without upper-level constraints and whose lower-level problem is a strongly convex mathematical program with linear constraints. Under adequate assumptions, convergence to a Clarke stationary point may be proved. However no computational experience is reported.

3.1. *Description of the algorithm*

To describe our algorithm, we first assume that an initial solution is available (see details below). We build a *linear-quadratic* model of problem (1) around local linear approximations of F, G and g, and a quadratic approximation of the lower-level objective f. If $(\bar{x}_1, \bar{x}_2)$ denotes the current iterate, we set up the model functions:

$$
\begin{aligned}
F_m(x_1, x_2) &= F(\bar{x}_1, \bar{x}_2) + c_1^T(x_1 - \bar{x}_1) + c_2^T(x_2 - \bar{x}_2), && (7)\\
G_m(x_1) &= G(\bar{x}_1) + A_1(x_1 - \bar{x}_1), && (8)\\
f_m(x_1, x_2) &= f(\bar{x}_1, \bar{x}_2)\\
&\quad + d_1^T(x_1 - \bar{x}_1) + d_2^T(x_2 - \bar{x}_2)\\
&\quad + \frac{1}{2}(x_1 - \bar{x}_1)^T Q_{11}(x_1 - \bar{x}_1)\\
&\quad + (x_1 - \bar{x}_1)^T Q_{12}(x_2 - \bar{x}_2)\\
&\quad + \frac{1}{2}(x_2 - \bar{x}_2)^T Q_{22}(x_2 - \bar{x}_2), && (9)\\
g_m(x_1, x_2) &= g(\bar{x}_1, \bar{x}_2) + H_1(x_1 - \bar{x}_1) + H_2(x_2 - \bar{x}_2), && (10)
\end{aligned}
$$

where

$$
\begin{aligned}
c_1 &= \nabla_{x_1} F(\bar{x}_1, \bar{x}_2) \in \mathbb{R}^{n_1}, \quad c_2 = \nabla_{x_2} F(\bar{x}_1, \bar{x}_2) \in \mathbb{R}^{n_2},\\
A_1 &= J_{x_1} G(\bar{x}_1) \in \mathbb{R}^{m_1 \times n_1},\\
d_1 &= \nabla_{x_1} f(\bar{x}_1, \bar{x}_2) \in \mathbb{R}^{n_1}, \quad d_2 = \nabla_{x_2} f(\bar{x}_1, \bar{x}_2) \in \mathbb{R}^{n_2},\\
H_1 &= J_{x_1} g(\bar{x}_1, \bar{x}_2) \in \mathbb{R}^{m_2 \times n_1}, \quad H_2 = J_{x_2} g(\bar{x}_1, \bar{x}_2) \in \mathbb{R}^{m_2 \times n_2},\\
Q_{11} &= \nabla^2_{x_1 x_1} f(\bar{x}_1, \bar{x}_2) \in \mathbb{R}^{n_1 \times n_1}, \quad Q_{12} = \nabla^2_{x_1 x_2} f(\bar{x}_1, \bar{x}_2) \in \mathbb{R}^{n_1 \times n_2},\\
Q_{21} &= \nabla^2_{x_2 x_1} f(\bar{x}_1, \bar{x}_2) \in \mathbb{R}^{n_2 \times n_1}, \quad Q_{22} = \nabla^2_{x_2 x_2} f(\bar{x}_1, \bar{x}_2) \in \mathbb{R}^{n_2 \times n_2}.
\end{aligned}
$$

The above notation yields the *linear-quadratic* bilevel program:

$$
\begin{aligned}
\min_{x_1, x_2} \quad & F_m(x_1, x_2)\\
\text{s.t.} \quad & G_m(x_1) \le 0\\
& \min_{x_2} \quad f_m(x_1, x_2)\\
& \text{s.t.} \quad g_m(x_1, x_2) \le 0.
\end{aligned}
\tag{11}
$$

Due to its combinatorial structure, problem (11) can be reformulated as a mixed-integer program (MIP) and solved using either specialized algorithms (see Jaumard et al. [46]) or an off-the-shelf MIP software.

To transform (11) into an equivalent MIP, we first remove the constant terms from (7) and (9) and set

$$\bar{G} = G(\bar{x}_1) - A_1\bar{x}_1,$$

and

$$\bar{g} = g(\bar{x}_1, \bar{x}_2) - H_1\bar{x}_1 - H_2\bar{x}_2.$$

The upper and lower-level functions are then expressed as

$$\begin{aligned} F_{m'}(x_1, x_2) &= c_1^T x_1 + c_2^T x_2, \\ G_{m'}(x_1) &= A_1 x_1, \\ f_{m'}(x_1, x_2) &= r_1^T x_1 + r_2^T x_2 + \frac{1}{2}\left(x_1^T\ x_2^T\right)\begin{pmatrix} Q_{11} & Q_{12} \\ Q_{21} & Q_{22} \end{pmatrix}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix}, \\ g_{m'}(x_1, x_2) &= H_1 x_1 + H_2 x_2, \end{aligned}$$

where

$$r_1 = d_1 - Q_{21}^T\bar{x}_2 - Q_{11}^T\bar{x}_1 \in \mathbb{R}^{n_1}$$

and

$$r_2 = d_2 - Q_{12}^T\bar{x}_1 - Q_{22}^T\bar{x}_2 \in \mathbb{R}^{n_2}.$$

The problem (11) can then be rewritten as:

$$\begin{aligned} \min_{x_1} \quad & F_{m'}(x_1, x_2) \\ \text{s.t.} \quad & G_{m'}(x_1) \leq -\bar{G} \\ & \min_{x_2} \quad f_{m'}(x_1, x_2) \\ & \text{s.t.} \quad g_{m'}(x_1, x_2) \leq -\bar{g}. \end{aligned}$$

Incorporating the trust-region constraint with respect to the upper-level vector x_1, we derive an equivalent MIP:

$$\min \quad c_1^T x_1 + c_2^T x_2 \tag{12a}$$

$$\text{s.t.} \quad \|x_1 - \bar{x}_1\|_\infty \leq \Delta_k, \tag{12b}$$

$$A_1 x_1 \leq -\bar{G}, \tag{12c}$$

$$H_1 x_1 + H_2 x_2 \leq -\bar{g}, \tag{12d}$$

$$\lambda \geq 0 \in \mathbb{R}^{m_2}, \tag{12e}$$

$$\lambda_i \leq M z_i, \quad i = 1, \ldots, m_2, \tag{12f}$$

$$[-\bar{g} - H_1 x_1 - H_2 x_2]_i \leq M(1 - z_i), \quad i = 1, \ldots, m_2, \tag{12g}$$

$$z_i \in \{0, 1\}, \quad i = 1, \ldots, m_2, \tag{12h}$$

$$\frac{1}{2}\left(Q_{12}^T + Q_{21}\right)x_1 + Q_{22}x_2 + H_2^T \lambda + r_2 = 0. \tag{12i}$$

The trust-region constraint is expressed as (12b), while (12c) represents the primal feasibility of the upper-level problem. Primal feasibility of the lower-level problem is ensured by (12d); dual feasibility corresponds to (12e) and (12i). Finally, the complementarity constraint

$$\lambda_i[-\bar{g} - H_1 x_1 - H_2 x_2]_i = 0, \quad i = 1, \ldots, m_2$$

is linearized and replaced by (12f), (12g) and (12h), where the parameter M is a 'large number' that ensures that the MIP is equivalent to the model bilevel program. The existence of such constant follows from a result of Anandalingam and White [5].

Let (x_1^m, x_2^m) denote a global solution of (12). In order to evaluate the true value of this solution, one must compute the lower-level reaction to x_1^m, i.e., the optimal solution of

$$\begin{aligned} \min_{x_2} \quad & f\left(x_1^m, x_2\right) \\ \text{s.t.} \quad & g\left(x_1^m, x_2\right) \leq 0, \end{aligned} \tag{13}$$

which we denote by x_2^*. Since a *global* solution of (13) is required, the lower-level problem will be assumed to be convex, unless one is willing to solve to optimality a non-convex program, using global optimization techniques. This completes the description of the trust-region subproblem which, starting from a given point $(\bar{x}_1, \bar{x}_2)$, generates a solution (x_1^m, x_2^*) of a model of (1) around $(\bar{x}_1, \bar{x}_2)$. We are now in position to summarize the complete algorithm.

1. *Initialization.* Initial values are provided for the trust-region radius Δ_0, parameters η_1, η_2, γ_1 and γ_2 satisfying

   $$0 < \eta_1 \leq \eta_2 < 1 \quad \text{and} \quad 0 < \gamma_1 < 1 < \gamma_2.$$

   Values are also provided for M, $\Delta_{\min}$, $k_{\max}$ and ε. It is further assumed that an initial feasible point $(x_1^{(0)}, x_2^{(0)})$ is available. This point may for instance be deduced from the *high point*, that is, $x_1^{(0)}$ is set to the partial solution (in x_1) of the single-level program:

   $$\begin{aligned} \min_{x_1, x_2} \quad & F(x_1, x_2) \\ \text{s.t.} \quad & G(x_1) \leq 0 \\ & g(x_1, x_2) \leq 0 \end{aligned} \tag{14}$$

   and $x_2^{(0)}$ is the solution of the lower level problem corresponding to $x_1^{(0)}$. Next, the iteration counter k is set to 0.

2. *Computation of the model.* Construct $\bar{G}$, $\bar{g}$ and the non-constant terms in the model functions F_m, G_m, f_m and g_m.
3. *Solving the subproblem.* Solve problems (12) and (13) for their respective solutions (x_1^m, x_2^m) and x_2^*.
4. *Updating radius and iterate.* Compute the ratio of achieved versus predicted reduction

$$\rho_k = \frac{F\big(x_1^{(k)}, x_2^{(k)}\big) - F\big(x_1^m, x_2^*\big)}{F_m\big(x_1^{(k)}, x_2^{(k)}\big) - F_m\big(x_1^m, x_2^m\big)}.$$

   If $\rho_k < \eta_1$, the model is deemed inaccurate: set $(x_1^{(k+1)}, x_2^{(k+1)}) := (x_1^{(k)}, x_2^{(k)})$ and $\Delta_{k+1} := \gamma_1 \Delta_k$.
   Otherwise, set $(x_1^{(k+1)}, x_2^{(k+1)}) := (x_1^m, x_2^*)$. If $\rho_k \geq \eta_2$, $\Delta_{k+1} := \gamma_2 \Delta_k$.
5. *Stopping criteria.* The optimization process is stopped if one of the following criteria is met:

   – $\|x_1^{(k+1)} - x_1^{(k)}\| < \varepsilon$ after a successful iteration,
   – the achieved reduction equals the predicted one, i.e. $\rho_k \approx 1$, and the predicted reduction $F_m(x_1^{(k)}, x_2^{(k)}) - F_m(x_1^m, x_2^m)$ is small (e.g. smaller than 1.0e-10),
   – a predetermined number `mxnuns` of consecutive unsuccessful iterations has been reached,
   – the trust-region radius becomes smaller than some predetermined threshold $(\Delta_{k+1} < \Delta_{\min})$,
   – the number of iterations exceeds some predetermined number (i.e. $k + 1 > k_{\max}$).

   If none of the above conditions is satisfied, set $k := k + 1$ and return to Step 2.

3.2. *Joint upper-level constraints*

Joint upper-level constraints $G(x_1, x_2) \leq 0$ are transparent to the lower-level 'player' and must naturally be induced by the upper-level vector x_1. Finding a feasible point that respects these constraints and is compatible with the lower-level reaction is, theoretically, as difficult as solving a bilevel program. Actually, these constraints can make the feasible set of the bilevel program disconnected. Since these constraints are not in the direct control of the leader player, their inclusion in the trust-region model could lead to infeasible solutions, once the lower-level reaction is properly taken into account. Further research might be dedicated to an adaptation of our algorithm for dealing with this type of problems.

4. Computational experience

BIPA (Bilevel Programming with Approximation methods) is a C ANSI code for solving problems of type (1), based on the above-described methodology. The software is designed around existing and efficient tools for addressing subproblems (12) and (13). This enables us to focus our programming effort on both the approximation and trust-region aspects. The MIP (12) is solved using the `CPXmipopt` routine from the CPLEX Callable Library

Table 1. Test problems: data and references.

Problem type	Problem name	n_1	n_2	m_1	m_2	Reference
Linear-quadratic	`AiyoShim84P2`	2	2	5	6	Aiyoshi and Shimizu [2]
Quadratic-quadratic	`Bard88Ex1`	1	1	1	4	Bard [7]
	`Bard88Ex2`	4	4	9	12	Bard [7]
	`Bard88Ex3`	2	2	3	4	Bard [7]
	`Dempe92`	1	1	0	1	Dempe [28]
	`DeSilva78`	2	2	0	4	DeSilva [67]
	`FalkLiu95`	2	2	0	4	Falk and Liu [34]
Nonlinear	`BIPA2`	1	1	1	4	New problem
	`BIPA3`	1	1	2	2	New problem
	`BIPA4`	1	1	2	2	New problem
	`BIPA5`	1	2	1	6	New problem
Network design	`NDP1`	5	5	5	8	See [26]
	`NDP2`	5	5	5	8	See [26]
Toll-setting	`TOLL1`	3	8	3	13	Labbé et al. [52]
	`TOLL2`	3	18	3	28	New problem
	`TOLL3`	3	18	3	28	New problem
	`TOLL4`	2	4	2	6	Brotcorne [17]
	`TOLL5`	1	4	1	6	Brotcorne [17]
MPEC	`Outrata94P1`	1	2	2	4	Outrata [58]

[44], while problem (13) is solved using the C ANSI version of Spellucci's DONLP2 code (see [69, 70] for the underlying algorithm and [68] for DONLP2 users guide). An in-depth description of BIPA is provided in [26].

We tested BIPA on a set of representative instances, which are listed in Table 1 below. A complete description of all these test problems may be found in the user's guide of BIPA [26]. Some problems involving linear or quadratic functions are borrowed from the literature. Others are new problems created by customizing some of the former ones through the introduction of nonlinearities. A third set of test problems is based on two applications of bilevel programming to transportation modeling and planning, namely a continuous variant of the network design problem and a toll-setting problem. We also tested BIPA on a problem introduced by Outrata [58] in a paper dedicated to MPECs. Note that some of the problems do not satisfy the assumption on the convexity of the lower-level problem.

The values of the various parameters are reported in Table 2, where `mxnuns` denotes the maximum number of unsuccessful iterations. Trust-region parameters agree with those recommended by Conn et al. [27], with the exception of the initial trust-region radius, for which we could observe that a larger value often leads to an increase in the number of iterations. The issue of fine tuning the values of $k_{\max}$ and $\Delta_{\min}$ proved to be non-critical. Indeed, the associated stopping criteria (number of iterations too large or trust-region radius

Table 2. Parameters for numerical experiments.

Initial trust-region radius	Δ_0	10.0
Other trust-region parameters	η_1	0.01
	η_2	0.90
	γ_1	0.6
	γ_2	1.4
MIP-parameter	M	1.0e+2
Stopping criteria	$k_{\max}$	50
	$\Delta_{\min}$	1.0e−6
	ε	1.0e−6
	`mxnuns`	5

too small) were seldom met in practice. Finally, we emphasize the role of the parameter M appearing in the MIP (12). Ideally, M should be chosen as small as possible, in order that the linear programming relaxation of the MIP be as tight as possible, thus improving the quality of the lower bound and accelerating the implicit enumeration process. However, if M assumes a value that is too small, the optimal solution could be missed and, in an extreme situation, the MIP could be infeasible. Although our computational experiments showed that the value $M = 100$ was 'uniformly' suitable, this issue, which is problem dependent, would certainly deserve a more thorough analysis.

The numerical results are summarized in Table III. All tests were performed on a Sun Ultra-10/300 computer (512M RAM). The starting points for the algorithm correspond to those given in [26]. For the problems taken from the literature, we used the initial points provided in the relevant papers.[1] Otherwise, we started the algorithm from the point obtained from the high point (see (14)).

Generally speaking, all problems are solved in a small number of iterations. Note that the number of calls for functions evaluating F, G, f, g and their derivatives is equal to the number of iterations so we did not include these numbers in Table 3. The last two columns provide the total and the average (per iteration) number of enumeration nodes required by CPLEX to solve the MIP reformulations. Due to its powerful preprocessing capabilities and the small size of the MIP problems, CPLEX could often uncover an optimal solution at node 0, i.e., without resorting to branching.

BIPA is able to find the optimal solution of all problems from the literature as well as for the NDP and TOLL problems. For problem `FalkLiu95`, our solution improves[2] on that reported in [34]. For the new problems (`BIPA2`-`BIPA4`), we report the best solution found.

For most problems the optimal solution is found very early and the subsequent iterations are used to check optimality. Even for problems requiring a larger number of iterations, good solutions were found rapidly. For instance, $F(x_1, x_2) = -6598.4687$ after only 10 iterations for problem `Bard88Ex2`, $F(x_1, x_2) = -1$ after 7 iterations for `DeSilva78a`, and $F(x_1, x_2) = -2.25$ after 20 iterations for problem `FalkLiu95b`.

Toll-setting problems constitute a very particular class of problems for which specific methods are developed. Actually, building a linear-quadratic model of these problems might deteriorate their structure since it ignores the bilinear structure of the lower-level. However

Table 3. Results with BIPA.

		CPU	Objectives at solution		Enumeration nodes	
Problem name	Iter.	time	$F(x_1^*, x_2^*)$	$f(x_1^*, x_2^*)$	Total	Average
AiyoShim84P2	2	0.40	0.00	200.00	0	0.00
Bard88Ex1	7	1.12	17.00	1.00	3	0.42
Bard88Ex2	32	6.22	−6600.00	57.48	19	0.59
Bard88Ex3	1	0.20	−12.68	−1.02	0	0.00
Dempe92	6	0.94	31.25	4.00	0	0.00
DeSilva78a	13	2.10	−1.00	0.00	1	0.07
DeSilva78b	7	1.23	−1.00	0.00	7	1.00
FalkLiu95a	47	7.76	−2.25	0.00	0	0.00
FalkLiu95b	45	7.14	−2.25	0.00	10	0.22
BIPA2	7	1.16	17.00	2.00	3	0.43
BIPA3	1	0.17	2.00	24.02	0	0.00
BIPA4	2	0.33	88.79	−0.77	0	0.00
BIPA5	16	2.53	2.75	0.57	3	0.19
NDP1	6	1.09	300.50	419.80	0	0.00
NDP2	6	1.04	142.90	81.95	0	0.00
TOLL1	10	2.36	−7.00	12.00	33	3.30
TOLL2	9	4.19	−9.00	32.00	163	18.11
TOLL3	10	4.48	−24.00	81.00	105	10.50
TOLL4	9	1.32	−8.00	14.00	0	0.00
TOLL5	8	1.16	−11.00	34.00	4	0.50
Outrata94P1a	12	1.88	3.2077	−20.5293	0	0.00
Outrata94P1b	10	1.69	3.2077	−20.5293	0	0.00

the algorithms performs well and produces the optimal solution with a very reasonable amount of computational work.

Some comments about problem Outrata94P1 are in order. This problem was originally introduced by Outrata [58] and is part of a series of four problems. These problems represent MPECs where the variational inequality corresponds to the optimality conditions of the lower-level and hence they involve multipliers for the lower-level constraints. Only the first of these four problems fits our framework. The remaining three involve those multipliers in the upper-level objective, which leads to a class of problems slightly different from the one we study.

5. Comments and future work

The originality of our algorithm is twofold, in that it combines techniques from nonlinear programming (trust-region methods) and combinatorial optimization (mixed-integer programming), and preserves the bilevel structure of the approximated model. The numerical results collected so far reflect the good behavior of the algorithm and the robustness of the associated code.

At this point, we would like to make two comments about the numerical testing of bilevel programs. First, the number of test problems treated in this paper, as well as the size of these instances, is small. This is due to the fact that very few instances of large-scale, unstructured, nonlinear bilevel programs are available in the literature. Actually, some of the examples addressed in this paper are the largest having been addressed by a software that does not take into account the specific structure of the problems. While there exist generators of bilevel problems (see Calamai and Vicente [18–20], Calamai et al. [21] or Vicente et al. [74]), these are limited to linear or quadratic problems, and are obviously irrelevant to our approach. To our knowledge, there exist but two collections of test problems: the first is described in Chapter 9 of Floudas et al. [38] and contains 19 problems, most of them linear or quadratic; the second is the *MacMPEC collection* maintained by Leyffer [53], which contains about 130 MPECs, only a handful of them being equivalent to bilevel programs.

Our second comment is concerned with the lack of a proper testing environment (including a possible proper modeling language) for solving general bilevel programs. Such an environment should provide all the necessary tools for evaluating objective functions, gradients, . . . and would allow to avoid coding these functions from scratch.

The convergence study of the algorithm presented in this paper is the subject of ongoing research. Beyond the theoretical justification of our method, a further benefit from this might be to provide more appropriate stopping criteria than those listed at Step 5 of the algorithm.

Finally, in view of the promising results obtained by our algorithm, we will extend the methodology to mathematical problems with (monotone) variational inequality constraints, a subclass of MPEC of great practical importance. Since no robust software is currently available for general variational inequalities, we intend to adapt DONLP2 for this purpose.

Acknowledgments

Work of the first author was supported by a grant from the *Ministère de la Communauté Française* (Belgium). The first author also thanks the *Centre de Recherche sur les Transports* for having provided a cordial environment. Computational support from Serge Bisaillon and Luc Rocheleau is gratefully acknowledged. Finally, the authors are indebted to the Editor and to the anonymous referees for their comments and suggestions which improved this paper.

Notes

1. The starting point for problems `DeSilva78a` and `FalkLiu95a` is (0, 0, 1, 1) while `DeSilva78b` and `FalkLiu95b` start from (1, 1, 1, 1). `Outrata94P1a` is started with $x_1^{(0)} = 0$ and `Outrata94P1b` takes $x_1^{(0)} = 10$.
2. Note that we had to set $mxnuns = 6$ to be able to find the optimal solution with the second starting point.

References

1. E. Aiyoshi and K. Shimizu, “Hierarchical decentralized systems and its new solution by a barrier method,” IEEE Transactions on Systems, Man, and Cybernetics, vol. 11, pp. 444–449, 1981.

2. E. Aiyoshi and K. Shimizu, “A solution method for the static constrained Stackelberg problem via penalty method,” IEEE Transactions on Automatic Control, vol. 29, pp. 1111–1114, 1984.
3. F.A. Al-Khayal, R. Horst, and P. M. Pardalos, “Global optimization of concave functions subject to quadratic constraints: An application in nonlinear bilevel programming,” Annals of Operations Research, vol. 34, pp. 125–147, 1992.
4. G. Anandalingam and T. Friesz, “Hierarchical optimization: An introduction,” Annals of Operations Research, vol. 34, pp. 1–11, 1992.
5. G. Anandalingam and D.J. White, “A solution method for the linear static stackelberg problem using penalty funcitons,” IEEE Transactions on Automatic Control, vol. 35, pp. 1170–1173, 1990.
6. A. Auslender, Optimisation: Méthodes numériques, Masson, Paris, 1976.
7. J.F. Bard, “Convex two-level optimization,” Mathematical Programming, vol. 40, pp. 15–27, 1988.
8. J.F. Bard, Practical Bilevel Optimization—Algorithms and Applications, Kluwer Academic Publishers, 1998.
9. J.F. Bard and J. Falk, “An explicit solution to the multi-level programming problem,” Computers and Operations Research, vol. 9, pp. 77–100, 1982.
10. J.F. Bard and J.T. Moore, “A branch and bound algorithm for the bilevel programming problem,” SIAM Journal of Scientific and Statistical Computing, vol. ll, no. 2, pp. 281–292, 1990.
11. O. Ben-Ayed, “Bilevel linear programming,” Computers and Operations Research, vol. 20, pp. 485–501, 1993.
12. Z. Bi, P.H. Calamai, and A.R. Conn, “An exact penalty function approach for the linear bilevel programming problem,” Technical Report No. 167-0-310789, Department of Systems Design Engineering, University of Waterloo, 1989.
13. Z. Bi, P.H. Calamai, and A.R. Conn, “An exact penalty function approach for the nonlinear bilevel programming problem,” Technical Report No, 180-0-170591, Department of Systems Design Engineering, University of Waterloo, 1991.
14. W. Bialas and M. Karwan, “Two-level linear programming,” Management Science, vol. 30, pp. 1004–1020, 1984.
15. W. Bialas, M. Karwan, and J. Shaw, “A parametric complementarity pivot approach for two-level linear programming,” Technical Report 80-2, State University of Now York at Buffalo, Operations Research Program, 1980.
16. J. Bracken and J. McGill, “Mathematical programs with optimization problems in the constraints,” Operations Research, vol. 21, pp. 37–44, 1973.
17. L. Brotcorne, “Approches opérationnelles et stratégiques des problémes de trafic routier,” PhD thesis, Université Libre de Bruxelles, Brussels, Belgium, 1998.
18. P.H. Calamai and L.N. Vicente, “Generating linear and linear-quadratic bilevel programming problems,” SIAM Journal on Scientific Computing, vol. 14, pp. 770–782, 1993.
19. P.H. Calamai and L.N. Vicente, “Algorithm 728: FORTRAN subroutines for generating quadratic bilevel programming problems,” ACM Transactions on Mathematical Software, vol. 20, pp. 120–123, 1994.
20. P.H. Calamai and L.N. Vicente, “Generating quadratic bilevel programming test problems,” ACM Transactions on Mathematical Software, vol. 20, pp. 103–119, 1994.
21. P.H. Calamai, L.N. Vicente, and J.J. Júdice, “A new technique for generating quadratic programming test problems,” Mathematical Programming, vol. 61, pp. 215–231, 1993.
22. W. Candler and R. Norton, “Multilevel programing,” Technical Report 20, World Bank Development Research Center: Washington DC, 1977.
23. W. Candler and R. Townsley, “A linear two-level programming problem,” Computers and Operations Research, vol. 9, pp. 59–76, 1982.
24. L.M. Case, An l_1 Penalty Function Approach to the Nonlinear Bilevel Programming Problem, PhD thesis, University of Waterloo: Ontario, Canada, 1999.
25. B. Colson, “Mathematical programs with equilibrium constraints and nonlinear bilevel programming problems,” Master’s thesis, Department of Mathematics, FUNDP, Namur, Belgium, Sept. 1999.
26. B. Colson, “BIPA (Bilevel programming with approximation methods): Software guide and test problems,” Technical Report CRT-2002-38, Centre de Recherche sur les Transports, Université de Montréal, Montréal, QC, Canada, Oct. 2002.
27. A.R. Conn, N.I.M. Gould, and Ph.L. Toint, Trust-Region Methods, SIAM Publications: Philadelphia, Pennsylvania, USA, 2000.

28. S. Dempe, “A necessary and a sufficient optimality condition for bilevel programming problems,” Optimization, vol. 25, pp. 341–354, 1992.
29. S. Dempe, Foundations of Bilevel Programming, vol. 61 of Nonconvex Optimization and its Applications, Kluwer Academic Publishers: Dordrecht, The Netherlands, 2002.
30. S. Dempe, “Annotated bibliography on bilevel programming and mathematical programs with equilibrium constraints,” Optimization, vol. 52, pp. 333–359, 2003.
31. J.E. Dennis Jr and R.B. Schnabel, Numerical Methods for Unconstrained Optimization and Nonlinear Equations, Prentice-Hall: Englewood Cliffs, New Jersey, USA, 1983. Reprinted as Classics in Applied Mathematics 16, SI AM Publications: Philadelphia, Pennsylvania, USA, 1996.
32. T. Edmunds and J.F. Bard, “Algorithms for nonlinear bilevel mathematical programs,” IEEE transactions on Systems, Man, and Cybernetics, vol. 21, no. 1, pp. 83–89, 1991.
33. F. Facchinei, H. Jiang, and L. Qi, “A smoothing method for mathematical programs with equilibrium constraints,” Technical Report R03-9.6, Università di Roma “La Sapienza,” Dipartimento di Informatica e Sistemistica, 1996.
34. J.E. Falk and J. Liu, “On bilevel programming, Part I: general nonlinear cases,” Mathematical Programming, vol. 70, no. 1, pp. 47–72, 1995.
35. R. Fletcher and S. Leyffer, “Nonlinear programming without a penalty function,” Mathematical Programming, vol. 91, pp. 239–269, 2002.
36. R. Fletcher and S. Leyffer, “Numerical experience with solving MPECs by nonlinear programming methods,” Numerical Analysis Report NA/210, Department of Mathematics, University of Dundee, Dundee, Scotland, 2002.
37. R. Fletcher, S. Leyffer, D. Ralph, and S. Scholtes, Local Convergence of SQP Methods for Mathematical Programs with Equilibrium Constraints, Numerical Analysis Report NA/209, Department of Mathematics, University of Dundee, Dundee, Scotland, 2002.
38. C.A. Floudas, P.M. Pardalos, C.S. Adjiman, W.R. Esposito, Z.H. Gumus, S.T. Harding, J.L. Klepeis, C.A. Meyer, and C.A. Schweiger, Handbook of Test Problems in Local and Global Optimization, Kluwer, 1999.
39. J. Fortuny-Amat and B. McCarl, “A representation and economic interpretation of a two-level programming problem,” Journal of the Operational Research Society, vol. 32, pp. 783–792, 1981.
40. M. Fukushima, “Equivalent differentiable optimization problems and descent methods for asymmetric variational inequality problems,” Mathematical Programming, vol. 53, pp. 99–110, 1992.
41. M. Fukushima and J.-S. Pang, “Complementarity constraint qualifications and simplified B-stationarity conditions for mathematical programs with equilibrium constraints,” Computational Optimization and Applications, vol. 13, pp. 111–136, 1999.
42. P. Hansen, B. Jaumard, and G. Savard, “New branch-and-bound rules for linear bilevel programming,” SIAM Journal on Scientific and Statistical Computing, vol. 13, no. 5, pp. 1194–1217, 1992.
43. S. Hsu and U. Wen, “A review of linear bilevel programming problems,” in Proceedings of the National Science Council, Republic of China, Part A: Physical Science and Engineering, vol. 13, pp. 53–61, 1989.
44. ILOG CPLEX Division, CPLEX User’s Guide.
45. Y. Ishizuka and E. Aiyoshi, “Double penalty method for bilevel optimization problems,” Annals of Operations Research, vol. 34, pp. 73–88, 1992.
46. B. Jaumard, G. Savard, and J. Xiong, “An exact algorithm for convex quadratic bilevel programming,” Draft paper, Ecole Polytechnique de Montréal, January 2000.
47. H. Jian and D. Ralph, “Smooth sqp methods for mathematical programs with nonlinear complementarity constraints,” Manuscript, Department of Mathematics and Statistics, University of Melbourne, Dec. 1997.
48. J. Júdice and A. Faustino, “The solution of the linear bilevel programming problem by using the linear complementarity problem,” Investigação Operational, vol. 8, pp. 77–95, 1988.
49. J. Júdice and A. Faustino, “A sequential LCP method for bilevel linear programming,” Annals of Operations Research, vol. 34, pp. 89–106, 1992.
50. C.D. Kolstad, “A review of the literature on bi-level mathematical programming,” Technical Report LA-10284-MS, Los Alamos National Laboratory, Los Alamos, New Mexico, 1985.
51. C.D. Kolstad and L.S. Lasdon, “Derivative estimation and computational experience with large bilevel mathematical programs,” Journal of Optimization Theory & Applications, vol. 65, pp. 485–499, 1990.

52. M. Labbé, P. Marcotte, and G. Savard, “A bilevel model of taxation and its applications to optimal highway pricing,” Management Science, vol. 44, pp. 1595–1607, 1998.
53. S. Leyffer, MacMPEC—Ampl collection of Mathematical Programs with Equilibrium Constraints, Available at http://wvw-unix.mcs.anl.gov/~leyffer/MacMPEC/.
54. G. Liu, J. Han, and S. Wang, “A trust region algorithm for bilevel programming problems,” Chinese Science Bulletin, vol. 43, no. 10, pp. 820–824, 1998.
55. P. Loridan and J. Morgan, “Weak via strong Stackelberg problem: New results,” Journal of Global Optimization, vol. 8, pp. 263–287, 1996.
56. Z.-Q. Luo, J.-S. Pang, and D. Ralph, Mathematical Programs with Equilibrium Constraints, Cambridge University Press, 1996.
57. A. Migdalas, “Bilevel programming in traffic planning: Models, methods and challenge,” Journal of Global Optimization, vol. 7, pp. 381–405, 1995.
58. J. Outrata, “On optimization problems with variational inequality constraints,” SIAM Journal on Optimization, vol. 4, no. 2, pp. 340–357, 1994.
59. J.V. Outrata, M. Kočvara, and J. Zowe, Nonsmooth Approach to Optimization Problems with Equilibrium Constraintsé. Kluwer Academic Publishers, 1998.
60. D. Ralph, Optimization with Equilibrium Constraints: A Piecewise SQP Approach, Manuscript, Department of Mathematics and Statistics, University of Melbourne, Nov. 1998.
61. G. Savard, Contribution à la programmation mathématique à deux niveaux. Ph.D. thesis, Ecole Polytechnique de Montréal, Université de Montréal, April 1989.
62. G. Savard and J. Gauvin, “The steepest descent direction for the nonlinear bilevel programming problem,” Operations Research Letters, vol. 15, pp. 265–272, 1994.
63. S. Scholtes, “Convergence properties of a regularisation scheme for mathematical programs with complementarity constraints,” SIAM Journal on Optimization, vol. 11, pp. 918–936, 2001.
64. S. Scholtes and M. Stöhr, “Exact penalization of mathematical programs with equilibrium constraints,” SIAM Journal on Control and Optimization, vol. 37, no. 2, pp. 617–652, 1999.
65. K. Shimizu and E. Aiyoshi, “A new computational method for Stackelberg and min-max problems by use of a penalty method,” IEEE Transactions on Automatic Control, vol. 26, pp. 460–466, 1981.
66. K. Shimizu, Y. Ishizuka, and J.F. Bard, Nondifferentiable and Two-Level Mathematical Programming, Kluwer Academic Publishers, 1997.
67. A.H. De Silva, Sensitivity Formulas for Nonlinear Factorable Programming and their Application to the Solution of an Implicitly Defined Optimization Model of US Crude Oil Production, Ph.D. thesis, George Washington University, 1978.
68. P. Spellucci, D0NLP2 Users Guide. Technical University at Darmstadt, Department of Mathematics, 64829 Darmstadt, Germany.
69. P. Spellucci, “A new technique for inconsistent problems in the SQP method,” Math. Meth. of Oper. Res., vol. 47, pp. 355–400, 1998.
70. P. Spellucci, “An SQP method for general nonlinear programs using only equality constrained subproblems,” Mathematical Programming, vol. 82, pp. 413–448, 1998.
71. N.V. Thoai, Y. Yamamoto, and A. Yoshise, “Global optimization method for solving mathematical programs with linear complementarity constraints,” Discussion Paper No. 987, Institute of Policy and Planning Sciences, University of Tsukuba, Japan, May 2002.
72. H. Tuy, A. Migdalas, and P. Varbrand, “A global optimization approach for the linear two-level rjrogram,” Journal of Global Optimization, vol. 3, pp. 1–23, 1993.
73. L.N. Vicente and P.H. Calamai, “Bilevel and multilevel programming: A bibliography review,” Journal of Global Optimization, vol. 5, no. 3, pp. 291–306, 1994.
74. L.N. Vicente, P.H. Calamai, and J.J. Júdice, “Generation of disjointly constrained bilinear programming test problems,” Computational Optimization and applications, vol. 1, pp. 299–306, 1992.
75. L.N. Vicente, G. Savard, and J.J. Júdice, “Discrete linear bilevel programming problem,” Journal of Optimization Theory and Applications, vol. 89, pp. 597–614, 1996.
76. U. Wen and S. Hsu, “Linear bi-level programming problems—A review,” Journal of the Operational Research Society, vol. 42, pp. 125–133, 1991.