Efficiently solving linear bilevel programming problems using off-the-shelf optimization software

INFORMS 2018

S. Pineda [1] H. Bylling [2] J. M. Morales [1]

[1]University of Malaga (Spain)
[2]University of Copenhagen (Denmark)

November 6, 2018

Bilevel programming

- Decentralized environments: multiple decisions maker with divergent objectives that interact with each other in a hierarchical organization
- Simplest case: one leader and one follower (Stackelberg game)
- A Stackelberg game can be mathematically formulated as a bilevel problem (BLP)

$$
\begin{aligned}
\min_{x} \quad & F(x,y) && \text{(1a)}\\
\text{s.t.} \quad & G_i(x,y) \geqslant 0, \quad \forall i && \text{(1b)}\\
& \min_{y} \quad f(x,y) && \text{(1c)}\\
& \quad \text{s.t.} \quad g_j(x,y) \geqslant 0, \quad \forall j && \text{(1d)}
\end{aligned}
$$

- Even if $F(x,y)$, $f(x,y)$, $G_i(x,y)$ and $g_j(x,y)$ are linear, the BLP is proven to be NP-hard[1]

[1] Jeroslow 1985; Bard 1991.

Bilevel programming applications

BLP is widely used in energy and power system applications:

- Electricity grid security analysis[2]
- Transmission expansion planning[3]
- Strategic bidding of power producers[4]
- Generation capacity expansion[5]
- Investment in wind power generation[6]
- Market equilibria models[7]

---

[2] Motto, Arroyo, and Galiana 2005.

[3] Garces et al. 2009; Jenabi, Fatemi Ghomi, and Smeers 2013.

[4] Ruiz and Conejo 2009; Zugno et al. 2013.

[5] Wogrin, Centeno, and Barquín 2011; Kazempour et al. 2011.

[6] Baringo and Conejo 2014; Maurovich-Horvat, Boomsma, and Siddiqui 2015.

[7] Pozo and Contreras 2011; Ruiz, Conejo, and Smeers 2012.

Methods to solve bilevel programming

Methods to solve BLP can be divided into two main categories:

- Dedicated methods[8]
  - Efficient and globally optimal
  - Hard to implement in commercial optimization software
- Plug-and-play methods[9]
  - Straightforward implementation in commercial optimization software
  - High computational burden and locally optimal
  - Most common: reformulate as single-level and Fortuny-Amat (bigM)

[8] Bialas and Karwan 1984; Shi, Lu, and Zhang 2005; Calvete, Galé, and Mateo 2008; Li and Fang 2012; Sinha, Malo, and Deb 2013; Jiang et al. 2013; Bard and Falk 1982; Bard and Moore 1990; Hansen, Jaumard, and Savard 1992; Shi et al. 2006.

[9] Fortuny-Amat and McCarl 1981; Ruiz and Conejo 2009; Gabriel and Leuthold 2010; Siddiqui and Gabriel 2012; Scholtes 2001; Ralph and Wright 2004; White and Anandalingam 1993; Hu and Ralph 2004; Lv et al. 2007; Fletcher and Leyffer 2004.

Linear bilevel problem

If all functions are linear, the resulting linear bilevel problem (LBLP) can be generally formulated as

$$\begin{aligned}
\min_{x} \quad & c_1 x + d_1 y && \text{(2a)}\\
\text{s.t.} \quad & A_1 x + B_1 y \leqslant b_1 && \text{(2b)}\\
& \min_{y} \quad c_2 x + d_2 y && \text{(2c)}\\
& \quad \text{s.t.} \quad A_2 x + B_2 y \leqslant b_2 \quad (\lambda) && \text{(2d)}
\end{aligned}$$

We assume $B_1 = 0$ to avoid disconnected feasible regions[10]

[10] Colson, Marcotte, and Savard 2005; Shi, Zhang, and Lu 2005.

Linear bilevel problem

Since the lower-level optimization problem is linear, it can be replaced with its KKT optimality conditions

$$
\begin{aligned}
\min_{x,y,\lambda} \quad & c_1 x + d_1 y && \text{(3a)}\\
\text{s.t.} \quad & A_1 x + B_1 y \leqslant b_1 && \text{(3b)}\\
& d_2 + \lambda B_2 = 0 && \text{(3c)}\\
& b_2 - A_2 x - B_2 y \geqslant 0 && \text{(3d)}\\
& \lambda \geqslant 0 && \text{(3e)}\\
& \lambda \left(b_2 - A_2 x - B_2 y\right) = 0 && \text{(3f)}
\end{aligned}
$$

Without complementarity conditions (3f), problem (3) would be linear. Thus, all methods differ on how to deal with this constraints.

Plug-and-play methods

- Special order sets type 1 (SOS1)
- Fortuny-Amat with bigM (FA)
- Regularization (REG)
- Proposed method (REG-FA)

SOS1 variables

- This method declares Special Ordered Sets of type 1 (SOS1)[11]

$$
\begin{aligned}
s_j(1) &= (b_2 - A_2 x - B_2 y)_j \,, \quad \forall j \\
s_j(2) &= \lambda_j, \quad \forall j
\end{aligned}
$$

- This method explores using a binary tree all combinations of the complementarity constraints and therefore ensures global optimality

[11] Siddiqui and Gabriel 2012.

Fortuny-Amat with bigM

The complementarity constraints are reformulated as

$$
\begin{aligned}
\min_{x,y,\lambda,u} \quad & F(x,y) = c_1 x + d_1 y && \text{(5a)}\\
\text{s.t.} \quad & A_1 x + B_1 y \leqslant b_1 && \text{(5b)}\\
& d_2 + \lambda B_2 = 0 && \text{(5c)}\\
& b_2 - A_2 x - B_2 y \geqslant 0 && \text{(5d)}\\
& \lambda \geqslant 0 && \text{(5e)}\\
& b_2 - A_2 x - B_2 y \leqslant (1-u) M_1 && \text{(5f)}\\
& \lambda \leqslant u M_2 && \text{(5g)}\\
& u \in \{0,1\} && \text{(5h)}
\end{aligned}
$$

where u is a vector of binary variables of appropriate size and M_1, M_2 are large enough scalars that need to be adjusted.

Fortuny-Amat with bigM

BigM are usually adjusted by the following trial-and-error procedure:

1. Select initial values for M_1 and M_2.
2. Solve model (5) using MIP solver (CPLEX).
3. Find a j' such that $u_{j'} = 0$ and $(b_2 - A_2x - B_2y)_{j'} = M_{1j'}$. If such a j' exists, increase the value of $M_{1j'}$ and go to step 2). Otherwise, go to step 4).
4. Find a j' such that $u_{j'} = 1$ and $\lambda_{j'} = M_{2j'}$. If such a j' exists, increase the value of $M_{2j'}$ and go to step 2). Else, the solution to (2) *is assumed* to correspond to the optimal solution of the original bilevel problem (1).

This method may fail and provide highly suboptimal solutions!!

Fortuny-Amat with bigM

Let us consider the following linear bilevel problem:

$$\begin{aligned}
\max_{x \in \mathbb{R}} \quad & z = x + y \\
\text{s.t.} \quad & 0 \leqslant x \leqslant 2 \\
& \min_{y \in \mathbb{R}} \quad y \\
& \text{s.t.} \quad y \geqslant 0 \quad (\lambda_1) \\
& \qquad\quad x - 0.01y \leqslant 1 \quad (\lambda_2)
\end{aligned}$$

It is easy to verify that the optimal solution to this problem is $z^* = 102, x^* = 2, y^* = 100, \lambda_1^* = 0, \lambda_2^* = 100$.

Fortuny-Amat with bigM

We can reformulate it using bigM as follows:

$$\max_{x,y} \quad z = x + y$$

s.t. $0 \leqslant x \leqslant 2$

$x - 0.01y \leqslant 1$

$1 - \lambda_1 - 0.01\lambda_2 = 0$

$y, \lambda_1, \lambda_2 \geqslant 0$

$\lambda_1 \leqslant u_1 M_1^D$

$y \leqslant (1 - u_1) M_1^P$

$\lambda_2 \leqslant u_2 M_2^D$

$-x + 0.01y + 1 \leqslant (1 - u_2) M_2^P$

$u_1, u_2 \in \{0, 1\}$

For $M_{1,2}^P = 200$ $M_{1,2}^D = 50$

Case	u_1	u_2	x	y	λ_1	λ_2	z
1	0	1	Infeasible				
2	1	1	1	0	Multiple$^{(*)}$		1
3	1	0	1	0	1	0	1
4	0	0	Infeasible				

$^{(*)}$ $0 \leqslant \lambda_1 \leqslant 50, 0 \leqslant \lambda_2 \leqslant 50, 1 - \lambda_1 - 0.01\lambda_2 = 0$

- Case 2 includes $\lambda_1 = 0.5, \lambda_2 = 50$
- CPLEX always provides Case 3
- Since $\lambda_1 < 50, \lambda_2 < 50$, Case 3 is assumed to be global optimal!!

Regularization approach

All feasible points of (3) are nonregular (nonlinear solvers fail even to find a local optimal solution). This problem can be regularized as follows[12]:

$$
\begin{aligned}
\min_{x,y,\lambda} \quad & F(x,y) = c_1 x + d_1 y && \text{(8a)}\\
\text{s.t.} \quad & A_1 x + B_1 y \leqslant b_1 && \text{(8b)}\\
& d_2 + \lambda B_2 = 0 && \text{(8c)}\\
& b_2 - A_2 x - B_2 y \geqslant 0 && \text{(8d)}\\
& \lambda \geqslant 0 && \text{(8e)}\\
& \lambda \left(b_2 - A_2 x - B_2 y\right) \leqslant t && \text{(8f)}
\end{aligned}
$$

- Parameter t is iteratively decreased to 0
- Strong theoretical and empirical convergence properties[13]
- Only guaranteed to provide local optimal solutions

[12] Scholtes 2001; Ralph and Wright 2004.

[13] Scholtes 2001.

Proposed approach

- The regularization method is fast, but only provides local optimal solutions
- The bigM method achieves global optimality provided that large constants are set to proper values
- The proposed method aims to combine both approaches. We propose to use information about the local optimal solution to set the **large constants** and find initial values of the **binary variables** as follows:
  1. Solve (3) using regularization to obtain a local optimal solution
  2. Select a value of $\mathcal{M} > 1$
  3. Set $M_1 \leftarrow \mathcal{M} \max_j\{(b_2 - A_2 x - B_2 y)_j\}$ and $M_2 \leftarrow \mathcal{M} \max_j\{(\lambda)_j\}$
  4. Set initial values of binary variables u as follows. If $(b_2 - A_2 x - B_2 y)_j > 0$, then $u_j = 0$. If $\lambda_j > 0$, then $u_j = 1$
  5. Solve mixed-integer problem (5)

Computational results

We compare the proposed method with existing ones using 300 randomly generated examples of different sizes:

$$\begin{aligned} \min_{x} \quad & c_1 x + d_1 y \\ \text{s.t.} \quad & A_1 x + B_1 y \leqslant b_1 \\ & \min_{y} \quad c_2 x + d_2 y \\ & \text{s.t.} \quad A_2 x + B_2 y \leqslant b_2 \end{aligned}$$

	n	m	p	q	r
Small	50	50	25	25	25
Medium	100	100	50	50	50
Large	200	200	100	100	100

$$\begin{aligned} c_1 &= |\mathcal{N}(1,n)| \\ d_1 &= |\mathcal{N}(1,m)| \end{aligned} \quad A_1 = \begin{pmatrix} \mathcal{N}(p,n) \\ -I \end{pmatrix} \quad B_1 = \begin{pmatrix} \mathbf{0} \\ \mathbf{0} \end{pmatrix} \quad b_1 = \begin{pmatrix} \mathcal{N}(p,1) \\ \mathbf{0} \end{pmatrix}$$

$$\begin{aligned} c_2 &= |\mathcal{N}(1,n)| \\ d_2 &= |\mathcal{N}(1,m)| \end{aligned} \quad A_2 = \begin{pmatrix} \mathcal{N}(q,n) \\ \mathbf{0} \\ \mathbf{0} \end{pmatrix} \quad B_2 = \begin{pmatrix} \mathcal{N}(q,m) \\ \mathcal{N}(r,m) \\ -I \end{pmatrix} \quad b_2 = \begin{pmatrix} \mathcal{N}(q,1) \\ \mathcal{N}(r,1) \\ \mathbf{0} \end{pmatrix}$$

Computational results

	Small (n=50)				Medium (n=100)				Large (n=200)			
	#opt	#inf	time (s)	gap(%)	#opt	#inf	time(s)	gap(%)	#opt	#inf	time(s)	gap(%)
SOS1	98	2	1	0.00	90	0	4656	0.27	27	0	17652	2.05
FA-5	11	24	12	7.65	8	7	5385	7.15	0	20	21415	4.57
FA-50	98	2	5	0.00	94	2	5495	0.04	53	1	16873	0.31
FA-100000	27	2	0	6.44	11	0	0	10.19	2	0	1	9.86
REG	61	2	0	0.55	41	0	1	0.52	30	1	14	0.29
REG-FA-10	98	2	5	0.00	99	0	2353	0.00	72	0	13258	0.10

- #opt: Number of problems solved to optimality (out of 100)
- #inf: Number of infeasible problems (out of 100)
- time: average time of 100 problems
- gap: average gap with respect to global optimal solution of 100 problems

Computational results

	Small (n=50)				Medium (n=100)				Large (n=200)			
	#opt	#inf	time (s)	gap(%)	#opt	#inf	time(s)	gap(%)	#opt	#inf	time(s)	gap(%)
SOS1	**98**	2	1	0.00	**90**	0	4656	0.27	**27**	0	17652	2.05
FA-5	11	24	12	7.65	8	7	5385	7.15	0	20	21415	4.57
FA-50	98	2	5	0.00	94	2	5495	0.04	53	1	16873	0.31
FA-100000	27	2	0	6.44	11	0	0	10.19	2	0	1	9.86
REG	61	2	0	0.55	41	0	1	0.52	30	1	14	0.29
REG-FA-10	98	2	5	0.00	99	0	2353	0.00	72	0	13258	0.10

- SOS1 works fine for small size problems
- For large problems, SOS1 reach the maximum time of 6 h

Computational results

	Small (n=50)				Medium (n=100)				Large (n=200)			
	#opt	#inf	time (s)	gap(%)	#opt	#inf	time(s)	gap(%)	#opt	#inf	time(s)	gap(%)
SOS1	98	2	1	0.00	90	0	4656	0.27	27	0	17652	2.05
FA-5	**11**	24	12	7.65	**8**	7	5385	7.15	**0**	20	21415	4.57
FA-50	**98**	2	5	0.00	**94**	2	5495	0.04	**53**	1	16873	0.31
FA-100000	**27**	2	0	6.44	**11**	0	0	10.19	**2**	0	1	9.86
REG	61	2	0	0.55	41	0	1	0.52	30	1	14	0.29
REG-FA-10	98	2	5	0.00	99	0	2353	0.00	72	0	13258	0.10

- FA-5 leads to infeasible problems since bigM are not large enough
- Numerical inestabilities occur for FA-100000
- FA-50 provides the best results for this approach

Computational results

	Small (n=50)				Medium (n=100)				Large (n=200)			
	#opt	#inf	time (s)	gap(%)	#opt	#inf	time(s)	gap(%)	#opt	#inf	time(s)	gap(%)
SOS1	98	2	1	0.00	90	0	4656	0.27	27	0	17652	2.05
FA-5	11	24	12	7.65	8	7	5385	7.15	0	20	21415	4.57
FA-50	98	2	**5**	0.00	94	2	**5495**	0.04	53	1	**16873**	0.31
FA-100000	27	2	0	6.44	11	0	0	10.19	2	0	1	9.86
REG	61	2	0	0.55	41	0	1	0.52	30	1	14	0.29
REG-FA-10	98	2	5	0.00	99	0	2353	0.00	72	0	13258	0.10

- The computational time for FA approach dramatically increases with problem size

Computational results

	Small (n=50)				Medium (n=100)				Large (n=200)			
	#opt	#inf	time (s)	gap(%)	#opt	#inf	time(s)	gap(%)	#opt	#inf	time(s)	gap(%)
SOS1	98	2	1	0.00	90	0	4656	0.27	27	0	17652	2.05
FA-5	11	24	12	7.65	8	7	5385	7.15	0	20	21415	4.57
FA-50	98	2	5	0.00	94	2	5495	0.04	53	1	16873	0.31
FA-100000	27	2	0	6.44	11	0	0	10.19	2	0	1	9.86
REG	61	2	0	**0.55**	41	0	1	**0.52**	30	1	14	**0.29**
REG-FA-10	98	2	5	0.00	99	0	2353	0.00	72	0	13258	0.10

- Local optimal solutions are quite close to the global optimal solutions
- The size of the problem does not significantly affect computational time

Computational results

	Small (n=50)				Medium (n=100)				Large (n=200)			
	#opt	#inf	time (s)	gap(%)	#opt	#inf	time(s)	gap(%)	#opt	#inf	time(s)	gap(%)
SOS1	98	2	1	0.00	90	0	4656	0.27	27	0	17652	2.05
FA-5	11	24	12	7.65	8	7	5385	7.15	0	20	21415	4.57
FA-50	98	2	5	0.00	94	2	5495	0.04	53	1	16873	0.31
FA-100000	27	2	0	6.44	11	0	0	10.19	2	0	1	9.86
REG	61	2	0	0.55	41	0	1	0.52	30	1	14	0.29
REG-FA-10	**98**	2	5	0.00	**99**	0	2353	0.00	**72**	0	13258	0.10

- The proposed approach achieves the optimal solution in most problems and achieves the lower average gap at reasonable computational times

Computational results

For $n = 100$, we change the scaling and sparsity of matrices and vectors

	Full matrix, good scaled				Sparse matrix, good scaled				Full matrix, bad scaled			
	#opt	#inf	time (s)	gap(%)	#opt	#inf	time(s)	gap(%)	#opt	#inf	time(s)	gap(%)
SOS1	90	0	4656	0.27	86	0	4293	0.48	56	0	18419	7.02
FA-5	8	7	5385	7.15	7	12	4370	8.75	0	100	-	-
FA-50	94	2	5495	0.04	92	2	4283	0.02	0	100	-	-
FA-100000	11	0	0	10.19	10	0	0	10.58	3	0	1	202.40
REG	41	0	1	0.52	45	0	1	0.67	4	41	4	6.68
REG-FA-10	99	0	2353	0.00	97	1	1644	0.01	82	6	10702	0.16

- Having sparse matrices do not significantly affect the comparison
- FA works particularly bad for bad scaled problems for any value of bigM
- The proposed method outperforms existing ones in bad scaled problems

Conclusions

If you are solving a linear bilevel problem you have the following options:

- Dedicated methods: efficient but hard to code
- SOS1: global optimal, but computational time is extremely high
- REG: fast but only provides local optimal solutions
- FA with bigM: easy to implement, but setting bigM with trial-and-error method may provide suboptimal solutions
- Try to find better ways to set large constants as the one we propose

Thanks for the attention!

Questions?

OASYS
Optimization and Analytics for Sustainable energY Systems

S. Pineda, H. Bylling and J. M. Morales, "*Efficiently solving linear bilevel programming problems using off-the-shelf optimization software*", in **Optimization and Engineering**, 19(1), 187-211, 2018.

References I

Bard, J and J Moore (1990). "A Branch and Bound Algorithm for the Bilevel Programming Problem". In: *SIAM J. Sci. Stat. Comput.* 11.2, pp. 281–292.

Bard, J. F. (1991). "Some properties of the bilevel programming problem". In: *J. Optim. Theory Appl.* 68.2, pp. 371–378.

Bard, Jonathan F. and James E. Falk (1982). "An explicit solution to the multi-level programming problem". In: *Comput. Oper. Res.* 9.1, pp. 77–100.

Baringo, Luis and Antonio J. Conejo (2014). "Strategic Wind Power Investment". In: *IEEE Trans. Power Syst.* 29.3, pp. 1250–1260.

Bialas, W. F. and M. H. Karwan (1984). *Two-Level Linear Programming*.

Calvete, Herminia I., Carmen Galé, and Pedro M. Mateo (2008). "A new approach for solving linear bilevel problems using genetic algorithms". In: *Eur. J. Oper. Res.* 188.1, pp. 14–28.

References II

Colson, Benoît, Patrice Marcotte, and Gilles Savard (2005). "Bilevel programming: A survey". In: *4OR* 3.2, pp. 87–107.

Fletcher, Roger and Sven Leyffer (2004). "Solving mathematical programs with complementarity constraints as nonlinear programs". In: *Optim. Methods Softw.* 19.1, pp. 15–40.

Fortuny-Amat, José and Bruce McCarl (1981). "A Representation and Economic Interpretation of a Two-Level Programming Problem". In: *J. Oper. Res. Soc.* 32.9, pp. 783–792.

Gabriel, Steven A. and Florian U. Leuthold (2010). "Solving discretely-constrained MPEC problems with applications in electric power markets". In: *Energy Econ.* 32.1, pp. 3–14.

Garces, L.P. et al. (2009). "A Bilevel Approach to Transmission Expansion Planning Within a Market Environment". In: *IEEE Trans. Power Syst.* 24.3, pp. 1513–1522.

Hansen, P., B. Jaumard, and G. Savard (1992). "New branch-and-bound rules for linear bilevel programming". eng. In: *SIAM J. Sci. Stat. Comput.* 13.5, pp. 1194–1217.

Hu, X. M. and D. Ralph (2004). "Convergence of a Penalty Method for Mathematical Programming with Complementarity Constraints". In: *J. Optim. Theory Appl.* 123.2, pp. 365–390.

Jenabi, Masoud, Seyyed Mohammad Taghi Fatemi Ghomi, and Yves Smeers (2013). "Bi-Level Game Approaches for Coordination of Generation and Transmission Expansion Planning Within a Market Environment". In: *IEEE Trans. Power Syst.* 28.3, pp. 2639–2650.

Jeroslow, Robert G. (1985). "The polynomial hierarchy and a simple model for competitive analysis". In: *Math. Program.* 32.2, pp. 146–164.

Jiang, Yan et al. (2013). "Application of particle swarm optimization based on CHKS smoothing function for solving nonlinear bilevel programming problem". In: *Appl. Math. Comput.* 219.9, pp. 4332–4339.

References IV

Kazempour, S. Jalal et al. (2011). “Strategic Generation Investment Using a Complementarity Approach”. In: *IEEE Trans. Power Syst.* 26.2, pp. 940–948.

Li, Hecheng and Lei Fang (2012). “An evolutionary algorithm for solving bilevel programming problems using duality conditions”. In: *Math. Probl. Eng.* 2012.

Lv, Yibing et al. (2007). “A penalty function method based on Kuhn-Tucker condition for solving linear bilevel programming”. In: *Appl. Math. Comput.* 188.1, pp. 808–813.

Maurovich-Horvat, Lajos, Trine K. Boomsma, and Afzal S. Siddiqui (2015). “Transmission and Wind Investment in a Deregulated Electricity Industry”. In: *IEEE Trans. Power Syst.* 30.3, pp. 1633–1643.

Motto, A.L. L, J.M. M Arroyo, and F.D. D Galiana (2005). “A Mixed-Integer LP Procedure for the Analysis of Electric Grid Security Under Disruptive Threat”. In: *IEEE Trans. Power Syst.* 20.3, pp. 1357–1365.

References V

Pozo, David and Javier Contreras (2011). "Finding Multiple Nash Equilibria in Pool-Based Markets: A Stochastic EPEC Approach". In: *IEEE Trans. Power Syst.* 26.3, pp. 1744–1752.

Ralph, Daniel and Stephen J. Wright (2004). "Some properties of regularization and penalization schemes for MPECs". en. In: *Optim. Methods Softw.* 19.5, pp. 527–556.

Ruiz, C. and A.J. J Conejo (2009). "Pool Strategy of a Producer With Endogenous Formation of Locational Marginal Prices". In: *IEEE Trans. Power Syst.* 24.4, pp. 1855–1866.

Ruiz, Carlos, Antonio J. Conejo, and Yves Smeers (2012). "Equilibria in an Oligopolistic Electricity Pool With Stepwise Offer Curves". In: *IEEE Trans. Power Syst.* 27.2, pp. 752–761.

Scholtes, Stefan (2001). "Convergence Properties of a Regularization Scheme for Mathematical Programs with Complementarity Constraints". en. In: *SIAM J. Optim.* 11.4, pp. 918–936.

Shi, Chenggen, Jie Lu, and Guangquan Zhang (2005). "An extended Kth-best approach for linear bilevel programming". In: *Appl. Math. Comput.* 164.3, pp. 843–855.

Shi, Chenggen, Guangquan Zhang, and Jie Lu (2005). "On the definition of linear bilevel programming solution". In: *Appl. Math. Comput.* 160.1, pp. 169–176.

Shi, Chenggen et al. (2006). "An extended branch and bound algorithm for linear bilevel programming". In: *Appl. Math. Comput.* 180.2, pp. 529–537.

Siddiqui, S. and S. A. Gabriel (2012). "An SOS1-Based Approach for Solving MPECs with a Natural Gas Market Application". In: *Networks Spat. Econ.* 13.2, pp. 205–227.

Sinha, Ankur, Pekka Malo, and Kalyanmoy Deb (2013). "Efficient Evolutionary Algorithm for Single-Objective Bilevel Optimization". In: *arXiv* 1303.3901. arXiv: 1303.3901.

White, D. J. and G. Anandalingam (1993). "A penalty function approach for solving bi-level linear programs". In: *J. Glob. Optim.* 3.4, pp. 397–419.

Wogrin, Sonja, Efraim Centeno, and Julián Barquín (2011). "Generation capacity expansion in liberalized electricity markets: A stochastic MPEC approach". In: *IEEE Trans. Power Syst.* 26.4, pp. 2526–2532.

Zugno, Marco et al. (2013). "Pool Strategy of a Price-Maker Wind Power Producer". In: *IEEE Trans. Power Syst.* 28.3, pp. 3440–3450.