Open access • Journal Article • DOI:10.1287/OPRE.38.3.556

Computational Difficulties of Bilevel Linear Programming — Source link

Omar Ben-Ayed, Charles E. Blair

Institutions: University of Illinois at Urbana–Champaign

Published on: 01 May 1990 - Operations Research (INFORMS)

Topics: Linear programming, Exact algorithm, Analysis of algorithms and Heuristics

BEBR

FACULTY WORKING PAPER NO. 1432

Computational Difficulties of Bilevel Linear Programming

Omar-Ben Ayed
Charles E. Blair, III

College of Commerce and Business Administration
Bureau of Economic and Business Research
University of Illinois, Urbana-Champaign

BEBR

FACULTY WORKING PAPER NO. 1432

College of Commerce and Business Administration

University of Illinois at Urbana-Champaign

January 1988

Computational Difficulties of Bilevel Linear Programming

Omar Ben-Ayed, Graduate Student
Department of Business Administration

Charles E. Blair, III, Professor
Department of Business Administration

COMPUTATIONAL DIFFICULTIES OF BILEVEL LINEAR PROGRAMMING

Omar Ben-Ayed
Charles E. Blair, III
Department of Business Administration, University of Illinois at Urbana-Champaign, 339 Commerce West, Champaign, IL 61820

(January 1988)

We show, using small examples, that two algorithms previously published for the Bilevel Linear Programming problem (BLP) may fail to find the optimal solution and thus must be considered to be heuristics. A proof is given that solving BLP problems is NP-hard, which makes it unlikely that there is a good exact algorithm.

Bilevel Linear Programming (BLP) is a nested optimization model involving two problems, an upper one and a lower one; both problems have to be optimized given a jointly dependent set $S=\{(x,y)\geq 0: Ax+By\leq b\}$. The upper decision maker, who has control over x, makes his decision first, hence fixing x before the lower decision maker selects y. The general form of BLP can be defined as:

$$\begin{aligned} &\max_{x}\ c_1\cdot x + d_1\cdot y \\ &\text{where } y \text{ solves:} \\ &\quad \max\ d_2\cdot y \\ &\quad \text{such that:} \\ &\quad Ax + By \leq b \\ &\quad x,\ y \geq 0. \end{aligned} \tag{1}$$

In this paper, which is a part of a broader research on BLP [Ben-Ayed 1988], we study two algorithms: the Parametric Complementary Pivot Algorithm [Bialas-Karwan-Shaw 1980, and Bialas-Karwan 1984] and the Grid Search Algorithm [Bard 1983]; we show that those algorithms do not always find the optimal solution, and we point out some of their potential pitfalls. Finally, we prove that the problem of solving BLP is NP-hard; this a special case of a little-known result in Jeroslow [1985], with a simpler proof. The NP-hardness of BLP suggests that, as with integer programming problems (which are also NP-hard), algorithms involving some form of branching [Falk 1973, Gallo and Ülkücü 1977, Fortuny-Amat and McCarl 1981, Bialas and Karwan 1982, Papavassilopoulos 1982, Candler and Townsley 1982, Bard and Falk 1982, Bard and Moore 1987] are to be preferred.

1. The Parametric Complementary Pivot Algorithm (PCP)

The Parametric Complementary Pivot Algorithm (PCP) [Bialas-Karwan-Shaw 1980, and Bialas-Karwan 1984] is distinguished by its popularity and the large number of papers that refer to it. Most published BLP algorithms compare their efficiency to that of PCP.

When replacing the lower problem in (1) by its Kuhn-Tucker conditions after introducing dual variables u, slack variables z and surplus variables t, an equivalent formulation of the BLP problem can be obtained:

$$
\begin{aligned}
&\text{MAX } c_1 \cdot x + d_1 \cdot y \\
&\text{such that:} \\
&Ax + By + z = b \\
&B^T u - t = d_2 \\
&y_i t_i = 0 \\
&u_i z_i = 0 \\
&x,\ y,\ u,\ z,\ t \geq 0.
\end{aligned}
\tag{2}
$$

The PCP algorithm uses formulation (2). At each iteration, the algorithm tries to find a feasible solution that gives an objective function value α to the BLP problem by solving the following system:

$$
\begin{aligned}
&Ax + By + z = b \\
&\varepsilon I y + B^T u - t = d_2 \\
&c_1 \cdot x + d_1 \cdot y - s = \alpha \\
&y_i t_i = 0 \\
&u_i z_i = 0 \\
&x,\ y,\ u,\ z,\ t,\ s \geq 0
\end{aligned}
\tag{3}
$$

where s is a one-dimensional surplus variable, I is the identity matrix and ε is a positive scalar sufficiently small so that the solution to the above system is the same as when ε equals zero. In attempting to solve (3), Bialas et al. added the positive definite matrix εI to use a technique similar to that proposed by Wolfe [1959] in solving a system corresponding to convex quadratic programming problems.

Although the PCP algorithm may find the optimal solution for some BLP problems, this is not guaranteed. The following is an example for which PCP does not give the optimal solution:

$$\text{MAX } 1.5x_1 + 6y_1 + y_2$$
$$\text{where } y_1 \text{ and } y_2 \text{ solve:}$$
$$\text{MAX } y_1 + 5y_2$$
$$\text{such that:}$$
$$x_1 + 3y_1 + y_2 \leq 5$$
$$2x_1 + y_1 + 3y_2 \leq 5$$
$$x_1 \leq 1$$
$$x_1,\ y_1,\ y_2 \geq 0.$$

We will consider the problem of finding a solution with upper objective value $\alpha \geq 2$. The system of equations corresponding to (3) is:

$$x_1 + 3y_1 + y_2 + z_1 = 5$$
$$2x_1 + y_1 + 3y_2 + z_2 = 5$$
$$x_1 + z_3 = 1$$
$$.01y_1 + 3u_1 + u_2 - t_1 = 1$$
$$.01y_2 + u_1 + 3u_2 - t_2 = 5$$
$$1.5x_1 + 6y_1 + y_2 - s = 2$$
$$y_1 t_1 = y_2 t_2 = u_1 z_1 = u_2 z_2 = 0.$$

The PCP algorithm initializes by solving the LP obtained by ignoring the lower objective function. In this example, that gives $x_1 = y_2 = 0$, $y_1 = 1.667$. The complementary slackness conditions then require that $t_1=u_2=0$ and $u_1=.328$. The fifth constraint above is not satisfied, so we introduce an artificial variable w such that $.01y_1+u_1+3u_2-t_2+w = 5$. This gives the system:

$$
\begin{aligned}
y_1 + .333x_1 + .333y_2 + .333z_1 &= 1.667 \\
z_2 + 1.667x_1 + 2.667y_2 - .333z_1 &= 3.333 \\
z_3 + 1.0x_1 &= 1 \\
u_1 - .001x_1 - .001y2 + .333u_2 - .001z_1 - .333t_1 &= .328 \\
w + .001x_1 + .011y_2 + 2.667u_2 + .001z_1 + .333t_1 - 1.0t_2 &= 4.672 \\
s + .5x_1 + 1.0y_2 + 2.0z_1 &= 8.
\end{aligned}
$$

The algorithm performs pivoting operations on the above system in order to make w=0 while preserving the complementarity conditions. From the w-equation above, we see that entering x_1, y_2, u_2, z_1 or t_1 would decrease w. However, u_2 cannot enter because $z_2=3.333>0$. Similarly, z_1 and t_1 cannot enter. If we choose y_2 as the entering variable, z_2 leaves. At the next step, we may have u_2 enter (u_1 leaves), then z_1 enters to produce the system:

$$
\begin{aligned}
y_1 + .147x_1 - .059z_2 - .176s &= .059 \\
y_2 + .618x_1 + .353z_2 + .059s &= 1.647 \\
z_3 + 1.0x_1 &= 1 \\
u_2 - .001x_1 + 3.0u_1 + .001z_2 - 1.0t_1 + .002s &= .999 \\
w - .002x_1 - 8u_1 - .005z_2 + 3.0t_1 - 1.0t_2 - .006s &= 1.985 \\
z_1 - .059x_1 - .176z_2 + .471s &= 3.176.
\end{aligned}
$$

The only variable which would decrease w at this stage is t_1, which cannot enter because $y_1>0$. Thus, the PCP algorithm stops at this point with the conclusion that a solution to the problem with upper objective function value greater than or equal to 2 cannot be found. However, $x_1=y_2=1$, $y_1=0$ is such a solution.

In this small example, one could guess the optimal solution using hindsight. For example, if we temporarily allowed w to increase, we could have x_1 enter and y_1 leave in our last system, which would then allow t_1 to enter and give the desired solution. Also, we would have found the solution if x_1 entered instead of y_2 at the beginning. However this example is sufficient to show that the PCP approach is flawed. On larger problems, such "quick fixes" may not be available.

Bialas-Karwan-Shaw [1980] proposed for their algorithm a proof based on techniques similar to those used to prove Theorem 3 in Wolfe [1959]. However, the two situations are not identical. In particular, condition (e) in Bialas-Karwan-Shaw cannot be obtained in the same way as the corresponding condition in Wolfe´s paper, and this makes the proof invalid. A specific counter-example is available from the authors and is also included in Ben-Ayed [1988].

2. The Grid Search Algorithm (GSA)

The Grid Search Algorithm (GSA) was proposed by Bard [1983]. The author claimed that, for some τ^* between 0 and 1, the solution to a BLP problem is the same as the solution to the following parameterized LP:

$$\max_{x,y} \tau^*(c_1 \cdot x + d_1 \cdot y) + (1 - \tau^*)d_2 \cdot y$$

such that:

$$Ax + By \le b$$

$$x,\ y \ge 0.$$

In other words, by finding the value of τ^*, one can solve BLP as an equivalent LP. Unfortunately, the statement is not always true. For instance, there is no parameterized LP that gives the same optimal solution as the following BLP problem:

$$\max_{x} x + y$$

where y solves:

$$\max -y$$

such that:

$$4x + 3y \ge 19$$

$$x + 2y \le 11$$

$$3x + y \le 13$$

$$x,\ y \ge 0.$$

The GSA, intended to find τ^*, starts with the infeasible solution (3,4) when τ=1 (it is infeasible because substituting x by 3 and solving the lower problem would give a value for y that is different from 4). 3/5 is the only value of τ, between 0 and 1, that preserves the optimality of (3,4). The vertex (4,1) obtained with the new τ is feasible and is supposed to be the optimum according to GSA. However, the actual optimal solution is (1,5). In general, if the GSA is currently at the point $(\underline{x},\underline{y})$ such that: $d_2\underline{y}>d_2 y^*$ and $c_1\underline{x}+(d_1-d_2)\underline{y}=c_1 x^*+(d_1-d_2)y^*$, then the algorithm has no way to go to the optimal vertex (x^*,y^*).

Problems with GSA were independently found by F. A. Al-Khayyal, and P. Marcotte.

The GSA is very quick; it could be used to provide a lower bound for other algorithms such as those based on the branch and bound technique. However, this algorithm is risky for two reasons. First, as is the case for PCP, it does not tell whether the solution it gives is global or local. And second, it does not provide intermediate results (improved upper and lower bounds); if the algorithm is terminated before the stopping rule is met, no solution will be given, not even an approximation.

3. The BLP Problem is NP-Hard

The Knapsack Optimization problem can be defined as the problem of choosing from a given set of natural numbers $\{a_1, a_2, \ldots, a_n\}$ a subset that adds to the largest value not exceeding a given natural number β. It is well known that this problem is NP-hard (see for instance Garey and Johnson 1979). We now show that if we could always solve BLP quickly, we could solve Knapsack Optimization problem quickly.

One way to formulate the Knapsack Optimization problem is:

$$\begin{aligned} &\text{MAX} \sum_{i=1}^{N} a_i x_i \\ &\text{subject to:} \\ &\sum_{i=1}^{N} a_i x_i \leq \beta \\ &x_i = 0 \text{ or } 1. \end{aligned} \tag{4}$$

The requirement that x_i equal 0 or 1 can be enforced indirectly by allowing x_i to be any real between 0 and 1 and making the distance from x_i to the nearest integer as small as possible. That is, the constraints:

$$x_i = 0 \text{ or } 1$$

can be replaced by the requirement that x_i be an optimal solution to the problem:

$$\begin{aligned} &\text{MIN} \sum_{i=1}^{N} y_i \\ &\text{subject to:} \\ &y_i = \text{MIN} \{x_i, (1-x_i)\} \\ &0 \leq x_i \leq 1. \end{aligned}$$

Therefore the Knapsack Optimization problem (4) can be reformulated as the BLP:

$$\text{MAX} \sum_{i=1}^{N} a_i x_i - M \sum_{i=1}^{N} y_i$$

where the y_i solve:

$$\text{MAX} \sum_{i=1}^{N} y_i$$

such that: (5)

$$\sum_{i=1}^{N} a_i x_i \leq \beta$$

$$y_i \leq x_i$$

$$y_i \leq 1 - x_i$$

$$x_i \leq 1$$

$$x_i ,\ y_i \geq 0$$

where M is a large number to make the minimum of the sum of the y_i s equal to zero.

The following result shows that, if M is chosen sufficiently large, every non-integer x used to produce a feasible solution to the BLP (5) is inferior to an integer solution z. Since there are only finitely many feasible integer solutions, this implies the optimal solution to the (5) is integer.

For technical reasons, we will assume that $\text{MAX}\{a_i\} \geq 2$. We can do this since a Knapsack problem with all $a_i=1$ is trivial.

Theorem

Let $M > (\text{MAX}\{a_i\})^2$, $f(x)= \sum a_i x_i$, $g(x)= \sum \text{MIN}\{x_i ,1-x_i\}$. If x is feasible, and not all x_i are integer, then there is a feasible z with all z_i integer and:

$$f(x) - Mg(x) < f(z) - Mg(z) \qquad (6)$$

Proof

Let $Q = 1-1/\mathrm{MAX}\{a_i\}$. Our assumption implies that $Q \geq .5$. We modify the feasible x in a sequence of steps of three kinds:

(1) If $0 < x_j \leq Q$ for some $j=1$, make $x_j = 0$.

(2) If $Q < x_j, x_k < 1$ for some $j \neq k$, replace them by z_j, z_k so that $a_j z_j + a_k z_k = a_j x_j + a_k x_k$, $z_j + z_k \geq x_j + x_k$, and one of the two new values is 1 while the other is between 0 and 1.

(3) If $Q < x_j < 1$ for some j, with all other components integer, make $x_j = 1$.

Each of these steps increases the number of integer components of x, so we terminate with all components integer.

If we let z be obtained from x by a single use of step (1), $f(z) \geq f(x)-(\mathrm{MAX}\{a_i\})x_j$. If $x_j \leq .5$, $g(z) = g(x)-x_j$. If $.5 < x_j \leq Q$, $g(z) = g(x)+x_j-1 \leq g(x)-1/\mathrm{MAX}\{a_i\}$. In either case, (6) holds.

If z is obtained using step (3), we clearly have $f(z) > f(x)$ and $g(z) < g(x)$, so (6) is immediate.

If z is obtained using step (2) we have $f(z)=f(x)$. The requirement $z_j+z_k \geq x_j+x_k \geq 1$ implies that $g(z) < g(x)$ except in the special case in which $a_j=a_k$ and $z_j, z_k \geq .5$, in which case

$g(z)=g(x)$. However, we cannot obtain a z with all components integer by using only steps of this kind, since each such step leaves at least one component of x strictly between .5 and 1.

Thus, when we obtain z with all components integer, (6) will be satisfied. It remains to show that the final z is feasible, in particular that $\Sigma\, a_i x_i \le \beta$.

Steps (1) and (2) clearly preserve feasibility. To show that step (3) also does, note that $\Sigma\, a_i z_i < 1 + \Sigma\, a_i x_i \le 1+\beta$. Since β and $\Sigma\, a_i z_i$ are integer, $\Sigma\, a_i z_i \le \beta$. Q.E.D.

This result leaves little hope that a polynomial algorithm can be found for BLP, and suggests that the situation for BLP is similar to that for integer programming. In fact, BLP can be solved as a mixed integer programming problem [Fortuny-Amat and McCarl 1982], which makes it an NP-complete problem.

Acknowledgement: The authors were introduced to BLP by David E. Boyce, who applied Bilevel Programming to the study of Transportation Network Design problems [Boyce 1986 and LeBlanc-Boyce 1986].

References

BARD, J. F. 1983. An Efficient Point Algorithm for a Linear Two-Stage Optimization Problem. Operations Research 31, 670-684.

BARD, J. F. and J. E. FALK 1982. An Explicit Solution to the Multi-Level Programming Problem. Computers and Operations Research, 9, 1, 77-100.

BARD, J. F. and J. T. MOORE 1987. A Branch and Bound Algorithm for the Bilevel Programming Problem. Department of Mechanical Engineering, University of Texas, Austin, TX.

BEN-AYED, O. 1988. Bilevel Linear Programming: Analysis and Application to the Network Design Problem. Ph.D. Thesis. Department of Business Administration. University of Illinois, Urbana-Champaign.

BIALAS, W. F. and M. H. KARWAN 1982. On Two-Level Optimization. IEEE Transaction on Automatic Control AC-27, 1, 211-214.

BIALAS, W. F. and M. H. KARWAN 1984. Two-Level Linear Programming. Management Science 30, 8, 1004-1020.

BIALAS, W. F., M. H. KARWAN and J. P. SHAW 1980. A Parametric Complementary Pivot Approach for Two-Level Linear Programming. Research Report No. 80-2, Operation Research Program, Department of Industrial Engineering, State University of Buffalo.

BOYCE, D. E. 1986. Integration of Supply and Demand Models in Transportation and Location: Problem Formulations and Research Questions. Environment and Planning A 18, 485-489.

CANDLER, W. and R. TOWNSLEY 1982. A Linear Two-Level Programming Problem. Computers and Operations Research 9, 1, 59-76.

FALK, J. E. 1973. A Linear Max-Min Problem. Mathematical Programming 5, 169-188.

FORTUNY-AMAT, J. and B. McCARL 1981. A Representation and Economic Interpretation of a Two-Level Programming Problem. Journal of Operational Research Society 32, 783-792.

GALLO, G. and A. ÜLKÜCÜ 1977. Bilinear Programming: An Exact Algorithm. Mathematical Programming 12, 173-194.

GAREY, M. and D. JOHNSON 1979. Computers and Interactability. W. H. Freeman and Company, San Francisco.

JEROSLOW, R. B. 1985. The Polynomial Hierarchy and a Simple Model for Competitive Analysis. Mathematical Programming, 32, 146-164.

LEBLANC, L. J. and D. E. BOYCE 1986. A Bilevel Programming Algorithm for Exact Solution of the Network Design Problem with User-Optimal Flows. Transportation Research 20B, 3, 259-265.

PAPAVASSILOPOULOS, G. P. 1982. Algorithms for Static Stackelberg Games with Linear Costs and Polyhedra Constraints. Proceedings of the 21st IEEE Conference on Decision and Control, 2 and 3, 647-652.

WOLFE, P. 1959. The Simplex Method for Quadratic Programming. Econometrica, 27, 3, 382-398.