Open access • Journal Article • DOI:10.1007/BF00939562

Derivative evaluation and computational experience with large bilevel mathematical programs — Source link

Charles D. Kolstad, Leon S. Lasdon

Institutions: University of Illinois at Urbana–Champaign, University of Texas at Austin

Published on: 01 Jun 1990 - Journal of Optimization Theory and Applications (Springer New York)

Topics: Bilevel optimization and Implicit function

BEBR

FACULTY WORKING
PAPER NO. 1266

Derivative Evaluation and Computational Experience with Large Bi-Level Mathematical Programs

Charles D. Kolstad
Leon S. Lasdon

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

BEBR

FACULTY WORKING PAPER NO. 1266

College of Commerce and Business Administration

University of Illinois at Urbana-Champaign

July 1986

Derivative Evaluation and Computational Experience
with Large Bi-Level Mathematical Programs

Charles D. Kolstad, Assistant Professor
Department of Economics

Leon S. Lasdon
University of Texas at Austin

Abstract

A bi-level program is a mathematical program involving functions defined implicitly as solutions to another mathematical program. We discuss a method for extracting derivative information on the implicit function, which is especially efficient when the lower level problem has simple bounds on the variables and/or many inactive constraints. Computational experience on problems with up to 230 variables and 30 constraints is presented.

I. INTRODUCTION[1]

Over the past decade there has been an increase in interest in multi-level mathematical programming, and in particular bi-level mathematical programming. The bi-level problem consists of two parts, an upper and lower part. Define the upper level problem (denoted henceforth as "P1") as:

$$(\text{P1}): \qquad \min_{t} w(x,t) \tag{1a}$$

$$\text{s.t.} \quad f(x,t) \leq 0 \tag{1b}$$

where x(t) is implicitly defined by the lower-level problem:

$$(\text{B1}(t)): \quad x(t): \quad \min_{x} s(x,t) \tag{1c}$$

$$\text{s.t.} \quad g(x,t) \leq 0 \; . \tag{1d}$$

All variables and constraint functions may be vectors and all functions are assumed to be twice continuously differentiable in all arguments. A tremendous variety of applied problems, particularly economic problems, can be viewed as bi-level math programs. A Stackelberg duopoly or leader-follower continuous game (e.g., Chen and Cruz, 1972; Papavassilopoulos, 1981) can be viewed as a bi-level programming problem with the follower's problem corresponding to B1(t) and the leader's problem corresponding to P1. The principal-agent problem (Grossman and Hart, 1983) is also a bi-level programming problem. The principal (problem P1) specifies incentives or other controls for the agent who then acts according to B1. Outside the economics literature, the max-min problem (Danskin, 1966) is that of maximizing the minimum of some function and is thus a special case of bi-level programming.

Unfortunately, good solution methods for the bi-level problem are not generally available.[2] The implicitly defined function $x(t)$ may not be everywhere differentiable in which case the functions of the upper level problems will not be differentiable everywhere. In addition, without significant restrictions on the lower level problem, the upper level problem may not be convex.

Most algorithms for solving P1 use first derivatives of its objective and constraints. These are easily computed at points where $x(t)$ is a differentiable function of t, once $\nabla x(t)$ is known. In Fiacco (1968) and Fiacco and McCormick (1976), conditions for existence of $\nabla x(t)$ and methods for computing it are presented. The procedure for computing ∇x requires solving a linear system of size $n+m$, where n is the dimension of x and m the dimension of g. Moreover, simple bounds on the variables must be included in g. The purpose of this paper is to show how this procedure can be adapted efficiently to large problems, where B1 may have hundreds of x variables and/or constraints, and where many components of x have simple bounds. In addition, many of the constraints in g may be inactive at the optimum. Then, the size of the linear system which must be solved to compute ∇x can be significantly reduced. The purpose of this paper is to show how this reduction can be accomplished, and to present computational experience on problems B1 with up to 230 x variables, but only 2 or 3 t variables. We derive a system of linear equations for $\nabla x(t)$ whose dimension at a point $\bar{t}$ is $r+\ell$, where r is the number of active constraints of $B1(\bar{t})$ (not including bounds) and ℓ is the number of components of $x(\bar{t})$ not at a bound. By not including inactive constraints and variables at bound, the complexity of the $\nabla x(t)$ calculation can be dramatically reduced. We illustrate with a set of test problems involving a coal market cartel. The MINOS/augmented algorithm of Murtagh and Saunders is used to solve B1(t). P1 is treated as a differentiable problem and, being unconstrained, is solved by the BFGS quasi-Newton method VA13A from the Harwell Subroutine Library. Although the computations are generally successful, some difficulties are caused by ignoring the nondifferentiability of P1. These problems are illustrated with 3 dimensional plots of w(x(t),t).

II. BACKGROUND

Most applications of bi-level programming that have appeared in the literature are in the economics realm, particularly central economic planning. The typical situation is that there is a planner with some social objective and a set of policy instruments to use for controlling one (or more) economic agents with different objectives. See Kolstad (1985) for a more thorough review of applications and algorithms.

In the context of the previously defined bi-level problem, the "policy" problem (P1) is given by (1a)-(1b), where the planner minimizes w(x,t) subject to the constraints of (1b). The planner can only affect his objective by adjusting the vector t, which may be a set of taxes, quotas or some other instrument. The subordinate problem is given by (1c)-(1d) and, following Candler and Townsley (1982), is termed the "behavioral" problem (B1(t)). Given a vector of policies, t, the subordinate agent is assumed to optimize his objective s(x,t) by adjusting the vector x. Obviously whatever x is chosen in the subordinate problem influences the planner's objective.

In the economics literature the subordinate problem (B1(t)) often serves a very specific purpose. It has been known for some time that the operation of a portion of a competitive economy can be simulated using mathematical programming (Takayama and Judge, 1971). It is thus common that the subordinate problem (B1(t)) is a single mathematical program simulating the decentralized market processes of a competitive economy. The policy problem might be to choose a tax or quota to achieve social objectives. The effect of a per-unit tax on such an economy can be simulated by subtracting a term for tax payments from the objective. A quota system applied in an economically efficient manner can be simulated by adding appropriate constraints to the problem.[3] It is within this framework that most economic applications of bi-level mathematical programming occur: an overall social objective (the planning problem) subject to equilibrium in a market economy (the behavioral problem) with communication between the two levels through taxes, quotas or some other set of policy instruments.

In spirit, the bi-level problem has a long history in economics--social objectives vs objectives of individual agents. The earliest explicit discussion in the economics literature of bi-level math programming appears to be Candler and Norton (1977a). They consider a numerical example of a milk producing monopoly in the Netherlands, regulated by the government. The behavioral problem represents the objectives of the monopoly as that of maximizing revenue from sales of milk, butter and cheeses. The government is assumed to have a composite objective involving consumer prices, government outlays and farm income. Other applications of bi-level programming have been suggested by Candler et al. (1981), principally in the area of development planning.

Another set of problems in the area of environmental regulation has motivated several authors to research the question of bi-level programming. The problem is to drive polluters to efficient levels of emissions through an emissions tax. The same tax (per unit of emissions) applies to many different sources of pollution in a region even though each source contributes in a different way to concentrations of pollution in the environment, due to locational differences and transport of pollutants by the environment. This problem was encountered by Schenk et al. (1980) for the case of water pollution and Kolstad (1986) for air pollution.

A very different problem was explored by Falk and McCormick (1982): that of a cooperative game. Their problem is that of an imperfect cartel of several countries in the international coal market. Since in an imperfect cartel, side-payments are not permitted, cartel objectives may not be to maximize joint profits. Falk and McCormick utilize Nash's solution to this bargaining problem. If u_i is the i^{th} cartel member's gain from joining the cartel (relative to his profit in a noncooperative setting), then the Nash solution is to maximize $\prod_i u_i$, the product of the u_i's. Falk and McCormick formulate this as a bi-level problem, utilizing a very simple competitive model of coal trade as the subproblem B1(t). The upper level problem (P1) is Nash's product of individual gains from cartelization, $\prod_i u_i$. Using a numerical example with a two-member cartel, Falk and McCormick demonstrate that two local maxima exist for the overall problem only one of which is a global maximum.

At least a dozen different algorithms for solving the bi-level problem appear in the literature. Most fall into three classes. One class is concerned exclusively with the linear bi-level problem. These algorithms are concerned with efficiently moving from one extreme point of B1 to another until an optimum is found (Bialas and Karwan, 1982; Candler and Townsley, 1982; Papavassilopoulos, 1981). Another set of algorithms utilizes the Kuhn-Tucker-Karush conditions on the subproblem B1 as constraints on the overall problem, thus turning the bi-level problem into a nonconvex single mathematical program (Bard, 1983a, b; Bard and Falk, 1982; Fortuny-Amat and McCarl, 1981; Bialas and Karwan, 1982). A third set of algorithms is based on descent approaches for the policy problem with gradient information from the subproblem acquired in a variety of ways (Shimizu and Aiyoshi, 1981; DeSilva, 1978). It is in this latter class that the solution algorithm of this paper falls.

III. COMPUTING $\nabla x(t)$

The major problem in solving P1 is that $x(t)$ is defined implicitly as the solution to $B1(t)$. We consider only cases where $x(t)$ is a differentiable function of t almost everywhere, and focus on the problem of computing $\nabla x(t)$. If $\nabla x(t)$ is known, first derivatives of $w(x(t),t)$ and $f(x(t),t)$ are easily computed. If $x(t)$ is a point-to-point map then it is generally continuous for all t of interest (Hogan, 1973); however, ∇x is usually not continuous at points t where the active set of B1(t) changes. If this fact is ignored, P1 may be treated as a differentiable problem, for which several efficient algorithms are available. Problems caused by ignoring the nondifferentiable nature of P1 are discussed in section IV. We now consider the procedure for computing ∇x.

∇x is computed using an adaptation of the methods and theory presented in Fiacco (1976). We first rewrite B1 to isolate simple bounds and include equality constraints:

$$(B2(t)): \quad \min_x s(x,t) \tag{2a}$$

$$\text{s.t.} \quad g(x,t) \left\{ \begin{matrix} = \\ \le \end{matrix} \right\} 0 \tag{2b}$$

$$\ell \le x \le u \tag{2c}$$

where some of the constraints in (2b) may be equalities and others may be inequalities. Let $B2(\bar{t})$ have a solution $(x^*,\pi^*,w^*) = z^*$ where π^*,w^* are multipliers for (2b) and (2c) respectively. The following assumptions, taken from Fiacco (1976), guarantee that z^* is a continuously differentiable function of t for all t in a neighborhood of $\bar{t}$.

Assumption 1

1. The solution z^* is unique and satisfies the second order sufficiency conditions.
2. Gradients of all active constraints in $B1(\bar{t})$ (including bounds) are independent.
3. Strict complementarity holds, i.e., any active inequality constraint (including bounds) has a positive multiplier.

Following the development in Fiacco (1976), we develop formulas for $\nabla x(t)$.

Theorem 1: Let Assumption 1 hold at $\bar{t}$. Without loss of generality, let $x^*(\bar{t})$ be partitioned into (y,z) such that all components of y lie strictly between their bounds and all components of z are at a bound. Partition the constraints $g(x^*,t)$ into binding constraints $b(x^*,t)$ and non-binding constraints $n(x^*,t)$. Similarly, partition π^* into (u,v), corresponding to binding and non-binding constraints respectively (thus $v = 0$). For any element of t, say t_k, the derivatives $\frac{dx^*}{dt_k}$ and $\frac{d\pi^*}{dt_k}$ satisfy:

$$\frac{dz}{dt_k} = 0 \tag{3a}$$

$$\frac{\partial v}{\partial t_k} = 0 \tag{3b}$$

$$\begin{pmatrix} \nabla_y^2 L & [\nabla_y b]' \\ \nabla_y b & 0 \end{pmatrix} \begin{pmatrix} \frac{dy}{dt_k} \\ \frac{du}{dt_k} \end{pmatrix} = - \begin{pmatrix} [\nabla_y \frac{\partial L}{\partial t_k}]' \\ \frac{\partial b}{\partial t_k} \end{pmatrix} \tag{3c}$$

where all derivatives are evaluated at $(x^*,\bar{t})$, L is the Lagrangian function of B1,

$$L = s(x,t) + \sum_{i \in B^*(\bar{t})} u_i b_i(x,t) = s(x,t) + u'b(x,t) \tag{4a}$$

and $B^*(\bar{t})$ is the set of active constraints not including bounds

$$B^*(\bar{t}) = \{ i \mid g_i(x^*,\bar{t})=0 \} \tag{4b}$$

Pf: From assumption 1, there exists a neighborhood of $\bar{t}$ within which all inactive constraints remain inactive and all active constraints remain active. Thus, (3a) and (3b) follow directly, and the Kuhn-Tucker-Karush conditions can be written for B2(t), ignoring bounds and inactive constraints. The Lagrangian of this problem (B2(t)) is (4a), and the first-order optimality conditions are

$$\nabla_y L = \nabla_y s(x^*,\bar{t}) + u'\nabla_y b(x^*,\bar{t}) = 0 \tag{5a}$$

$$b(x^*,\bar{t}) = 0. \tag{5b}$$

Note that if variable bounds were included in (4a), they would not appear in (5). This is the rationale for excluding them from (5).

By assumption 1, (5) holds for all points t in some neighborhood of $\bar{t}$. Hence the first derivatives of (5) will respect to any component of t, say t_k, may be set to zero in this neighborhood, yielding

$$\nabla_y^2 s \frac{dy}{dt_k} + \frac{\partial}{\partial t_k}(\nabla_y s)' + \sum_{i \varepsilon B^*(\bar{t})} u_i [\nabla_y^2 b_i \frac{dy}{dt_k}$$

$$+ \frac{\partial}{\partial t_k}(\nabla_y b_i)'] + (\nabla_y b)' \frac{du}{dt_k} = 0 \tag{6a}$$

$$\nabla_y b \frac{dy}{dt_k} + \frac{\partial b}{\partial t_k} = 0 . \tag{6b}$$

where all derivatives are evaluated at (x*,t).

Note that there are the same number of equations (6a) as elements of y. There are the same number of equations (6b) as active constraints. These equations can be rewritten as

$$\{\nabla_y^2 s + \sum_{i\varepsilon B^*(\bar{t})} u_i \nabla_y^2 b_i\} \frac{dy}{dt_k} + (\nabla_y b)' \frac{du}{dt_k}$$

$$= - \frac{\partial}{\partial t_k} [\nabla_y s + u'\nabla_y b]' \tag{7a}$$

$$\nabla_y b \frac{dy}{dt_k} = - \frac{\partial b}{\partial t_k} \tag{7b}$$

The bracketed part of (7a) is $\nabla_y L$ and the term in braces is $\nabla_y^2 L$ where L is defined in (4b), so (7) can be rewritten to yield (3c). The matrix in (3c) is nonsingular because the second order sufficiency conditions for B2 are satisfied at $\bar{t}$ (see Fiacco and McCormick (1968)). Thus (3c) can be solved for the required derivatives.

Theorem 1 is the basic result of this section and constitutes the core of the algorithm used to solve the bi-level programming problem. The difference between (3c) and (2.3) in Fiacco (1976) is the elimination of inactive constraints and variables at bounds from the calculations. This can greatly decrease the complexity of computing $\nabla_t x(t)$, particularly in problems involving a large number of nonbasic variables or inactive constraints (e.g., spatial equilibrium problems).

IV. COMPUTATIONAL EXPERIENCE

Our test problems are variants of those encountered by Falk and McCormick (1982) in that we consider a spatial model of the international coal market as our behavioral problem B1(t). The model and data are described more fully in Kolstad and Abbey (1984). In essence, we are considering a spatial market for a single good with multiple producers and consumers trading through a costly transportation network. Each consumer has a constant elasticity demand function for the good (i.e., $q = Ap^{\varepsilon}$) and each producer's incremental (marginal) costs are a linear function of quantity produced.

A subset of the producers in this market join together to form a perfect (i.e., side payments allowed) cartel. The cartel's problem is to determine how much to raise the price of their product, over cost, in order to maximize cartel profits. The policy problem below, P3, is merely a maximization of cartel monopoly profits, where, for producer i, t_i is the unit-price markup over marginal cost, $s_i(t)$ is the resulting quantity produced and the set C identifies the producers that belong to the cartel. Note that P3 is unconstrained:

$$\text{(P3)} \qquad \max_{t_i} \sum_{i \in C} t_i s_i(t) \qquad (7)$$

where $s_i(t)$, sales from producer i, is determined by the market. Thus $s_i(t)$ can be found by solving

$$\text{(B3(t))}: \quad \max_{r,s,q} \Big[\sum_{j=1}^{J} \alpha_j \int_0^{r_j} x^{\beta_j}\, dx - \sum_{i=1}^{I} \int_0^{s_i} (a_i + b_i x)\, dx$$

$$- \sum_{i=1}^{I} \sum_{j=1}^{J} \tau_{ij} q_{ij} - \sum_{i \in C} t_i s_i \Big] \qquad (8a)$$

$$\text{s.t.} \qquad \sum_j q_{ij} \leq s_i \qquad i = 1,\ldots,I \qquad (8b)$$

$$\sum_i q_{ij} \geq r_j \qquad i = 1,\ldots,J \qquad (8c)$$

$$s_i, r_j, q_{ij} \geq 0 \qquad i = 1,\ldots,I; \quad j = 1,\ldots,J \qquad (8d)$$

The first term of the objective function of B3(t) is the area under the (constant elasticity) demand curves for all the demand regions (j). The second term is the area under the (linear) marginal cost curves for all the suppliers (i). The third term is the transport cost for moving the good (coal) from i to j where q_{ij} is quantity moved from i to j. The fourth term represents the cartel monopoly profits. Equation (8b) assures that shipments out of supplier i are no more than supplier i's production. Equation (8c) assures that consumer j receives at least as much of the good as he demands in total. In most cases, all of the constraints in these two sets will be binding.

We have solved three versions of this problem using an algorithm with gradient information obtained as described in section III. All three problems have the same structure but are of different size. In problems A and B the cartel has two members whereas in problem C there are three cartel members. In problem A there are six producers and consumers, whereas in problems B and C there are thirty (see Table I).

To solve the subproblem B3, we used MINOS/Augmented (4.0), a general-purpose efficient nonlinear program solver (Murtaugh and Saunders, 1981). Default values of tolerances, etc. were used which implies among other things, that a conjugate-gradient algorithm was used.

The choice of algorithm to solve the upper level problem is particularly important because of the time-consuming nature of solving the subproblem. In our example, problem P1 is unconstrained. Thus we chose to use the VA13A code from the Harwell Subroutine Library, a BFGS variable metric algorithm, with a convergence tolerance[4] of 10^{-5}. Eqn (3b) was solved using the LINPAK linear algebra routines.

Details of the iteration-by-iteration convergence of the algorithm for these three examples are given in Table II. Although a moderately tight convergence tolerance was specified, the algorithm converged fairly rapidly and in a modest amount of time (5 to 60 seconds on a Cray I - see table I). However, the performance of the algorithm depended on the starting point. As a case in point, in problem B, when the algorithm starts at (0,0), a flat spot in the objective function is encountered leading to a local rather than global maximum.

These difficulties can be better understood by examining problem A graphically. Figure 1 shows the behavior of s_1 and s_2 as functions of t_1 and t_2. As can be seen, there are flat spots. These lead to flat spots in the objective of P3 as shown in Figure 2 where the objective function is plotted as a function of t_1 and t_2 and shown from two perspectives. Starting at (0,0), no problems are encountered, as indicated in Table II. However, if the problem is started at (34,18), the algorithm may terminate without finding the global maximum (Fig. 2). It is quite possible that one might believe a solution had been found.

The existence of such flat spots is not a quirk of the test problem considered here. There are two reasons for flat spots. If a cartel markup rate is too high, the producers will sell nothing. Thus the ridge along $t_2 = 18$, $t_1 \geq 24$ corresponds to $q_1 = 0$. Adjustment in t_1 above 24 lead to no change in q_1 nor in cartel profits. When problem B is started at (0,0) it rapidly wanders out to (33,54) and stays at that flat spot, even though the answer is "between" these two values. Another reason for flat spots is that producer one and two's markets may not be strongly interrelated; i.e., as producer 1 raises his markup, he will drive consumers to other producers, but possibly not to producer 2. Thus $\partial q_2/\partial t_1 = 0$.

One way of avoiding these problems is to check second order sufficiency conditions at any optimum. This may not be satisfactory, however, not only because of the difficulty in obtaining second derivative information from B3(t), but also because these conditions may not be satisfied at or near a (nonunique) global optimum if it occurs at a flat spot.

V. CONCLUSIONS

The primary contribution of this paper is in our presentation of an efficient method of obtaining derivative information on solutions of large nonlinear programs in order that large bi-level programs may be easily solved. We have also tested our methods in several contexts and found that while the method works and is efficient, there are potential pitfalls, principally related to encountering flat spots and sharp ridges in the objective function of the policy, or first-level, mathematical program.

FOOTNOTES

[1]Computational support from Robert Bivins and Myron Stein is gratefully acknowledged. Comments from Jon B ard are appreciated. Work supported in part by US Department of Energy through the Los Alamos National Laboratory.

[2]It is important to distinguish the general bi-level programming problem from the many decomposition techniques which have been in use for a number of years (particularly the methods of Dantzig and Wolf and Geoffrion). These methods are all concerned with breaking down a large mathematical program into a number of smaller, more tractable units. An important aspect of these methods is a coincidence among the objectives of the multiple levels. In all cases the decomposed program can also be written as a single mathematical program. This is not the case for the general problem of bi-level programming.

[3]A quota is a restriction on overall output from a particular sector of the economy. Within an optimization model of a competitive economy, it would be represented as a constraint on aggregate output. In practice, the quota would have to be translated to the firm level through a license system or some other mechanism. For an aggregate constraint to realistically represent the action of a quota, the licenses must be allocated to firms in an economically efficient manner. This can be assured by allowing private trading of licenses among firms.

[4]Convergence achieved when changes in each independent variable of 10^{-5} fail to improve objective function.

REFERENCES

J. Bard, "An Algorithm for Solving the General Bilevel Programming Problem, Mathematics of Operations Research, 8: 260-272 (1983).

J. Bard, "An Efficient Point Algorithm for a Linear Two-Stage Optimization Problem," Operations Research, 31: 670-684 (1983).

J. F. Bard and J. E. Falk, "An Explicit Solution to the Multi-Level Programming Problem," Comput. & Ops. Res., 9 (1), 77-100 (1982).

W. F. Bialas and M. H. Karwan, "On Two-Level Optimization," IEEE Trans. Auto. Cont., AC-27 (1), 211-214 (Feb. 1982).

Wilfred Candler and Roger D. Norton, "Multi-Level Programming," World Bank Development Research Center Discussion Paper #20, Washington, DC (Jan. 1977).

W. Candler, J. Fortuny-Amat and B. McCarl, "The Potential Role of Multilevel Programming in Agricultural Economics," Amer. J. Agric. Econ. 63 (3), 521-531 (1981).

W. Candler and R. Townsley, "A Linear Two-Level Programming Problem," Comput. & Ops. Res., 9 (1), 59-76 (1982).

C. I. Chen and J. B. Cruz, Jr., "Stackelberg Solution for Two-Person Games with Biased Information Patterns," IEEE Trans. Auto. Cont., AC-17 (6), 791-798 (1972).

J. W. Danskin, "The Theory of Max-Min with Applications," J. SIAM Appl. Math. 14 (4), 641-664 (1966).

James E. Falk and Garth P. McCormick, "Mathematical Structure of the International Coal Trade Model," US Department of Energy Report DOE/NBB-0025, Washington, DC (Sept. 1982).

A. V. Fiacco, "Sensitivity Analysis for Nonlinear Programming Using Penalty Methods," Math. Prog., 10, 287-311 (1976).

A. V. Fiacco and G. P. McCormack, Nonlinear Programming: Sequential Unconstrained Minimization Techniques (John Wiley, New York, 1968).

J. Fortuny-Amat and B. McCarl, "A Representation of a Two-Level Programming Problem," J. Opl. Res. Soc., 32, 783-792 (1981).

S. J. Grossman and O. D. Hart, "An Analysis of the Principal-Agent Problems," Econometrica, 51 (1), 7-45 (Jan. 1983).

W. W. Hogan, "Point-to-Set Maps in Mathematical Programming," SIAM Review, 15 (3), 591-603 (1973).

C. D. Kolstad and D. S. Abbey, "The Effect of Market Conduct on International Steam Coal Trade," Eur. Econ. Rev., 24, 29-59 (1984).

C. D. Kolstad, "Empirical Properties of Economic Incentives and Command-and-Control Regulations for Air Pollution Control," forthcoming in Land Economics (1986).

C. D. Kolstad, "A Review of the Literature on Bi-level Mathematical Programming," Los Alamos National Laboratory Report LA-10284-MS, Los Alamos NM (October, 1985)--available from NTIS, Springfield, Virginia.

J. Marschak, "Economic Measurement for Policy and Prediction," in W. C. Hood and T. C. Koopmans (eds.) Studies in Econometric Method Cowles Foundation Monograph #14 (Yalé University Press, New Haven, 1953).

Bruce A. Murtagh and Michael A. Saunders, "A Projected Lagrangian Algorithm and its Implementation for Sparse Nonlinear Constraints," Systems Optimization Laboratory Report SOL 80-1R, Department of Operations Research, Stanford University, Stanford, CA (February 1981).

G. P. Papavassilopoulos, "Solution of Some Stochastic Nash and Leader-Follower Games," SIAM J. Cont. & Opt., 19 (5), 651-666 (Sept. 1981).

G. W. Schenk, W. F. Bialas and M. H. Karwan, "A Multilevel Programming Model for Determining Regional Effluent Charges," forthcoming, Water Resources Research.

K. Shimizu and E. Aiyoshi, "A New Computational Method for Stackelberg and Min-Max Problems by Use of a Penalty Method," IEEE Trans. on Auto. Control, AC-26 (2), 460-466 (Apr. 1981).

Anura 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," D. Sc. dissertation, George Washington University, Washington, DC (Jan. 1978).

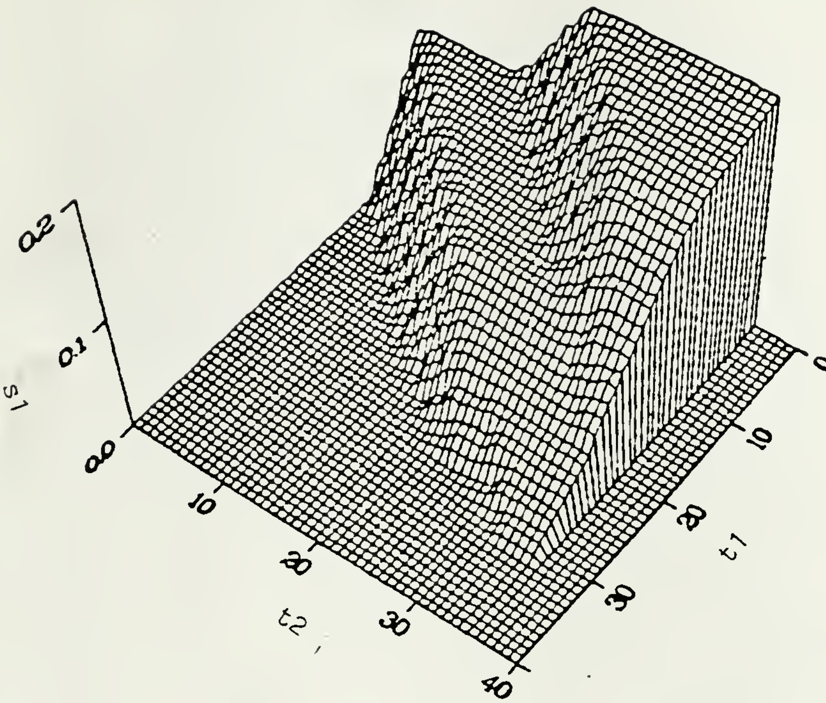
T. Takayama and G. Judge, Spatial and Temporal Price and Allocation Models (North-Holland, Amsterdam, 1971).

TABLE I: CHARACTERISTICS OF THE TEST PROBLEMS

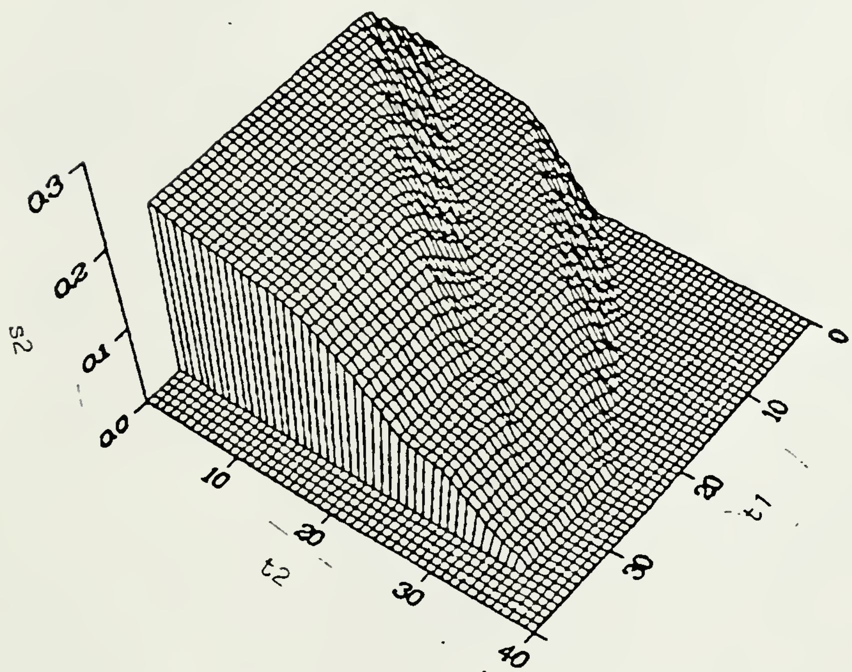
	Problem A	Problem B	Problem C
Subproblems			
Constraints	6	30	30
Variables	14	230	230
I	4	10	10
J	2	20	20
C	$\{1,2\}$	$\{1,2\}$	$\{1,2,3\}$
CPU Time in Seconds,			
CRAY I	5.269	35.178 ((20,15) starting point)	59.036

TABLE II: ITERATION LOG FOR SOLUTION OF THREE TEST PROBLEMS

Iteration	Cumulative Subproblem Calls	P3 Objective	t_1	t_2	t_3
Problem A:					
0	1	0	0	0	
1	5	3.09805	17.46	18.85	
2	9	3.16631	15.41	25.93	
3	12	3.37351	18.29	23.51	
4	18	3.38987	18.38	23.88	
5	19	3.41301	17.45	23.82	
6	22	3.41751	16.21	23.89	
7	23	3.42640	16.32	23.89	
9	26	3.42762	16.32	23.83	
11	29	3.42882	16.32	23.85	
14	35	3.42903	16.32	23.86	
15	36	3.42903	16.32	23.86	
Problem B:					
0	1	5.19018	20.00	15.00	
1	2	5.36747	19.79	16.00	
2	6	6.03113	44.10	38.80	
3	7	6.49260	35.64	30.56	
4	9	6.59798	37.56	32.61	
5	10	6.62623	36.72	33.22	
6	15	6.68452	36.49	31.99	
8	23	6.68869	36.65	32.00	
11	31	6.69085	36.63	32.00	
13	37	6.69097	36.63	32.00	
Problem B (started at (0,0)):					
0	1	0	0	0	
1	3	3.27900	10.00	6.79	
3	5	5.03343	18.55	42.62	
5	11	6.09682	34.02	55.21	
7	17	6.13779	33.04	54.41	
9	26	6.14104	32.95	54.37	
12	37	6.14360	32.94	54.37	
Problem C:					
0	1	6.26882	20.00	15.00	10.00
1	3	7.46470	20.15	23.69	12.64
2	5	9.99682	56.32	55.55	34.93
3	9	9.99879	56.37	55.60	35.08
4	11	10.0234	56.32	55.51	35.34
6	18	10.0778	56.29	55.42	35.91
7	22	12.1085	44.66	43.51	90.26
8	31	12.5706	51.86	50.79	57.21
10	41	12.6235	51.76	50.77	57.26
13	48	12.6396	51.77	51.10	55.60
16	56	12.6468	51.72	51.23	55.01
18	62	12.6488	51.76	51.23	55.04



(a) s1



(b) s2

Figure 1: s1 and s2 as functions of t1 and t2, Problem A

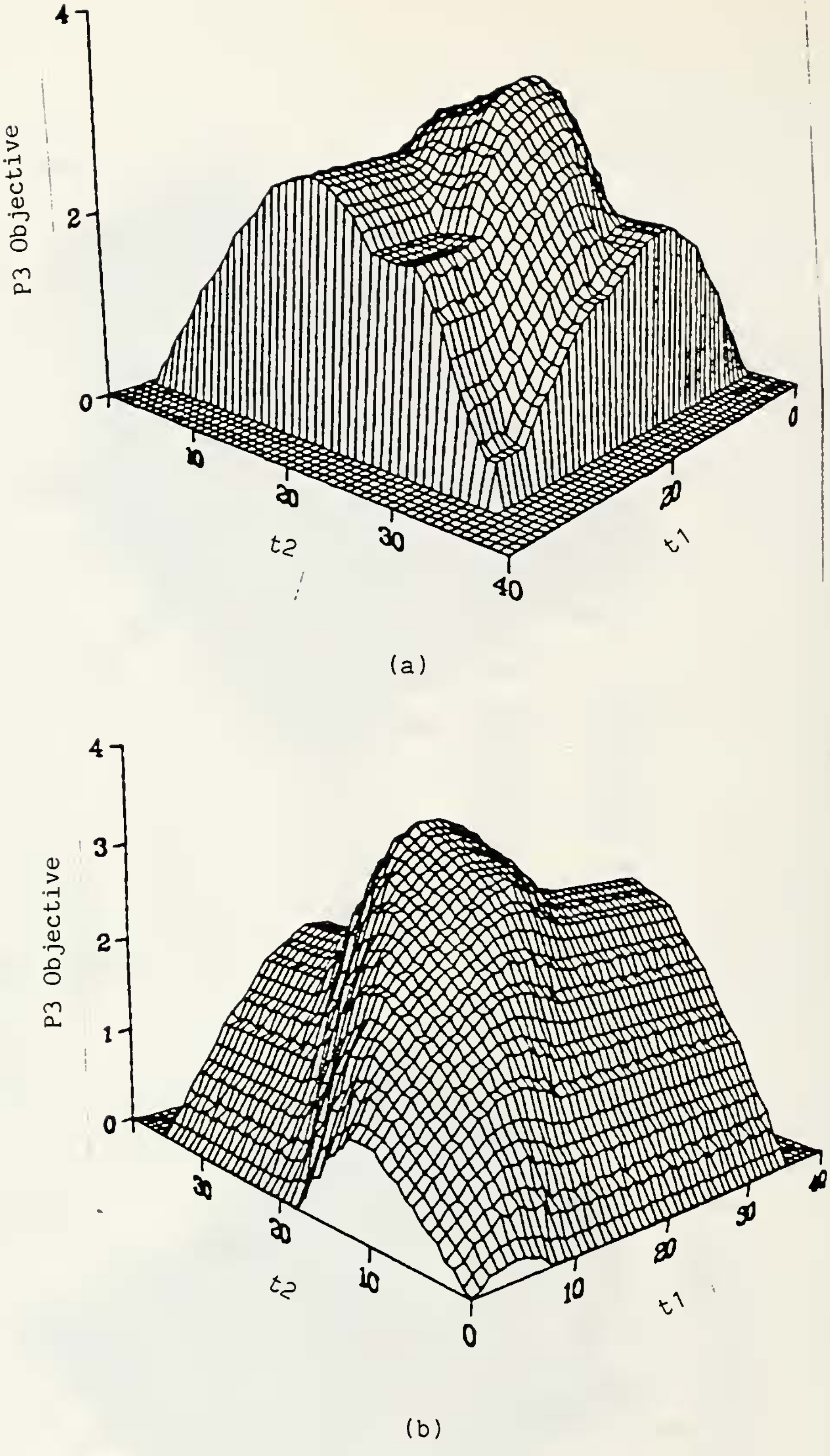


Figure 2 : P3 Objective as function of t1 and t2, from two perspectives: Problem A