Disc. paper 407
ISSN 0316-5078

INSTITUTE FOR ECONOMIC RESEARCH

QUEEN'S UNIVERSITY

Kingston, Ontario, Canada K7L 3N6

TWO NOTES:

1. On User Solution Strategy for Mixed-Integer Linear Programming Models

2. On the Solution of Spatial Price and Allocation Models

#407

by

John Rowse

Queen's University and
the University of Calgary

November 1980

A Note on User Solution Strategy for Mixed-Integer Linear Programming Models

ABSTRACT

Large scale mixed-integer linear programming (MILP) models may easily prove extraordinarily difficult to solve, even with efficient commercially-implemented MILP solution codes. Drawing on experience gained in solving and analyzing three intertemporal investment planning MILP models for electric power supply, this note offers several practical suggestions for reducing computer solution times for general production-allocation MILP models. Solution time reduction stems from judicious use of the powerful computational capabilities of existing commercial LP codes in conjunction with information known or to be learned by the practitioner about the model's structure.

Introduction

Mixed-integer linear programming (MILP) problems have been formulated and solved for over two decades.[1] As with all OR methods, MILP's have increasingly been applied in diverse fields, due in part to the diffusion of knowledge relating to their usefulness (see, for example, [5] [9] [10] [11] [18]) and in part to the extension of commercial LP codes to solve MILP's (see, for instance, [2] [4] [8]). A safe prognosis would seem to be for further growth in their popularity. What seems to be a stark fact of life at present, however, is that it is possible to formulate interesting and useful large MILP models which can be solved to optimum only within a computing time limit beyond the reach of most research budgets. Therefore, virtually any methods that allow reduction of solution time for large problems are to be welcomed, provided they do not force the practitioner to incur enormous costs in other ways, such as requiring one to acquire expertise in algorithmic design and implementation. Woolsey [16] [17] offers excellent advice on practical aspects of formulating and solving integer programming models, and most manuals for commercial LP packages likewise provide useful guidelines for formulating and framing MILP models for computer processing. Typically, however, the most efficient use of such systems requires substantial knowledge of the character of the optimum before solution commences. This note complements the advice offerred in these sources by making several common sense suggestions concerning an MILP solution strategy, with the powerful computational capabilities of existing commercial LP codes in mind. Though couched in terms of an intertemporal investment planning model, these suggestions are intended to be fully general for production-allocation type models, and are relevant whether the objective of model formulation is outright optimization, suboptimization to explore the neighborhood of the optimum, or simulation of the model to ascertain its properties. (See [10]). Further, they are intended for large-scale problems, which minimizes their utility for MILP models solvable readily by existing codes.

The Application Motivating the Suggestions

The experience gained in analyzing three MILP models constructed to examine in depth the extent to which co-operation in power supply between two neighboring electric utilities in Canada might potentially reduce joint costs of supply, motivates the following suggestions. The two utilities involved, Manitoba Hydro and the Saskatchewan Power Corporation, are natural candidates for joint expansion, since their supply systems are highly complementary: the former is almost totally hydro oriented, while the latter is predominantly thermal oriented. MILP formed a natural analytical tool for pinpointing and quantifying supply economies, in part because the approach was known and well understood (see [1] [13] [14] [15]). In addition, the investment alternatives considered were discrete and construction of certain of the prospects was conditional upon construction of others, implying the need to model precedence relationships. Such relationships were easily included in an MILP framework. Three cost-minimizing models were formulated, one for each of the utility systems planning for self-sufficiency in supply -- to be solved to provide a reference standard for measuring supply economies -- and a third for joint planning. The joint model allowed for three conceptually different sources of economies: (1) staggering or sequencing of capacity additions between the utilities; (2) one utility could draw upon surplus reserves of the other in order to permit deferral of capacity installation; and (3) phasing of generation expansion whereby the joint system could first develop the most favorable alternatives, then the next most attractive, and so on. Due to the complexity of the problem, the third model possessed 393 constraints and 459 decision variables (excluding slacks), of which 175 variables were discrete. Thus, the joint model constituted a very large MILP model, and the initial expectation was that it would prove exceedingly difficult if not impossible to solve. In fact, even the much smaller self-sufficiency models proved difficult to solve individually.

Suggestions for an MILP Solution Strategy

I Assemble the model in a manner facilitating easy model revision.

An integral part of this procedure should be the careful selection of a naming convention for variables and constraints. The naming convention should readily lend itself to expanding or contracting the number of decision variables and constraints, and through its mnemonics be immediately comprehensible to the analyst, since interface between the user and the solution procedure lies at the heart of solution time reduction. The flexibility option allows the analyst scope to change model structure and size -- such as the number of demand and/or supply nodes, transshipment points, investment alternatives, time periods, etc. -- as information is learned about the model feasible point set (FPS). MILP model size and complexity are seldom engraved in stone. Rather, they will normally vary with the questions to be answered, the solution detail considered necessary, and in no small degree the extent of research-budget funds earmarked for computing.

For the power supply application, circumstances suggested writing a flexible matrix generator program from the start. The wisdom of this step was really appreciated, however, only when intensive computing was undertaken. At the outset, model size was purposely considered elastic, because of expected difficulties with computer solution. Quantification of economies demanded a horizon distant enough to allow the joint system to evolve toward the most attractive supply configuration for both utilities. The legacy of history in the form of existing capacity meant that such movement could only occur slowly, and thus a lengthy planning period appeared warranted. Another consideration was that each utility plans generation installation to meet an annual winter peak. Thus, economies deriving from the staggering of generation capacity increments could not be accurately measured with a time period in excess of one year. A model with a distant horizon embracing many short time periods in sequence therefore appeared necessary. The needs of solvability dictated a model as small as possible, which resulted in the following compromise: a ten-year planning period, with individual one-year time periods. Simultaneously, the FORTRAN matrix generator program was written to allow easy model shrinkage or expansion when information on computer solution time became available. For instance, such model elements as the number of time periods, the number of investment alternatives considered during each time period, and the load duration curve approximation were all taken to be parameters for the matrix generator.

II Explore model FPS in detail before invoking the MILP solution algorithm of a commercial LP system.

Under favorable conditions of model structure, it may be possible to determine an optimal solution through FPS exploration alone, and to subsequently allow the algorithm to merely <u>verify</u> optimality. Failing such a favorable outcome, there are still three major advantages to exploring the FPS and identifying several feasible solutions: (1) useful information will be gained concerning the setting of solution controls for the commercial MILP algorithm, including selection of the best objective function value found as input for an initial bound; (2) the character of the unknown optimum may be transformed from "murky" to "dimly outlined," or even to "relatively clear"; and (3) the analyst will doubtless gain much better understanding of the model's functioning -- that is, how various components interact, or how certain right-hand-side elements drive the model -- which will prove valuable in subsequent sensitivity analysis pertaining to the assumed costs of construction, operation or inventory policy, demand requirements, warehouse capacities, processing capabilities, the discount rate, or the distance of the planning horizon.

Exploration of the FPS can be made easily with the use of the bounds-setting facilities of commercial LP systems, combined with the restart capabilities of these codes. Such features allow the user to perform several of the unsophisticated steps of a branch-and-bound (B-B) algorithm by making explicit decisions on branching at each iteration. While it may seem paradoxical for the user to consider exercising intimate control of a sequence that can easily be directed by a programmed set of instructions, the analyst will have (or be able to acquire) knowledge of the model's structure, which will immediately permit the elimination of particular decision strategies. Such information may be impossible to communicate in detail to the B-B algorithm, and indeed may not even be consciously understood by the analyst prior to FPS exploration.

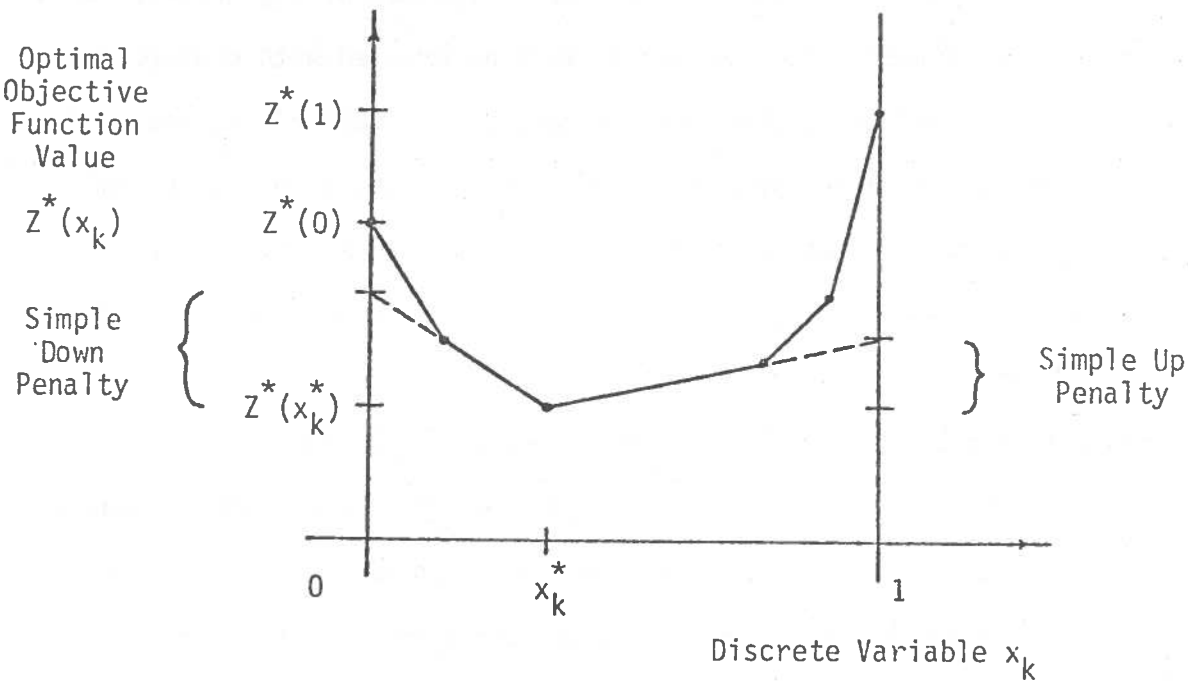
For the application, the need for FPS exploration became apparent shortly after solution of one of the self-sufficiency models was attempted.

Although great care was taken in initially formulating the model, on first try the MILP solution algorithm used up one hour of CPU time with only small discernible progress. From the continuous optimum -- at which all integer requirements for the discrete variables were relaxed -- the algorithm moved rapidly toward a part of the decision tree that intuition, but not the algorithm, would immediately recognize as far from optimal. With this finding, future algorithmic performance was feared to be so poor that direct intervention by the analyst was believed essential to secure even a "good" suboptimal solution.

The method of FPS exploration followed several of the steps of mechanized branching and bounding. First, the basis associated with the continuous optimum was stored on a separate disk file (*not* on the problem workfile, to prevent its loss during subsequent program manipulations), and the optimal values of the discrete variables were examined. Nearly all of these variables were binary (0-1) and many were fractional valued, such as x_k^* in Figure 1. The process of branching on the discrete variables, or setting the integer variables sequentially to their upper or lower limits, was then begun. The branching steps were easily performed using the Burroughs commercial LP system TEMPO. During model assembly in industry-standard MPS format, all discrete variables had been assigned upper bounds in the BOUNDS section. At each branching step it was a simple matter to fix a particular discrete variable at an integral value using an FX bound declaration in the (revised) BOUNDS section. Reoptimization using the immediately preceding basis to restart then followed. Progress through the decision tree was associated with an increasing number of FX declarations, and a diminishing number of discrete variables allowed to vary continuously. Termination at

FIGURE 1

Hypothetical Graph of Optimal Objective Function Value as a Function of Discrete Variable x_k



NOTES TO ACCOMPANY FIGURE 1

1. The MILP objective is to minimize Z.
2. The continuous optimum yields x_k^* for discrete variable x_k and $Z^*(x_k^*)$ for Z.
3. The convex piecewise linear function $Z^*(x_k)$ graphs Min Z as a function of x_k when all other decision variables are allowed to adjust freely.
4. Kinks in the graph of $Z^*(x_k)$ correspond to basis changes.
5. The "simple down penalty" estimates the degradation in $Z^*(x_k)$ when x_k is forcibly reduced from x_k^* to 0; the "simple up penalty" estimates this degradation when x_k is forcibly increased from x_k^* to 1.
6. For detailed discussion of the relationship of this figure to branch-and-bound solution algorithms, see [6].

a feasible solution occurred when all discrete variables took on integral values.

Conventional wisdom of first branching on the most important discrete variables was followed,[2] with identification of key integer variables proving straightforward: such variables were associated with prospective generating stations having the greatest generation capabilities and cost. Quite naturally, the model sought to defer for as long as possible the construction of these stations. Branching on the discrete variables, and the concomitant degradation of the objective function at each step (exemplified by the rise of Z^* in Figure 1 from $Z^*(x_k^*)$ to either $Z^*(0)$ or $Z^*(1)$ depending on the bound taken), were continued until a feasible solution was obtained. Only hunch and economic intuition regarding the intertemporal evolution of the power supply systems were used in making the branching decisions; no effort was directed toward computing up or down penalties at each step. Such effort might have speeded up FPS exploration. At each iteration the optimal basis was updated, and when a feasible solution was found, the basis was saved on disk for future sensitivity analysis.

Derivation of subsequent feasible solutions was usually started at the continuous optimum, but occasionally it was started at some stage intermediate between the continuous optimum and a previous integral feasible solution. One method found particularly effective for rapidly identifying feasible solutions was first to branch on the most important variables, then to enforce integrality requirements moving forward in time: initially for time t=1, next for t=2 and so on to the model horizon T=10. What this approach amounted to was modified myopic capacity expansion on a year-by-year basis to the horizon. When a feasible solution was identified, the values for the discrete variables of the last several time periods (say M through T) were recorded, and return was made to the continuous optimum. These discrete variables were then fixed according to the values recorded, and branching decisions were once again made moving forward in time. Such an approach yielded answers to the question: presuming all investments from periods M through T are known, what investments and sequencing from periods 1 through M-1 appear most attractive? The method allowed a reduction in problem size (that is, the number of discrete variables allowed to vary freely at the start of branching declined) and a better feasible solution than the initial one was frequently obtained.

Standard algorithmic practice of using the objective function value of the best feasible solution as a bound in pruning the decision tree also was followed throughout the exploration phase.

III Use the information gained through FPS exploration to pare down the FPS by eliminating as many variables and constraints as possible.

This measure should go a long way toward reducing solution time, or transforming an unsolvable problem into one that is solvable. The ability to easily alter the MILP model will prove highly useful for this purpose. For instance, it may have been learned that investments A through D are best made "early," and E through K best made "late." Consequently, it may be possible to modify the model to delete choice of investments A through D late, and E through K early, which would make model size more manageable for the actual B-B algorithm. Paring of this precise nature was performed in the application; the number of time periods was left unchanged but the number of alternatives considered in any time period was reduced. This step was instrumental in allowing the self-sufficiency models to be solved to optimum. There is, however, a real danger of eliminating optimal or near-optimal decision strategies if paring is not performed judiciously, or if it is not made with an eye to future sensitivity analyses that could make the solution strategies eliminated more attractive.

IV Use the information gained through FPS exploration during subsequent post-optimality analysis.

Since most MILP models are formulated to examine decision problems in depth, and not merely to determine an optimum for a single set of given conditions, the FPS scrutiny performed earlier should pay handsome dividends for testing model sensitivity to certain assumptions. For instance, changes in costs of facility construction or operation will normally give rise to certain qualitative expectations in terms of desired construction or sequencing. These expectations may immediately be verified or refuted in part by evaluating the (modified) objective functions of the group of feasible solutions found earlier. During this evaluation process, strong hints may also emerge as to the extent of the tilt of the optimum toward or away from certain decision strategies by the changes made. In the application, major importance was attached to the discount rate, since long-lived capital intensive hydro alternatives were considered among the investment prospects. Sensitivity analysis with respect to the discount rate altered only the objective function coefficients and not the FPS, thereby leaving all previous integral-feasible solutions feasible. The information extracted from these feasible solutions proved centrally important in demonstrating that joint supply economies were broadly insensitive to the discount rate. In a similar manner, a glimpse at the consequences of changes in resource availabilities or demand requirements will be possible if at least several of the feasible solutions retain their feasibility. Even if feasibility is lost in all solutions, some learning may still prove possible, since the commercial LP code will typically pinpoint the rows in which infeasibilities occur, and implicitly suggest possible modifications to such solutions for feasibility to be regained. In the application, the latter occurred when sensitivity analysis of the time path of future power demands was undertaken.

Even if little learning about the new optimum is possible from what has gone before, the fact that FPS exploration has been performed previously will mean that subsequent model exploration proceeds much faster, as the routine becomes more familiar to the analyst.

Conclusion

MILP models will doubtless continue to be applied, extended, and solved. While generalized network formulations of such models (see [7]) may help to reduce solution times, ever-larger models will be assembled for solution, most likely in an environment where a commercially-implemented MILP algorithm is the principal solution device. For such models the suggestions set forth here should assist in shrinking solution times, and thus lighten the burden of computing on the research budget.

FOOTNOTES

1. See [12] for an early study of MILP formulation, [3] for a current exposition of standard model formulations, and [6] for a discussion and comparison of algorithmic solution procedures.

2. For discussion of the conventional wisdom on branching, including the use of branching priorities, see [6] and [2] [4] [8].

REFERENCES

[1] Anderson, D. "Models for Determining Least-Cost Investments in Electricity Supply." Bell Journal of Economics and Management Science, Vol. 3, No. 1 (Spring 1972), pp. 267-299.

[2] Burroughs Corporation. TEMPO User's Manual. 1975.

[3] Bradley, S.; A. Hax; and T. Magnanti. Applied Mathematical Programming. Reading, Mass.: Addison-Wesley, 1977.

[4] Control Data Corporation. APEX-III Reference Manual. 1975.

[5] Gaballa, A., and W. Pearce. "Telephone Sales Manpower Planning at Qantas." Interfaces, Vol. 9, No. 3 (May 1979), pp. 1-9.

[6] Geoffrion, A., and R. Marsten. "Integer Programming Algorithms: A Framework and State-of-the-Art Survey." Management Science, Vol. 18, No. 7 (March 1972), pp. 465-491.

[7] Glover, F.; J. Hultz; and D. Klingman. "Improved Computer-Based Planning Techniques. Part II." Interfaces, Vol. 9, No. 4 (August 1979), pp. 12-20.

[8] Honeywell Corporation. Mathematical Programming System User Guide. 1977.

[9] Jones, G., and J. Wilson. "Optimal Scheduling of Jobs on a Transmission Network." Management Science, Vol. 25, No. 1 (September 1979), pp. 98-104.

[10] Kendrick, D., and A. Stoutjesdijk. The Planning of Industrial Investment Programs: A Methodology. Baltimore: Johns-Hopkins, 1978.

[11] Mairs, T., et al. "On a Production Allocation and Distribution Problem." Management Science, Vol. 24, No. 15 (November 1978), pp. 1622-1630.

[12] Markowitz, H., and A. Manne. "On the Solution of Discrete Programming Problems." Econometrica, Vol. 25, No. 1 (January 1957), pp. 84-110.

[13] Noonan, F., and R. Giglio. "Planning Electric Power Generation: A Non-Linear Mixed Integer Model Employing Benders Decomposition." Management Science, Vol. 23, No. 9 (May 1977), pp. 946-956.

[14] Rowse, J. "Toward Optimal Capacity Expansion for an Electric Utility: The Case of Saskatchewan Power." Canadian Journal of Economics, Vol. 11, No. 3 (August 1978), pp. 447-469.

[15] Scherer, C. "Estimating Peak and Off-Peak Marginal Costs for an Electric Power System: An Ex Ante Approach." Bell Journal of Economics, Vol. 7, No. 2 (Autumn 1976), pp. 575-601.

[16] Woolsey, R. "Some Practical Aspects of Solving Integer Programming Problems." Chapter 17 of S. Zionts. Linear and Integer Programming. Englewood Cliffs, N.J.: Prentice-Hall, 1974.

[17] Woolsey, R. "How to Do Integer Programming in the Real World." Chapter 13 of H. Salkin. Integer Programming. Reading, Mass.: Addison-Wesley, 1975.

[18] Zierer, T.; W. Mitchell; and T. White. "Practical Applications of Linear Programming to Shell's Distribution Problems." Interfaces, Vol. 6, No. 4 (August 1976), pp. 13-26.

A Note on the Solution of Spatial Price and Allocation Models

Models of spatial and temporal price and allocation have enjoyed widespread acceptance in agricultural economics and continue to be employed, refined and extended in the literature. In this journal, for instance, contributions have been made by Yaron et al. [16], Ghosh [2], Baumes and McCarl [1] and Jenson and Piedrahita [4]. It is the purpose of this note to draw attention to a versatile and computationally efficient nonlinear programming system called MINOS, as yet little known outside the operations research community, for the solution of nonlinear price and allocation models -- including the widely applied special class of quadratic programming models. This software system has several features which very strongly commend its use to the practitioner and have already led to its adoption for solving nonlinear models by Manne [7] and Rowse [10], among others. A skeletal outline of several of the principal advantages of the MINOS system follows and an example of its use in solving a small nonlinear model is provided.

Since specific details on the nature and construction of MINOS are readily available elsewhere (Murtagh and Saunders [8, 9] and Saunders [12]), only concise observations on technical requirements are necessary here.* MINOS requires the user to specify, in FORTRAN, all nonlinear entries in the objective function and derivatives of these elements, and at present accepts only linear constraints. Specific advantages of the solution system for the user include (i) compatibility with scientific computers of major manufacturers, allowing widespread dissemination of the package and in consequence the portability of computer-programmed nonlinear models; (ii) compatibility in input format requirements with major commercial linear programming systems, allowing preparation and extensive debugging of the model constraint set by these efficient systems prior to actual solution using MINOS; (iii) convenient restart options facilitating intensive post-optimality analysis; (iv) modularity in assembly of the package, allowing the user with computing and operations research expertise to assemble his own algorithmic package for the solution of problems not manageable within the scope of MINOS presently by calling MINOS optimization subroutines as part of a customized solution procedure; and (v) arguably most important of all for the practitioner, detailed information and support documentation on its use. An additional important advantage is the freedom to formulate a spatial model as either a primal (flow) or a dual (pricing) program, since the primal (or the dual) may prove easier to solve or simply more tractable than its counterpart. Furthermore, from a computational standpoint, for MINOS "...the general algorithm comes close to being 'ideal' on quadratic programs, without undue inefficiency or any specialized code." (Murtagh and Saunders [9, p. 59]). Given the number of quadratic spatial models currently in use, this latter advantage is obviously a major one.

To illustrate the capabilities of MINOS, the following variant of the generalized transportation problem (as defined by MacKinnon [6]) was solved. Supply and demand functions for a single homogeneous commodity are summarized as follows:

Supply	Demand
$S_1 = -200 + 120\ P_1^{1.6}$	$D_1 = 500 + 280\ P^{-0.7}$
$S_2 = -100 + 75\ P_2^{0.9}$	$D_2 = 320 + 235\ P^{-1.2}$
$S_3 = -80 + 90\ P_3$	$D_3 = 230 + 190\ P^{-0.4}$
$S_4 = -30 + 25\ P_4^{0.8}$	

where subscripted P's denote supply prices and unsubscripted P's denote demand prices.

Demand sinks and supply sources are separated spatially, and transport costs must be incurred whenever commodity flows take place. Per-unit transport costs t_{ij} of shipping the commodity from source i to sink j are given by the matrix T:

$$\begin{bmatrix} 0.32 & **** & 0.80 \\ 0.72 & 0.52 & 0.24 \\ 0.20 & 0.48 & 0.78 \\ 0.10 & 0.40 & 0.60 \end{bmatrix}$$

The starred entry in this matrix is nonconstant and hence has been excluded. Letting x denote the flow from source 1 to sink 2, per-unit transport costs are represented by the increasing function $0.22\ x^{0.1}$. Such rising unit costs may be thought of as the outcome of congestion on crowded transport routes, for instance. This case of nonlinear transport cost has been included merely to exemplify the solution capability of MINOS; nonlinear costs for all flows could have been included if desired. Incorporation of declining unit transport costs would have presented no difficulty for solution by MINOS, but would have required a careful user search of the model feasible point set to determine an optimum. In technical terms, the maximization problem to be solved would then no longer have been concave and thus any local optimum determined by MINOS could not be guaranteed to constitute a global optimum.

Following Takayama and Judge [14], the mathematical programming problem formulated was to maximize net social payoff attaching to production, transportation and consumption of the commodity; in algebraic terms the problem was cast into a form similar to the primal program set forth in Rowse [11]. Since demand and supply functions were invertible directly, the problem was formulated in the quantity domain as one with nineteen variables and seven constraints and the optimal solution was found to be:

OPTIMAL SUPPLIES, DEMANDS, AND PRICES

	Supply		Demand	
Source/Sink	Price	Quantity	Price	Quantity
1	3.771665	803.7700	4.091660	604.4307
2	4.331660	180.5758	4.168243	362.3764
3	3.891667	270.2501	4.571700	333.4481
4	3.991661	45.6594	--------	--------

OPTIMAL FLOWS AND DELIVERED PRICES

From Source \ To:	Sink 1	Sink 2	Sink 3
1	288.5213 (4.091665)	362.3764 (4.168252)	152.8723 (4.571665)
2	0.0 (--)	0.0 (--)	180.5758 (4.571660)
3	270.2501 (4.091667)	0.0 (--)	0.0 (--)
4	45.6594 (4.091661)	0.0 (--)	0.0 (--)

Both supply and demand quantities and prices appear to be accurate at least to three decimal places.

Two characteristics of this solution deserve mention. First, sink 1 forms the principal demander, as a glance at its demand function suggests, while supply sources 1 and 4 are the major and minor suppliers respectively, again as intuition would suggest from a glance at their supply functions. Second, an integral property of the solution is that if the flow x_{ij} from source i to sink j is positive, then the delivered price at sink j equals the supply price at source i plus the per-unit transport cost. Algebraic statement of this equality is in fact none other than a dual constraint of the generalized transportation problem represented as a nonlinear program, and the solution property may be verified directly by calculating the sum of supply price and unit transport cost for each nonzero transport flow. These sums have been computed and included in the above table in parentheses for comparison purposes. (Unit transportation cost for the flow between source 1 and sink 2 was determined to be 0.396587144.) Delivered and demand prices in the solution found by MINOS are thus seen to be identical to at least three decimal places for all flows.

This particular type of spatial model is one of the simplest non-linear programming problems solvable using MINOS. It is a very small problem and has featured no processing of production inputs into commodity outputs via a linear technology, no transshipment of commodities, and no barriers to trade such as floors or ceilings on allowable flows or per-unit taxes, all of which can be represented readily within the framework acceptable to MINOS. Computer solution required 6.52 seconds of CPU time on the Burroughs B6700 computer at Queen's University (including all central processing time requirements for problem data input and solution output), and only default settings for user-accessible tolerance parameters were utilized. Judicious setting of these tolerances would have allowed greater accuracy, had it been desired.

While no single solution system such as MINOS can ever claim to be most efficient for solving all conceivable types of nonlinear models, the appeal of this package deriving from its strong user orientation is immense. MINOS should facilitate empirical extensions of nonlinear models and its widespread dissemination should allow a degree of uniformity in nonlinear solution methods across research institutions not heretofore possible, leading to virtual portability of computer-programmed nonlinear models. Moreover, the further development of MINOS to accept nonlinearities in the constraint set, which is already well advanced (see Lasdon and Meeraus [5] and Waren and Lasdon [15, pp. 439-40, 445-6]), will allow further empirical extensions of nonlinear models and should encourage theoretical extensions of such models as well. Finally, the solution system will permit much greater flexibility in the imaginative use of mathematical programming methods, exemplified by the forecasting analysis by Shumway and Chang [13] and the joint-production analysis by Griffin [3].

---

*Information on the availability of the MINOS system for research purposes can be obtained by contacting Dr. Michael A. Saunders, Department of Operations Research, Stanford University.

REFERENCES

1. Baumes, H. and B. McCarl, "Linear Programming and Social Welfare: Model Formulation and Objective Function Alternatives," Canadian Journal of Agricultural Economics 26 (1978), 53-60.

2. Ghosh, S., "A Quadratic Programming Interregional Trade Model: An Application and a Perspective," Canadian Journal of Agricultural Economics 20 (1972), 65-77.

3. Griffin, J., "The Econometrics of Joint Production: Another Approach," Review of Economics and Statistics 59 (1977), 389-397.

4. Jenson, E. and A. Piedrahita, "A Note to Explain the Correct Use of the RAND Quadratic Programming Program: An Illustration of Farm Planning Under Risk," Canadian Journal of Agricultural Economics 27 (1979), 61-69.

5. Lasdon, L. and Meeraus, "Solving Nonlinear Economic Planning Models Using GRG Algorithms," Paper presented at the TIMS/ORSA Joint National Meeting, Spring 1979.

6. MacKinnon, J., "An Algorithm for the Generalized Transportation Problem," Regional Science and Urban Economics 5 (1975), 445-64.

7. Manne, A., "ETA: A Model for Energy Technology Assessment," Bell Journal of Economics 7 (1976), 379-406.

8. Murtagh, B. and M. Saunders, "MINOS: A Large-Scale Nonlinear Programming System - User's Guide," Technical Report SOL 77-9, Department of Operations Research, Stanford University, 1977.

9. Murtagh, B. and M. Saunders, "Large-Scale Linearly Constrained Optimization," Mathematical Programming 14 (1978), 41-72.

10. Rowse, J., "Intertemporal Pricing and Investment for Electric Power Supply," Bell Journal of Economics 11 (1980), 143-65.

11. Rowse, J., "Solving the Generalized Transportation Problem," Regional Science and Urban Economics, forthcoming.

12. Saunders, M., "MINOS System Manual," Technical Report SOL 77-31, Department of Operations Research, Stanford University, 1977.

13. Shumway, C. and A. Chang, "Linear Programming versus Positively Estimated Supply Functions: An Empirical and Methodological Critique," American Journal of Agricultural Economics 59 (1977), 344-57.

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

15. Waren, A. and L. Lasdon, "The Status of Nonlinear Programming Software," Operations Research 27 (1979), 431-56.

16. Yaron, D., Y. Plessner and E. Heady, "Competitive Equilibrium - Application of Mathematical Programming," Canadian Journal of Agricultural Economics 13 (1965), 65-79.