Cowles Foundation Paper 763

OR FORUM

# MATHEMATICAL PROGRAMMING AND ECONOMIC THEORY

**HERBERT E. SCARF**

*Yale University, New Haven, Connecticut*

(Received November 1989; accepted January 1990)

This paper, which is based on the remarks offered during a plenary address at the May 1989 CORS/TIMS/ORSA meeting in Vancouver, discusses the analogy between economic institutions and algorithms for solving mathematical programming problems. The simplex method for solving linear programs can be interpreted as a search for market prices that equilibrate the demand for factors of production with their supply. A possible interpretation in terms of the internal organization of a large firm is offered for Lenstra's integer programming algorithm.

I would like to take this opportunity to describe my own very personal thoughts about the relationship between mathematical programming and economic theory. These two topics were intimately linked during the marvelous initial burst of activity in mathematical programming some 40 years ago. Some of the major figures in the early development of our field—Arrow, Leontief, and Koopmans—were themselves economists. Others, such as Kuhn and Tucker, Kantorovich, Gomory, Dantzig, and, certainly, von Neumann were sufficiently close to economics to be alert to the relationship between mathematical programming and the problem of the optimal allocation of resources, which is at the heart of economic theory. Since that time, however, the two subjects have drifted so far apart that many scholars in the field are unaware of their historical connections, the similarity of their basic themes, and their enormous potential for fruitful mutual stimulation.

Microeconomic theory studies the interaction of individual economic agents with private and frequently competitive goals. The subject deals with the institutions of private property, the benefits of decentralized profit maximization, the distribution of income arising from the provision of goods and services, and the role played by prices in equilibrating supply and demand.

The primary topic in mathematical programming seems, at first glance, to be quite different. Mathematical programming is concerned with developing algorithms for the efficient numerical solution of discrete and continuous maximization problems, and, as such, is apparently unrelated to the institutional considerations of economic theory.

What I will do in this paper is to try to bring these two subjects together and remind readers of their common features. In particular, I will stress the close relationship between algorithms and economic institutions, suggest that it may be fruitful to view economic institutions as highly specialized computational procedures, and to view numerical algorithms as the analogs of economic activity engaged in by individuals or firms.

## 1. THE THEORY OF ECONOMIC EQUILIBRIUM

One of the major themes of microeconomic theory is that the producing and consuming units of the economy respond—in a decentralized fashion—to prices that are determined in competitive markets. In conventional economic analysis, we typically divide the basic units in the economy into two classes. One class consists of consumers, who own all of the assets of the economy either directly or indirectly through the ownership of financial claims or shares in manufacturing entities. The second class of economic agents are producers, whose business is to transform productive inputs into those goods and services that are valued by consumers themselves ,or used as intermediary goods by other producers.

There may be many ways for a given producer to take his factors of production—labor of varying skills,

*Subject classification* Programming, mathematical· the relationship between algorithms and economic institutions



0030-364X/90/3803-0377 $01 25

capital, a great variety of raw materials, energy and other inputs—and transform them into outputs. Food can be produced on small plots with primitive implements, or on large farms that make use of the most advanced forms of agricultural machinery. Steel can be produced in plants of varying sizes, either operating independently or integrated with enterprises using steel as inputs. During the Great Leap Forward in China, it was even proposed that steel be manufactured in individual backyards. How are these choices to be made?

Economic theory usually makes the assumption that the individual producers in the economy are faced with competitive prices for all of the factors of production and with competitive prices for the outputs of production. If all of the input and output prices are known by the firm, then any particular production plan will have a profit associated with it: the value of output at these prices minus the cost of those factors used in production. It is then customary to assume that the goal of the manufacturing entity is to select, from the list of all possible plans available to it, the particular production plan that maximizes profit.

Prices also enter into the consumer side. Each consuming unit owns its share of the assets of the economy and is able to evaluate its income or wealth once the prices of these assets are known. Given the income of each consumer and the prices of goods to be purchased, the individual consumer's demand for the outputs of production can be specified as well defined functions of price. Adding the individual demand functions, we obtain the market demand functions, which tell us the quantity demanded of each of the goods and services in the economy as a function of the entire set of prices faced by consumers.

Market demands arise from the consumer side of the economy, market supplies from the producer side. At an arbitrary selection of prices, it need not be true that the demand for each commodity is equal to its supply. If the price of apples is too high, consumers may wish to spend their income on oranges, and if the price of clothing is too low in comparison with wages and the cost of materials, manufacturers may not be able to cover their costs of production. Only certain prices—equilibrium prices—will equilibrate the demand and supply for all commodities. It is these prices—and these prices alone—that permit the economy to function in the decentralized fashion celebrated by economic theory.

This intellectual construction—this paradigm—of decentralized competitive behavior is extremely flexible and provides a framework of analysis for a great variety of economic problems. To take one example, it can be used to discuss the consequences of a change in some significant parameter of the economy, such as the abrupt increases in the cost of imported oil experienced by the United States twice in the last 15 years (in 1973–1974 and 1979–1980), changes with extraordinary consequences for the future development of the U.S. and world economies. To analyze this experience we design a formal mathematical model of the economy in which the price of imported oil appears as an exogenous parameter. On the consumer side of the economy we specify the assets—for example, labor, capital and durable goods—owned by each class of consumers and their preferences for the goods and services potentially available to them. To model the production side, we need an explicit mathematical description of the techniques available to producers, given perhaps by an input/output table for the economy as a whole, or by a series of activity-analysis matrices for each of the firms in the economy. We then solve the pre- and post-shock variants of the model for equilibrium prices, the choice of productive techniques, the distribution of income, and other variables of economic interest. A mathematical presentation of the model of equilibrium is required for this exercise in comparative statics to be carried out on the computer.

Ideally, the general equilibrium model can be used to describe the efficient allocation of resources and the selection of production plans in a socialist economy, such as the Soviet Union, as well as in an economy in which private initiative and individual gain are the motivating forces for economic decisions. The general equilibrium model formed the basis for the fascinating discussion of economic planning in the early decades of this century. One of the major figures in this debate was the Italian economist Barone, who was skeptical about the use of the equilibrium model on the ground that the computational difficulties were insurmountable. Barone described the production side of the economy by an activity-analysis model. He realized that, if we knew precisely which activities were to be used at equilibrium, then relative prices for all the goods and services in the economy could be determined by solving the system of linear equations that said the profit associated with each of these activities was zero. For Barone, and for subsequent participants in this debate about socialist planning, the difficulty in applying the equilibrium model to a socialist economy was that no computational procedure existed for solving the vast number of nonlinear equations and inequalities required to select precisely the correct set of activities to be used at equilibrium.

On the face of it, Barone's objection seems no longer to be valid, given the emergence of the modern computer and the development of efficient computer codes for calculating equilibrium prices. Models with, say, one hundred variables can be solved quite readily on a personal computer. Given the value of this calculation, it would seem desirable to use a dozen supercomputers full time to provide Soviet planners with the prices and production decisions that would allocate resources in an optimal fashion. Like all administrators, Soviet politicians are presumably reluctant to give up political power. A bank of Cray supercomputers would be a trivial investment if the results of numerical computations could avoid the inefficiencies of the Soviet economy without sacrificing centralized political control of economic decisions and without allowing the vast disparities in income that are an inevitable consequence of private economic initiative.

Why is it, then, that we see the emergence of the institution of private markets to solve the economic problems faced by one centralized economy after another? It is astonishing how much political power is being sacrificed today in the Soviet Union, Poland, Hungary, and other countries, in the hope of dramatic improvements in economic conditions. In my opinion, the major attraction of markets over centralized calculation, for Gorbachev and his economic reformers, it not so much the mathematical difficulty of a single equilibrium calculation; it is rather that these computations must be performed over and over again, in real time, in the face of constantly changing economic circumstances. The economy is in continual flux, with new possibilities constantly emerging, and mathematical solutions to the equilibrium equations will at best represent the solutions to yesterday's problems. If we are to be responsive to the novel conditions of daily life—and to engage the energies and skills of millions of self-interested economic actors—it may be necessary to use the market as an algorithm for solving the equilibrium equations rather than solving these equations themselves on the computer.

Suppose that the system is in equilibrium and that someone discovers a new way to make sausages out of sawdust, or a new way to transport electrical energy using superconducting wires. Shall this new activity be used? The planners could recalculate the equilibrium on the supercomputer. Or they could make use of a theorem of economic theory—perhaps the most important result of microeconomic analysis—that provides an immediate necessary and sufficient condition for an affirmative answer to the question: can all consumers be made better off if the new activity is used? The condition is amazingly simple; all consumers can be made better off if and only if the new activity makes a positive profit at the current equilibrium prices. I do not know whether Gorbachev is impressed by mathematical theorems, but the fact that the market test of profitability is the precise test for a Pareto improvement is the intellectual justification for decentralized markets. And this market test can be carried out by self-interested, economically motivated individuals, rather than on the computer, if we are willing to tolerate inequalities in the distribution of income.

## 2. MATHEMATICAL PROGRAMMING

What does this have to do with mathematical programming? On the face of it, mathematical programming is concerned with an entirely different set of issues than those I have just mentioned. Mathematical programming is about the maximization of a function of several variables subject to a set of constraints. The primary example of a constrained maximization problem is a linear program, in which the objective function is a linear function of the variables and the constraints are themselves a series of linear inequalities. The solution of a linear program seems to be an exercise in applied mathematics and apparently has nothing to do with prices, profit maximization, and decentralized economic decisions.

At present, the two major contenders as algorithms for solving linear programming problems are the simplex method, invented by Dantzig some forty years ago, and the new interior-point methods introduced by Karmarkar within the last 5 years. There may be considerable debate as to the computational merits of these two methods for solving any particular linear program. But, from an economic point of view, the simplex method is the clear winner in the sense that the steps of the simplex method are capable of the most striking economic interpretation. At each step of the simplex method a trial solution to the linear program is proposed. To test for the optimality of this solution, we find those prices that yield a profit of zero for the activities in use, and use them to calculate the profitability of the remaining activities. The trial solution is optimal if none of the remaining activities make a positive profit; if one of them is profitable, we simply increase the level of its use from zero, making compensating changes in the previous activity levels until one of them falls to zero. The algorithm continues until a trial solution is found that passes the pricing test for optimality.

The simplex method mimics the search for decentralized prices that equilibrate the supply and demand for factors of production. A visitor from another planet who was taught the simplex method for the solution of maximization problems would immediately be led to the introduction of competitive markets. With no knowledge whatsoever of the long historical development of the institutions of competition, our visitor would be able to answer a vital question: If the economy is in equilibrium—in the sense that the optimal values of the variables have been determined—and a change in economic circumstances presents a new activity for possible use, can the new activity be used so as to increase the value of the objective function? The visitor would know immediately that a necessary and sufficient condition for the use of this activity is that it make a positive profit at the old equilibrium prices. Prices and the institution of competitive markets, not obviously associated with the simple mathematics of maximization, arise in the most natural way in solving optimization problems.

I remember a conversation that I had many years ago with Tjalling Koopmans, when linear programming models were being considered as a tool for socialist planning. At one point Tjalling said, "Suppose that the giant linear program for the Soviet Union is solved on the computer. Should we tell the individual firms the specific production plans that the model instructs them to use, or should we simply give them the prices for their inputs and outputs and let them make their own decisions?" I think that perhaps we should do neither. Instead, we should suggest that the institution of competitive markets be used to decide on the merits of the novel economic possibilities that firms will be facing over and over again in the future.

## 3. INTEGER PROGRAMMING AND THE ECONOMICS OF LARGE-SCALE PRODUCTION

Now I will turn this discussion of markets on its head and talk about what is for me one of the major difficulties in the competitive solution to the problem of resource allocation. Both linear programming and the classical model of equilibrium make an extremely important—and, to my way of thinking, extremely restrictive—assumption about the production side of the economy. Both of these formulations require that production exhibit constant returns to scale: that the mix of inputs needed to produce a particular assortment of outputs be unchanged as the scale of production varies; that it is just as efficient to manufacture steel in our own backyards as it is to use a fully integrated assembly line. This is a terribly restrictive assumption, which excludes the possibility of economies of scale and forces us to ignore one of the central features of economic life in the twentieth century: the large industrial firm whose size is based on the economic advantages of large-scale production.

Economists have been concerned for many years about the need to incorporate the possibility of increasing returns to scale in their analytic formulations. An older school of economists held the opinion that efficiencies of large-scale production were caused by indivisibilities, that is, large, lumpy aggregates of capital—assembly lines, railroad and telephone networks, bridges—whose economic advantages could not be realized at low levels of production. Lerner (1944), for instance, devoted two chapters of his famous book, *The Economics of Control*, to the study of indivisibilities. I quote from Chapter 15 (p. 176) to illustrate his position on this subject:

> We see then that indivisibility leads to an expansion in the output of the firm, and this either makes the output big enough to render the indivisibility insignificant, or it destroys the perfection of competition Significant indivisibility destroys perfect competition.

In the classical case of constant returns to scale, there is essentially no theory of the firm, because the firm can progressively be disaggregated into smaller and smaller units, which then interact with each other by means of market prices. If increasing returns to scale or indivisibilities prevail, however, the firm cannot be disaggregated into competitive units without a substantial loss of efficiency. In the case of constant returns to scale, institutional arrangements such as competitive markets are directly suggested by numerical methods for the solution of linear programming problems. If the analogy were to be maintained, we would expect corresponding insights about the internal organization of large firms from the study of decision methods for solving maximization problems involving indivisibilities. And one of the central concerns in mathematical programming at present is precisely the study of maximization problems in which the production-possibility set involves indivisibilities, increasing returns to scale, or other forms of nonconvexity—that is, integer programming.

Indivisibilities are introduced into a linear programming problem by requiring that some, or all, of the activity levels take on integral values, rather than arbitrary real values. Linear programming problems that require some of the activity levels to be integral are known as integer programs. As you all know, the first algorithm for solving the general integer programming problem was introduced in the late 1950s by Gomory. There were some serious problems with this and other early computational methods. The methods were not robust: A slight change in one of the parameters of the problem could transform an easy problem into an intractable one. In contrast to the simplex method, which performs remarkably well on most linear programming problems, integer programming algorithms were capricious and unreliable. And, perhaps even more significant for economic theory, none of these algorithms seemed capable of being interpreted—by even the most sympathetic student—in meaningful economic terms.

Early researchers in this area were acutely aware of the vital relationship of integer programming to economic theory: Koopmans and Beckmann (1957) wrote an early paper on indivisibilities, and Gomory and Baumol (1960) published a joint paper on a possible economic interpretation of Gomory's cutting plane algorithm. By the late 1960s, however, the origins of discrete programming problems in economic theory were in the process of being forgotten by practitioners in the field. The search for tractability led to groupings and classifications of integer programs that were based solely on their mathematical properties; less and less reference was made to economic considerations. The terms *indivisibility*, *factor endowment*, *capital*, and *degrees of substitution* were slowly replaced by a new set of concepts: *graphs*, *network flows*, *matching problems*, and *matroids*. Economic theorists and scholars in discrete mathematics became, in time, unable to converse with one another, despite the essential underlying connections between these two disciplines.

## 4. COMPLEXITY THEORY

In the 1970s an important intellectual event took place: the development of the field of *computational complexity*. A new way of looking at the intrinsic complexity of a discrete programming problem was introduced; it involves classifying problems as easy if the time required for their solution is a *small* function (a polynomial) of the time required to describe the problem, or as hard if this is not the case.

An example of an easy problem is that of maximizing the flow of material through a railroad network with capacity constraints on each link of the network. This problem can be solved quite readily for networks of large size. On the contrary, the traveling salesman problem—which calls for the construction of a tour through a set of cities so as to minimize the total traveling time—is hard, and it becomes prohibitively expensive to obtain the precise optimal solution as the number of cities increases. Easy problems are routine and presumably can be carried out by human beings without the extraordinary intellectual and conceptual investment required by a hard problem. This is a point that will come up again.

In the last decade, my own research has been directed toward studying the general integer program from the point of view of complexity theory. To illustrate this point of view, let me return to the earlier discussion of the role played by prices in solving linear programs. For such problems, prices have their customary economic interpretation as *marginal value products*—the marginal change in the optimal value of output if a particular factor of production is increased by a small amount. But, as we have seen, prices are also used to determine whether a specific feasible solution, one that satisfies the constraints of the problem, is actually the optimal solution. Given a feasible solution to a linear program, we find the prices that yield a zero profit—net of all costs, including the rental of capital—for the activities being used. Then a necessary and sufficient condition that the proposed feasible solution be optimal is that all the remaining activities make a profit less than or equal to zero when their profitability is evaluated at these same prices.

This test for optimality is not available for integer programs; there simply need not be a set of prices that yields a zero profit for the activities in use at the optimal solution. Let us look at the following example of an integer program with a single constraint and two nonnegative integer variables.

$$\begin{aligned} &\text{Maximize} && x + 3y \\ &\text{subject to} && 2x + 3y \leq 5 \\ &x, y \geq 0 && \text{and integral.} \end{aligned}$$

The solution to the corresponding linear program, with no requirement of integrality for the activity levels, is $(x, y) = (0, 5/3)$, and the price of the constraint—the optimal dual variable—is equal to one. At this price, the second activity makes a profit of zero and the first activity, which is not used, has a negative profit. But the optimal solution for the integer program is $(x, y) = (1, 1)$; both activities are used and there is no price at all that yields a zero profit for the two activities simultaneously (see Figure 1).

This is, of course, not an accident of this particular example. Except for very special integer programs, there will not be a vector of prices that provides a profit of zero for the activities used in the optimal solution and a negative profit for the remaining activities. Moreover, if we have solved a specific integer program by one device or another, and a new activity is discovered, there is no conclusive pricing test to tell

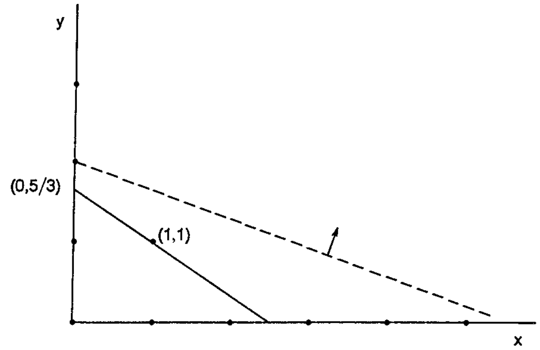


**Figure 1.** An example of an integer program.

us whether the new activity can be used to improve the objective value.

A similar difficulty arises in the equilibrium formulation that I discussed earlier. Suppose that the economic system is in equilibrium at certain prices and that a new activity is discovered that can only be used at an integral level. Is its profitability at the equilibrium prices a necessary and sufficient condition for a Pareto improvement—for the possibility that everyone can be made better off using this new activity? The answer, unfortunately, is no! And if several activities are discovered simultaneously, all of which must be used at integral levels, improvements may require the use of a complex mixture of both profitable and unprofitable activities. The market test is simply not available to us in the presence of indivisibilities.

The market test fails because the firm, whose technology is based on an activity-analysis model with integral activity levels, cannot be decentralized without losing the advantages of increasing returns to scale. The large firm has an integral organization of managerial and productive tasks, which cannot be replaced by competitive markets that are internal to the firm. We cannot decentralize the large firm by assuming that the subdivisions of the firm trade outputs and factors with each other using competitive prices. But, if we return to our metaphor about the relationship between computational procedures and economic institutions and view the large firm as an algorithm for the solution of integer programming problems, can some hints about the internal structure of the firm be obtained by examining numerical algorithms?

## 5. LENSTRA'S ALGORITHM

Let us look at the algorithm proposed by Lenstra several years ago (Lenstra 1983). In the language of complexity theory, integer programming is what is known as an NP-complete problem: If there is a polynomial algorithm for integer programming, then virtually every problem that we can think of is easy to solve—a quite unlikely possibility. Lenstra's algorithm provides a polynomial algorithm for integer programming when the number of integral variables is fixed in advance. It also provides a sharp theoretical description of the complexity of integer programming and it is also possible that the algorithm may have practical as well as theoretical significance. Several of us are now programming a variant of Lenstra's algorithm to see whether it is actually useful in solving the general mixed integer program with, say, thirty or forty integral variables. It is too early to give a definitive verdict, but I am very optimistic.

Lenstra's algorithm makes heavy use of the branch of mathematics known as the Geometry of Numbers. This subject, invented by the distinguished mathematician Hermann Minkowski almost 100 years ago, is undergoing a remarkable revival, owing primarily to its potential application to the study of discrete programming problems.

To appreciate the novelties of the Geometry of Numbers, it may be useful to contrast it with the classical arguments of linear programming. The primary mathematical tool used in linear programming is convex analysis. The constraint set defined by a series of linear inequalities is a convex body, and the existence of prices that support an optimal solution is a direct application of the separating hyperplane theorem. When indivisibilities are present, the corresponding activity levels are restricted to integer values, and the vector of possible activity levels lies in the lattice of integers in $n$-dimensional space. The major mathematical problem in the theory of discrete programming is to find out, in an efficient way, whether a given convex body contains a lattice point.

It is an elementary mathematical observation that a convex body may have an arbitrarily large volume and yet be free of lattice points. But if the body is *symmetric* about the origin (see Figure 2), it will contain a lattice point other than zero if its volume is sufficiently large. Minkowski's fundamental result is that there will be a lattice point, different from zero, in a symmetric convex body lying in $n$-dimensional space if the volume of the body is greater than $2^n$. Minkowski's theorem is applied, in an indirect way, in Lenstra's algorithm.

Lenstra begins by casting the integer program in the form mentioned above: Does a given convex body contain a lattice point? Let us consider as our convex body the triangle in the plane with integral vertices $(1, 0)$, $(0, 1)$ and $(15, 17)$ and suppose that our question

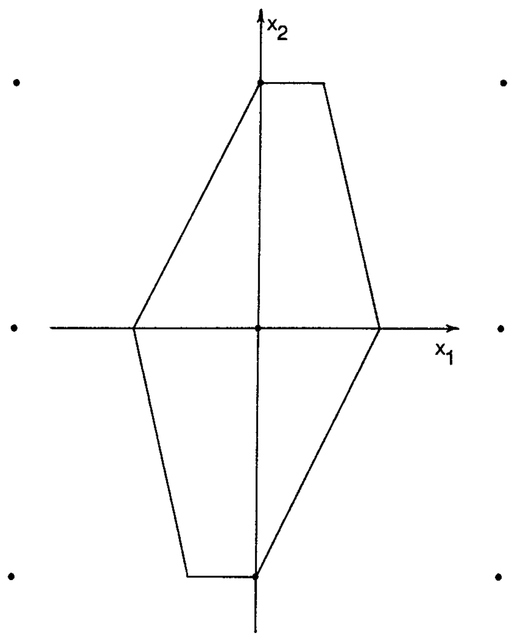


**Figure 2.** A convex body symmetric about the origin.

is whether the triangle contains an integral vector other than one of its vertices (see Figure 3). (This particular example is, of course, trivial to analyze; we may actually write an explicit formula for the number of lattice points in a planar triangle with integer vertices. Think of this, rather, as an illustration of a general convex polyhedron in $n$-space.) The most naive approach is to enclose the triangle in the rectangular box $0 \leq x_1 \leq 15$, $0 \leq x_2 \leq 17$, set the first coordinate equal to each of its 16 possible values, and for each of these first coordinates, check to see whether there is an integral value of the second coordinate that satisfies the linear inequalities defining the body. This is the basic idea of a branch-and-bound algorithm.

But we can do better. If we make the following unimodular transformation of coordinates, which carries lattice points into lattice points

$$y_1 = -x_1 + x_2$$

$$y_2 = 5x_1 - 4x_2$$

the body has the form of Figure 4, with vertices $(-1, 5)$, $(1, -4)$ and $(2, 7)$, and the rectangular box containing the body is considerably smaller. We have only 4 possible values of the first coordinate, rather than our previous 16 values.

What Lenstra does for the general problem is to construct a unimodular transformation such that either

1. the body is sufficiently large so that it clearly contains a lattice point, or
2. the rectangular box containing the body in the new coordinate system is small in at least one coordinate, say, the first one.

In the first case, the algorithm terminates with a lattice point. In the second, the lattice points lying in the body have a small number of possible values for their first coordinate, and the problem is then reduced to the study of a small number of $(n - 1)$-dimensional problems.

The algorithm takes the form of a decision tree (see Figure 5): Find the good unimodular transformation and consider, in turn, each one of the small number of integer programs involving $n - 1$ variables. Repeating the process, each of these problems leads to a small number of integer programs with $n - 2$ variables. Ultimately, we are led to integer programs with a single variable, which are, of course, trivial to solve. It is necessary to consider all branches in the decision tree, but there are ample opportunities for parallel processing, since the computations to be carried out on distinct branches can be done simultaneously.

Other algorithms for solving integer programs—such as branch-and-bound methods—also make use of decision trees. But the Lenstra algorithm is the only

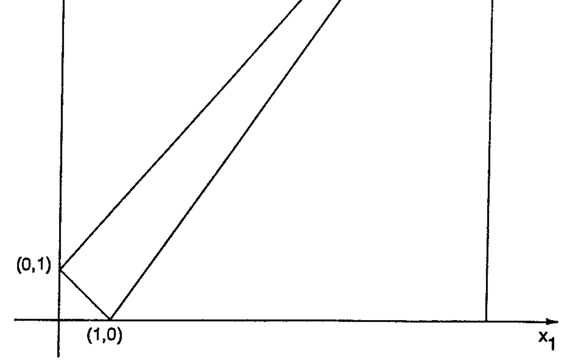


**Figure 3.** The triangular convex body enclosed in a rectangular box.

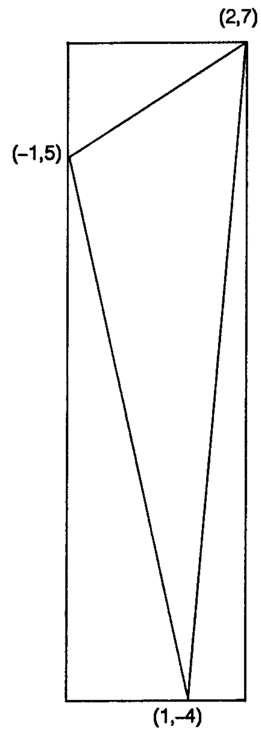


**Figure 4.** The triangular convex body after the unimodular transformation of coordinates, together with its rectangular box of reduced size.

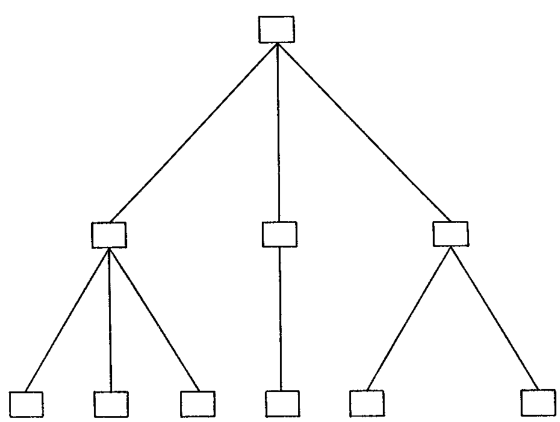

**Figure 5.** The decision tree for Lenstra's algorithm.

one that I am aware of—for the general integer program, rather than the special case of 0, 1 problems—in which the work at each node and the number of branches emanating from each node are both polynomial in the data of the problem, at least if the number of variables is fixed.

The work at each node of the decision tree involves finding a unimodular transformation so that the body is relatively thin in a particular direction. This can be done by means of the Lovász basis-reduction algorithm (see Lovász 1986), which executes in polynomial time for a fixed number of variables. In the second case, in which a lattice point cannot be determined directly, the number of branches emanating from the node also can be shown to be a polynomial function of the data of the problem. If we use the metaphor of complexity theory, which suggests that easy problems are those that can be solved routinely by human beings, negotiating through the decision tree is in itself routine, even for a general integer program.

## 6. THE ORGANIZATION OF THE FIRM

The computational procedure seems to be capable of an interpretation in terms of the managerial tasks faced by the firm. If the large-scale firm is viewed as an algorithm for solving maximization problems based on an activity-analysis model with integral activity levels, the decision tree may be taken as a representation of the internal organization of the firm. We can imagine a human being sitting at each node of Figure 5, who performs the routine calculations required at the node, and transmits the results of these calculations to the small number of subordinates at each of the successor nodes. Accepting the metaphor means accepting the equivalence between the arithmetical operations required by an algorithm and the more usual paperwork performed at a desk or handling of materials on the shop floor. But we have accepted a metaphor of this sort before in discussing the equivalence between the simplex method and the market's search for equilibrium prices.

Of course, Lenstra's algorithm—as just described—solves one particular integer program in much the same way that the equilibrium calculation in the socialist planning office solved for a vector of equilibrium prices under a particular set of circumstances. If the decision tree is meant to be a representation of the internal organization of a large firm whose technology involves substantial indivisibilities, this organization must have some stability in the face of changing economic circumstances. Suppose, for example, that we are interested in solving not one specific integer program but rather a family of similar problems, say, the family of integer programs with a fixed technology matrix, but with many different right-hand side vectors. It can be shown, using some recent results by

Kannan, Lovász and Scarf (1988), that a decision tree with polynomial work and a polynomial number of branches at each node may be constructed to depend only on the activity-analysis model and to be fully independent of the right-hand side of the integer program. With such a decision tree, a shock to the environment need not require a total redesign of the internal organization of the firm. It may also be shown that the tree will change only slightly if the technology matrix is altered slightly, whether by the revision of one of its coefficients or by the discovery of a new process. We actually see these small changes taking place in the performance of the algorithm itself—in the structure of the tree—as we change the numerical values of the problem.

Such a flexible tree cannot be constructed in polynomial time for variable $n$. But, once it is constructed, the computations at each node and the number of successor nodes are actually polynomial in $n$ as well as the other data of the problem, as long as the number of inequalities is less than the number of variables by a fixed amount. Constructing the decision tree can be viewed as an expensive investment activity, which provides a firm with flexibility in the face of changes in its economic environment. There is a considerable tradeoff between a flexible design capable of withstanding substantial changes in the parameters of the problem and a less expensive design tailored to a particular problem. It is much less costly to construct a decision tree for a particular problem rather than one that deals simultaneously with a large set of alternatives. But if a specific tree were constructed and the economic environment changed in a significant fashion—if, say, there were a substantial change in the price of imported oil—then a new decision tree, and perhaps a reorganization of the firm's administrative structure, might have to be recalculated at considerable cost. The achievement of flexibility—the construction of an organization or computer code capable of solving a large number of relatively similar problems in real time—may merit its additional cost in the face of the uncertain and constantly changing circumstances that are an ever present aspect of modern economic life.

## ACKNOWLEDGMENT

The preparation of this paper was supported by the Program in Discrete Mathematics at the Cowles Foundation at Yale University and by National Science Foundation grant 8807167.

## REFERENCES

Gomory, R. E., and W. J. Baumol. 1960. Integer Programming and Pricing. *Econometrica* **28,** 521–550

Kannan, R. L., L. Lovász and H. E. Scarf. 1988. The Shapes of Polyhedra. Cowles Foundation Discussion Paper No. 833, Yale University, New Haven, Conn. *Math. Opns Res* (to appear).

Koopmans, T. C., and M. Beckmann. 1957 Assignment Problems and the Location of Economic Activities. *Econometrica* **25,** 53–76.

Lenstra, H W., Jr. 1983. Integer Programming With a Fixed Number of Variables. *Math Opns Res* **8,** 538–548.

Lerner, A. 1944. *The Economics of Control.* Macmillan, New York.

Lovász, L. 1986. An Algorithmic Theory of Numbers, Graphs, and Convexity. CBMS-NSF Regional Conference Series in Applied Mathematics 50, Society for Industrial and Applied Mathematics, Philadelphia.