



Operations Research

Publication details, including instructions for authors and subscription information:
<http://pubsonline.informs.org>

Linear Programming

George B. Dantzig,

To cite this article:

George B. Dantzig, (2002) Linear Programming. Operations Research 50(1):42-47. <https://doi.org/10.1287/opre.50.1.42.17798>

Full terms and conditions of use: <https://pubsonline.informs.org/Publications/Librarians-Portal/PubsOnLine-Terms-and-Conditions>

This article may be used only for the purposes of research, teaching, and/or private study. Commercial use or systematic downloading (by robots or other automatic processes) is prohibited without explicit Publisher approval, unless otherwise noted. For more information, contact permissions@informs.org.

The Publisher does not warrant or guarantee the article's accuracy, completeness, merchantability, fitness for a particular purpose, or non-infringement. Descriptions of, or references to, products or publications, or inclusion of an advertisement in this article, neither constitutes nor implies a guarantee, endorsement, or support of claims made of that product, publication, or service.

© 2002 INFORMS

Please scroll down for article—it is on subsequent pages



With 12,500 members from nearly 90 countries, INFORMS is the largest international association of operations research (O.R.) and analytics professionals and students. INFORMS provides unique networking and learning opportunities for individual professionals, and organizations of all types and sizes, to better understand and use O.R. and analytics tools and methods to transform strategic visions and achieve better outcomes. For more information on INFORMS, its publications, membership, or meetings visit <http://www.informs.org>

LINEAR PROGRAMMING

GEORGE B. DANTZIG

Department of Management Science and Engineering, Stanford University, Stanford, California 94305-4023

The Story About How It Began: Some legends, a little about its historical significance, and comments about where its many mathematical programming extensions may be headed.

Linear programming can be viewed as part of a great revolutionary development which has given mankind the ability to state general goals and to lay out a path of detailed decisions to take in order to “best” achieve its goals when faced with practical situations of great complexity. Our tools for doing this are ways to formulate real-world problems in detailed mathematical terms (models), techniques for solving the models (algorithms), and engines for executing the steps of algorithms (computers and software).

This ability began in 1947, shortly after World War II, and has been keeping pace ever since with the extraordinary growth of computing power. So rapid has been the advance in decision science that few remember the contributions of the great pioneers that started it all. Some of their names are von Neumann, Kantorovich, Leontief, and Koopmans. The first two were famous mathematicians. The last three received the Nobel Prize in economics.

In the years from the time when it was first proposed in 1947 by the author (in connection with the planning activities of the military), linear programming and its many extensions have come into wide use. In academic circles decision scientists (operations researchers and management scientists), as well as numerical analysts, mathematicians, and economists have written hundreds of books and an uncountable number of articles on the subject.

Curiously, in spite of its wide applicability today to everyday problems, it was unknown prior to 1947. This is not quite correct; there were some isolated exceptions. Fourier (of Fourier series fame) in 1823 and the well-known Belgian mathematician de la Vallée Poussin in 1911 each wrote a paper about it, but that was about it. Their work had as much influence on Post-1947 developments as would finding in an Egyptian tomb an electronic computer built in 3000 BC. Leonid Kantorovich’s remarkable 1939 monograph on the subject was also neglected for ideological reasons in the USSR. It was resurrected two decades later after the major developments had already taken place in the West. An excellent paper by Hitchcock in 1941 on

the transportation problem was also overlooked until after others in the late 1940’s and early 1950’s had independently rediscovered its properties.

What seems to characterize the pre-1947 era was lack of any interest in trying to optimize. T. Motzkin in his scholarly thesis written in 1936 cites only 42 papers on linear inequality systems, none of which mentioned an objective function.

The major influences of the pre-1947 era were Leontief’s work on the Input-Output Model of the Economy (1933), an important paper by von Neumann on Game Theory (1928), and another by him on steady economic growth (1937).

My own contributions grew out of my World War II experience in the Pentagon. During the war period (1941–45), I had become an expert on programming-planning methods using desk calculators. In 1946 I was Mathematical Advisor to the US Air Force Comptroller in the Pentagon. I had just received my PhD (for research I had done mostly before the war) and was looking for an academic position that would pay better than a low offer I had received from Berkeley. In order to entice me to not take another job, my Pentagon colleagues, D. Hitchcock and M. Wood, challenged me to see what I could do to mechanize the planning process. I was asked to find a way to more rapidly compute a time-staged deployment, training and logistical supply program. In those days “mechanizing” planning meant using analog devices or punch-card equipment. There were no electronic computers.

Consistent with my training as a mathematician, I set out to formulate a model. I was fascinated by the work of Wassily Leontief who proposed in 1932 a large but simple matrix structure which he called the *Interindustry Input-Output Model* of the American Economy. It was simple in concept and could be implemented in sufficient detail to be useful for practical planning. I greatly admired Leontief for having taken the three steps necessary to achieve a successful application:

1. Formulating the inter-industry model.
2. Collecting the input data during the Great Depression.
3. Convincing policy makers to use the output.

Leontief received the Nobel Prize in 1976 for developing the input-output model.

Subject classification: Professional: comments on.

Area of review: ANNIVERSARY ISSUE (SPECIAL).

For the purpose I had in mind, however, I saw that Leontief's model had to be generalized. His was a steady-state model and what the Air Force wanted was a highly *dynamic model*, one that could change over time. In Leontief's model there was a one-to-one correspondence between the production processes and the items being produced by these processes. What was needed was a model with *alternative activities*. Finally it had to be computable. Once the model was formulated, there had to be a practical way to compute what quantities of these activities to engage in that was consistent with their respective input-output characteristics and with given resources. This would be no mean task since the military application had to be *large scale*, with thousands of items and activities.

The *activity analysis* model I formulated would be described today as a time-staged, dynamic linear program with a staircase matrix structure. *Initially there was no objective function*; broad goals were never stated explicitly in those days because practical planners simply had no way to implement such a concept. Noncomputability was the chief reason, I believe, for the total lack of interest in optimization prior to 1947.

A simple example may serve to illustrate the fundamental difficulty of finding a solution to a planning problem once it is formulated. Consider the problem of assigning 70 men to 70 jobs. Suppose a value or benefit v_{ij} would result if the i th man is assigned to the j th job. An *activity* consists in assigning the i th man to the j th job. The restriction are: (i) each man must be assigned a job (there are 70 such), and (ii) each job must be filled (also 70). The level of an activity is either 1, meaning it will be used, or 0, meaning it will not. Thus there are 2×70 or 140 restrictions, 70×70 or 4900 activities with 4900 corresponding 0-1 decision variables x_{ij} . Unfortunately there are $70!$ different possible solutions or ways to make the assignments x_{ij} . The problem is to compare the $70!$ solutions with one another and to select the one which results in the largest sum of benefits from the assignments.

Now $70!$ is a big number, greater than 10^{100} . Suppose we had a computer capable of doing a million calculations per second available at the time of the big bang fifteen billion years ago. Would it have been able to look at all the $70!$ combinations by the year 1990? The answer is no! Suppose instead it could perform at nano-second speed and make 1 billion complete assignments per second? The answer is still no. Even if the Earth were filled with such computers all working in parallel, the answer would still be negative. If, however, there were 10^{40} Earths circling the sun each filled solid with nanosecond speed computers all programmed in parallel from the time of the big bang until the Sun grows cold, then perhaps the answer might be, yes.

This easy-to-state example illustrates why up to 1947, and for the most part even to this day, a great gulf exists between man's aspirations and his actions. Man may wish to state his wants in complex situations *in terms of a general objective to be optimized* but there are so many different ways to go about it, each with its advantages and

disadvantages, that it would be impossible to compare all the cases and choose which among them would be the best. Invariably, man in the past has turned to a leader whose 'experience' and 'mature judgment' would guide the way. Those in charge like to do this by issuing a series of ground rules (edicts) to be executed by those developing the plan.

This was the situation prior to 1946. In place of an explicit goal or objective function, there were a large number of *ad hoc* ground rules issued by those in authority to guide the selection. Without such rules, there would have been, in most cases, an astronomical number of feasible solutions to choose from. Incidentally, "Expert System" software which is very much in vogue today makes use of this *ad hoc* ground-rule approach.

All that I have related up to now about the early development took place before the advent of the computer, more precisely, before in late 1946 we were aware that the computer was going to exist. But once we were aware, the computer became vital to our mechanization of the planning process. So vital was the computer, that our group arranged (in the late 1940's) that the Pentagon fund the development of computers.

To digress for a moment, I would like to say a few words about the electronic computer itself. To me, and I suppose to all of us, one of the most startling developments of all time has been the penetration of the computer into almost every phase of human activity. Before a computer can be intelligently used, a model must be formulated and good algorithms developed. To build a model, however, requires the axiomatization of a subject matter field. In time this axiomatization gives rise to a whole new mathematical discipline which is then studied for its own sake. Thus, with each new penetration of the computer, a new science is born.

Von Neumann notes this tendency to axiomatize in his paper on *The General and Logical Theory of Automata*. In it he states that automata have been playing a continuously increasing role in science. He goes on to say:

Automata have begun to invade certain parts of mathematics too, particularly but not exclusively mathematical physics or applied mathematics. The natural systems (e.g., central nervous system) are of enormous complexity and it is clearly necessary first to subdivide what they represent into several parts which to a certain extent are independent, elementary units. The problem then consists of understanding how these elements are organized as a whole. It is the latter problem which is likely to attract those who have the background and tastes of the mathematician or a logician. With this attitude, he will be inclined to forget the origins and then, after the process of axiomatization is complete, concentrate on the mathematical aspects.

By mid-1947, I had formulated a model which satisfactorily represented the technological relations usually encountered in practice. I decided that the *ad hoc* ground rules had to be discarded and replaced by an explicit objective function. I formulated the planning problem in mathematical terms using a set of axioms. The axioms concerned

the relations between two kinds of sets: the first was the set of items being produced or consumed and the second, the set of activities or production processes in which these items would be inputted or outputted in fixed proportions providing these proportions are non-negative multiples of each other. The resulting system to be solved was the minimization of a linear form subject to linear equations and inequalities. The use (at the time it was proposed) of a linear form as the objective function to be extremized was a novel feature of the model.

Now came the nontrivial question: Can one solve such systems? At first I assumed the economists had worked on this problem since it was the problem of allocation of scarce resources. I visited T. C. Koopmans in June 1947 at the Cowles Foundation (which at that time was at the University of Chicago) to learn what I could from the mathematical economists. Koopmans became quite excited. During World War II, he had worked for the Allied Shipping Board on a transportation model and so had the theoretical as well as the practical planning background necessary to appreciate what I was presenting. He saw immediately the implications for general economic planning. From that time on, Koopmans took the lead in bringing the potentialities of linear programming models to the attention of other young economists who were just starting their careers. Some of their names were Kenneth Arrow, Paul Samuelson, Herbert Simon, Robert Dorfman, Leonid Hurwicz, and Herbert Scarf, to name but a few. Some thirty to forty years later four of them received the Nobel Prize for their research.

Seeing that economists did not have a method of solution, I next decided to try my own luck at finding an algorithm. I owe a great debt to Jerzy Neyman, the leading mathematical statistician of his day, who guided my graduate work at Berkeley. My thesis was on two famous unsolved problems in mathematical statistics which I mistakenly thought were a homework assignment and solved. One of the results, published jointly with Abraham Wald, was on the Neyman-Pearson Lemma. In today's terminology, this part of my thesis was on the existence of Lagrange multipliers (or dual variables) for a semi-infinite linear program whose variables were bounded between zero and one and satisfied linear constraints expressed in the form of Lebesgue integrals. There was also a linear objective to be extremized.

Luckily the particular geometry used in my thesis was the one associated with the columns of the matrix instead of its rows. This column geometry gave me the insight which led me to believe that the *simplex method* would be a very efficient solution technique. I earlier had rejected the method when I viewed it in the row geometry because running around the outside edges seemed so unpromising.

I proposed the simplex method in the summer 1947. But it took nearly a year before my colleagues and I in the Pentagon realized just how powerful the method really was. In the meantime, I decided to consult with the great, Johnny von Neumann to see what he could suggest in the way of solution techniques. He was considered by many as the

leading mathematician in the world. On October 3, 1947, I met him for the first time at the Institute Advanced Study at Princeton.

John von Neumann made a strong impression on everyone. People came to him for help with their problems because of his great insight. In the initial stages of the development of a new field like linear programming, atomic physics, computers, or whatever, his advice proved invaluable. After these fields were developed in greater depth, however, it became more difficult for him to make the same spectacular contributions. I guess everyone has a finite capacity, and Johnny was no exception.

I remember trying to describe to von Neumann (as I would to an ordinary mortal) the Air Force problem. I began with the formulation of the linear programming model in terms of activities and items, etc. He did something which I believe was uncharacteristic of him. "Get to the point," he snapped at me impatiently. Having at times a somewhat low kindling point, I said to myself, "OK, if he wants a *quickie*, then that's what he'll get." In under one minute I slapped on the blackboard a geometric and algebraic version of the problem. Von Neumann stood up and said, "Oh that!" Then, for the next hour and a half, he proceeded to give me a lecture on the mathematical theory of linear programs.

At one point, seeing me sitting there with my eyes popping and my mouth open (after all I had searched the literature and found nothing), von Neumann said:

I don't want you to think I am pulling all this out of my sleeve on the spur of the moment like a magician. I have recently completed a book with Oscar Morgenstern on the theory of games. What I am doing is conjecturing that the two problems are equivalent. The theory that I am outlining is an analogue to the one we have developed for games.

Thus I learned about *Farkas' Lemma*, and about *Duality* for the first time. Von Neumann promised to give my computational problem some thought and to contact me in a few weeks which he did. He proposed an iterative nonlinear scheme. Later Alan Hoffman and his group at the Bureau of Standards (around 1952) tried it out on a number of test problems. They also compared it to the simplex method and with some proposals of T. Motzkin. The simplex method came out a clear winner.

As a result of another visit in June 1948, I met Albert Tucker, who later became head of the Mathematics department at Princeton. Soon Tucker and his students Harold Kuhn and David Gale and others like Lloyd Shapley began their historic work on game theory, nonlinear programming and duality theory. The Princeton group became the focal point among mathematicians doing research in these fields.

The early days were full of intense excitement. Scientists, free at last from war time pressures, entered the post-war period hungry for new areas of research. The computer came on the scene at just the right time. Economists and mathematicians were intrigued with the possibility that the fundamental problem of optimal allocation of scarce

resources could be numerically solved. Not too long after my first meeting with Tucker there was a meeting of the Econometric Society in Wisconsin attended by well-known statisticians and mathematicians like Hotelling and von Neumann, and economists like Koopmans. I was a young unknown and I remember how frightened I was with the idea of presenting for the first time to such a distinguished audience the concept of linear programming.

After my talk, the chairman called for discussion. For a moment there was the usual dead silence; then a hand was raised. It was Hotelling's. I must hasten to explain that Hotelling was fat. He used to love to swim in the ocean and when he did, it is said that the level of the ocean rose perceptibly. This huge whale of a man stood up in the back of the room, his expressive fat face took on one of those all-knowing smiles we all know so well. He said: "*But we all know the world is nonlinear.*" Having uttered this devastating criticism of my model, he majestically sat down. And there I was, a virtual unknown, frantically trying to compose a proper reply.

Suddenly another hand in the audience was raised. It was von Neumann. "Mr. Chairman, Mr. Chairman," he said, "if the speaker doesn't mind, I would like to reply for him." Naturally I agreed. Von Neumann said: "The speaker titled his talk 'linear programming' and carefully stated his axioms. If you have an application that satisfies the axioms, well use it. If it does not, then don't," and he sat down. In the final analysis, of course, Hotelling was right. The world is highly nonlinear. Fortunately systems of linear inequalities (as opposed to equalities) permit us to approximate most of the kinds of nonlinear relations encountered in practical planning.

In 1949, exactly two years from the time the Linear programming was first conceived, the first conference (sometimes referred to as the Zero Symposium) on mathematical programming was held at the University of Chicago. Tjalling Koopmans, the organizer, later titled the proceedings of the conference, *Activity Analysis of Production and Allocation*. Economists like Koopmans, Kenneth Arrow, Paul Samuelson, Leonid Hurwitz, Robert Dorfman, Nicholas Georgescu-Roegen, and Herbert Simon; mathematicians like Albert Tucker, Harold Kuhn, and David Gale; and Air Force types like Marshall Wood, Murray Geisler, and myself all made contributions.

The advent or rather *the promise* that the electronic computer would exist soon, the exposure of theoretical mathematicians and economists to real problems during the war, the interest in mechanizing the planning process, and last but not least the availability of money for such applied research all converged during the period 1947–1949. The time was ripe. The research accomplished in exactly two years is, in my opinion, one of the remarkable events of history. The proceedings of the conference remains to this very day an important basic reference, *a classic*!

The simplex method turned out to be a powerful theoretical tool for proving theorems as well as a powerful computational tool. To prove theorems it is essential that the

algorithm include a way of avoiding degeneracy. Therefore, much of the early research around 1950 by Alex Orden, Philip Wolfe and myself at the Pentagon and by J. H. Edmonson as a class exercise in 1951 and by A. Charnes in 1952 was concerned with what to do if a degenerate solution is encountered.

In the early 1950's many areas which we collectively call *Mathematical Programming* began to emerge. These subfields grew rapidly with linear programming playing a fundamental role in their development. A few words will now be said about each of these.

Nonlinear Programming began around 1951 with the famous Karush-Kuhn-Tucker Conditions which are related to the Fritz-John Conditions (1948). In 1954, Ragnar Frisch (who later received the first Nobel prize in economics) proposed a nonlinear interior point method for solving linear programs. Earlier proposals such as those of von Neumann and Motzkin can also be viewed as interior methods. Later in the 1960's, G. Zoutendijk, T. Rockafellar, P. Wolfe, R. Cottle, Fiacco and McCormick, and others developed the theory of nonlinear programming and extended the notions of duality.

Commercial Applications were begun in 1952 by Charnes, Cooper and Mellon with their (now classical) optimal blending of petroleum products to make gasoline. Applications quickly spread to other commercial areas and soon eclipsed the military applications which started the field.

Software—The Role of Orchard-Hays. In 1954, William Orchard-Hays of the Rand Corporation, wrote the first commercial-grade software for solving linear programs. Many theoretical ideas such as ways to compact the inverse, take advantage of sparsity, and guarantee numerical stability were first implemented in his codes. As a result his software ideas dominated the field for many decades and made commercial applications possible. The importance of Orchard-Hays' contributions cannot be overstated for it stimulated the entire development of the field and transformed linear programming and its extensions from an interesting mathematical theory into a powerful tool that changed the way practical planning was done.

Network Flow Theory began to evolve in the early 1950's. Flood, Ford and Fulkerson in 1954, Hoffman and Kuhn in 1956 developed its connections to graph theory. Recent research on combinatorial optimization benefited from this early research.

Large-Scale Methods began in 1955 with my paper "Upper Bounds, Block Triangular Systems, and Secondary Constraints." In 1959-60 Wolfe and I published our papers on the *Decomposition Principle*. Its dual form was discovered by Benders in 1962 and first applied to the solution of mixed integer programs. It is now extensively used to solve stochastic programs.

Stochastic Programming began in 1955 with my paper “Linear Programming under Uncertainty” (an approach which has been greatly extended by R. J.-B. Wets in the 1960’s and J. R. Birge in the 1980’s). Independently, at almost the same time in 1955, E. M. L. Beale proposed ways to solve stochastic programs. Important contributions to this field have been made by A. Charnes and W. W. Cooper in the late 1950’s using chance constraints, i.e., constraints which hold with a stated probability. Stochastic programming is one of the most promising fields for future research, one closely tied to large-scale methods. One approach that the author, Peter Glynn and Gerd Inflanget investigated (1990), combines Benders’ decomposition principle, with ideas based on importance sampling and the use of parallel processors.

Integer Programming began in 1958 by R. E. Gomory. Unlike the earlier work on the traveling salesman problem by D. R. Fulkerson, S. M. Johnson and Dantzig, Gomory showed how to systematically generate the ‘cutting’ planes. Cuts are extra necessary conditions which when added to an existing system of inequalities guarantee that the optimization solution will solve in integers. Ellis Johnson of IBM extended the ideas of Gomory. Egon Balas and many others have developed clever elimination schemes for solving 0-1 covering problems. Branch-and-bound has turned out to be one of the most successful ways to solve practical integer programs. The most efficient techniques appear to be those which combine cutting planes with branch-and-bound.

Complementary Pivot Theory was started around 1962–63 by Richard Cottle and Dantzig and greatly extended by Cottle. It was an outgrowth of Wolfe’s method for solving quadratic programs. In 1964 Lemke and Howson applied the algorithm to bimatrix games. In 1965 Lemke extended the approach to other nonconvex programs. Lemke’s results represent a historic breakthrough into the nonconvex domain. In the 1970’s, Scarf, Kuhn, and Eaves extended this approach once again to the solving of fixed point problems.

Computational Complexity. Many classes of computational problems, although they arise from different sources and appear to have quite different mathematical statements can be “reduced” to one another by a sequence of not-too-costly computational steps. Those that can be so reduced are said to belong to the same *equivalence class*. This means that an algorithm that can solve one member of a class can be modified to solve any other in the same equivalence class. The *computational complexity* of an equivalence class is a quantity which measures the amount of computational effort required to solve the most difficult problem belonging to the class, i.e., its *worst case*. A non-polynomial algorithm would be one which requires in the worst case a number of steps not less than some exponential expression like Ln^m , $n!$, 100^n , where n and m refer to the row and column dimensions of the problem and L to the number of bits needed to store the input data.

Polynomial-Time Algorithms. For a long time it was not known whether or not linear programs belonged to a non-polynomial class called “hard” (such as the one the traveling salesman problem belongs to) or to an “easy” polynomial class (like the one that the shortest path problem belongs to). In 1970, Victor Klee and George Minty created an example that showed that the classical simplex algorithm would require an exponential number of steps to solve a worst-case linear program. In 1978, the Russian mathematician L. G. Khachian developed a polynomial-time algorithm for solving linear programs. It is an *interior method* using *ellipsoids* inscribed in the feasible region. He proved that the computing time is guaranteed to be less than a polynomial expression in the dimensions of the problem and the number of digits of input data. Although polynomial, the bound he established turned out to be too high for his algorithm to be used to solve practical problems.

Karmarkar’s algorithm (1984) was an important improvement on the theoretical result of Khachian that a linear program can be solved in polynomial time. Moreover his algorithm turned out to be one which could be used to solve practical linear programs. At the present time (1990), interior algorithms are in open competition with variants of the simplex method. It appears likely that commercial software for solving linear programs will eventually combine pivot type moves used in the simplex methods with interior type moves.

ORIGINS OF CERTAIN TERMS

Here are some stories about how various linear programming terms arose. The military refer to their various plans or proposed schedules of training, logistical supply and deployment of combat units as a *program*. When I first analyzed the Air Force planning problem and saw that it could be formulated as a system of linear inequalities, I called my paper *Programming in a Linear Structure*. Note that the term ‘program’ was used for linear programs long before it was used as the set of instructions used by a computer. In the early days, these instructions were called *codes*.

In the summer of 1948, Koopmans and I visited the Rand Corporation. One day we took a stroll along the Santa Monica beach. Koopmans said: “Why not shorten ‘Programming in a Linear Structure’ to ‘Linear Programming’?” I replied: “That’s it! From now on that will be its name.” Later that day I gave a talk at Rand, entitled “Linear Programming”; years later Tucker shortened it to *Linear Program*.

The term *Mathematical Programming* is due to Robert Dorfman of Harvard, who felt as early as 1949 that the term *Linear Programming* was too restrictive.

The term *simplex method* arose out of a discussion with T. Motzkin who felt that the approach that I was using, when viewed in the geometry of the columns, was best described as a movement from one simplex to a neighboring one. Mathematical programming is also responsible for many terms which are now standard in

mathematical literature, terms like *Arg Min*, *Arg Max*, *Lexico-Max*, *Lexico-Min*. The term *dual* is an old mathematical term. But surprisingly the term *primal* is new and was proposed by my father Tobias Dantzig around 1954 after William Orchard-Hays stated the need for a word to call the original problem whose dual was such and such.

SUMMARY OF MY OWN EARLY CONTRIBUTIONS

If I were asked to summarize my early and perhaps my most important contributions to linear programming, I would say they are three:

(1) Recognizing (as a result of my wartime years as a practical program planner) that most practical planning relations could be reformulated as a system of linear inequalities.

(2) Replacing the set of ground rules for selecting good plans by an objective function. (Ground rules at best are only a means for carrying out the objective, not the objective itself.)

(3) Inventing the simplex method which transformed the rather unsophisticated linear-programming model of the economy into a basic tool for practical planning of large complex systems.

The tremendous power of the simplex method is a constant surprise to me. To solve by brute force the assignment problem (which I mentioned earlier) would require a solar system full of nano-second electronic computers running from the time of the big bang until the time the universe grows cold to scan all the permutations in order to select the one which is best. Yet it takes only a moment to find the optimum solution using a personal computer and standard simplex or interior method software.

In retrospect it is interesting to note that the original problem that started my research is still outstanding—namely the problem of planning or scheduling dynamically over time, particularly planning dynamically under uncertainty. If such a problem could be successfully solved it could (eventually through better planning) contribute to the well-being and stability of the world.

By 1990, stochastic programming has become a very exciting field of research and application with research

taking place in many countries. This active and difficult field has already solved some important long term planning problems. I believe that progress depends on ideas drawn from many fields. For example, our group at Stanford is working on a solution method, which combines the Nested Decomposition Principle, Importance Sampling, and the use of parallel processors.

Prior to linear programming it was not practical to explicitly state general goals and so objectives were often confused with the ground rules for the solution. Ask a military commander what the goal is and he would say, "The goal is to win the war." Upon being pressed to be more explicit, a Navy man might say, "The way to win the war is to build battleships," or, if he is an Air Force general, he might say, "The way to win is to build a great fleet of bombers." *Thus the means to attain the objective becomes the objective in itself*, which in turn spawns new ground rules as to how to go about attaining the means such as how best to go about building bombers. These means in turn become confused with goals.

From 1947 on, the notion of what is meant by a goal has been adjusting to our increasing ability to solve complex problems. As we near the end of the 20th Century, planners are becoming more and more aware that it is possible to optimize a specific objective while, at the same time, hedging against a great variety of unfavorable contingencies which might happen and taking advantage of any favorable opportunity that might arise.

The ability to state general objectives and then be able to find optimal policy solutions to practical decision problems of great complexity is the revolutionary development I spoke of earlier. We have come a long way to achieving this goal but much work remains to be done, particularly in the area of uncertainty. The final test will come when we can solve the practical problems which originated the field back in 1947.

This article originally appeared in History of Mathematical Programming: A Collection of Personal Reminiscences, 1991, J. K. Lenstra, A. H. G. Rinnooy Kan, and A. Schrijver (eds.), Elsevier Science Publishers B.V., Amsterdam, The Netherlands. Copyright is held by the author and is reprinted here with his permission.