Operations Research

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

Feature Article—Survey of Nonlinear Programming Applications

Leon S. Lasdon, Allan D. Waren,

To cite this article:
28(5):1029-1073. http://dx.doi.org/10.1287/opre.28.5.1029

Full terms and conditions of use: http://pubsonline.informs.org/page/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.

© 1980 INFORMS

Please scroll down for article—it is on subsequent pages

INFORMS is the largest professional society in the world for professionals in the fields of operations research, management science, and analytics. For more information on INFORMS, its publications, membership, or meetings visit http://www.informs.org
Feature Article

Survey of Nonlinear Programming Applications

LEON S. LASDON

University of Texas, Austin, Texas

ALLAN D. WAREN

Cleveland State University, Cleveland, Ohio

(Received November 1979; accepted May 1980)

Several factors imply an increase in the use of nonlinear optimization models. We face serious problems of declining productivity and increasingly scarce, expensive raw materials. Computers are becoming cheaper and faster, and more efficient nonlinear programming (NLP) algorithms are being developed. This paper attempts to illustrate the potential of NLP by describing the application of nonlinear programming models to three classes of problems: petrochemical industry applications, nonlinear networks, and economic planning. Problems in the petrochemical industry ranging from product blending, refinery unit optimization, and unit design to multiphase production, and distribution planning are discussed. The nonlinear networks topic includes electric power dispatch, hydroelectric reservoir management, and problems involving traffic flow in urban transportation networks. In economic planning, we describe NLP applications involving large dynamic econometric models, a variety of static equilibrium models, and submodels of larger planning systems. In each area we consider the problem and its nonlinear model, the various algorithms and software systems which have been applied, and (where available) the benefits derived. Some obstacles are identified which currently inhibit the effective use of nonlinear optimization models, and research aimed at overcoming these obstacles is suggested. A comprehensive bibliography is included.

THERE ARE several classes of problems where nonlinear programming (NLP) has had significant impact. Those discussed in this survey are (1) oil and chemical industry applications, (2) nonlinear network problems, and (3) economic planning models. The first category includes problems in production and distribution planning, product blending, optimal operation of various processing units, and problems involving enhanced recovery from oil and gas reservoirs. The section on nonlinear networks focuses on optimization problems in electric power, water, and gas distribution networks, and equilibrium models of urban traffic flow. The third section offers descriptions of several classes of economic planning applications including optimal control of dynamic econometric models, single period macroeconomic development planning models, NLP.
formulations used in energy and agriculture, and a modeling system used in higher educational planning.

The models and algorithms described in Sections 1 and 2 have been used to achieve benefits in actual problem situations. In macroeconomic planning, almost all optimization applications are still in an experimental stage. Thus far, their chief contribution has been to yield insight into model behavior.

This paper can only hope to introduce the reader to these complex and diverse classes of problems. Many important details have necessarily been omitted, but the bibliography provides a means for further investigation. We have tried to achieve a balanced presentation, including the problem situation, its mathematical models, solution algorithms and software systems used, and results. Whenever possible, factors limiting the usefulness of NLP have been identified, and research aimed at overcoming the obstacles to implementation has been cited. Several important areas, such as engineering design, are not included. There are numerous excellent publications on this topic (Avriel (1973, 1979), Fox (1971), Bandler (1979), and Brayton (1978)). We have also excluded the use of NLP in statistics (Tapia (1978)).

1. NLP APPLICATIONS IN THE OIL AND CHEMICAL INDUSTRY

Linear and separable programming have been used extensively by the oil and chemical industry for many years. Applications range from product blending and product distribution (Zierer (1976)) to minimum cost development of oil and gas fields (Aronofsky (1957) and Babayev (1975)) and integrated production-distribution systems (Durrer (1977)). An increasing number of models involve nonseparable nonlinearities. This section describes a sample of these models, as well as the NLP algorithms used to solve them. NLP has also been applied to subproblems arising from the minimum cost development of offshore oil fields (Devine (1972), Frair (1975) and Lilien (1973)), but these applications are not discussed here.

Models for Optimal Planning and Operating Decisions

Given their history of extensive LP use, it was natural for oil and chemical companies to adopt NLP algorithms based on linear programming. Griffith (1961) of Shell Oil described a successive linear programming (SLP) algorithm for solving NLP's arising from refinery optimization. The objective and constraint functions were assumed to have the form $ax + f(y)$, where $a$ is a constant row vector and $f$ is a differentiable nonlinear function of the "nonlinear variables" $y$. Given initial values $y_0$, the nonlinear functions $f$ are approximated by the Taylor series terms.
Nonlinear Programming Applications

The objective is to minimize the function \( f(y) \) + \( \nabla f(y) \Delta y \). Upper and lower bounds (called step bounds) are imposed on \( \Delta y \) to ensure that the linear approximations are valid. The resulting LP is solved. If the new point \( y_1 = y_0 + \Delta y \) is acceptable, the procedure is repeated at \( y_1 \). If not, the step bounds are reduced and the LP is resolved. Various heuristics, described by Griffith (1961), Batchelor (1976), and Beale (1974), can be used for determining if \( y_1 \) is acceptable and for reducing (or increasing) the step bounds. Results given by Griffith (1961) describe application of this basic SLP algorithm to problems with up to 30 nonlinear variables and 100 nonlinear constraints.

Batchelor (1976) described a more sophisticated SLP algorithm which uses a mixture of SLP and reduced gradient concepts. It exploits the fact that, if the nonlinear problem is linearized about a point satisfying the problem equations (but not necessarily the bounds on the variables), the reduced costs (reduced gradient) relative to any basis can be used to derive a search direction in nonbasic space. In Batchelor (1976), a conjugate gradient method is used to obtain this search direction. A one-dimensional search procedure is initiated, and at each new point within the search, the problem is relinearized and the LP resolved. This LP solution may be viewed as one step of Newton’s method for restoring feasibility. Further details are given by Beale (1978).

Cheifetz (1974) at Gulf Oil applied an early version of the Batchelor and Beale method to a planning problem for oil field development. For long term planning the optimization model determines how much each reservoir should produce in each time period, how much gas or water should be injected to maintain reservoir pressures, and the level and timing of new plant and equipment outlays. Nonlinearities arise because reservoir productivity in time period \( t \) depends nonlinearly on cumulative production prior to \( t \), as well as on total gas and water injection. The model also contains many linear constraints in each period specifying total production from all reservoirs, product quality, and various capacity limits. Cheifetz reports run times of 2–3 minutes per major iteration for a 5-period model and 20 minutes for a 20-period model on a Univac 1108. He states that “acceptable solutions” may take 10–15 iterations and indicates that the algorithm is often terminated short of optimality.

The current version of the Batchelor and Beale algorithm (Batchelor (1976)) has been applied by SCICON Inc. to nonlinear programming problems of the Kuwait Oil Company. Their Reservoir Development Model (H. M. Ali (1978)) determines optimal investment policy, oil production rates, and gas and water injection into each reservoir over a 48-year horizon with 4-year time periods. The model is designed to minimize the net present worth of the discounted capital and operating costs of the facilities needed to provide specified amounts of oil for export in each period over the time horizon. The largest problem solved has 2336 constraints and 4392 linear variables. Typical problems have about 400
nonlinear variables and the LP matrix has about 20,000 nonzero elements. Solution typically requires about 50 major iterations with each iteration taking about 15 minutes of CPU time on a Univac 1108. The bulk of the time is consumed in the SCICONIC LP code. To the authors' knowledge, this is the only NLP code for large, sparse problems now available commercially.

A widely used SLP code, called POP II (for Process Optimization Program) was developed initially by Colville (1963). The current SHARE version (Smith (1965)) has been available since 1965 and has been extensively used and modified. A good description is available in Himmelblau (1972). Marathon Oil uses their version of POP for a variety of purposes. Their earliest applications involved nonlinear regression analysis, primarily for building process models, and these applications still exist. For example, in 1975, they used POP to determine approximately a dozen parameters for a hydro-cracking (Unicracking) model. More recent problems involve the development of the Cook Inlet Alaskan gas fields. This area contains three gas fields and many wells. Delivery contracts, over many years, call for the delivery of gas to a number of different locations. Decisions to be made involve the drilling and location of new wells and the addition of compressors to existing fields. Although this dynamic problem has some integer variables, it was formulated (in 1976) as a continuous problem with 10 time periods, 519 variables and 253 constraints.

The largest NLP model in use at Marathon involves the optimization of a complete refinery and it has a hierarchical structure. The first level consists of approximately 20 major FORTRAN simulation models for the refinery processes. These models provide input data for an LP system which handles the mass balance and other "bookkeeping" functions required to coordinate the individual models. The entire system is then embedded within POP.

Peterson (1971) discusses the application of an SLP code, developed at Chevron Research by Wise (1973), to the planning and operation of the Pascagoula refinery of the Standard Oil Company of Kentucky. The refinery included a two state Crude Unit, an Isomax, Catalytic Reformer, Steam/Methane H₂ Plant, Fluid Catalytic Cracker and an Alkylation Plant. These units were described by large FORTRAN simulation programs. In 1968, the plant (and model) was expanded, first by the addition of a large Paraxylene complex, later by adding a new Crude Unit, Isomax and Hydrogen Plant. Nonlinearities arise because the process models for these units are very nonlinear and also because nonlinear blending models are used. The SLP program was used for day-to-day process planning and also for expansion planning studies for the units added during the 1968–1971 period. In addition to stock evaluation and facilities planning,
the SLP system is regularly used to evaluate the effect of finished product specification changes on refinery profits.

Chevron has developed several special purpose SLP programs (Bod-ington (1973, 1979)). These include Batch Blending, and Fluid Catalytic Cracker, Reformer, and Refinery Optimization. The 1979 blending program is used by most Standard Oil Company of California refineries and is run about 400 times per month. It consists of a long range program with a 1-week time horizon and a short range program for daily runs. Nonlinearities arise chiefly because the product properties, such as octane number and vapor pressure, are nonlinear functions of the amounts and properties of each component in the blend. The short range program is needed because the stock properties and availabilities may differ from the values assumed by the long range program. Many factors can cause deviations from the long range values. For example, the immediate use of certain stocks may be required to make room in storage tanks or some amounts of old stock may remain in a tank, requiring modification of the blending specifications in order that original requirements are met.

Prabhakar (1970) and Buzby (1974) describe Union Carbides' application of a successive linear programming approach to a very large chemical process model. The optimization results are used to determine how Union Carbide should operate its olefins plants and chemical complex over the next time period (for example, 3 months), in order to minimize its costs, while meeting all its customer requirements. Since the model is representative of others in the oil and chemical industry it will be described in some detail.

The raw materials—natural gas concentrates and light petroleum fractions—enter each olefins plant at the thermal cracking units. The thermal cracking units decompose the inputs into six main product components plus by-products and unreacted input. This process can be controlled by adjusting temperatures, feed rates, etc. The output of various components per unit input is a highly nonlinear function of the process variables.

From the cracking units, the outputs are fed to a train of distillation columns for which the separation properties and capacities of each column are nonlinear functions of the input composition, feed rate and operating parameters. A typical plant has more than 30 or 40 distinct units that are needed to complete the separation. In addition to the distillation columns there are heat exchangers, heaters, coolers and units to split the streams. To further complicate the models, unreacted feed materials from various stages are recycled to the cracking units.

This whole process is described by a set of FORTRAN programs. This FORTRAN simulation model treats the raw material quantities and the process control settings as inputs, and provides, as outputs, the amount of each chemical component that will be generated, as well as the total
olefin plant cost. In addition, the simulator provides values for operating variables which, because of equipment limitations, must be restricted in value. The simulator does not impose these restrictions.

Each olefin plant output is sent to a nearby chemical plant. The chemical plants are described by a set of linear equations involving olefin outputs and linear variables representing the amounts of the various products produced, transferred from one shipping point to another, purchased externally, sold to a customer, or sold as surplus. The linear equations represent material balance constraints, customer requirements, and equipment capacity limits.

Letting \( y \) be the inputs of the olefin plants and \( x \) the linear variables for the chemical plants, the NLP problem is to minimize \( f(y) + c \cdot x \) subject to material balance \( (-P(y) + A;x = 0) \) and demand \( (A;x = b) \) constraints as well as bounds on the variables \( (R(y) \leq r, y \geq 0, 0 \leq x \leq u) \). The functional relationships between \( y \) and \( f(y) \), \( P(y) \) and \( R(y) \) are described by complex FORTRAN simulation models. In the specific case reported, the olefin subsystem involved 160 products, 8 plants, 40 terminals, 60 warehouses, 160 customer centers, and the linear production and distribution model had 500 products, 8 plants, 50 production facilities, 40 warehouses and terminals, and 120 customer centers.

The SLP algorithm used by Union Carbide utilizes the IBM LP Code MPSX to solve the LP subproblems. They report that this system solves the above problem, with 2200 linear constraints and 10,000 linear variables, in less than 3 hours on an IBM 370/165. Interestingly, the initial data reduction takes 6 hours, the initial LP solution takes 1 hour and involves 3,500 simplex iterations, and subsequent LP's (approximately 20 are required) take about 8 minutes and involve very few (average of 23) simplex iterations.

Several of the problems discussed above involve the flow of products through an oil refinery or chemical plant. In such problems, two or more intermediate products must often be mixed together ("pooled") prior to being blended into final products. Reasons for pooling include limited storage tank availability, the need to transport intermediate products in a single tank car, pipeline, etc.

A simple pooling problem is described by Haverly (1978–1980). It has three intermediate products, \( A, B, \) and \( C \), with known sulfur qualities, and two final products, \( X \) and \( Y \). \( A \) and \( B \) must be pooled; then the mixture can be blended with \( C \) to produce final products \( X \) and \( Y \). The constraints of the problem include upper limits on sulfur quality and on production quantity for products \( X \) and \( Y \) as well as material balance equations. Since the amounts of \( A \) and \( B \) to be pooled are decision variables, the pool sulfur quality, \( Q \), is unknown. The variable \( Q \) multiplies other variables in the sulfur quality constraints for products \( X \) and \( Y \), so the model is nonlinear.
The distinctive mathematical feature of this model is that it becomes linear if the pool sulfur quality, \( Q \), is assigned a fixed value. Problems of this type (linear when a subset of variables is fixed) are often solved by an algorithm called recursion (Haverly, 1978). Initial values are assigned to the nonlinear variables, the resulting LP is solved, and new values are calculated for the nonlinear variables. For example, in the pooling problem described above, a new value for \( Q \) may be computed if the quantities of \( A \) and \( B \) entering the pool (determined by the LP solution) are known. If the new values for the nonlinear variables differ significantly from the previous values, the process is repeated.

The simple pooling problem described above is used by Haverly (1978–1980) to illustrate the pitfalls involved in solving nonlinear problems by recursion. Two formulations of the model are presented, the first being a direct statement of the problem involving only one nonlinear variable, the pool sulfur quality, \( Q \). Haverly shows that for all initial estimates of \( Q \) in its feasible range, recursion converges to a single local optimum which is far from the global optimum. He then presents a more complicated recursion model involving an additional nonlinear variable ("recursion coefficient") and two additional "correction vectors" that serve to distribute the error made in estimating pool quality to the pool destinations. When applied to this model, recursion yields a global optimum from most of the starting points tested.

Lasdon (1979) shows that generalized reduced gradient and successive linear programming algorithms, applied to the pooling problem, show some significant advantages over recursion, although they have not yet been thoroughly tested on problems of significant size. White and Trierwireler (1980) describe Socal's experience and growing use of Haverly's "distributed recursion" approach. Their initial model, applied to a large coastal refinery, has 21 pooled intermediate blends; each with 3-9 recursion coefficients, for a total of 100 nonlinear variables. They expect the finished model to have approximately 60 pooled blends and 350 variables.

Models for Optimal Design Decisions

The models discussed previously deal with optimal planning or operational decisions. NLP models have also been used for optimal design. Consider the problem of temperature control of fluid streams in a continuous processing system. Often the emerging stream from a particular unit must be heated or cooled prior to entering some other process unit. Furnaces, coolers and heat exchangers are used to alter stream flow temperatures, with heat exchangers being particularly desirable in terms of energy conservation considerations.

There are many stream flows through a plant's complex of heating and cooling devices. The relationships between temperatures, heat contents
and flows are nonlinear, as is the function describing the cost of the heating/cooling units. Bryant (1979) of Standard Oil (Indiana) recently described an optimization design system that determines the parameters for heat exchangers, furnaces, and coolers to be built for a given process and temperature configuration. The system accounts for existing units as well as future units to be built.

Because of the complexity of the overall system, the computer time to evaluate the configuration cost (i.e., the objective function) is quite large. The authors have developed an approximating function for this cost which is inexpensive to evaluate. When its optimum is found, the true objective is optimized starting from the optimal solution to the approximate problem.

2. NONLINEAR NETWORKS

A. Electrical Power Network Models

An electrical power system consists of power generating plants, transmission lines, transformers, capacitors, switches, loads and tie lines connecting one system to neighboring ones. Power enters the transmission system from nuclear, hydro and/or thermal plants and from neighboring systems and leaves the system through loads, transmission line and other losses, and transfers to neighboring systems. In some cases, power storage units (i.e., pumped hydro or compressed air) are also included.

The primary objective of power system operation and control is to ensure that “user’s demands are met at the lowest cost compatible with adequate service quality—as measured in terms of satisfactory continuity in supply and of sufficiently small frequency and voltage deviations—and with acceptably low deterioration in the environment” (Quazza (1976)). This objective explicitly specifies an optimization problem with a cost function to be minimized and a variety of constraints describing the physical system and limits on acceptable performance. Meeting this objective by properly controlling the individual components of the power system is an extraordinarily complex task.

One of the difficulties associated with power system control is the sheer physical size of the system. The network may have several thousand nodes, may span hundreds of miles and involve plants with capacities in the billions of watts. To ensure continuity of supply, many thousands of constraints—in terms of tolerable voltage, current or power—must be satisfied. Environmental and ecological considerations can introduce many additional constraints.

Another major difficulty in dealing with electrical power systems is the tremendous range of time intervals over which various processes need to be controlled. At one extreme are problems relating to power system expansion and planning in which the time horizon may be as long as 10-
50 years (for nuclear plant construction). At the other extreme lie problems such as the impact of electromagnetic transients along a power line caused by lightning, in which the period of interest may be only milliseconds.

As is common in many similar situations, the system is modeled in different ways over different time periods with the implicit assumption that coupling effects can be ignored. Some initial efforts have been made to study the control of power systems from a hierarchical point of view, with each lower level primarily responsible for control actions over a shorter time horizon than the level above (DyLiacono (1967, 1974)). The various time frames give rise to different kinds of optimization problems. At the lowest level, controls usually involve high speed direct control by electromechanical devices such as switches and governors. At this level some optimization problems, primarily involving the sequence in which switching actions occur, have been formulated as integer programming problems (Sasson (1974)); but nonlinear programming has not been applied in any significant fashion.

At the next level of interest, involving time spans from minutes to hours, we have the problem of optimal dispatch—determining how much power should be produced by each of the currently active generating plants. Economic dispatch, reactive dispatch, as well as pollution dispatch, are all specialized versions of the more general optimal dispatch case. For time spans ranging from hours to weeks, the primary problem of interest is unit commitment and hydrothermal dispatch. The broad problem is to determine which power plants should be on-line (i.e., committed), and their average level of operation over each subinterval of time. The goal is to meet system-demand (assumed constant over each subinterval) at minimum cost, taking into account start-up and shutdown costs as well as other operating and physical constraints. Integer and dynamic programming formulations have been proposed (Dillon (1978), Muckstadt (1977), and Varamontes (1978)), but many recent and successful efforts have used nonlinear programming.

In an even longer time frame there is the problem of long term fuel and water management policies based on forecasts of future loads, water availability and fuel costs. This problem has also been formulated and solved as a nonlinear programming problem, although the most common approach has been to use dynamic programming (Harley (1978) and Twisdale (1979)). Expansion planning problems involving decisions to build and/or expand generating plants, and to expand the transmission network, have also been formulated as optimization problems. In these areas the primary solution strategies have involved linear (Sawey (1977) and Wall (1979)) and dynamic programming (Adams (1973) and Varamontes (1978)); although some use has been made of nonlinear programming (Bessiere (1970) and Sachdeva (1973)). Excellent bibliographies are
found in Fischl (1975), Knight (1974), Rettberg (1976), and Rosenthal (1977).

**Power-Flow Network Models**

In deriving a mathematical model of the power transmission system for use in optimal dispatch and unit commitment studies, it is usual to make several assumptions concerning both short and long term effects. Loads are assumed to be constant over the time interval of interest (from minutes to hours), and to act as constant power sinks. We assume as given the set of generating plants which are on-line and the values of real powers of the hydroelectric units (if any); we ignore “transient” (short term) phenomena in altering the output power of a given plant from its current value to a new value (this problem is considered in Chang (1971)); we assume frequency is constant, and we represent transmission lines by lumped, constant, linear network parameters. Okamura (1975) has developed a model which does incorporate effects such as voltage dependence of loads and system frequency deviation. Furthermore, although the system is a three-phase system, we assume that it is balanced and all three phases behave identically. This allows us to model only a single phase. In reality the phases are usually somewhat unbalanced (recently there have been efforts to incorporate such effects, see Snyder (1975)). If all three phases are treated simultaneously, the network model can triple in size.

The system is represented as a network of \( n + 1 \) buses (busbars or nodes) connected by \( m \) transmission lines, some of which may include transformers. The generators and loads are assumed connected at the buses. The network conservation laws give rise to a real and reactive power \( (P_i, Q_i) \) balance equation at each bus, in terms of the bus voltage magnitude and phase \( (V_i, \theta_i) \). The solution of these \( 2n + 2 \) nonlinear equations, given half of the variables, for the remaining variables is called the load flow problem. Load flow analysis is the most frequently used computer program for power systems applications (Sasson (1975)).

The most common approach to solving these equations is Newton's method. Writing the equations as \( F(x) = 0 \), the method iterates from a current point, \( x^* \), to a new point, \( x^* + \Delta x \), where the correction \( \Delta x \) is the solution of the very sparse linear system \( J \Delta x = -F(x^*) \) and \( J \) is the Jacobian matrix of \( F \). This system can be solved very rapidly by sparsity-programmed, ordered triangularization and back-substitution (Tinney (1967a, 1967b)). There are several versions and possible enhancements to the general Newtonian method with the best current approach, due to Stott, called the fast decoupled method (Despotovic (1974) and Stott (1972, 1974b)). This has been successfully applied to problems with as many as 7000 nodes (Sasson (1975)).
Some researchers (Sasson (1970) and Wallach (1968)) have experimented with nonlinear programming approaches for the load flow problem. Generally these approaches take the problem and convert it into an unconstrained minimization problem with objective function \( f_0(x) \) being the sum of the squares of the equations. The fact that the Newton direction is a direction of descent for the sum of squares \( f_0 \), has been used to enhance the convergence properties of Newton's method. Instead of using \( \Delta x \) as the correction, \( \alpha \Delta x \) is used where \( \alpha \) is determined by a one-dimensional search in the Newton direction. Various line search strategies, in this context, are discussed in Gross (1975) and Sasson (1971b). A good review and an excellent bibliography for the load flow problem, through 1974, is contained in Stott (1974a). More recent nonlinear programming approaches to the load flow problem have been within the context of an optimal load flow which we discuss shortly.

**Optimal Dispatch**

The optimal dispatch problem is to determine the allocation of power demand among the generating units available to minimize operating costs subject to various constraints. The economic dispatch problem is a special subclass of problems in which reactive power generation is assumed fixed and only the real power generation is to be determined. If the system involves both hydro and thermal plants then it is assumed that the hydro generation has been previously defined. Moreover, it is assumed that the thermal units that are to take on load are committed; therefore, start-up and shutdown costs can be ignored.

Early approaches neglected power losses in the transmission network. These power losses depend on the values for the powers at generator \( i, P_i \) and hence the total power required consists of two components—the system load, which is assumed known, and the losses \( P_{\text{loss}}(P) \). In the early 1950's Kirchmayer (1951, 1952) used the approximate “B-constants formulation” to represent transmission losses by the quadratic loss formula

\[
P_{\text{loss}} = B_0 + \sum_i B_i P_i + \sum_i \sum_j P_i B_{ij} P_j
\]

where the \( B \) matrix is square and symmetric. He minimized \( \sum_i F_i(P_i) \) subject to \( \sum_i P_i = P_{\text{load}} + P_{\text{loss}} \) and \( P_i^{\text{min}} \leq P_i \leq P_i^{\text{max}} \). The necessary conditions for a stationary point (ignoring the bounds) give the “coordination equations,” \( (dF_i(P_i)/dP_i) \cdot PF_i = \lambda \) where the penalty factor \( PF_i = (1 - \partial P_{\text{loss}}/\partial P_i)^{-1} \) and \( \lambda \) is the Lagrange multiplier for the equality constraint. This formulation is still in extensive use today, since in many cases it is sufficient to calculate the penalty factors prior to performing an economic dispatch and to consider them to be constant (Happ (1974)). However, the approximations introduced by the loss formula may give poor results, especially as the conditions of the system become different.
from the ones assumed in their computation. In addition, the use of this approach makes it difficult to incorporate other constraints.

**Optimal Load Flow**

If losses are accounted for by including the load flow equations as equality constraints for the optimal dispatch problem, then this version is usually called the optimal load flow problem. Carpentier (1962) formulated this problem as a nonlinear program. He determined the real power generations to minimize the operating cost and included equipment limitation constraints as problem inequalities. He used a variant of the Gauss-Seidel iterative procedure to solve the Kuhn-Tucker conditions as a system of simultaneous equations. The method was found to converge erratically, if at all, unless the initial guess for the variables was sufficiently close to the optimum (Carpentier (1968)).

In 1967, Dopazo published an exact penalty factor approach to the economic dispatch problem. Rather than using the $B$-constants formulation for approximating the system losses, he used the load flow equations to compute exact expressions for $\frac{\partial Q_{\text{flow}}}{\partial Q}$, and substituted these in the coordination equations.

Early in 1968, Peschon used Carpentier's formulation of the optimal load flow problem, but for the case of reactive power optimization alone. He used a reduced gradient approach to solve this problem. Later in 1968, Dommel and Tinney at the Bonneville Power Administration described a reduced gradient method without basis changes for solving the optimal power flow problem. They solved the power balance equations for the basic variables using a sparsity oriented Newton load flow method. The technique was successfully applied to a large real world problem (328 node, 493 branch network). Because all inequality constraints on the unknowns were handled by penalty terms, the authors were able to specify their basic and nonbasic variables initially and no basis update procedures were required. As recently as 1977, this paper was evaluated as "one of the most important that has so far been advanced in solution techniques of the optimal load flow problem" (Happ (1977)).

Abadie and Carpentier (1969, 1970b) at Electricité de France developed a Generalized Reduced Gradient algorithm and applied it to the optimal power flow problem. (Actually it was developed in the context of power systems and then extended to general nonlinear programming problems.) The optimization problem considered was basically the same one used by Dommel and Tinney. However Abadie and Carpentier handled the inequality constraints, other than simple bounds, by introducing slack variables and treating all constraints as equalities. The problem variables were divided into basic and nonbasic ones, with the basics being selected to be distant from their bounds. During the course of optimization the
basic variables may reach their bounds requiring basis changes to ensure continuing satisfaction of such contraints.

Peschon (1972) reported on computational experience in applying the GRG method to an expanded problem incorporating security constraints. These are typically bounds on line currents and on real power flows in the lines, quantities which are nonlinear functions of the bus variables. Results were reported for several systems, the largest being one with 100 nodes. The Dommel-Tinney approach was extended by Alsac (1974) to include contingency constraints. The equality constraints are the new load flow equations for the network, assuming the contingent failure occurred, and the inequalities are the new operating limits for this failure condition. They report results for an adaptation of the IEEE 30-bus standard load-flow test system with 33 contingency cases.

Carpentier (1973) also considered the optimal load-flow problem with contingency constraints and reported on the application of a modification of the earlier GRG method to a 92-node network with 117 lines. The contingency conditions can add more than 10,000 inequality constraints but only a few will be binding at optimality. Carpentier uses a relaxation approach and uses the contingent load-flow equations only to evaluate the system variables in order to check the contingent inequalities. Only those inequality constraints which are near or exceed their bounds are included. He solves this “reduced” problem, checks the relaxed constraints, and resolves a new problem if necessary. He reports an average computation time of less than 25 seconds on a CDC 6600.

Sasson (1971a) compared the approaches used by Dopazo (1967), Dommel (1968), and Sasson (1969a, 1969b, 1973), as well as a method developed at Electricite de France, limited to linear objective functions, and a method developed by El-Abiad and Jaimes (1969) which is similar to the Dommel-Tinney approach. He concludes that the Dommel-Tinney and Sasson methods are the most attractive, computationally and theoretically, and that the other methods lack flexibility and generality in addition to an inferior performance record on the test problems.

However, interest has continued in solving the economic dispatch version of the optimal load flow problem by exact versions of the B-constants or penalty factor method (the Dopazo approach), primarily for on-line real time applications where its speed and simplicity are particularly attractive. Happ (1974) reports tests of an exact penalty factor calculation procedure on the 118 bus IEEE system. DyLacch (1972, 1977) describes an implementation of this approach for real time use at the Cleveland Electric Illuminating Company. CEI uses the Stott fast decoupled load flow to solve the load flow equations and they compute the exact penalty factors by a sparsity oriented triangular factorization process. The complete optimization process takes “40–50 seconds on the Sigma 5 computer for our 240 bus model.” Wollenberg and Stadlin (1974)
also use a load flow analysis to compute the $B$-constants. They linearize security and environmental constraints and the power balance equality constraint, giving a separable convex program. Inequality constraints are incorporated as penalty terms. They use the Dantzig-Wolfe (Dantzig (1961) and Lasdon (1970)) decomposition technique. For a recent tutorial on the penalty factor approach, see Alvarado (1978).

Some of the concepts from the Hessian matrix approach of Sasson (1973) and the Dommel-Tinney method (1968) have been combined by Rashed (1974). More recently, Barcelo (1977) described a real time implementation of an optimal power flow with security constraints using a penalty function and Hessian approach. He reported computational results on a 1200-bus model with 1667 transmission lines, where each transmission line has a nonlinear upper and lower bounded inequality constraint specified to limit current flow. These constraints are incorporated by penalty functions. Beginning the iterative process from a solved case, with a 1% load increment, a solution was obtained within 33.6 seconds on an IBM 370/155.

The work by Sasson (1969a, 1969b) has been extended by Henser (1971) where tap ratios for transformers were introduced as additional variables. Billinton (1973) also used Sasson's approach, but used it to solve subproblems resulting from a decomposition of the optimum load flow problem. The decomposition arises naturally since, in a power system, there are strong couplings between the real powers and voltage phase angles $(P, \theta)$, and between the voltage magnitudes and reactive power $(V, Q)$, whereas these sets are weakly coupled.

A similar decomposition scheme was also proposed by Nicholson (1973). He first solves a real power economic dispatch problem, and then a reactive power minimum loss problem. To solve the first problem he approximates the individual generator cost functions by quadratics, and given initial variable values he computes an initial value for the net power loss which is assumed constant. The nonlinear constraints are linearized, and the resulting quadratic programming problem is solved using Beale’s (1959) method. An updated net power loss is computed and, if this is not sufficiently close to the previous value, the process is repeated. The resulting real generator powers are then held fixed and a gradient search procedure is used to determine the reactive power dispatch for minimum loss. The overall procedure then iterates between these two subproblems until convergence is attained.

A quadratic programming approach has also been suggested by Reid and Hasdorff (1973) using Wolfe’s (1959) technique and also by Naboma and Freris (1973). More recently, Biggs (1977) solved the optimal power flow problem by a sequence of quadratic programs using the REQP (Biggs (1976)) method. He reported test results on a 23-bus system with 79 variables, and 67 binding constraints to be identified out of a total of
Nonlinear Programming Applications

190; and concluded that REQP appears promising and deserves further consideration.

A trend in the application of optimization techniques to the optimal load-flow problem is the increasing generality of the problem formulation, in terms of added constraints due to security and environmental considerations. As the problem size has increased, it has become more and more attractive to consider linearized versions. A recent survey paper by Stott (1979) provides a comprehensive review and bibliography of linear programming applied to the power dispatch problem.

In these approaches, advantage is usually taken of the natural decomposition of the variables into \( P - \theta \) and \( Q - V \) groups, and most researchers have concentrated on the \( P - \theta \) problem with \( Q \) and \( V \) assumed constant. Usually the load flow equations are linearized about the current operating point (for other possibilities refer to Stott (1979) and Chan (1978)) to yield an incremental model of the form \( \Delta P = H \cdot \Delta \theta \). Upper and lower bounds on the generator powers lead to constraints \( P^\text{min} \leq \Delta P \leq P^\text{max} \); security constraints are expressed as \( L^\text{min}_y \leq A^y \cdot \Delta \theta \leq L^\text{max}_y \) where \( A^y \) is a row vector. In addition a power balance equation of the form \( \sum \Delta P = 0 \) (if losses are ignored), or \( \sum \beta \Delta P = 0 \), with \( \beta \) being the incremental transmission loss factors (if changes in losses are to be accounted for), is imposed. The objective function can be either linearized, modeled as piecewise linear, approximated by quadratics, or kept as a more complicated, but separable, convex function.

The above formulation has been referred to as the "nonsparse formulation" since the matrix \( A \) is not sparse. If the \( \Delta \theta \)'s, rather than the \( \Delta P \)'s, are selected as the main problem variables then an equivalent formulation (Stott (1979)) yields a sparse system for the security constraints at the cost of a more complex approach to account for changes in active power losses. The lack of sparsity in the first formulation is probably not too significant since the best approach to solving the problem seems to be a relaxation approach (Lasdon (1970)). With this technique only those security constraints which are violated (or very near to bounds) at the current point are used. The optimization problem is solved, the remaining security constraints are evaluated and any nonsatisfied constraints are added, and the problem is resolved. According to Stott (1979), theoretically constraints could oscillate in and out of the critical subset, but this has not been observed in practice. Hobson (1977) in evaluating the efficacy of this method found that in the 462-bus example there were potentially over 85,000 line constraints, but only 10 were violated during a test run. The actual LP implementations will not be discussed here, other than to mention that the dual simplex algorithm is generally preferred, since a feasible solution to the dual problem is easy to obtain, and because the relaxation process introduces new unsatisfied constraints into the primal problems.

Copyright © 2001 All Rights Reserved
Thus far this presentation has not explicitly considered the fact that power systems seldom exist in isolation but are tied together. The interconnections between power systems are usually handled by a natural decomposition that cuts the network at the corporate boundaries between different power companies and represents tie powers as additional constraints (Peschon (1971)). Optimization within a one-company area will generally not be optimal for a multiarea system, or power pool; hence a larger problem, combining several systems, may need to be examined. For an excellent review of work related to the multiarea economic dispatch problem see the paper by Happ (1977). An earlier paper by Sasson and Merrill (1974) provides a good review of optimization in power systems up to 1974.

Hydrothermal Dispatch and Unit Commitment

The previous discussion dealt with optimal dispatch problems involving time periods measured in minutes or hours. We now consider some medium term planning problems in hydroelectric power systems, with time periods measured in hours or weeks. For a mixed hydro and thermal generation power system, the problem is to decide which units should be on-line and what their level of operation should be in each time period. Given the dimensionality of the problem it is usual, as in Bonaert (1971), to decompose it into two subproblems: one involving the hydro system operation, and the other the thermal plants and electrical networks. The optimal reservoir management policy (hydroelectric subproblem) is determined by minimizing an appropriate hydro cost function subject to system constraints. Coupling of this problem to the thermal dispatch problem is achieved through the objective function and the inclusion of electrical transmission losses into the forecast loads to be met. The thermal subsystem is then optimized with the previously computed values for hydropower output and the process repeated until the changes are sufficiently small.

Various approaches have been used for solving the overall problem as well as the subproblems. For a review of work up to 1974 see Sasson (1974) and Gruhl (1975). Rosenthal (1977a) has compiled a comparative bibliography on operations research applied to reservoir management that is available on request. Since the most successful applications of nonlinear programming to these problems have been for hydroelectric systems, we consider only the hydro problem.

Hydroelectric systems consist of hydraulically and electrically interconnected reservoirs and hydroelectric generating plants. Planning and control models cover various time horizons. Typically a mid-range planning model would have a 1-2-year time horizon, subdivided into time periods from a week to a month. For better resolution for short term
control, a short term time horizon can then be one or two mid-range periods in length, and this horizon is again subdivided into smaller intervals. Gagnon and Bolton (1978), for example, consider a short term planning model with a 1-week time horizon subdivided into 8-hour increments.

The mathematical model for the system involves water balance equations at reservoirs and generating plants, functions describing power generation as a function of various water levels and flows, and constraints due to physical limits, desired operations, and irrigation, navigation, recreation and conservation considerations. Several objective functions have been proposed, generally nonlinear and nonseparable. Hanscom (1976, 1977) at Hydro-Quebec uses an objective function which minimizes the “cost” of meeting forecast weekly loads (with the deficit met by a thermal plant). The objective includes a reward for final reservoir storage, used for coordination with long term planning models. Rosenthal (1977b) uses a similar objective, but treats final water levels on reservoirs as prescribed conditions. At the Bonneville Power Administration (Gagnon (1978), Hicks (1974) and Baxter (1975)) the emphasis has been on maximizing the energy remaining in the system at the last period, with all other considerations incorporated into the constraints. In Gagnon (1974) the objective function includes two components: one designed to provide a minimum average power deficit over the planning period, and the other to distribute this deficit as uniformly as possible throughout the period.

These formulations have linear flow conservation constraints, plus bounds on reservoir levels and outflows. The BPA, Rosenthal, and Hanscom approaches all solve the resulting nonlinear network problem by different versions of a reduced gradient method. The BPA group selected the water storage contents as permanent nonbasic variables, and the flows as basic. Bounds on the flows were handled by exterior penalty terms, and bounds on the nonbasics were handled by a simple projection method. Conjugate gradient and steepest descent methods were used, with the conjugate gradient methods being superior (Gagnon (1974)). Their largest test problem included 38 reservoir plants, 45 run of river plants, 49 time periods, 5929 variables, 4067 equality and 11,172 inequality constraints. On a CDC 6400 computer the solution took about 1 hour.

Hydro-Quebec reversed the BPA choices and treated the reservoir outflows as nonbasic, and the water storages as basic variables. They avoid basis changes by projecting onto the bounds outflows by solving a quadratic “direction finding problem.” Computational results indicate convergence to an optimum “from scratch” in 10 CPU minutes for a 10-reservoir, 52-time period problem on an IBM 370/168.

Rosenthal (1977b) uses an implementation of the Convex Simplex Method which exploits the network structure of the flow conservation
equations (see the following section on traffic assignment problems for more details and other application of this approach). He used a model of the TVA system with 6 reservoirs and 52 periods. His code, running on a DEC system 10 computer, took 9 minutes and 35 seconds of CPU time to achieve a final result somewhat better than that obtained by an earlier TVA nonlinear programming algorithm, running for 30 CPU minutes on an IBM 360/50.

B. Traffic Assignment and Computer Network Models

In electrical power network problems, physical laws determine the equations governing power flow. When formulated in terms of real quantities, these equations are nonlinear. In the hydroelectric power problems of Section 2A, the mathematical model involves the flow of water between reservoirs. The associated inventory equations are linear. In this section, we consider the flow of vehicles on the arcs of a transportation network. These flows also obey a linear system of flow conservation constraints. There are no physical laws to determine which of the many possible solutions to these linear equations will actually occur in a particular transportation network. However, there are certain equilibrium principles, phrased as nonlinear objective functions, which have yielded accurate results in several tests.

To model the equilibrium flow of traffic in a transportation network, the area served by the network is divided into zones, and a study is made to determine the number of trips made between each pair of zones during a particular time of the day (for example, morning or evening rush hours). The zones and the associated transportation network are modeled as a directed graph. In the node-arc formulation of the problem given by A. Ali (1978), the flows, \( x^k_j \), are the number of vehicles on arc \( j \) originating at node \( k \). The flow conservation constraints are

\[
Ax^k = r^k, \quad k = 1, K
\]

where \( A \) is the node-arc incidence matrix of the network, \( x^k \) is the vector of all \( x^k_j \)'s, and \( r^k \) is a vector of supplies and demands, with a supply at node \( k \), demands at all nodes that are destinations for traffic which originates at \( k \), and zeros elsewhere.

The objective function is based on one of Wardrop’s two principles of traffic equilibrium (Wardrop (1952)). Average travel time for a vehicle on arc \( j \), \( t_j(f_j) \), is assumed to be a nondecreasing function of the total number of vehicles, \( f_j \), traveling on that arc. The total flow, \( f_j \), is denoted by

\[
f_j = \sum_k x^k_j.
\]

In Wardrop’s first principle (user optimum) the equilibrium distribution of traffic is such that any other path a vehicle might choose between its origin and destination has an equal or higher travel time than the path
Nonlinear Programming Applications

actually chosen. Under the second principle (system optimum) total travel time of all vehicles is minimal. Beckmann (1956) shows that principle 1 leads to the objective function

$$z_1 = \sum_j \int_0^{y_j} t_j(y) \, dy.$$  \hfill (3)

That is, if (3) is minimized subject to (2), (1), and $x^k \geq 0$ for all $k$, the Kuhn-Tucker optimality conditions imply that the first principle is satisfied. Principle 2 leads directly to the objective

$$z_2 = \sum_j t_j(f_j) \cdot f_j.$$  \hfill (4)

The function $z_1$ is convex in the $x_j^{ik}$'s if $t_j$ is nondecreasing and continuous (Abadie (1970a)). If, in addition to these properties, $t_j$ is also convex, $z_2$ is convex over the nonnegative orthant. An arc-chain formulation of the problem using these same objectives is given by Florian (1974, 1976).

Florian (1976) gives an expression for the “volume delay” function $t_j$ used in a study of Winnipeg, Canada. LeBlanc (1976) and Nguyen (1974) specify a delay function used by the U.S. Bureau of Public Roads as

$$t_j(z) = a_j(1 + 0.15 \cdot (z/c_j)^4)$$  \hfill (5)

where $a_j$ is the travel time on arc $j$ at mean free speed and $c_j$ is the design, or practical capacity, of the arc. Both expression (5) and the function given by Florian (1976) are strictly convex and increasing for $z \geq 0$. Hence the problem of minimizing $z_1$ or $z_2$ subject to (1), (2) and $x^k \geq 0$ for all $k$ is a convex program with linear multicommodity flow constraints (1).

The only coupling of commodities is through the objective function, because of the relations (2). Practical problems of this type can be very large. As pointed out by Nguyen (1974), a problem with 50 zones (supply-demand nodes), 500 arcs, and a total of 100 nodes, would have 5,000 constraints (100 per commodity times 50 commodities) and 25,000 variables (one for each arc-zone combination).

Algorithms for solving traffic assignment problems reduce dimensionality difficulties by exploiting problem structure. Several researchers, for example LeBlanc (1975), and Florian (1976), have used the Frank-Wolfe (1956) method. The direction finding subproblem of this algorithm, when applied to traffic assignment problems, reduces to finding, for each $k$, the shortest route from origin $k$ to all of its destinations. Although the rate of convergence of this method is quite slow (Frank (1956) and Wolfe (1970)), each iteration may be performed quickly, even for large problems. Florian (1976) reported that in problems with 140 zones, a total of 1,035 nodes, and 2,789 arcs, the Frank-Wolfe algorithm converged in 15–18 iterations, used about 700 seconds/run on a CDC Cyber 74, and had an average cost/run of $315. In an application to the Chicago area having 999 zones, a total of 9,400 nodes, and 29,000 arcs (hence 9,390,600
constraints!), Eash (1978) reported convergence of Frank-Wolfe to a nearly optimal solution in 4 iterations. The implementation which produced these results is now available to practitioners in the UMTA/FHWA Urban Transportation Planning System (Eash, 1978).

The Convex Simplex Method has also been applied to traffic assignment problems (Florian (1976) and Nguyen (1974)). This is a reduced gradient method in which only one nonbasic variable at a time is varied. Although the rate of convergence of this procedure is slow (Luenberger (1973)), it may also be tailored to the network structure. Computations require maintaining basis trees and manipulating simple cycles in the graph for each commodity. Such operations may be carried out very quickly using techniques developed for linear network flow problems (Bradley (1977)). Florian (1976) reports an average time/run on the problem referred to above of about 500 seconds (Cyber 74) and an average cost/run of about $290.

There is some controversy over the need for highly accurate solutions to traffic assignment problems. Some transportation analysts suspect that suboptimization may be superior to optimization in these problems, since in reality not everyone knows the least time route. If this is true, then the slow ultimate rate of convergence of the convex simplex and Frank-Wolfe methods is not a serious drawback.

Florian (1976) also describes validation tests of these models, in which computed equilibrium solutions were compared with observed data taken in Winnipeg, Canada, in 1971. Results showed good correspondence between actual and predicted traffic on the arcs, with the goodness of fit improving as arc volume increased. The correspondence between observed and predicted arc travel times was slightly worse, but still reasonable. Origin to destination accessibility times were found by computing shortest routes from each origin to each destination, using only arcs which had positive flow. These were compared to route travel times computed using observed traffic volumes, and found to be in close agreement. In general, traffic assignment models were judged completely adequate for planning purposes.

Equilibrium traffic assignment models can be extended in several ways. Beckman (1956) and Florian (1974) consider problems where the number of trips between each origin-destination pair (fixed in the models discussed previously) is a given function of the travel time between the pair. An optimization problem is defined whose optimal solution satisfies Wardrop's first principle. The objective of this problem is analogous to the social welfare functions discussed in Manne (1979) for finding economic supply-demand equilibria. Multiple modes of travel are considered in Florian (1977) and Abdulal (1978).

Problems of optimal routing in packet-switched computer communication networks may also be modeled using this framework. In such
networks, messages are segmented into packets. The packets are stored in queues at intermediate nodes, and are forwarded to other nodes when communication channels become free. Here, the variables $x_j^k$ are the time average flow (measured in bits/second) of all packets on channel $j$ bound for destination $k$. The objective may be either to minimize the average (over time and over all arcs) delay per packet; or to maximize system throughput, subject to an upper limit on average delay. Expressions for these objectives and an example can be found in Cantor (1974).

Other Nonlinear Network Flow Problems

Several other classes of problems may be formulated as optimization of a nonlinear objective subject to linear flow conservation equations. Prominent among these is the pipe network analysis problem described in Collins (1977, 1978). The network under consideration is an interconnection of reservoirs, pumps, pipes, and valves through which water flows to satisfy demands at some nodes. Optimal design or operation of such networks is discussed by Alperovits (1977). Here the problem is one of analysis—that is, to find the flows in the network, given that the pressure at some reservoir nodes is fixed, and that there are fixed inflows at some reservoir nodes and desired outflows at other demand nodes. In Collins (1977, 1978) it is shown that, just as with traffic assignment problems, an optimization problem may be defined whose solution yields the equilibrium flows and pressures. The constraints of this problem are conservation constraints as in (1) with a single commodity, i.e., $K = 1$, plus $x_j^k \geq 0$. The objective has the form of $z_1$ in (3), where $f_j(y)$ now represents the pressure drop (or gain) along arc $j$ when the arc flow is $y$. As pointed out by Collins (1977), $f_j$ is usually increasing when the arc represents a passive element, e.g., a pipe or valve, but may not be increasing if arc $j$ represents a pump. Hence $z_1$ may not be convex in all cases, although it was in the example discussed in Collins (1978). In the preceding case, a problem with 452 nodes and 530 arcs (representing a scaled-down version of the Dallas, Texas, water distribution system) was solved using the Frank-Wolfe, convex simplex, separable programming, and Newton algorithms. Of the first three methods, separable programming was by far the fastest, although somewhat inferior in accuracy to the others. The convex simplex method was comparable at higher accuracies. Newton's method, often used to solve such problems in practice, performed well for initial estimates near the equilibrium solution, but exhibited erratic behavior for starting points farther from optimality.

The problem of finding steady state currents and voltages in a nonlinear resistive electrical network may be formulated as a NLP, in much the same way as for water distribution networks. Problems with nonlinear objectives and linear network constraints also arise in several other
contexts. The hydroelectric energy scheduling problems discussed earlier have this form, and the convex simplex approach of Rosenthal (1977b) exploits the network structure. As pointed out by A. Ali (1978), transportation problems with random demands also have this structure, as do dynamic production scheduling problems with nonlinear costs (Ratliff (1976)).

3. ECONOMIC PLANNING

Econometric Models

Econometric models are systems of difference equations having the form

$$g_t(y_t, u_t, z_t, y_{t-1}, u_{t-1}, z_{t-1}, \ldots, y_{t-r}, u_{t-r}, z_{t-r}, p) = e_t, \quad t = 1, T. \quad (6)$$

In (6), $g_t$ is an $m$-vector of functions, $y_t$ is a vector of $m$ endogenous variables, and $u_t$ is a vector of $n$ instrument or control variables. The vectors $z_t, p$, and $e_t$ represent the exogenous variables, model parameters, and equation residuals, respectively. According to Klein (1975), econometric models are usually intended to investigate economic stabilization policies in the short to medium term, up to approximately a 5-year planning horizon. Time periods, indexed by $t$, are typically quarters.

The endogenous vector $y_t$ contains those variables deemed necessary to describe the state of the economy. They may be roughly categorized as prices (interest rates, price indices, wage rates), stocks or quantities (gross national product, consumption, employment, investment, inventories, savings, corporate assets) and flows (taxes, dividends, government transfer payments, imports and exports, disposable income). A medium sized model could contain 50-100 such variables (see Athans (1976)), while large models range from around 200 to over 1000 endogenous variables (Nepomiastchy (1978)). The model contains as many equations ($m$) as there are endogenous variables, and it is assumed to be solvable recursively for $y_t, y_{t-1}, \ldots, y_{t-r}$ if values for all the other quantities in (6) are specified. These include the initial conditions $y_0, \ldots, y_{1-r}$, and values for $u_t$ and $z_t$ for $t = 1 - r$ to $t = T$.

The variables $u_t$ are a subset of those quantities which are wholly or partially under the control of policy makers. They may often be classified as fiscal (government spending, tax rates, government or business transfer payments) and monetary (treasury bill interest rate, federal funds discount rate, bank reserves). There are typically far fewer of these than there are endogenous variables. For example, in the experiments reported in Palash (1977) using the MIT-PENN-SSRC(MPS) econometric model, $y_t$ has about 200 components while $u_t$ has 2 components (government nondefense nonwage spending and the treasury bill rate). The exogenous
variables include such quantitites as export price indices, population, “dummy” variables to represent the effects of unusual conditions like strikes or oil embargos, and some components of endogenous quantities, e.g. defense spending and transfer payments made by state and local governments.

The model equations $g_i$ are of two types: behavioral equations and identities. The identities define one variable in terms of others, e.g. GNP = $C + I + \Delta INV$ specifies GNP as the sum of consumption, investment, and the change in inventories. Identities usually contain only a few variables, and are often linear. However, operations such as division by a price index may introduce nonlinearities. Behavioral equations often include demand and production functions. Their structure is determined by a mix of theoretical and empirical principles, and they are fully specified by choosing the parameters $p$ so that the variable explained by the equation fits the historical data. The quantities $e_i$ are the errors which inevitably result in this fit. Behavioral equations usually contain relatively few variables per equation (perhaps an average of 5) and often involve simple nonlinearities, for example, logs, products, or quotients. These arise from such operations as division by price indices to transform from nominal to real terms, the use of log-linear relations, and multiplying employment by a wage rate to obtain income. Values of variables from past periods (lagged variables) are often included to model the slow response of economic systems. Usually the different classes of variables $y$, $u$, and $z$ will appear in the model equations with different lags, ranging from 1 to, say, 20 periods. For simplicity we have shown only the maximum lag, $r$, in (6).

Most models have significant proportions of both behavioral and definitional equations. This, and the properties of these equations, imply that large models are sparse and mostly linear, and that the nonlinearities are fairly simple, drawn from a small class of nonlinear functions. The model equations can be conveniently grouped into blocks according to the variables they explain. The appendix of Athans (1976) groups its 37 behavioral and 47 definitional equations into these classes: Consumption Expenditures, Private Fixed Investment, the Labor Market, Financial Markets, Prices and Wages, and National Income and its Distribution.

Optimization problems subject to the model equations (6) are stochastic control problems. Most current work with large models uses the “certainty equivalence” or deterministic approach. In this strategy, the random residuals, $e_i$, are replaced by known values, usually either their expected values (zero) or in experiments over periods of the past, their historical values. Forecasts or historical values are used for the exogenous variables $z_i$, and the parameters $p$ are determined by statistical fitting procedures. Suppressing the dependence on these quantities and introducing an objective function, we consider the problem
Lasdon and Waren

(OCP)

\[
\text{minimize } f(y, u)
\]

subject to

\[ g_t(y_t, u_t, y_{t-1}, u_{t-1}, \ldots, y_{t-r}, u_{t-r}) = 0, \quad t = 1, T. \] \hspace{1cm} (7)

In the above, \( y \) and \( u \) are vectors composed of the subvectors \( y_1, \ldots, y_T \) and \( u_1, \ldots, u_T \), respectively.

Problem OCP has several features which distinguish it from "standard" discrete time optimal control problems (see Bryson (1969) for standard formulations). The most significant of these is that the model equations (7) are implicit, that is, \( g_t \) does not specify \( y_t \) as an explicit function of its other arguments. Hence the model equations must be solved numerically, first for \( y_t \), then \( y_2, \) etc., given initial values for the \( y \)'s and a set of \( u_t \) values. Econometric models are most often used in this simulation mode, to answer "what if" questions regarding the future effects of various policies, under a specified set of exogenous conditions (Klein, 1975).

Another distinctive model feature is that \( r \), the maximum lag in (7), will almost always be greater than 1. Although (7) can be placed in "standard" first order form \((r = 1)\) by defining new variables equal to the \( y_{t-1}'s \) for \( r > 1 \), the resulting system has many new variables and equations. Most investigators choose to work with (7) directly.

In most problems solved to date, the objective \( f \) is a "goal" function; either quadratic or piecewise quadratic. Its terms are weighted squared deviations of components of \( y \) or \( u \) from specified desired values (see Norman (1974), Holbrook (1974), and Chow (1975)). There are typically only a few key endogenous variables appearing in the objective, for example, the unemployment with inflation rates used by Craine (1976). One-sided quadratics, with or without indifference bands, are sometimes used (see McCarthy (1977) and Palash (1977)) to penalize for inflation and unemployment values above (but not below) the target levels. Palash recommends using exponential functions to capture such one-sided preferences in an objective that is infinitely differentiable. Control variables often appear in the objective also, reflecting the desire to keep instruments, such as government spending or interest rates, from deviating significantly from previous period values or from a smooth "ideal" path. McCarthy (1977) suggests restricting the controls to be polynomials of degree \( k \) over time by forcing the \((k + 1)st\) time difference to be zero. This assures smooth variations in control and reduces the dimensionality of the problem in \( u \)-space. Craine (1977-78) approximately enforces the polynomial restriction by allowing the \((k + 1)st\) difference to differ from zero, in each time period, by an error variable; and by introducing terms in the squares of these variables into the objective. Both papers give computational results using the 200 equation MPS model.
Problem OCP may also contain inequality constraints. The most common are simple bounds on the state and control variables. More complex inequality restrictions may be placed in this form by adding slack variables, and viewing the resulting equality as another model equation. Most problems solved thus far do not involve inequalities, therefore we will postpone their consideration.

Solution Algorithms

Several classes of algorithms have been applied to OCP. Haurie (1974) uses an Augmented Lagrangian method in a problem involving a 39 equation nonlinear model of Canada. Bock (1977-78) solves OCP by applying Newton's method directly to the Kuhn-Tucker necessary conditions for optimality. However, the majority of published algorithms for solving OCP are reduced gradient methods. They differ from the Newton and Augmented Lagrangian approaches in that the model equations are satisfied at each iteration, rather than only at optimality. The model equations (7) are assumed to define \( y \) as a differentiable function of \( u, y = h(u) \), and the reduced objective function (also assumed differentiable) is \( F(u) = f(h(u), u) \). OCP is then a problem of unconstrained minimization of \( F(u) \). Various unconstrained optimization algorithms have been applied: Steepest Descent by Fair (1974) and Normal (1974); Conjugate Gradient (McCarthy (1977), Norman (1974), and Mantell (1977-78); Variable Metric or Quasi-Newton (Norman (1974), Mantell (1977-78), and Athans (1976)); and various approximations of Newton's method (Garbade (1974), Chow (1976), and Rustem (1978)). Within these algorithms, various line search strategies have been employed. Even in models with many equations, for example, \( m > 100 \), the number of controls is usually small, that is, \( n = 2 \) or 3. Thus, for typical values of \( T \) in the range 10-20, the dimension of \( u, nT \) is small, so manipulation of \( nT \)-dimensional matrices as required by Quasi-Newton or Newton methods is not prohibitive.

All these algorithms require the reduced gradient \( \nabla F \). Fair (1974) approximates \( \nabla F \) by finite differences:

\[
\frac{\partial F}{\partial u_j} \approx \frac{f(h(u + \Delta u_j), u + \Delta u_j) - f(h(u), u)}{\Delta u_j},
\]

where \( \Delta u_j \) is a vector of zeros except for a small positive value in position \( j \). This leads to an algorithm which is easy to implement; a simulation routine which computes \( h(u) \), given \( u \), is interfaced with an unconstrained minimizer that accepts \( \nabla F \) and produces new values for \( u \). However, since the \( h \)-values are obtained numerically, inaccuracies in \( h \) limit the accuracy of \( \nabla F \), which, in turn, limits the accuracy of the optimization. This problem gets worse as model size increases (Norman (1974)), since accurate model solutions become increasingly expensive, especially if (as
Lasdon and Waren

is usual in this approach) a Gauss-Seidel method is used to solve the model equations. Fair (1974) reports satisfactory solution of problems with $m = 19, 6 \leq T \leq 10$ and $1 \leq n \leq 4$, using various unconstrained minimizers. In Nepomia
cstch (1978) and McCarthy (1977), problems involving the approximately 200 equation MPS model with $T = 7$ (Norman (1974)) and 11 (McCarthy (1977)) and $n = 2$ or 3 are solved using (8) and the Gauss-Seidel method.

A more complex but more accurate method of computing $\nabla F$ is through the fundamental equations which define it (Abadie (1970a)). Applied to OCP these become

\begin{align}
\mathbf{b} & = \mathbf{a}f/\mathbf{a}y, \\
\nabla F & = \mathbf{a}f/\mathbf{a}u - \mathbf{g}g/\mathbf{a}u
\end{align}

where $g = (g_1, g_2, \ldots, g_T)$ and $B = \mathbf{g}g/\mathbf{a}y$. Since $g_i$ depends only on $y_i, y_{i-1}, \ldots, y_{i-r}$, $B$ is square lower-block triangular, with diagonal blocks $\mathbf{a}g_i/\mathbf{a}y_i = B_i$, and $B$ is nonsingular if and only if each $B_i$ is nonsingular.

Nonsingularity of $B$ at a point $(y, u)$ satisfying (7) is a sufficient condition that the implicit function $h$ exists in some neighborhood of $u$; therefore we assume that $B$ is nonsingular. Equation (9) can then be solved recursively: first for $\mathbf{b}_r$, then for $\mathbf{b}_{T-1}$, etc., where $\mathbf{b}_r$ is the subvector of $\mathbf{b}$ corresponding to $g_r$. The equations yielding $\mathbf{b}_r$ have coefficient matrix $B_r$.

All matrices $B_i$ have the same zero-nonzero pattern, and will be sparse for medium and large models. Sparse matrix methods permit solution of these linear systems rapidly with modest storage requirements. In Nepomia
cstch (1977-78), experiments with the 130 equation STAR model of France showed this analytic method of computing $\nabla F$ to be from 40 ($T = 10$) to 106 ($T = 30$) times faster than the finite difference approach.

The most time consuming step in a reduced gradient solution of OCP is simulation, solving the model equations (7) recursively to find the $y$'s for a given vector $u$. This must be done several times during each line-

search to evaluate the reduced objective $F$. Older algorithms, which use the finite-difference approximation (8) to $\nabla F$, usually use a variant of the Gauss-Seidel method (Ortega (1970)) for simulation. It is simple, but often slow (Norman (1974) and Mantell (1977-78)), and sometimes unrealistic (Norman (1974)). Recent approaches using (9) and (5) to compute $\nabla F$ have used some variant of Newton's method to solve the model equations, e.g., Drud (1977, 1977-78), Mantell (1977-78), and Nepomias
tch (1977-78). This is a natural choice, since both Newton's method and equations (4) use the basis matrix $B$. A reduced gradient implementation of this type is more complex than one using (8) and the Gauss-Seidel method. First derivatives of the model equations, with respect to all arguments, are required (derivatives may be approximated by finite differences), as are sparse matrix routines. There are several advantages,
however. Newton's method is independent of the ordering of the equations. It converges in 1 iteration for linear models (not true for Gauss-Seidel), and should require significantly fewer iterations for nonlinear models. In Nepomiasihty (1977–78), a variant of Newton's method required from \( \frac{2}{3} \) to \( \frac{4}{3} \) the time of Gauss-Seidel to perform a 10-period simulation of the 130 equation STAR model, the advantage increasing as higher accuracy was required. In Mantell (1977–78), Newton's method required 2.88 iterations per time period to simulate a 27 equation model with \( T = 13 \), while Gauss-Seidel required 69.2.

Problems with upper and/or lower bounds on \( u \) are easily solved by reduced gradient methods; only a few changes in the unconstrained minimization algorithm need be made. Bounds on the endogenous variables cause more difficulties. Building on earlier work by Abadie (1970a), Drud (1977, 1977–78) describes a reduced gradient algorithm including basis changes. If a component of \( y \) hits a bound, it becomes independent (nonbasic) and some free variable in the current time period becomes dependent (basic). Drud assumes that such an interchange can always be made. An alternative which is easier to implement (but may be slower) is to use Augmented Lagrangian methods to transform the problem into a sequence of problems without endogenous variable bounds.

Computational optimal control experiments have been performed using medium to large-size models. In Palash (1977), the 200 equation MPS model was used to investigate the effect of the objective function specification and the finite planning horizon, \( T \) (that is, "end effects") on optimal solutions. Two planning periods were used: the volatile period 1971.1–1975.1 and the more "normal" period 1965.1–1969.1. Exogenous variables and equation residuals were set to their historical values. The objectives all reflected the tradeoff between inflation and unemployment rates, while control variables were Treasury bill interest rates, and Federal Government nondefense, nonwage expenditures. Experiments indicated that optimal solutions using a linear objective were more affected by the value of \( T \) than those employing truncated quadratics or exponentials, but end effects were also significant with these. The various objectives used, penalized either values of inflation and unemployment above 0 and 4.5%, respectively (deviations from their "long run" values), or deviations from the value of inflation at the start of the horizon. Solutions in each case were presented and analyzed.

Craine (1978) used the MPS model to investigate monetary policy over the volatile period 1973.3–1975.1. The control variable is money supply M1, while the objective is truncated quadratic, penalizing values of \( u \) above 4.8%, \( p \) above 2.5%, period to period variations of the treasury bill rate larger than 0.15, and deviations of M1 from a constant 5.1% growth path. Fixed policies (independent of the current state of the the economy), feedback decision rules that depend on the state, and policies derived by
optimizing the objective were compared. The "optimal" policies were computed by solving a new optimization problem with \( T = 8 \) each quarter, and using only the controls in the first period. In each of these problems, the values of future exogenous variables \( z_t \) and residues \( e_t \) were forecast values which had been available in the initial quarter, while initial conditions and past exogenous variables were set to actual historical values. Hence the information used was the information that would have been available to policymakers at that time. Results showed that, using the objective as a measure of effectiveness, this "feedback optimal" policy was not as effective as "actual" policy (which cautiously used feedback to improve things in the very short run) or fixed policies. Crane ascribes this mainly to large errors in the exogenous variable forecasts and (perhaps) to structural errors in the model, since the U.S. economy was experiencing rapid change and severe exogenous "shocks" during that period.

Athens (1976) used an 84-equation model to investigate the ability of "feedback optimal" policies (called sequential open loop optimal (SOLO) by Athens) to reduce end effects. A problem with \( T = 16 \) (1969.1-1973.1), 2 controls (tax rate and unborrowed reserves) and 2 target variables (inflation and unemployment rates) was posed. The full 16-period solution was compared to a sequence of 8-period SOLO solutions, in which only the first 2 period's control values were used. Exogenous variables were set at historical values, and initial conditions for each 8-period problem were model generated values. The objective was quadratic, and included cost of control terms. Comparison of the 16-period and SOLO optimal policies showed very close agreement between the inflation and unemployment paths, but substantially different paths for the control variables.

Shapiro (1976) summarizes the results of one of the more ambitious computational projects, a model comparisons effort sponsored by the National Bureau of Economic Research. Nine econometric models, all currently in active use and under constant development, were tested over the period 1971.1-1975.1. Target endogenous variables were unemployment and inflation rates, balance of trade, and GNP, while controls were the treasury bill and personal tax rates and government spending. Exogenous variables and residuals were set to their historical values. Each group of modelers devised policies in their own way; not all used formal optimization methods. Results showed significant differences among the models. Only two models yielded solutions with both lower average unemployment and inflation rates than historically occurred—all others had lower unemployment and higher inflation. The control paths had significantly higher variance than historical paths. Shapiro comments that, if the best current models yield solutions which are dominated by model differences, then this is even more likely to be true for the smaller models which are often used in optimal control experiments. Hence the
validity of such solutions for policy purposes is questionable. However, optimal control has great potential value in evaluating and discriminating between different models, and for gaining insight into their behavior.

The application of optimization methods to real policy decisions is still in its infancy. Two major reasons for this are: the highly political and sensitive nature of economic policy decisions, and the imperfect ability of current economic models to represent real economic behavior. In addition, Klein (1975) cites the problems inherent in encoding policymakers' preferences in a single objective, computational problems in solving OCP, and the need to include stochastic effects. He also discusses the tendency of economic agents to respond to control actions so as to nullify their effect, which implies a change in model structure as \( u \) varies. His conclusion is that optimal control can be useful in policy decisions, but only if used in an adaptive framework, where many OCP problems are solved, and judgmental input is used in determining each successive problem. This is the same framework required to make models useful in the simulation mode (Klein (1975)). Clearly, several years of research and experience are needed before optimization methodology begins to have real impact.

Recent applications of multiobjective methodology promise to alleviate problems of modeling policymakers' preferences. Wallenius (1978) applies two interactive multiobjective methods to a 13-equation linearized model of the Finnish economy with \( T = 1 \) year. High level government officials and members of the Confederation of Finnish industries participated in the exercises, and continuing use in the latter organization is reported. Bock (1977–78) rejects the usual goal function, and replaces OCP by a feasibility problem, involving the model equations (2) and a system of inequalities. These require that the "target" endogenous and control variables lie above or below certain satisfaction levels. If this problem is feasible, new more ambitious levels may be specified. If infeasible, Bock suggests that the inequalities be ordered in terms of relative importance, and that a sequence of feasibility problems be investigated, in which successively less important bounds are imposed. If any problem in this sequence is infeasible, an OCP is solved in which the amount of infeasibility in the constraint last imposed is minimized, subject to the previous constraints. The minimal infeasibility is used to adjust the bound value so that the new system is feasible, and the process continues. As mentioned earlier, Newton's method is used to solve each problem generated, with inequalities converted to equations by squared slack variables, i.e., \( x \leq u \) is transformed to \( x + s^2 = u \).

The "adaptive" use of optimization methods (including their use in a multiobjective framework) requires efficient, flexible, easy to use software. Recent work in Denmark (Drud (1977–78)), Germany (Bock (1977–78)), Canada (Bird (1976)), England (Rustem (1978)), France (Nepomi-
astchy (1978)), and the United States (Mantell (1977–78)) should, when complete, produce optimization software satisfying these criteria. However, such software must be imbedded in a larger system, analogous to the matrix generator-report writer systems used in linear programming. Such modeling systems facilitate the construction and manipulation of models, using natural language, interface easily with regression and optimization routines and data bases, and provide flexible output capabilities. Nepomiaschy (1978) proposes the construction of such a system in France, while Drud (1978a, 1978b) discusses the model-optimizer interface, including the automatic generation of derivatives through computer manipulation of symbolic expressions. In the United States, work by Bisschop and Meeraus (1977) at the World Bank is in progress to develop a General Algebraic Modeling System which will greatly facilitate the process of building and solving optimization models.

Other Economic Planning Models

In addition to the problems discussed above, there have been many other applications of nonlinear programming in economic planning. Some of these deal with planning problems in developing countries. An example is the class of “Yulgok” models described by Inman (1978). Model variables are (for each economic sector): production, fixed investment, exports, competitive and noncompetitive imports, and inventory accumulation. Constraints are: (a) production must meet or exceed demand, (b) production must not exceed sectoral capacity, (c) investment is limited by available funds, and (d) the balance of payments is limited. The objective is to maximize household consumption of outputs of all sectors. All variables represent values of the various economic aggregates in the terminal year of an 8-year planning horizon (1974–1981), under the assumption that fixed investment, production, and gross domestic product will grow at constant percentage rates over the horizon. The (unknown) growth rates are to be chosen by the optimization, and this leads to exponential nonlinearities in several subsets of the constraints.

Formulating the model in terms of terminal year quantities eliminates the “end-effect” problem discussed in the previous section. There is no tendency for this model to neglect investment in later years of the horizon, so no special restrictions need be placed on terminal capital stock.

With s economic sectors, the Yulgok models have 4s + 5 constraints (2s + 1 nonlinear) and 6s + 4 variables (2s + 2 appear nonlinearly). A 53-sector version is currently being used at the Korean Ministry of Energy and National Resources to investigate issues involving the intersectoral allocation of cumulative investment over the period 1974–1981 and the intersectoral patterns of foreign trade in 1981. In these applications, a
Nonlinear Programming Applications

recursive linear programming scheme is used to solve the model, but versions with up to 5 sectors have been solved using the generalized reduced gradient program GRG2 (Lasdon (1978b)) and a GRG code designed to handle large problems, MINOS/GRG (Lasdon (1978b)).

Another class of development planning models, PROLOG, is described in Norton (1978) and Inman (1977). These are price-endogenous, static, partial, equilibrium models (see Manne (1979)). A version with a single consumer, no investment, and no government sector (discussed in Norton (1978)) has the following constraints: (a) production of each commodity must meet or exceed demand, (b) resources used in production must not exceed given levels, (c) demand for each commodity is a function of all commodity prices and consumers real income, (d) real income definition, and (e) the price of each commodity must not exceed the value of the resources used in its production (both commodity and resource prices are endogenous). The demand functions (c) may be linear or nonlinear, and if y is real income, and x and p, are the quantity consumed and price of commodity i, respectively, (d) is \[ y = \sum p_i x_i / \sum \alpha_i p_i \] where the denominator is a price deflator term. The “budget shares” \( \alpha \) may be fixed or may be specified by the expression \[ \alpha_i = p_i x_i / \sum p_i x \] which introduces an additional nonlinearity. The objective (to be maximized) is the total value of goods consumed minus the value of resources used, and it is also nonlinear. Many versions of this simplified PROLOG model have been solved, using the GRG2 program mentioned earlier. Details of the experiments may be found in Norton (1978).

A more complex class of PROLOG models, intended for application to the Korean economy, is described by Inman (1977). These PROLOG models have seven economic sectors, three income groups, and two factors of production. They contain several additional sets of equations in addition to those described above, for example, investment and capital accumulation, savings behavior, and international trade. They contain some dynamic equations describing the accumulation of various quantities over time, such as capital stock, the labor force, and financial assets of various income groups, and are meant to be solved recursively over time. That is, in each year, a static optimization problem is solved, some of whose data is derived from the previous year’s solution and from forecasts of exogenous quantities. The optimal solution is meant to predict actual economic behavior in that year. In Inman (1977), some preliminary solutions for Korea are given over the period 1974–1976, using several alternative versions of the model equations. Piecewise linearization and a commercial LP code were used to solve each problem, but smaller nonlinear PROLOG versions have been solved in this recursive mode using GRG2. Direct solution of the nonlinear model has the advantages of simplicity and of no linearization error. Solution of larger more realistic versions using MINOS/GRG is contemplated in the future.
In solving both the PROLOG and YULGOK classes of models using GRG2, several software features have proven particularly useful. These features include: the use of finite difference approximations to derivatives (so the user need not code first derivatives of the problem functions); the ability to add, delete, restrict, or relax constraints easily by changing their type indicator in a specification file; and the ability to revise only a portion of a problem’s data and then solve the new problem, starting from the optimal solution to the previous one. Using these and other features, over 1,000 problems have been solved. These were of the PROLOG, YULGOK, and CHENERY classes. The CHENERY class is described by Lasdon (1978a), who also gives some typical results regarding problem sizes and iteration counts.

A significant class of nonlinear planning models can be formulated as quadratic programs, requiring the optimization of a quadratic function subject to linear constraints. The principles supporting these models were originated by Samuelson (1952), and are discussed by Takayama (1971) and Manne (1979). Applications include coal production and distribution (Dux (1977)), more general energy models (Glassey (1978), Cherniavsky (1974)), and agricultural planning (Heady (1975)).

Imbedded NLP’s and Planning Systems

There are many instances in which optimization problems are components of a larger model. An example is AGRIMOD (Campbell (1978) and Levis (1977)), a dynamic simulation model of the U.S. food production system over a 10–20-year horizon. It is intended for analysis of proposed agricultural and energy policies, as well as weather scenarios. Two modules within this model use nonlinear programming. In the farm input demand model, the input used to grow crops—land, fertilizer, and machinery utilization—are determined for all existing crop-region pairs (machinery utilization is measured in gallons of diesel fuel per acre per year). The objective function of this submodel is net expected profit summed over all crops and regions. In computing costs and revenues, the prices of input and crop output are those prices which farmers expect to prevail when crops are sold or when input is purchased (the expectation process is modeled endogenously). A nonlinear production function gives crop yield per acre in each region as a function of fertilizer and machinery utilization. The constraints form a transportation system, limiting total land use for each region and for each crop, plus upper and lower bounds on each variable. The problem has 129 variables and up to 18 constraints, and is solved using the generalized reduced gradient code GRG1 (Lasdon, 1978c).

This optimal allocation problem is used to form demand functions for the three nutrients that combine to yield fertilizer. The variables repre-
senting land and machinery allocation are fixed at their optimal values. Fertilizer demand for each crop-region pair can be determined as a function of fertilizer price by setting first derivatives of the objective with respect to the fertilizer allocation variables to zero. Summing demand functions over all crops and regions yields total fertilizer demand; applying fixed ratios yields total demand for each of the three nutrients, as a joint function of their prices. Using supply functions from other modules of the model, nutrient prices are determined by solving the 3-equation system equating nutrient supply and demand. This coupling of an optimization model and a supply-demand balance is found in several current planning models, e.g. the PIES national energy model (Federal Energy Administration (1976) and Manne (1979)).

AGRIMOD also solves a nonlinear program in its wholesale farm commodities market module. Here, the fact that soybeans, when crushed, yield two final products in fixed proportions (oil and meal) implies that the demand functions for these products must have the same ratio. The assumption that the soybean industry is perfectly competitive implies that net profit per pound of soybeans must equal an exogenously specified fixed value (no excess profits). Hence there are two equality constraints in this equilibrium calculation. The GRG code in Lasdon (1978c) permits rapid solution of this problem by defining the objective (to be minimized) as the sum over all commodities of the square of excess demand.

Nonlinear programming algorithms are also components of software systems used for constructing and solving planning models. Huckfeldt (1976) describes a planning system which has been used by several states in post-secondary education planning. It is a set of FORTRAN subroutines which allow a user to construct and either simulate or optimize a discrete-time dynamic planning model. In educational applications, the state variables of the model might be numbers of students or faculty, amounts of facilities available, or financial balances. Control variables might be student admissions, tuition, financial aid, or faculty hiring. The objective is usually a goal function (as in the econometric model optimization problems discussed previously), a weighted sum of squares. Each term in the sum is the deviation of a certain performance measure from a user-specified target. The measures may relate to educational quality (faculty/student ratio), financial stability (operating funds balance), or access to higher education (ratio of admissions to total student population). The optimization subroutine uses a bounded variable version of the Fletcher-Reeves conjugate gradient method.

For state level planning, institutions are usually grouped into sectors (2-year, 4-year, private, public, etc.). In attempting to estimate the impact of various state financing policies, for example, one might specify these policies by exogenous variables in a model of the sector, specify a goal function for the sector, and solve the dynamic optimization problem for
a set of institutional decisions over the planning horizon. Varying the exogenous variables and solving a sequence of optimization problems yields insight into how closely goals can be met under various policy alternatives. Varying the weights in the objective allows tradeoff studies among competing goals.

In the applications discussed by Huckfeldt (1977), Colorado and Maryland used the system in the simulation mode only, evaluating state financing plans and proposed tuition increases for out-of-state students. Analysts in the New York State Education Department used the optimization capability as well. Their model had five educational sectors, involving SUNY, CUNY, and private institutions. The model was built to investigate the relationships between state funding, student enrollments and costs, and institutional costs and revenues. Further details may be found in the New York State Department of Education (1977) report. An application to planning problems in the Boulder County Sheriff’s Department is described by Huckfeldt (1978). The system is now available through the National Policy Analysis Center, Boulder, Colorado.

4. CURRENT PROBLEMS AND FUTURE RESEARCH

Algorithms and Software

Although nonlinear optimization models have made important contributions in several areas, they have had much less overall impact than linear programming or mixed integer models. Until recently, a major factor limiting the use of NLP has been the lack of readily available high quality software. This remains a problem, but there has been considerable software development in the last 5 years. Several NLP solution codes in current use are mentioned in this paper, and further information is available in the recent survey by Waren (1979). Almost all available software is intended for small to medium size (less than 100 constraints or variables) problems, but there has been some recent work in developing codes for large sparse NLP’s. See Lasdon (1978b) and Rosen (1977) for further details.

Significant software improvements require the design of faster, more robust algorithms. For smaller problems, Augmented Lagrangian, Sequential Quadratic Programming and GRG methods seem to be the main contenders (Waren, 1979). New developments are occurring rapidly. More information on the relative performance of algorithms is needed; several comparative studies have already been performed (Waren (1979)) and more are in progress. In addition, some work has been done on algorithms for NLP’s with special structure, for example, the algorithms for optimal control and nonlinear network problems discussed earlier in this paper.

Some problem formulation languages currently under development
Nonlinear Programming Applications

promise to ease significantly the task of building, revising, and describing the solution of nonlinear optimization models. Such languages play a role analogous to that of the matrix generator/report writer systems which are widely used in linear programming. An ideal language will allow a user to state the problem in a language natural to him or her, it will have good data handling and reporting features, and will generate all required derivatives automatically. Some recent work on such languages has been described in this paper, at the end of the section on control of econometric models. Related work can be found in Fouser (1975) and Thames (1975).

Of course, many special-purpose problem-formulation languages currently exist for financial planning, chemical process design and analysis, simulation, electrical network analysis, etc. If optimization routines were imbedded in such languages, many new optimization applications would result. Daughety (1979) describes a simulation-NLP combination, while Roy (1980) describes the interfacing of a number of optimization codes with the financial planning system IFPS. The resulting system, IFPOS, is currently being tested by several organizations.

Modeling

Of course, having good solution tools available is only part of the problem. The other part concerns nonlinear models—they require considerable sophistication to build, validate, and maintain. Expertise in sufficient quantity is probably not available in many organizations, even those now doing OR work. Several major NLP projects in the 1960’s failed mainly because of this difficulty, and because of the uncertain payoffs to be obtained from using NLP. A related problem is that many managers—even many engineers—are not aware of the possible benefits and capabilities of NLP models, and do not recognize problem situations where NLP methods can help. All these problems are part of an “educational gap.” More “success stories”—articles on successful NLP applications—will help to fill this gap. NLP methodology has not been well represented in our major journal on practice, Interfaces, but perhaps this situation will improve. The reader is referred to Bodington (1973) and Mancini (1979) for a description of recent successful NLP applications in two major oil companies.

REFERENCES


Lasdon and Waren


E. M. L. Beale. 1974. A Conjugate Gradient Method of Approximation Program-
Nonlinear Programming Applications


Copyright © 2001 All Rights Reserved
Nonlinear Programming Applications


Lasdon and Waren

CONF-750867, NTIS. U.S. Dept. of Commerce.


Lasdon and Waren


Lasdon and Waren

43, College of Business Administration, University of Tennessee, Knoxville.
T. TAKAYAMA AND G. G. JUDGE. 1971. Spatial and Temporal Price and Allo-
Nonlinear Programming Applications


Copyright © 2001 All Rights Reserved