223

7th MATHEMATICAL PROGRAMMING SYMPOSIUM 1970

THE HAGUE - THE NETHERLANDS - SEPTEMBER 14 - 18

ABSTRACTS

N·H
P·C

1970

NORTH-HOLLAND PUBLISHING COMPANY — AMSTERDAM

S.N. AFRIAT, University of North Carolina, Chapel Hill.

The Progressive Support Method for Convex Programming*

A solution to a convex problem P, to find a maximum $\bar{x}$ of $f(x)$ subject to $g(x) \leq 0$ where $f(x)$, $g(x)$ are functions which are concave, convex respectively, is approached by solutions of a series of polyhedral convex problems $P_m = P(x_1, \ldots, x_m)$, where x_m is the solution of $P_{m-1} = P(x_1, \ldots, x_{m-1})$. The polyhedral problems, since they are expressible as standard linear problems, can be treated by the simplex algorithm. The problem P_m is to find a maximum x_{m+1} of $f_m(x)$ subject to $g_m(x) \leq 0$, where

$$f_m(x) = \min[\sigma_1(x), \ldots, \sigma_m(x)],$$

$$g_m(x) = \max[\rho_1(x), \ldots, \rho_m(x)],$$

and $\sigma_r(x)$, $\rho_r(x)$ are linear supports of $f(x)$, $g(x)$ at x_r. The function $f_m(x)$ defines a polyhedral concave function circumscribed to the concave function $f(x)$ at the points $x_1, \ldots, x_m$. Similarly, $g_m(x)$ is a polyhedral convex function circumscribed to $g(x)$ at the points. Then P_m is a polyhedral convex problem and defines a support for P based on points $x_1, \ldots, x_m$.

This process, with a different formulation but no essential difference, is exhibited by the Algorithm IV of Cheney and Goldstein [1]. They cite earlier appearances of the basic idea, including the cutting plane method of Kelley [2]. With

$$\varphi(x,t) = \max[t - f(x), g(x)],$$

the problem P can be put in a standard form:

$$\max[t : \varphi(x,t) \leq 0].$$

The presently considered support method, as it applies to the problem put in this standard form, gives the same process as Kelly's method. There is a diversity of ways in which the cutting plane

* SIAM J. Numer. Anal. Vol.7, No. 3, September 1970.

principle can be elaborated, and relative merit must depend upon peculiarities of the problem dealt with. But a separate advantage of the present development is that, in an extended version, to be considered later, it can simultaneously determine dual solutions for problems with several constraints. Discussion is also provided for questions such as concern the starting procedure and the decision of boundedness and consistency. Of practical importance is a condensation of the process which is pointed out, by which constraints are retired as new ones are introduced, so the successively treated linear problems remain strictly limited in size. Also remarked is the advantageous application of the process to linear problems which have a large number of constraints.

[1] E.W. CHENEY and A.A. GOLDSTEIN, Newton's method for convex programming and Tchebycheff approximation, Numer. Math., 1(1959), pp. 253-268.

[2] J.E. KELLEY, The cutting plane method for solving convex programs, SIAM J. Appl. Math., 8(1960), pp. 703-712.

LEIF APPELGREN, Salén Shipping Companies, Stockholm.

Computational Experience of a Vessel Scheduling Algorithm using Column Generation and Branch-and-Bound Techniques.

The column generation part of this algorithm is described in Transp. Science 3, No. 1, 1969. As this LP algorithm does not guarantee integer solutions, a branch-and-bound algorithm has been added.

The problem concerns the assignment of a set of vessels to a set of cargoes. It is assumed that two cargoes are never carried simultaneously by the same vessel. The cargoes are characterized by size, type, loading and discharging ports and dates. The vessels have different size, type, speed and initial position.

The column generation algorithm iterates between a master LP routine and a subprogram using dynamic programming. In the dynamic programming routine, an optimal path is generated for each vessel using cargo values obtained from the dual variables of the master program. The master program combines the previously generated vessel paths into an optimum schedule under the constraints that each vessel follows exactly one path and that each cargo is carried exactly once.

If the planning period is so short that each vessel can carry one cargo during the period, the LP problem is an ordinary assignment problem which always has an optimal integer solution. This is not the case, however, when the vessels can carry sequences of two or more cargoes during the period. In the branch-and-bound algorithm, one fractional variable is selected from the previous solution. This variable corresponds to a specific cargo sequence C_1, C_2, ..., for a specific vessel V. One of these cargoes, say C_1, is selected, and a branching is made into the two alternatives that Vessel V must/must not carry cargo C_1. The column generation algorithm is then applied on these alternatives, which are controlled by variation of the value of cargo C_1 for vessel V in the dynamic programming routine. If even the new solutions are fractional, a new vessel/cargo combination is selected and the process is repeated.

The algorithm is presently run on a CD 6600 computer where the running time until the first LP optimum is approximately five minutes for a six week problem with about 110 vessels and 120 cargoes. The branch-and-bound algorithm has to be used when the first LP optimum is fractional, which it is in about 50 % of the cases for this problem size. In these cases between two and ten constrained problems have to be solved, but the running time for these problems is only about 10 % of the time for the initial solution since they use the previously generated LP matrix and the old solution as a starting basis.

MORDECAI AVRIEL, Technion, Haifa.

Nonlinear Programming with r-convex Functions.

In recent years several extensions and generalizations of convex functions have appeared in the mathematical programming literature. The purpose of this paper is to present a unified theory of nonlinear programs involving a large class of functions including convex functions and most of the known extensions.*

Denote by $M_r(a^1, a^2; q)$ the weighted rth mean of two nonnegative numbers a^1 and a^2. Then we have the well known formula;

$$M_r(a^1, a^2; q) = [q_1(a^1)^r + q_2(a^2)^r]^{1/r}$$

$$q_1 \geq 0, q_2 \geq 0, q_1 + q_2 = 1$$

Let f be a real valued nonnegative function defined on a convex subset C of R^n. We call f r-convex if given $x^1 \varepsilon C$, $x^2 \varepsilon C$

$$f(q_1 x^1 + q_2 x^2) \leq M_r(f(x^1), f(x^2); q)$$

for any two numbers $q_1 \geq 0$, $q_2 \geq 0$ such that $q_1 + q_2 = 1$.
For $r = 1$ the above relation reduces to the definition of ordinary nonnegative convex functions. The well known relation

$$M_r(a^1, a^2; q) \leq M_s(a^1, a^2; q)$$

for $s > r$ determines a ranking of r-convex functions culminating in quasiconvex functions for $r \to + \infty$.

In this paper we first present some algebraic and geometric properties of r-convex functions, then discuss necessary and sufficient conditions for optimality in nonlinear programs with r-convex functions. Finally, relations between r-convex functions and other known extensions of convexity are developed.

* e.g. pseudoconvex functions.

M.L. BALINSKI, City University of New York and IBM France, Neuilly-sur-Seine.

A Totally Feasible Method for Linear Programming.

Consider the pair of dual linear programs (in "canonical form"):

$$(P) : \max\{cy;\ Ay \leq b,\ y \geq 0\} \text{ and}$$

$$(D) : \min\{xb;\ xA \geq c,\ x \geq 0\}.$$

Three conditions are necessary and sufficient for y^* and x^* to be optimal solutions: (i) y^* feasible ("primal feasible"), (ii) x^* feasible ("dual feasible"), and (iii) $x^*(b - Ay^*) + (x^*A - c)y^* = 0$, i.e., each term of the summation has at least of its factors equal to zero ("complementary orthogonality").

The simplex method [4] begins with a feasible y-solution, a paired x-solution satisfying the complementary orthogonality property, and maintains these conditions while working to obtain a feasible x-solution. The dual simplex method [1] begins with a feasible x-solution, a paired y-solution satisfying the complementary property, and maintains these conditions while working to obtain a feasible y-solution. The method of this paper fills the evident gap: it begins with feasible x- and y-solutions and maintains these conditions while working to obtain complementary orthogonality. As such, it is related to recent work on "complementary pivot theory" (e.g., [3], [6]) although the method itself was inspired by the existence of three similarly related methods for transportation problems (e.g., a primal method [1], a dual or Hungarian method [5], a totally feasible method [2]).

The interest of this approach is largely computational: in instances where good initial guesses at feasible solutions for both programs can be made - such as is the case for a pair of mixed strategies for matrix games or for industrial problems which are resolved periodically with only slightly changed parameters - it would be reasonable to expect that a method improving such good starts would be preferable to methods which "forget" half of this information. An experimental study is under way to evaluate this "expectation" and will be reported upon.

[1] Balinski, M.L. and R.E. Gomory, "A Primal Method for the Assignment and Transportation Problems", Management Science, Vol. 10 (1964), pp. 578-593.

[2] Briggs, F.E.A., "A Dual Labelling Method for the Transportation Problem", Operations Research, Vol. 10 (1962), pp. 506-517.

[3] Cottle, Richard W. and George B. Dantzig, "Complementary Pivot Theory of Mathematical Programming", Mathematics of the Decision Sciences, Vol 1, American Mathematical Society, 1968, pp. 115-136.

[4] Dantzig, G.B., "Minimization of a linear Function of Variables Subject to Linear Inequalities", in Activity Analysis of Production and Allocation, John Wiley & Sons, 1951, pp. 339-347.

[5] Kuhn, H.W., "The Hungarian Method for the Assignment Problem", Naval Research Logistics Quarterly, Vol. 3 (1956), pp. 83-97.

[6] Lemke, C.E., "Bimatrix Equilibrium Points and Mathematical Programming", Management Science, Vol. 11 (1965), pp. 681-689.

[7] Lemke, C.E., "The Dual Method of Solving the Linear Programming Problem", Naval Research Logistics Quarterly, Vol. 1 (1954), pp. 36-47.

JOSEPH L. BALINTFY, University of Massachusetts, Amherst.

Mathematical Programming Models and Techniques for Optimum Human Diets.

Acceptable and economical human diets can be planned by mathematical programming approaches and computers. The new technique offers significant cost saving advantages with nutritionally controlled diets especially for volume feeding systems and institutions. The problem is solved by constructing mathematical models for menu planning. The variables in these models are the Cartesian products of the menu items and the days involved in a planning period. The constraints are derived from dietary, palatability (variety) and structural considerations. The formulation leads to large scale and presently unsolvable mixed integer programming problems with some stochastic elements. A special multiple choice programming algorithm was developed and applied to find near-optimum menus by a fast multi-stage solution process. An alternative linear programming model with bounded variables and heuristic scheduling was also found to be applicable in certain cases. The rapid acceptance and application of these models initiated a new computer technology for the food service industry.

E.M.L. BEALE and J.P. VIT, Scientific Control Systems Limited, London.

Selecting subsets by Dynamic Programming.

In spite of recent improvements in general mixed integer algorithms (see for example Tomlin (1970)), other procedures for finding optima to special classes of non-convex problems remain important. One such class is the selection of an optimal subset from a finite set of elements. An enumerative tree search method for such problems is described by Beale, Kendall and Mann (1967) and in more general terms by Beale (1970).

Gortsko (1966) discusses a dynamic programming approach to the problem of depot location in one dimension. At first sight this work, while interesting, seems outside the mainstream of mathematical programming. But this is not really the case. The problem reduces mathematically to the subdivision of a line into non-overlapping intervalls so as to minimize an objective function consisting of a sum of functions of the selected intervalls. (These functions represent the cost of servicing that particular section of the line from a single depot located in the best position for this task). For numerical dynamic programming work we must select a finite number N of possible break-points between successive intervals and we must select an optimum subset out of them. Dynamic programming solves this problem very easily, because the objective function C to be minimized is of the form

$$C = \sum_{k=1}^{r+1} a_{i_{k-1} i_k} \tag{1}$$

where $i_0 = 0$, $i_1, \ldots i_r$ denote the selected points in ascending order, $i_{r+1} = N+1$, and a_{jk} denotes the cost associated with the single interval from the $j^{\underline{th}}$ to the $k^{\underline{th}}$ point.

Problems with this precise cost structure may be rare. But in many problems of selecting a subset from some ordered set one can define quantities a_{jk} such that

$$C \geq C_0 + \sum_{k=1}^{r+1} a_{i_{k-1} i_k} \tag{2}$$

for any selection. The quantity C_0 then represents the cost when all points are selected, and a_{jk} represents the net increase over this cost when all points between the j^{th} and the k^{th} are omitted.

When (2) holds, dynamic programming can be used to provide a lower bound on the possible cost and a suggested selection. This analysis can be refined by using a tree-search algorithm, continuing to a guaranteed optimum if desired. The branches of the tree represent preselecting a single point to be either in or out of the final solution. With any such preselections a new set of a_{jk} can be computed such that (2) provides a better bound on the cost of other selections. Dynamic programming can therefore be used to evaluate each node of the tree, and to derive bounds on unexplored nodes.

[1] E.M.L. Beale (1970) "Selecting an Optimum Subset" in Integer and Nonlinear Programming. Edited by J. Abadie (North Holland, Amsterdam).

[2] E.M.L. Beale, M.G. Kendall and D.W. Mann (1967). "The discarding of variables in multivariate analysis". Biometrika 54 pp. 357-366.

[3] A. Gortsko (1966) Mathematical Models and Optimal Planning, Nauka Novosibirsk

[4] J. Tomlin (1970) "Branch-and-Bound Methods for Integer and Non-Convex Programming", in Integer and Nonlinear Programming. Edited by J. Abadie (North Holland, Amsterdam).

MARTIN J. BECKMANN, Technische Hochschule, München, and Brown University, Providence, and
KAILASH C. KAPUR, General Motors Research Laboratories, Warren, Mich.

On Duality and Conjugacy in Non-linear Programming and some Applications.

The purpose of this paper is to prove and apply a duality theorem which is of considerable interest in transportation planning and utility analysis.

Duality Theorem

Let f(x) be concave and let the set (1) $S = [x | x = 0, Ax \leqq c]$ be bounded. Then

$$(2) \qquad \sup_{x \in S} f(x) = f(\hat{x}) = \inf_{u \geqq 0} [u'c - \emptyset(u)]$$

where $\emptyset(u)$ is a generalized conjugate function of f(x)

$$(3) \qquad \emptyset(u) = \inf_{x \in S} [u'Ax - f(x)].$$

This duality theorem differs from the conventional ones by using the conjugate function concept [Fenchel, Rockafellar]. Of course, this function is known explicitly only in special cases. Several will be discussed here: The quadratic, the separable, and the Cobb-Douglas cases.

For a negative definite quadratic objective function $f(x) = 1/2\ x'Qx + b'x$ with linear constraints $Ax \leqq c$ the conjugate function turns out to be

$$\tilde{\emptyset}(u) = 1/2\ (u'A - b')\ Q^{-1}\ (Au - b)$$

and the dual problem is

$$\inf_{u \geqq 0} [u'c - \tilde{\emptyset}(u)] .$$

For a separable differentiable objective function

$$\sum_k \int_0^{x_k} p_k(x)dx$$

and constraints $Ax \leqq c$ the conjugate function is

$$\emptyset^*(u) = \sum_k \int_0^{\sum_i u_i a_{ik}} g_k(p)dp$$

where the functions $g_k(p)$ are the inverse functions to $p_k(x)$. The conjugate function $\emptyset^*$ is an alternative representation of a consumer surplus. This has applications to transportation systems planning.

Application of quadratic, separable and Cobb-Douglas maximization functions are given to utility maximization by households subject to budjet constraint and to (unconstrained) profit maximization by firms.

MANDELL BELLMORE, The Johns Hopkins University, Baltimore, and

H. DONALD RATCLIFF, The University of Florida.

Set Covering and Involutory Bases.

Some new properties associated with the special class of integer programs known as weighted set covering problems are derived. While it is well known that an optimal integer solution to the set covering problem is a basic feasible solution to the corresponding linear program, we show that there exists an optimal basis which is involutory (i.e., $B = B^{-1}$).

This property and others are used to develop a new algorithm which uses strong cutting planes. The cutting planes are strong in the sense that they exclude both integer and non integer solutions. Computational experience is presented.

R. BENAYOUN, J. de MONTGOLFIER, J. TERGNY, SEMA (Metra International), Paris, and
O. LARITCHEV, Profsojusnaja 81, Moscow.

Linear Programming with Multiple Objective Functions

This paper describes a solution technique for Linear Programming problems with multiple objective functions. In this type of problem it is often necessary to replace the concept of "optimum" with that of "best compromise". In contrast with method dealing with a priori weighted sums of the objective functions, the method described here involves a sequential exploration of solutions. This exploration is guided to some extent by the decision maker who intervenes by means of defined responses to precise questions posed by the algorithm. Thus, in this manmodel symbiosis, phases of computation alternate with phases of decision. The process allows the decision maker to "learn" to recognize good solutions and the relative importance of the objectives. The final decision (best compromise) furnished by the manmodel system is obtained after a minimum of successive optimisations.

Three major classes of problems are envisaged: those where the relative importance of the objective functions is quantified, where it is known but not quantifiable, and where it is completely unknown. In every case, the intervening phases of computation, guiding the exploration of solution space, are easily programmed on the computer. The method is illustrated by an application in Manpower Management. The model is a LP with four different linear objective-functions:

1. Employee satisfaction;
2. Employee efficiency;
3. Costs (salary, recruiting, training,...);
4. "Slack" between programme allocations and forecast staff requirements.

A practical numeric example is examined, with the help of a standard Linear Programming code.

M. BENICHOU, J.M. GAUTHIER, P. GIRODET, G. HENTGES, G. RIBIERE, O. VINCENT, IBM France, Paris.

Mixed Integer Linear Programming.

This paper first presents a branch-and-bound type method for solving mixed integer linear programming problems; new rules for branching and bounding are used.

This method has been implemented in a code, which appears as a module of the IBM Mathematical Programming System/360. The implementation presents original features to give full efficiency to the branch-and-bound search. Numerous parameters in the code enable the user not only to produce and prove optimal integer solutions but also to explore the set of integer solutions.

At last, extensive numerical results and comparisons between various rules of branching and bounding are presented.

Several models of real problems have already been solved. See attached first results where:

Problem 1, 2, 3, 4 are investment models
Problem 5, is the model of a bank
Problem 6, is a production planning model.

IBM 360 MODEL	PROOF OF OPTIMALITY			OPTIMAL INTEGER SOLUTION				CONTINUOUS OPTIMUM		PROBLEM CHARACTERISTICS				PROBLEM N°
	ITERATION N°	NODE N°	TIME (C.P.U.) (minutes)	FUNCTIONAL VALUE	ITERATION N°	NODE N°	TIME (C.P.U.) (minutes)	NUMBER OF NON INTEGRAL INT. VAR.	FUNCTIONAL VALUE	NUMBER OF INT. VAR.	NUMBER OF NON ZERO ELEMENTS	NUMBER OF INT. AND CONT. VAR.	NUMBER OF ROWS	
50	486	60	10.3	323 380	345	29	6.34	4	337 095	39	2941	172	137	1
75	180	86	0.11	119 439	124	70	0.07	6	126 476	25	329	44	29	2
75	176	57	0.12	1492.9	115	44	0.07	7	1558.3	14	394	34	38	3
75	649	95	2.70	288 588	304	35	1.15	18	331 516	25	1592	397	368	4
75	482	51	1.20	752 831	444	42	1.08	6	811 995	22	1920	340	405	5
75	1322	222	28.8	795 377	732	134	15.8	16	790 747	39	20028	1156	721	6

BERNARD BEREANU, Academy of the Socialist Republic of Romania, Bucharest.

Numerical Methods in Stochastic Linear Programming.

The problem of determining the probability distribution and (or) the moments of the optimum of a linear program with random coefficients with respect to an a priori distribution (distribution problem) is solved in principle [1]; but no efficient numerical method was published except when such coefficients are affine functions of a single random variable [2].

Here such a method is proposed which is operational if the number of random variables on which the coefficients (which may all be random) depend is not too large. However, the limitations on the dimensions of the program itself are only those imposed by existing computers on ordinary linear programs. The method is also applicable to nonstationary stochastic programming (the distribution of the coefficients change with time) and it is shown that the updating of this distribution involves relatively little additional running time.

The relation between the distribution problem and the two-stage programming (recourse problem) [3], [4], computer implementation of the algorithm (stationary and nonstationary case), running time, error estimation and efficiency are discussed. Sample problems are given and some computational experience reported.

The method is based on the following. Let

(1) $$\gamma(\xi) = \max_x c(\xi)x \text{ subject to } A(\xi)x \leqq b(\xi),$$

$x \geqq 0$, where ξ is an absolutely continuous random r-vector with sample space $[a_k, b_k]$, $k \in \overline{1,r}$ and the components of $A(\xi)$, $b(\xi)$, $c(\xi)$ are affine functions of ξ (perhaps projections). Suppose that (1) defines a random variable, e.g., we have a positive stochastic program [1].

In the nonstationary case ξ is replaced by a multidimensional stochastic process with discrete parameter $\{\xi_t, t \in T\}$.

Here after the indices i,j,...,p take the integer values $\overline{1,n}$ and k takes the values $\overline{1,s}$, with $s = r-1$, and these limits will not be indicated.

Let A_i^n , h_i^n be the coefficients and nodes of the standard Gaussian quadrature formula with n nodes, $x_{ki}^n = 1/2\ (b_k - a_k)h_i^n + 1/2\ (a_k+b_k)$ and $D = 2^{-s} \prod_k (b_k-a_k)$. Put

$$F_n(z) = D \sum_i \cdots \sum_p A_i^n \cdots A_p^n\, F_\gamma(z|x_{1i}^n,\ldots,x_{sp}^n) f(x_{1i}^n,\ldots,x_{sp}^n),$$

$$M_n^\ell = D \sum_i \cdots \sum_p A_i^n \cdots A_p^n\, M^\ell(\gamma|x_{1i}^n,\ldots,x_{sp}^n) f(x_{1i}^n,\ldots,x_{sp}^n),$$

where f(.) is the marginal density of the vector $(\xi_1,\ldots,\xi_s)$ and

$$F_\gamma(z|x_{1i}^n,\ldots,x_{sp}^n),\ M^\ell(\gamma|x_{1i}^n,\ldots,x_{sp}^n)$$

are the conditional probability distribution function and the moments of order ℓ of $\gamma(\xi)$; then they can be computed for $\ell = 1,2$ with the computer program STOPRO of [2].

We have:

1. $F_n(z) \overset{u}{\to} F_\gamma(z)$ and $M_n^\ell \to M^\ell(\gamma)$ when $n \to \infty$ and $F_\gamma(\xi)$ and $M^\ell(\gamma)$ are the probability distribution function and the moments of order ℓ of $\gamma(\xi)$ (when such a moment exists).
2. Analogous relations hold for nonstationary programming and the inputs to STOPRO do not depend on t.
3. If, for each t, the rth component of ξ_t is independent of the remainder of the components and identically distributed, then $F_{\gamma_t}(z|x_{1i},\ldots,x_{sp})$ and $M^\ell(\gamma_t|x_{1i}^n,\ldots,x_{sp}^n)$ do not depend on t.

[1] B. Bereanu, Z.Wahrscheinlichkeitstheorie und Verw. Gebiete, 8, 148-152, 1967.

[2] B. Bereanu and G. Peeters, D.P. 6815, Center for Oper. Res. and Econometrics, Univ. of Louvain, 1968.

[3] G.B Dantzig, Management Sci., 1, 3-4, 197-206, 1955.

[4] D.W. Walkup and R.J. Wets, SIAM J. Appl. Math., 15, 1299-1314, 1967.

HANS BERGENDORFF, The Royal Institute of Technology, Stockholm.

An error estimate for discrete approximations of continuous state space dynamic programming.

The presented result is part of a work concerning controlled Markov-processes with incomplete state information. The model used is a finite state infinite time, discounted, controlled Markov process where the states are grouped into classes, superstates. The information that is available to the decisionmaker is the superstate plus a probability distribution over the hidden substates in the superstate. An important special case is optimal control with no information, viz. a process with only one superstate. Optimal control in that case can be found by solving a dynamic programming problem with a continuous state space (of probability distributions). If that problem is solved with dynamic programming on a discrete state space (where the discrete state space represents an ε-grid in the continuous state space), the error depends on the maximal variation of the partial derivatives of the true value function in the interval between the discrete points. The results show this dependence and give upper and lower bounds for the partial derivatives. Upper bounds for the partial derivatives are given by the state values of the corresponding optimally controlled Markov chain with perfect state information. If these upper bounds are given by $T(\bar{r})$ where $\bar{r}$ is the payoutvector in each stage, the lower bounds are given by $-T(-\bar{r})$.

ABRAHAM BERMAN, Northwestern University, Evanston, and Université de Montreal, Montreal, and
ADI BEN-ISRAEL, Technion, Haifa.

Linear Inequalities, Mathematical Programming, and Matrix Theory.

Solvability theorems for linear inequalities over cones and cones with interior are developed and applied to complex mathematical and to matrix theory.

UMBERTO BERTELE' and FRANCESCO BRIOSCHI, Politecnico di Milano, Milan.

New Developments in the Theory of Nonserial Dynamic Programming.

Consider the following optimization problem

$$\min_{X} F(X) = \min \sum_{i \in I} f_i(X^i)$$

where $X = \{x_1, x_2, \ldots, x_M\}$ is a set of discrete variables,

$$I = \{1, 2, \ldots, n\} \text{ and } X^i \subset X.$$

Each component $f_i(X^i)$ of the cost function $F(X)$ is specified by means of a stored table with $|X^i| + 1$ columns and $\sigma^{|X^i|}$ rows (it has been assumed that all the variables have the same range).
One ordered partition among all the possible ones of the variables of the set X is selected. For this partition the given optimization problem (the primary optimization problem) may be solved by dynamic programming.
Since an optimal assignment for X can can be obtained by all ordered partitions of X, it is clear that another optimization problem, the secondary optimization problem, arises. It consists in determining that partition that is the best from the point of view of the minimization of the computing time and of the storage requirements.

The solution of the secondary optimization problem is based on graph theoretical considerations.
The previous works on non-serial dynamic programming [1,2,3,4] are essentially concerned with finding a solution to the secondary optimization problem in the special case when each block of the ordered partitions consists of a single variable.
The consideration of this special class of decompositions is sufficient for minimizing a reasonable index of the computing time needed for the solution of the primary optimization problem.
However, for tackling the memory limitations it is essential to deal with a more general class of decompositions than the one considered here.
In this paper many interesting new results are given and their computational relevance is pointed out.

[1] F. BRIOSCHI and S. EVEN, Minimizing the Number of Operations in Certain Discrete Variable Optimization Problems, to appear in Operations Research. Technical Report 567, Division of Engineering and Applied Physics, Harvard University, Cambridge, Mass, August 1968.

[2] U. BERTELE' and F. BRIOSCHI, A New Algorithm for the solution of the Secondary Optimization Problem in Nonserial Dynamic Programming, Journal of Mathematical Analysis and Applications, Vol 27, No. 3, Sept. 1969, pp. 565-574.

[3] U. BERTELE' and F. BRIOSCHI, Contribution to Nonserial Dynamic Programming, Journal of Mathematical Analysis and Applications, Vol. 28, No. 2, Nov. 1969, pp. 313-325.

[4] U. BERTELE' and F. BRIOSCHI, A Theorem in Nonserial Dynamic Programming, Journal of Mathematical Analysis and Applications, Vol. 29, No. 2, Feb. 1970, pp. 351-353.

OLE BILDE, The Technical University of Denmark, Lyngby.

On the solution of some road design problems.

In recent years efforts have been made in order to develop methods and computer programs which can assist road design engineers in their work. This paper contributes to these efforts with some methods of mathematical programming applied to problems related to the determination of vertical road alignment.

Consider a road section with a prescribed horizontal alignment. The data of the terrain are known and the problem is to determine the vertical profile subject to the associated costs, the level of traffic security, and the aestetic values.

Based on some simplified assumptions on technical design, the method to be presented here tries to find a vertical profile which minimizes costs and meet certain demands to traffic security and aestetic values.

The solution method is iterative and each step consists of two stages. The first stage assumes a feasible vertical profile to be known, and a transportation problem which minimizes the costs of earthmoving is solved. In the second stage the shadow prices (dual variables) from the solution of the transportation problem and other costs related to the vertical alignment are used in seeking a new feasible profile with smaller costs. The procedure terminates when no improvement to the current solution can be found.

The earth to be transported from one station i to another station j on the road section must be shipped along the road. In order to make the solution applicable one has to ensure that all the earthwork on the part of the road section between the two stations is completed before the earth is transported from i to j.

In the mathematical formulation of the transportation problem this results in some non-linear restrictions. A special Branch-and-Bound method for the last problem has been developed. Using the set of non-linear constraints to cut down the number of arcs in the transportation network, the algorithm has proved to be very efficient for various problems with "real life" data.

LOUIS J. BILLERA, Cornell University, Ithaca, N.Y.

Some Recent Results in n-Person Game Theory.

The discussion will center mainly on some work on two solution concepts: the core for games without side payments and the nucleolus for games with side payments (characteristic function games). The core has become an important equilibrium concept in Mathematical Economics. The nucleolus is related to the theory of bargaining sets.

The core is the set of payoffs which cannot be blocked by any coalition of the players. Various sufficient conditions for a game without side payments to have a nonempty core have been derived. If the payoff sets are assumed to be convex, then one can charaterize the games with nonempty core in terms of the support functions of the payoff sets. This characterization generalizes the result for side payment games, which follows directly from linear programming duality.

The nucleolus is defined for games given by a characteristic function v. For each payoff x, arrange the numbers $v(S) - \sum_{i \varepsilon S} x_i$, for all coalitions S, in decreasing order. The resulting vector in R^{2^n} is denoted $\theta(x)$. The nucleolus is defined to be those payoffs x for which $\theta(x)$ is minimal in the lexicographical ordering of R^{2^n}. The nucleolus is always non-empty, is contained in the bargain set, and belongs to every non-empty ε-core. The nucleolus has recently been characterized in terms of balanced collections of coalitions (which arise in the study of the core). This characterization provides simple proofs of the uniqueness and continuity (as a function of v) of the nucleolus.

EMANUELE BIONDI, PIERCARLO PALERMO, CARMELO PLUCHINOTTA, Politecnico di Milano, Milan.

A heuristic Method for a Delivery Problem.

This paper deals with a delivery problem which has been treated widely in the literature.

There is a central warehouse in which there is stored a certain commodity that is to be distributed by common carriers to a number of customers at various destinations within some region.

In any given time period the customers' demands have to be satisfied in such a way to minimize the total transportation cost. The demand of a given customer must be satisfied in one delivery only. The shipper specifies the delivery schedule to be followed by the carriers taking into account the capacity constraints of the carriers and the customers' demands. For sake of simplicity, given the capacities of the carriers and the average demands of the customers, it is supposed that any carrier delivers the orders of k customers. It is supposed also that the transportation cost on a route depends only on the time ruiquired to travel along that route.

This problem has been treated in the literature as a set covering problem and exact algorithms have been developed for its solution. The drawback of that approach is the huge number of variables to be considered even for problems of moderate size. This justifies the search of different approaches and of heuristic methods of solution, as done in this paper.

The algorithm is based on the partition of the overall problem into the sub-problem of satisfying the demand of each customer; for each of them it is determined a set of customers which are likely to be supplied in the same delivery, in a good solution of the problem.

It is possible to give an upper bound for the difference of the values of the objective function corresponding to the "good" solution found by the algorithm and the optimal one. The algorithm has been tested on some practical problems with satisfactory results.

P. BOD, Hungarian Academy of Sciences, Budapest.

On an extremum problem concerning graphs (the generalized minimum length tree problem).

The following problem has been brought up in the "Problem Section" of the "Colloquium in Combinatorics" organised by the Bolyai János Mathematical Society in Balatonfüred last autumn.

Given the finite graph: $G = (X,U)$ with non negative lengths on its edges: $\ell(u) \geqq 0, \forall u \in U$, and a subset $X_1 \subset X$ of the vertex set.

It is to find such a $G_0 = (X_0,U_0)$ connected subgraph of G for which

$$X_1 \subset X_0 \subset X; \; U_0 \subset U$$

and

$$\sum_{u \in U_0} \ell(u) \to \min!$$

It is obvious that the problem involves: i) the minimum length spanned tree's one if

$$X_1 = X$$

and ii) the problem of the minimum length if

$$X_1 = \{x,y\}; \; x \neq y; \; x \in X; \; y \in X.$$

We are showing that G_0 always exists if X_1 lies in one connected component of G, and is necessarily a tree if

$$\ell(u) > 0, \forall u \in U.$$

A complete description type algorithm will be sketched which uses as subroutine an algebraic representation of the well-known algorithm due to Kruskal.

As the correct algorithm may necessitate an excessive number of iterations, an approximating algorithm will also be proposed.

GHEORGHE BOLDUR, Academy of the Socialist Republic of Romania, Bucharest.

Linear Programming Problems with complex Decision Conditions.

Consider an n variables and m relations system:

$$(1) \qquad \sum_{j=1}^{n} a_{ij}x_j = b_i, \qquad i=1,2,\ldots,m, \; j=1,2,\ldots,n, \; x_j \geq 0,$$

and:

- r objective functions with distinct signification by no means reducible to a common matter;
- s states of nature;
- t decision-makers with possibly different opinions.

If we simultaneously take into account all the objectives considering the opinions of the t decision-makers and all the states of nature, we get r.s.t objective complex functions:

$$(2) \qquad \text{opt } F_{hk\ell} = \text{opt} \sum_{j=1}^{n} c_{hk\ell j}x_j, \quad h=1,2,\ldots,r, \; k=1,2,\ldots,s, \; \ell=1,2,\ldots,t.$$

The system (1) and the functions (2) define a linear programming problem in complex decision conditions. To solve it we avail ourselves of the subjective utility, of the methods of solving the games against nature and of some elements in the group-decision theory.

First we sum the opinions of the decision-makers, thus obtaining r.s objective functions:

$$(3) \qquad \text{opt } F_{hk}^{G} = \text{opt} \sum_{\ell=1}^{t} \sum_{j=1}^{n} g_\ell \cdot c_{hk\ell j}x_j, \quad h=1,2,\ldots,r, \; k=1,2,\ldots,s.$$

g_ℓ are "competence coefficients" associated to each decision-maker.

Secondly we solve r.s linear programming problems determined by the system (1) and the functions (3); so we obtain r.s optimal values of the objective functions (3): $X_1, X_2, \ldots, X_{r.s}$. We also solve r.s linear programming problems determined by the system (1) and the functions:

$$(4) \qquad \text{pes } F^G_{hk} = - \text{opt } (-F^G_{hk})^{*)}$$

obtaining in this way r.s pessimal values of the efficiency functions $Y_1, Y_2, \ldots, Y_{r.s}$.

Using the Von Neumann-Morgenstern-method (or any other method), we estimate the utilities of the optimal and pessimal values found before:

Values of the objective function	X_1	X_2	$X_{r.s}$	Y_1	Y_2	$Y_{r.s}$
Utilities	$u(X_1)$	$u(X_2)\ldots$	$u(X_{r.s})$	$u(Y_1)$	$u(Y_2)\ldots$	$u(Y_{r.s})$

and we make the linear transformations:

$$(5) \qquad \begin{aligned} \alpha_{hk} X_{hk} + \beta_{hk} &= u(X_{hk}) \\ \alpha_{hk} Y_{hk} + \beta_{hk} &= u(Y_{hk}). \end{aligned}$$

In the following, the method will be differentiated according to the conditions of certainty, risk of incertainty.

We find a synthesis function for each state of nature:

---

*) If the functions (3) have not their both extreme value finite, we introduce new additional restrictions through which we get this condition.

$$(6) \qquad \text{opt} \sum_{h=1}^{r} \sum_{\ell=1}^{t} \sum_{j=1}^{n} d_h \cdot \alpha_{hk} \cdot g_\ell \cdot c_{hk\ell j} \cdot x_j + \sum_{h=1}^{r} d_h \cdot \beta_{hk},$$

where d_h are "importance coefficients" of the objective functions, and α_{hk} and β_{hk} are obtained by solving the systems (5); we solve s linear programming problems determined by (1) and (6) finding the solutions: $S_1, S_2, \ldots, S_s$.

We apply further the known schemes of the theory of games against nature (the case of certainty conditions is equivalent to k=1).

P. BONZON, University of Toronto, Toronto.

Combinatorial Dynamic Programming: a set theoretical approach.

It is well known that, in the discrete case, Dynamic Programming is identified with a combinatorial algorithm equivalent to a partial enumeration of feasible solutions. This paper is an attempt to provide a general framework for enumeration algorithms, in order to be able to give a normal definition of Dynamic Programming of combinatorial type. It distinguishes this particular algorithm from more general algorithms and gives necessary and sufficient conditions for its application.

We first consider the most general discrete optimization problem of the form: minimum $f(x_1,\ldots,x_n)$, where f is any function over $(x_1,\ldots,x_n) \in E$ and E any finite set.
i.e. an optimal solution must belong to a given set of feasible solutions. We describe a general recursive optimization procedure, which is equivalent to a total enumeration of E. This process of enumeration uses the equivalence classes of E defined by successive cuts which generalize the cuts of a graph. The function f itself is decomposed in a sequence of successive applications ϕ_k such that

$$
\begin{array}{llll}
(x_1,\ldots,x_n) & \overset{\phi_n}{\rightarrow} & \psi_n & \\
(\psi_n,x_1,\ldots,x_n) & \overset{\phi_{n-1}}{\rightarrow} & \psi_{n-1} & \text{with the } \phi_k \text{ increas-} \\
 & & & \text{sing functions of} \\
(\psi_k,x_1,\ldots,x_k) & \overset{\phi_{k-1}}{\rightarrow} & \psi_{k-1} & \psi_{k+1} \\
(\psi_2,x_1,x_2) & \overset{\phi_1}{\rightarrow} & \psi_1 & = f(x_1,\ldots,x_n)
\end{array}
$$

i.e. $f = \phi_1 \circ \phi_2 \circ \ldots \circ \phi_n$.

It is always possible to find such a sequence for any function by taking

$$\phi_k(\psi_{k+1},x_1,\ldots,x_{k+1}) \equiv \psi_{k+1}, \qquad k=n-1,\ldots,1$$

with

$$\psi_n = f(x_1,\ldots,x_n).$$

We then demonstrate how, by assuming a priori hypotheses for the structure of the successive cuts and applications, this general procedure is simplified to a shorter procedure involving only a partial enumeration of E. This shorter procedure happens to be identified with the algorithm of Dynamic Programming and thus the a priori hypotheses for E and f constitute a posteriori necessary and sufficient conditions for Dynamic Programming of combinatorial type.

Under these conditions, the function f must be defined by a sequence of applications of the form

$$\phi_k(\psi_{k+1},x_1,\ldots,x_{k+1}) = g_k(\psi_{k+1},x_k,x_{k+1}),$$

$$k = n-1,\ldots,1 \text{ with } \psi_n = g_n(x_n).$$

The condition for the cuts of E expresses the fact that the feasibility of any component x_k (possibly vectorial) can depend only on its direct neighbours x_{k-1} and x_{k+1}.

C.G. BROYDEN and W.E. HART, University of Essex, Wivenhoe Park, Colchester.

A new algorithm for constrained optimisation.

We describe a new algorithm for minimising the function of n variables $\phi(x)$ subject to the $m(<n)$ nonlinear equality constraints $f(x) = 0$. It is assumed for the purpose of this paper that both the gradient $g(x)$ of $\phi(x)$ and the Jacobian $F(x)$ of $f(x)$ are available as explicit expressions.

Following Greenstadt and Bard (Keele Optimisation Conference, 1968) we solve the (m+n)th order system of nonlinear equations in x en y

$$F^T(x)y = g(x) \tag{1}$$

$$f(x) = 0$$

where y is the vector of Lagrange multipliers. The method used is a quasi-Newton one (see "Quasi-Newton methods and their Application to Function Minimisation", C.G. Broyden, Math. Comp., 21, 368-381) but the matrix update employed is designed specifically to take advantage of the special features of the Jacobian $J(x,y)$ of the system. It was shown by Greenstadt and Bard that this latter had the form

$$J(x,y) = \left[\begin{array}{c|c} G(x,y) & F^T(x) \\ \hline F(x) & 0 \end{array}\right]$$

where $G(x,y)$ is an nth order matrix which is symmetric but not necessarily positive definite. Thus $J(x,y)$ itself is symmetric but not necessarily positive definite. The update used (which is a rank-2 update of the approximation to the inverse Jacobian) maintains the symmetry of the approximation to $J(x,y)$ and also maintains the zero lower righthand corner partition.

Because of the imposition of this structure upon the approximation to the Jacobian, rapid convergence in the neighbourhood of the solution is attained.

We also assume that an initial estimate x_0 is available for the independent variables (if one is not it may be obtained by the use of penalty function techniques coupled with one of the unconstrained minimisers described in "Quasi Newton Methods ...") . To obtain an initial estimate of y we solve the equation

$$F(x)F(x)^T y = F(x)g(x)$$

thereby obtaining the best y in the least-squares sense.
Another possibility of solving (1) when a good initial approximation is not available is to use a Davidenko path technique ("On a New Method of numerical solution of systems of non-linear equations", D.F. Davidenko, Doklady Akad. Nauk. SSSR 88, 601-602). This is clearly related to the S.U.M.T. and we hope to further investigate this relationship in the near future.

NATHAN BURAS, Technion, Haifa.

Investment Scheduling in the Development of Water Resources.

The analysis of multiple-purpose projects, single-structured or where several structures compose a complex water resource system, necessitates often the application of mathematical programming methods. These methods may be used in solving problems of hydrology, engineering, and applied economics.

The development of water resources in a given region requires large investments for which several projects may compete. The overall investment is planned for a development period of several years, a given portion of it being allocated each year. The question arises as to which projects (or parts of them) should be built every year.

Moreover, it is desirable to establish the optimal allocation of the overall development investment to each year of the development plan.

Given revenue-outlay curves for each of the projects considered for the regional development of water resources, this double optimization problem is formulated as a sequential decision model. It is assumed that at specified descrete times t (years), decisions have to be made regarding the yearly investment $\bar{x}_t$ and regarding its allocation among the i possible projects. The sequential decision model is solved using forward dynamic programming.

Defining $f_1(\bar{x})$ to be the maximum net revenue generated by an investment $\bar{x}$ at the first stage (year) and pursuing an optimal policy, one can determine its allocation among the technologically feasible projects. This optimal allocation may be displayed as an $m \times n$ array, m being the number of values admissible by $\bar{x}$ and n being the number of projects. At the second stage (year), $f_2(\bar{x})$ is determined so that development funds are allocated to projects showing most promise. An iterative correction may be necessary for the previous year investment allocation, which would optimize the overall net revenue. One proceeds in a similar fashion until the entire development period is covered and $f_T(\bar{x})$ is determined. The set of tables obtained as computer printout for each t will form the basis of the optimal investment scheduling for the regional development of water resources.

C. BURDET, Carnegie-Mellon University, Pittsburgh, on leave from ETH, Zürich.

A Class of Cuts and Related Algorithms for Integer Programming.

The efficiency of the cutting plane approach to solve optimization problems in integers depends, to a large extent, on the depth of the cuts.

In recent publications, Balas and Young have established in somewhat different contexts that valid cutting planes could be generated from the intersection of (basic) rays with adequately constructed hyperspheres. Later Balas, Bowman, Glover and Sommer recognized that these intersection cuts could be improved by replacing the sphere by a polyhedron whose faces are tangential to the sphere.

Here is family of polyhedra is introduced in replacement of the sphere; these polyhedra are not tangential to the hypersphere and produce cutting planes with particular characteristics which are discussed. The Gomory cuts are contained in this family of cutting planes as an extreme case; and in general, the family contains cuts deeper than any of the Gomory cuts. (Depth is meant here to be measured in the direction of the objective function).

Constructive procedures are presented for generating step by step cutting planes which become deeper and deeper at every step, until the deepest cut is obtained. This construction often relies on a partial enumeration schema of the set of integer solutions; the corresponding algorithms may thus be viewed as a compromis between the partial enumeration and the cutting plane techniques.

Finally some concluding remarks on numerical experiments and their computational efficiency, convergence and further open questions are included.

R. CHANDRASEKARAN, Case Western Reserve University, Cleveland.

A Special Case of the Complementary Pivot Problem.

The fundamental problem in linear programming, quadratic programming and bimatrix games is the following: Given a real m-vector q and a real m×m matrix M, find vectors w and z which satisfy:

(1) $w \geq 0,\ z \geq 0,\ w = Mz + q$ and

(2) $w'z = 0.$

So far the most general class for which there are algorithms is a class $\mathcal{L}$ defined by B.C. Eaves. In the present paper we provide a simple algorithm for a class Z which is not a subset of $\mathcal{L}$.(Z is a class of matrices defined by Fiedler & Ptak: On matrices with Non-positive off-diagonal Elements and Positive principal Minors - Czech. Math. Journal 12 (62), pp. 382-400.) It is also shown that if $M \in Z$, then feasibility of (1) implies the existence of a solution to the whole problem.

A. CHARNES, USC Inc. and the University of Texas, Austin,
W.W. COOPER, USC Inc. and Carnegie-Mellon University, Pittsburgh,
M.A. KEANE, Applied Devices Corporation, College Point, N.Y.,
E.F. SNOW, Applied Devices Corporation, College Point, N.Y., and
A.S. WALTERS, Carnegie-Mellon University, Pittsburgh.

A Mixed Goal Programming Model for PPBS in a Consumer Protection Regulatory Agency.

This work reports some new approaches to cost/benefit and related issues that have emerged in the course of studies for a public regulatory agency. The agency's operations are directed toward consumer protection, some parts of which were reduced, in one way or another, to dollar measures of potential benefits of proposed and existing regulations. Other parts of this agency's protection duties, however, involve objectives with components that are non-comparable, incommensurable, or both, and hence do not admit of such reductions. The formulations and interpretations utilized to deal with all of these-commensurable, incommensurable and non-comparable elements—are discussed in a way that relates them to past and possible future courses of development for mathematical programming applications.

A.R. COLVILLE, IBM Corporation, White Plains, N.Y.

Acceleration of LP Computations.

The economic importance of Linear Programming applications to major users of this technique has been increasing almost exponentially since the early days when LP was first used. A good part of this growth is due the generation of larger and larger models in the study of time-staged and multi-facility systems. This trend has led to an increasing emphasis on the efficiency of the linear programming system being used.

Since the original implementations of the revised simplex method using the product form of the inverse, numerous mechanisms have been proposed for increasing performance of these codes by tailoring them to specific computing systems, taking into account common model structures, improving data processing characteristics, etc. This paper will describe a number of approaches for accelerating LP computations which have been tested and proven. It will also discuss conjectures for future efforts in this area. Included in the topics to be covered will be:

1) Pre-conditioning the matrix
2) Obtaining better starting solutions
3) Speeding up iterations
4) Reducing the number of iterations
5) Improving inversion
6) Handling special structures

The influence of LP performance considerations and other special algorithmic and functional capabilities on the design of future Mathematical Programming Systems will also be discussed.

JAMES W. DANIEL, The University of Texas, Austin.

Approximate Minimization of Functionals by Discretization.

This paper presents the simple general theory for analyzing the convergence properties of solutions of discretizations of constrained continuous optimization problems including discretized penalization or regularization methods. As specific examples, we survey convergence results for discretizations of continuous optimal control problems including discretized penalization methods.

L. DEKKER, Delft University of Technology, Delft.

Hybrid Computation in the Field of Mathematical Programming.

This paper will illustrate possibilities of hybrid computation for solving problems in the field of mathematical programming.

In contrast to each other, digital and analogue computation are favorite with regard to respectively

- memory, control of the computation
- computing speed, solving ordinary differential equations.

A hybrid computer - as a cooperation between an analogue and a digital computer - is able to combine these features in practice.

Computing methods in mathematical programming often ask for a digital computer because of the memory capacity and the flexibility of control of the computation by means of a stored program. However, mostly these methods are very time-consuming, in the first place because a solution is determined iteratively and often moreover a search procedure has to be applied in order to assure that the absolute optimum is found in stead of a local optimum. In hybrid computation one can apply discrete as well as continuous iteration processes. A continuous iteration process is equivalent with solving ordinary differential equations until (approximately) the

steady-state solution is reached. Solving a continuous iteration process asks for an analogue computer because of the high computing speed and its elegance for solving ordinary differential equations.

So for solving problems in the field of mathematical programming hybrid computation is attractive. This will be illustrated by discussing briefly some hybrid-numerical methods. For example attention will be given to an iterative method for solving the nonlinear problem: minimize $f(x)$, subject to $g_i(x) = 0$, $i=1,\dots,m$ and where $x \in R^n$, $m < n$. This method can be formulated as a (convergent) discrete iteration process

$$H_r(x) = \{f(x) - y_{r-1}\}^2 + \alpha \sum_i g_i^2, \ r \geq 1$$

with

$$y_{r-1} = f(x_{r-1}),\ H_{r-1}(x_{r-1}) = \min_x H_{r-1}(x);\ y_0 < f^*,$$

where f^* represents the minimum of $f(x)$, subject to $g_i(x) = 0$. Determining of $\min\limits_x H_{r-1}$ can be realized as a continuous iteration process by applying the gradient-method. The above method is based on transforming the original problem into:

$$\min_y \{\min_x H(x,y)\},\ y \leq f^*$$

where

$$H(x,y) = \{f(x)-y\}^2 + \alpha \sum_i g_i^2.$$

Applying the gradient-method twice to the last problem the iterative method can also be formulated as a nesting of two continuous iteration processes in different time-scales.

Some practical results will be presented.

HANS-GEORG DIESS, SACLANT ASW Research Centre, La Spezia.

Game Theoretical Solution to an Aiming Problem.

This paper discusses the game theory solution to the following aiming problem:

An attacker receives information about the location of a target and launches a weapon. It is assumed that the target may be anywhere within an annulus with radii R_1, R_2, which depend on the weapon delivery time and the target evasion manoeuvres. Using a polar coordinate system R, β, the assumption is made that the target is uniformly distributed in β, but chooses R between the limits R_1, R_2, in order to maximize its chance of escape. The attacker will then distribute the weapon aimpoints uniformly in the angle β over its range 0, 2π. For a single weapon $\beta = 0$ is chosen arbitrarily and the problem is reduced to the choice of the radial coordinate X for the aimpoint. The pay-off for this two-person game, where both players have continuous strategies, is expressed by the probability of the target having a distance from the aimpoint of less than a fixed damage radius e. Pure and mixed strategy solutions are discussed and conditions are derived for the normalized parameters

$$\frac{2e}{R_2-R_1}\,,\ \frac{R_2}{R_2-R_1}$$

which allow one to determine the type of strategies for a given set of values of the parameters e, R_1 and R_2.

PIERRE DOULLIEZ, Société de Traction et d'Electicité, Belgium.
M.R. RAO, University of Rochester, Rochester, N.Y.

A Labeling Algorithm for a Multiterminal Network with Any One Arc Subject to Failure.

In this paper we consider a multiterminal network consisting of several demand nodes, each with an associated demand function increasing over time. The demand nodes are connected to a common source node through several intermediate nodes. Associated with each arc of the network are two values which represent the normal and reduced capacity of that arc. It is assumed that at any given instant at most one arc may have a reduced capacity. It is required to find the maximum time upto which all demands can be satisfied. A labeling algorithm is given to solve this problem and finiteness of the algorithm is proved.

IRINEL DRAGAN, C.O.R.E., Heverlee, Belgium.

An improvement of the lexicographical algorithm for solving discrete programming problems.

Consider the following problem: minimise $f_0(x_1,\dots,x_n)$, subject to

$$f_h(x_1,\dots,x_n) \le 0, \ (h=1,\dots,s), \ x_i \in \{0,1,\dots,p_i\},$$

$$(i=1,\dots,n),$$

where p_i are positive integers.

For solving this problem by the lexicographical algorithm, it is always necessary to carry out a previous transformation. The above stated problem thus becomes: find the "first" vector - in the sense of the lexicographical ordering - belonging to

$$X = \{(x_0,\ldots,x_n)/f(x_0,\ldots,x_n) \le g(x_0,\ldots,x_n)\},$$

$$x_i \in \{0,1,\ldots,p_i\},\ (i=0,1,\ldots,n)\}$$

where p_0 is a positive integer and $f(x_0,\ldots,x_n)$, $g(x_0,\ldots,x_n)$ are lexicographical monotone nondecreasing functions. The function $g(x_0,\ldots,x_n)$ is chosen from a certain subset G of the set of all lexicographical monotone nondecreasing functions.

In the present paper is shown that the number of necessary steps N, for solving the last problem by the lexicographical algorithm, depends on the chosen function $g \in G : N = N(g)$. A procedure is given for constructing a function $\bar{g}(x_0,\ldots,x_n) \in G$ in such a way that $N(\bar{g}) \le N(g)$, for all $g \in G$. Of course, when the ordering of the variables is changed the function $\bar{g}(x_0,\ldots,x_n)$ will be changed as well. Another procedure for selecting the appropriate ordering is given such that the optimal solution will be found after a minimal number of steps. Finally, a numerical example is presented, in which the two procedures and the lexicographical algorithm are used.

At the present time the described procedure is worked out for the linear case; the extension for the nonlinear case is not yet completed.

SALAH E. ELMAGHRABY, North Carolina State University, Raleigh, and MONMAHAN K. WIG, The Corning Glass Co., Corning, N.Y.

On the Application of Diophantine Programming Concepts to Stock-cutting Problems.

Two specific problems of the one-dimensional stock cutting activity are discussed. Problem I minimizes the maximum absolute deviation of realized production from desired production subject to upper and lower bounds on the total quantities produced of m different lengths. Problem II determines the stock length (or stock lengths) that minimizes the total material used and satisfies a

given demand schedule. The approach to both problems is through Diophantine arguments and dynamic programming formulations.

ALEXANDER J. FEDEROWICZ, Westinghouse Research Laboratories, Pittsburgh.

Asymptotic and Approximate Analytic Solutions to Geometric Programming Problems.

Zener's initial result (1961) showed that an analytic solution to a simple (i.e. zero degree of difficulty) geometric programming problem can always be obtained. This paper shows how an analytic solution to a complex G.P. problem can be obtained by solving either an approximate problem or an asymptotic problem both of which are L.P. problems. The interpretation of these L.P. problems, their duals and of the various cases which can occur in solving these L.P. problems are of interest. The asymptotic technique has been applied to transformer design equations; the approximate technique has been useful in obtaining answers to large Chemical Equilibrium problems and in obtaining starting point solutions to complex G.P. problems. These results highlight the close theoretical and computational relationships which exist between G.P. and L.P.

JACQUES A. FERLAND, Stanford University, Stanford.

Quasi-convexity and Pseudo-convexity of Quadratic Functions.

It is well known that quasi-convexity and pseudo-convexity play a "natural" role in nonlinear programming theory. Despite this, these notions lack utility because they have defining conditions involving infinitely many inequalities and are not easily checked.

Extending two recent works of Béla Martos, we prove that testing the quasi-convexity (pseudo-convexity) of a quadratic function

$$\phi(x) = \frac{1}{2} x^T D x + c^T x$$

on the nonnegative (semipositive) orthant can be reduced to an examination of finitely many conditions on the matrix

$$\begin{bmatrix} D & c \\ c^T & 0 \end{bmatrix}$$

associated with the function ϕ, D being a real square symmetric matrix of order n, c and x vectors of order n.

This criterion applies to the quasi-convexity of quadratic functions over convex sets larger than the nonnegative orthant. We are interested in the maximal domain of quasi-convexity for a quadratic function and give a characterization thereof. A similar analysis is pursued for pseudo-convexity.

JEAN CHARLES FIOROT, Faculté des Sciences, Lille.

Some Linear Inequalities in Z^n.

We propose to resolve practically some linear inequalities in integers i.e. to generate all the vectors with integer components which satisfy the following inequalities:

1) $Ax \geq b$ where A is a matrix with integer or rational entries - it will always be so in the following - of rank n, with n rows and n columns, $b \in \mathbb{R}^n$.

Then $Ax \geq b$ is a regular polyhedral cone whose vertex $\bar{x}$ is the unique solution (non necessarily integer) of $Ax = b$.

We introduce a particular set (noted P_f) of integer points called fundamental points all situated in a parallelotop the faces of which are parallel to those of the cone.

Theorem: Every integer point of a regular polyhedral cone of $\mathbb{R}^n$ is either a point of P_f or is translated from a point of P_f by integer translation vectors parallel to edges of the cone.

The set P_f and the translation vectors are perfectly determined: a construction of them is given. A simple formula gives the number of these fundamental points.

2) $Ax \geq b$, A of rank m with m rows and n columns, $m < n$, $b \in \mathbb{R}^m$.
   As before we introduce P_f and the linear varieties

$$K_j = \{x \mid A_i x = b_i,\ i \neq j\} \text{ and}$$

$$A^* = \{x \mid A_i x = b_i,\ i = 1,2,\dots,m\} .$$

A corollary is given.

3) In the case when $Ax \geq b$ define a polyhedral cone whose number of faces is greater than the dimension of the space, we decompose the cone into a union of regular cones having at most one face in common and we apply 1).

4) $b \leq Ax \leq c$, A of rank n, with n rows and n columns, b and c are vectors of $\mathbb{R}^n$. This defines a parallelotop of $\mathbb{R}^n$. We introduce a set of integer points (noted P_f^*), called fundamental points of the parallelotop which is a subset of P_f associated with one of the cones asymptotic to the parallelotop. A theorem is given.

5) We also treat the case of an unbounded parallelotop defined by

$$b_J \leq A_J x \leq C_J,\ b'_J \leq A_{J'} x,\ |J \cup J'| = n,$$

$$J \cap J' = \emptyset,\ A_{J \cup J'} \text{ of rank n.}$$

Or by:

$$b_J \leq A_J x \leq C_J,\ b_{J'} \leq A_{J'} x,\ J = \{1,2,\dots,s\},$$

$$J' = \{s+1,\dots,m\},\ m < n,\ A_{J \cup J'} \text{ of rank m.}$$

Application: Let us consider the problem:

PM : Max $\{fx \mid Ax \geq b\}$ where fx is a linear form, and $\{x \mid Ax \geq b\}$ is a cone. Let us suppose that the vertex $\bar{x}$ is the optimum for PM, then the optimum in integers is a point of P_f. This conclusion is identically extended to the parallelotop replacing the set P_f by P_f^*

J.S. FOLKERS and O.B. de GANS, Delft University of Technology, Delft.

Geometric Programming: Some hard Questions.

The literature on geometric programming may be summarized by a quotation from one of the publications: "When it works, it works admirably".
Trying to apply the method to some relatively simple problems, the authors were faced with the situation in which it did not work. This paper is a report on the analysis of the questions raised by that failure, which are mainly concerned with the use of Lagrange multipliers for the development of geometric programming.

A.M. GEOFFRION, University of California, Los Angeles.

Vector Maximal Decomposition Programming.

Many problems in large-scale mathematical programming, decentralized economic planning, and engineering can be cast in the following terms. There is a system composed of a number of subsystems, each seeking to optimize its own objective function by choice of its own variables but subject to some control by a coordinator of the system as a whole. The task of the coordinator is to exercise control over the subsystems in a way that achieves the most preferred vector maximum of the (possibly incommensurate) subsystem objective function values.

Two iterative coordination procedures are derived for a fairly broad class of nonlinear but convex systems, one "global" in its view at each iteration and the other "local". These procedures can be viewed as extensions of the Tangential Approximation and Large-Step Subgradient methods presented in a previous paper by the author ["Primal Resource-Directive Approaches for Optimizing Nonlinear Decomposable Systems," The RAND Corporation, RM-5829-PR, December, 1968; soon to be published in Operations Research]. The implications of each procedure for decentralized decision-making are examined. Some interesting possibilities are indicated for interactively guiding the coordinator when he cannot explicitly state his entire preference function. This in turn raises some very fundamental questions concerning how a decisionmaker can or ought to deal simultaneously with numerous criteria.

GERZSON KERI, Hungarian Academy of Sciences, Budapest.

A Modified Stepping-Stone Algorithm for the Transportation Problem.

The computational efforts required for the stepping-stone algorithm to solve the transportation problem

$$\sum_{j=1}^{n} x_{ij} = a_i \qquad (i = 1,2,\ldots,m)$$

$$\sum_{i=1}^{m} x_{ij} = b_j \qquad (j = 1,2,\ldots,n)$$

$$x_{ij} \geq 0 \qquad (i = 1,2,\ldots m;\ j = 1,2,\ldots,n)$$

$$\min \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij}\, x_{ij}$$

can be reduced by carrying out the transformations of the numbers $\delta_{ij} = z_{ij} - c_{ij}$ throughout basis changes in the following way.

Given a set H as a system of basis cells, let us consider the numbers $\delta_{ij} = z_{ij} - c_{ij}$ and let δ'_{ij} be their new values after a cell $(i_0,j_0) \in H$ leaving and another cell $(i_1,j_1) \notin H$ entering the basis. Now let us construct the sets C and D, which represent some rows and columns of the transportation tableau respectively, by the following algorithmic labelling process. First let us label the i_1st row and the columns of all basis cells, except the cell (i_0,j_0) lying in the i_1st row. At the general step let us label the rows of all basis cells lying in the previously labelled columns, provided that these rows have been unlabelled till now. Next let us label the columns of all basis cells, except the cell (i_0,j_0) lying in the rows labelled just now, provided that these columns have been unlabelled till now. The process terminates when either no further row or no further column can be labelled in this way. The indices of all labelled rows will form the set C and the indices of all labelled columns will form the set D. The transformation formulae for the numbers are as follows:

$$\delta'_{ij} = \delta_{ij} \qquad \text{if } i \in C \text{ and } j \in D,$$

$$\text{or } i \notin C \text{ and } j \notin D,$$

$$\delta'_{ij} = \delta_{ij} - \delta_{i_1,j_1} \quad \text{if } i \in C \text{ and } j \notin D,$$

$$\delta'_{ij} = \delta_{ij} + \delta_{i_1,j_1} \quad \text{if } i \notin C \text{ and } j \in C.$$

The method described here for the transformation of the numbers δ_{ij} is also inserted in the algorithm elaborated in details.

P.M. GHARE, Virginia Polytechnic Institute, Blacksburg.

Multi-step Gradient Methods for Non-linear Programming.

We consider a general mathematical programming problem of optimizing a function

$$F(X) = F(x_1,x_2,...x_n) \qquad 1.1$$

subject to constraints

$$g_i(X) = g_i(x_1, x_2, \ldots, x_n, b_i) \geq 0 \qquad i = 1 \ldots. m \qquad 1.2$$

and

$$x_i \geq 0 \qquad i = 1 \ldots. n. \qquad 1.3$$

Several subsets of this general problem have been solved by gradient methods. As a class the gradient methods are based on the existence of a continuous curve (a trajectory) from any point X^0 to the optimal X^* such that the F(X) is a monotonic (increasing or decreasing) function along the curve. Although it is not necessary, it is desirable that the entire trajectory lies within the feasible domain described by 1.2 and 1.3. In the steepest gradient methods the directions of the trajectory and the steepest gradient coincide. In deflected gradient methods a direction other than the direction of the steepest gradient is chosen.

One Step Finite Iteration Methods.

Most practical gradient method solutions to mathematical programs employ linear steps of finite length. The trajectory is replaced by segments of straight lines. The length of each step is obtained by optimizing the objective function along the chosen direction (full step) or as a multiple of the full step. The direction is chosen by a matrix transformation of the gradient direction. Thus a "one step finite iteration" method can be described as

$$S_k = H_k \nabla\phi(X_k, U_k)$$

$$a_k^* \rightarrow F(X_k + a_k^* S_k) \geq F(X_k + a_k S_k) \text{ for any } a_k$$

$$X_{k+1} = \gamma_k a_k^* S_k + X_k.$$

The designation "one step" implies that the optimization policy (γ_k, H_k) is determined for one step at a time and "finite" implies a finite movement $(X_k - \dot{X}_{k-1})$ for each step rather than infinitesimal movement ∂X in pure gradient methods. Examples of one step finite iteration procedures are

a) Steepest gradient methods when $\gamma_k = 1$ and $H_k = I$ for each k

b) Gradient Projection methods when H_k is a projection matrix and

c) Conjugate gradient methods when S_k is orthogonal to each $S_1 \mid 1 < k$.

Multi-Step Finite Iteration Methods.

In an s-step finite iteration method the optimization policy is determined for "s" steps simultaneously which are incorporated into a single computational process. Consequently an s-step method would be desirable if

a) the error (or suboptimization $|F(x^*) - F(X_k)|$) after one iteration of s-step method is smaller than s iterations of a one step method; and

b) the computational effort for one iteration of an s-step method is smaller than s iterations of a one step method.

Multi-step methods have been studied extensively in connection with the solutions to linear algebraic systems and the rate of convergence of multi-step methods is shown to be uniformly superior to corresponding one-step methods. Although multi-step methods have not been used for M.P. problems, the "Pattern Search" procedure used for unconstrained experimental optimization can be shown to be an elementary multi-step method.

The purpose of this communication is to extend the application of multi-step gradient methods to the solution of general mathematical programming problems. Successful multi-step solution techniques can be developed by expressing the Kuhn-Tucker optimality conditions as a system of equations (in the general case these would not be linear) and solving them by a multi-step gradient procedure. This paper describes the conditions for convergence and the rate of convergence in the generalized case. A simple 2-step algorithm is described for the Quadratic Programming problem as an illustration and possible extensions to experimental optimization are discussed.

FRANCO GIANNESSI, Universita' di Venezia, Venice.

A gradual algorithm for the resolution of linear programming problems of large scale.

The aim of this paper is first of all the description of an algorithm which lets us to solve a linear programming problem by considering the m constraining equations one at a time; that's we consider m linear programming problems which have the same objective function as the initial problem, and which have, as constraints, the first equation, the first two,..., the first m equations of the initial problem respectively, and obviously the nonnegativity inequalities. Knowing a solution of the first subproblem we go to a solution of the second subproblem, and so on to a solution of the last subproblem, which is a solution also of the initial problem.

Such an algorithm (which is a variant of Simplex Method) is different from the Cross-section Method (by J.J. Stone), because it maintains the optimality (instead of the feasibility) during the various iterations.

The algorithm we introduce has some advantages in respect of those already existing: it is more efficient from a computational viewpoint (we have a cut of about 50% in computational time and of about 30% in required memory in respect of Revised Simplex Method. Moreover, it does not degenerate and cannot have cycles.

The preceeding algorithm is especially interesting when applied to linear programming problems of large scale. In these problems, as it really happens, the matrix of constraining system has a large number of zero elements, which are not assigned at random, but they are arranged to form submatrices (blocks); this happens for instance in block angular, and block triangular problems. In such cases the algorithm we consider lets us operate only on non-zero blocks of the matrix of constraining system, by solving only at one time the subproblems which correspond to the preceeding blocks.

We observe that by such an algorithm it is possible to treat in the same way block angular, block triangular problems and every other problem containing some zero blocks located at random.

Till now we have experimented the algorithm on many numerical problems with an unexpected success.

At last an extension of such algorithm lets us consider in a similar way quadratic programming problems having zero blocks. The numerical experiments in this field are at the beginning.

The algorithm we have introduced has been applied with success to optimal control and stochastic programming problems.

P.E. GILL and W. MURRAY, National Physical Laboratory, Teddington.

A Numerically Stable Form of the Simplex Algorithm.

This paper is concerned with the solution of the following linear programming problem:

$$\min: c^T x$$

subject to

$$A^T x \geq b.$$

Standard implementations of the Simplex Method have been shown to be subject to computational instabilities. A numerically stable form of the Simplex Method is presented with storage requirements and computational efficiency comparable with those of the standard form.

The algorithm is based on the ability to recur from one iteration to the next the lower triangular matrix L_i where:

$$A_i^T A_i = L_i L_i^T ,$$

A_i being the matrix of coefficients of the active constraints at the ith iteration. Although the principle concern of the paper is not with constraints with a large number of non-zero elements, all necessary modification formulae for the extension to these cases are given.

The algorithm has the following features:

(i) The algorithm admits non-Simplex steps, for instance, a step can be made across the interior of the feasible region. This feature enables the method to be readily generalized to quadratic and non-linear programming.

(ii) The algorithm only needs an initial point which is feasible.

(iii) In addition to the storage needed to define the problem the algorithm requires at most half the storage required for the Revised Simplex Method.

(iv) The number of operations involved per iteration is at most O(nm) where n is the number of variables and m the number of constraints.

R.E. GOMORY and E.L. JOHNSON, I.B.M. Thomas J. Watson Research Center, Yorktown Heights, N.Y.

Some Continuous Functions related to Corner Polyhedra and their Applications to Integer Programming.

Methods will be presented for generating inequalities and terminating conditions useful in enumeration or branch-and-bound algorithms for integer programming. These methods avoid the complexity of the corner polyhedra corresponding to large groups and are numerically simple. The work is based on approximating any corner polyhedron using a class of continuous functions arising from a fixed corner polyhedron. The theory is extended to apply to mixed integer programs. Numerical experiments will be described to give an indication of the effectiveness of these methods.

S. GORENSTEIN, IBM N.Y. Scientific Center, New York.

An Algorithm for Project (job) Sequencing with Resource Contraints.

In this paper we present an algorithm for solving the project scheduling (machine scheduling is included in project scheduling) problem with resource constraints.

The machine scheduling problem has been formulated in several ways as an integer program; it is the formulation as a disjunctive graph that we are concerned with in this paper. This involves the insertion of disjunctive arc pairs into the network. A disjunctive arc pair is a pair of arcs between nodes, and only one of them is permitted to appear in a derived fixed network. A selection of one out of each pair of disjunctive arcs determines a fixed network which represents the sequence of processing on each machine. This network has a critical (longest) path. The optimal solution is a selection (out of all selections without circuits) from all the disjunctive arc pairs such that the resulting fixed network has the minimal critical path length; i.e., a minimaximal path is sought, or minimum overall processing time.

A further generalization is to allow for more than one machine of each type and to allow for more general jobs (projects) than sequencing through a set of machines. A multi-project network with resource constraints can be cast into this form. In this case we have to allow for the possibility that neither of the disjunctive arcs of a pair need appear in a particular network determined by a selection. However we need the additional condition that a network must be feasible with respect to resources to be eligible for consideration; that is, there must be sufficient resources to actually accomplish the processing represented by the network. The feasibility of a network can be related to the generalized coefficient of internal stability of its transitive closure. The generalization to more than one machine of each type or to more general resources requires a feasibility check, that is, those networks that are not feasible with respect to resources would have to be eliminated from consideration, and the critical path computation made only for acyclic, feasible networks.

The prospective advantage of this approach is the elimination of the need to consider individual time periods over the program horizon. An algorithm is presented which uses partial enumeration for what is essentially a mixed integer program. The algorithm employs a maximum flow computation as a check for feasibility with respect to available resources.

F.J. GOULD and JON W. TOLLE, University of North Carolina, Chapel Hill.

A Necessary and Sufficient Qualification for Constrained Optimization.

The following optimization problem with mixed constraints is considered: maximize

$$f(x),$$

subject to

$$g_i(x) \leq 0,\ i = 1,\ldots,m;\ r_i(x) = 0,\ i = 1,\ldots,k;\ \text{and}$$
$$x \in D \subset R^n,$$

where D is an arbitrary set in R^n. The objective and constraint functions are assumed to be continuous on some open set containing D and differentiable at a local optimum for the given problem. A weak qualification is given which insures that this problem satisfies the analogue of the Kuhn-Tucker conditions at the local optimum. The qualification is weaker than that of Mangasarian and Fromovitz, and for the problem with pure inequality constraints it is weaker than the qualifications of Abadie and of Arrow, Hurwicz and Uzawa. It is shown to be the weakest possible in the sense of being necessary and sufficient for Lagrange regularity of the above problem. In the special case of pure equality constraints, the new qualification is evidently necessary and sufficient for the classical Lagrange multiplier rule to be valid. In order to express the qualification, suppose x_0 is a local solution to the given problem. Let

$$I_0 = \{i: g_i(x_0) = 0, 1 \le i \le m\},$$

$$\bar{C_0} = \{x \in R^n: x^T \nabla g_i(x_0) \le 0, i \in I_0\},$$

and let $L_0^{\perp}$ be the orthogonal complement of the subspace spanned by

$$\nabla r_i(x_0), \ i = 1,\ldots,k.$$

Let A' be the polar cone of an arbitrary set $A \subset R^n$, let S be the constraint set of the given problem, and finally let $T(S,x_0)$ be the cone of tangents to S at x_0. Then the weak constraint qualification can be stated as

$$(C_0 \cap L_0^{\perp})' = T'(S,x_0).$$

G. GRAVES and ANDREW WHINSTON, Purdue University, Lafayette, Ind.

Application of Mathematical Programming to Regional Water Quality Management.

It has become apparent that major changes are desirable in the institutional structure for water quality in the United States, The desirable structure would be a regional or basin-wide water quality management authority.

A single agency would control all discharges (industrial and municipal) and operate all treatment plants in a region. It would contruct new regional treament plants in optimum locations and control the distribution and re-distribution of treated and partially treated wastes.

The authority would be responsible for finding and implementing the least cost solution of meeting the stream (or estuary) water quality goals. The question of setting water quality goals relates to social and economic desires and needs. This question need not be resolved by the authority but it could make valuable contributions to the rationality of the process. When a change in goals is proposed, the authority will determine the optimum solution to meet the changes in addition to the cost. Thus, an informed public should be better able to decide what quality of water it is willing to pay for.

It is the purpose of this work to present a planning tool (algorithm) to provide optimal solutions for the complex choices involved in balancing alternative methods for attaining water quality goals. As one might suspect, there are a tremendous number of alternatives that would achieve these desired goals in a body of water. One of the most commonly proposed solutions is for the polluters to increase their levels of treatment. This is also one of the most expensive solutions.

Within the framework of a proposed regional water quality management authority, we are going to investigate this problem with some powerful tools from non-linear programming and control theory. Note that for the Delaware estuary, which we are going to study, an authority such as proposed above (namely the Delaware River Basin Commission) already exists.

Considering the size and complexity of the problem, the cost of computer time, and assuming a reasonable number of computer runs, one of the prime goals of this work was to develop an algorithm that gives efficient solutions in a reasonable time. This became a formidable undertaking because of the many undesirable features inherent in our non-linear programming model of the physical situation. It has over 2000 variables and over 80 constraints. Some of the first partial derivatives are discontinuous and the transfer functions have such a wide range that scaling is extremely difficult.

J. GREENSTADT, IBM N.Y. Scientific Center, New York.

Variable-Metric Formulas from Variational Principles.

The main problem in variable-metric methods is to estimate the essential quantities that characterize a quadratic approximation to the function to be minimized. In those problems in which the gradient can be independently calculated, the essential quantities are the components of the (symmetric) Hessian matrix. Where the gradient cannot be independently calculated, it is necessary to estimate it (or some version of it) as well.

By defining a "best" correction (in the sense of minimizing a quadratic norm) to be made to the essential quantities after each step, and by imposing certain natural restrictions on these corrections (which are based on identities holding for quadratic functions), one is let to an equality-constrained variational problem, which is easily solved. By assigning various simple forms to certain weighting matrices appearing in the quadratic norm, one may generate various correction formulas (including the well-known Davidon formula). In the gradient-free case, formulas for the corrections to the estimated gradient and to the estimated Hessian result.

These formulas have been tested on a computer, and all converge and yield the correct Hessian at the end. (The exceptions are those cases where the minimum is not quadratic). The result for various well-known test functions will be shown.

MICHAEL D. GRIGORIADIS , IBM N.Y. Scientific Center, New York.

A Projective Method for a class of Structured Nonlinear Programming Problems.

This paper describes a partitioning method for solving the following structured nonlinear programming problem:

$$f(x_1^*,\dots,x_k^*,y^*) = \min \{f(x_1,\dots,x_k,y) \mid (x_1,\dots,x_k,y) \in S\}$$

where

$$S = \cap_{j=0}^{k} S_j$$

$$S_0 = \{x_j \in \mathbb{R}^{n_j};\ j=1,\dots,k;\ y \in Y\}$$

$$S_j = \{x_j \in \mathbb{R}^{n_j};\ y \in Y \mid B_j'x_j + D_j'y \le h_j\};\ j=1,\dots,k$$

$$Y = \{y \in \mathbb{R}^{n_0} \mid D_0(y) \le 0\}$$

$$f(x_1,\dots,x_k,y) = \Sigma_{j=1}^{k} f_j(x_j,y) + f_0(y)$$

B_j', D_j', h_j are (m_j,n_j) and (m_j,n_0)-matrices and m_j-vectors respectively and $D_0: \mathbb{R}^{n_0} \to \mathbb{R}^{m_0}$ is a given vector function.

The proposed algorithm uses the special structure of the constraints to reduce the given problem by elimination of variables. In variance to other methods proposed previously, this elimination is effected through the use of the general solution to an underdetermined system of linear equations representing the active constraints at a given feasible point. For weakly coupled systems ($n_0 << \Sigma_{j=1}^{k} n_j$), this arrangement provides a drastic reduction in the number of variables and the problem reduces to one with only n' variables. Under appropriate differentiability assumptions it is shown that a constrained stationary point of the overall problem is obtained by solving a sequence of the reduced nonlinear programs. Primal feasibility is maintained throughout the optimization procedure. Global optimality of the solution is then assured by additional convexity assumptions on f and D_0.

Specializations to quadratic functions f(x,y), matrix structures arising in distribution problems, implicit handling of upper and lower bounds on the variables (x,y) and the resulting algorithmic simplifications are also discussed. Computational experience and results for several structured quadratic programming problems are presented.

MICHAEL D. GRIGORIADIS and W.W. WHITE, IBM N.Y. Scientific Center, New York.

A Partitioning Algorithm for the Multi-commodity Network Flow Problem.

This paper presents an algorithm for solving the multi-commodity flow problem for directed networks. Appropriate partitions of the network into the usual master problem-subproblem divisionalization are used in the framework of a flow method. This approach differs from specializations of Dantzig-Wolfe decomposition and is expected to avoid the familiar "tailing" problems encountered with that method.

For treating the subproblem, the structure of the associated network is utilized. Any network technique which records the tree structure is satisfactory since the computations performed are essentially those of the usual network flow variety, e.g. as in the Out-of-Kilter method. Analysis of any nonbasic arc in a subproblem amounts to finding a cycle containing this arc along which flow can be augmented.

The "master" problem is a linear program which is treated by a modification of the simplex method. In certain cases, use of a modified dual simplex method is considered. For computational efficiency these algorithms utilize a working basis of considerably smaller dimension than the number of arcs in the given network. A set of secondary constraints is periodically examined during the optimization procedure. Additionally, the network structure of the subproblems and that of the bounding requirements allow the generation of nonbasic columns for the master problem in a simple fashion. Thus explicit representation of this problem is avoided.

The proposed algorithm is easily extended to handle the case where there are given restrictions on the weighted sums of the flow through the arcs. Some computational aspects, including programming and data handling considerations, in comparison to other existing methods are also discussed.

M. GUIGNARD, University of Lille, Lille, and
K. SPIELBERG, IBM N.Y. Scientific Center, New York.

The State Enumeration Method for Mixed Zero-One Programming.

This paper reports extensions of, and results obtained with, the algorithm proposed in "Search Techniques with Adaptive Features for Certain Integer and Mixed Integer Programming Problems", M. Guignard, K. Spielberg, IFIPS Congress 1968. Consider the problem

$$\min \zeta = \delta\xi + \gamma\eta$$

$$D\xi + C\eta \le \beta \qquad (1)$$

$$\xi \ge 0, \ \eta_k \in \{0,1\}, \ k = 1,2,\ldots,p$$

and an enumerative search procedure of the single branch type.

At a node ν of the search tree the situation is as follows. Certain components η_k are fixed at one (zero), i.e.: $k \in E^\nu$ ($k \in Z^\nu$). The others are yet "free", $k \in F^\nu$. The task is to devise effective devices for:

a) determining whether the node may be abandoned (return to a predecessor node),

b) determining what free variable may be eliminated at the node,

c) choosing a new branch of the tree, i.e. fixing some free variable arbitrarily at 1 or 0.

The essential notion is that of a "state", which is a partitioning of the set of free variables into three sets,

$$F^\nu = (F1^\nu, F2^\nu, F3^\nu).$$

The state is usually passed from a predecessor node with obvious modifications. It may be prespecified at the start of the computation and modified later in auxiliary problems.

Problem (1) is replaced by a sequence of "State Problems" S^ν, one for each ν. S^ν is (1) with

$$\eta_k = 1(0),\ k \in E^\nu + F1^\nu(Z^\nu + F2^\nu),\ 0 \le \eta_k \le 1,\ k \in F3^\nu.$$

The solution to (1) is eventually obtained as a solution to one of the state problems. Also, each state problem, whether feasible or not, furnishes an associated inequality for the η_k (related to those of J.F. Benders), which in turn is utilized in a-c above, as well as in the establishment of global bounds.

General programs have been written. One of them functions within a general linear programming, mixed integer programming system LPS/MIP developed at the IBM New York Scientific Center. Special problems of the Knapsack - and Plant Location Type have been considered. Numerical results will be discussed.

A well constructed program, in which the states are determined intelligently and the inequalities are exploited judiciously, one or several at a time, should yield good operational results. Branch-Bound Programming, for example, can be viewed as a simple, special form of a State Enumeration Algorithm, augmented by the usual penalty calculations.

MILTON M. GUTTERMAN, Standard Oil Company (Ind.), Chicago.

Efficient Implementation of a Branch-and-Bound Algorithm.

The general mixed integer programming problem can be solved, at least conceptually, by the branch and bound algorithm. In order to make this approach an effective computational tool, a number of techniques must be used to keep the computation time and cost within reasonable bounds. These techniques can be divided into four categories:

algorithmic choices (heuristics)
algorithmic shortcuts
data handling techniques
computation techniques.

The first two categories relate to the algorithm details; the last two to the computer programming work to implement the algorithm.

The author has just implemented a branch and bound algorithm which operates under MPS/360 on the IBM system 360. This paper describes the exact algorithm which was implemented, including the choices and shortcuts. The data handling and computation techniques which were incorporated to improve efficiency are also described.

During the development and implementation of this algorithm a number of techniques (primarily in the data handling area) were considered and rejected because they would take too long to develop. These techniques are also described.

PETER L. HAMMER, Université de Montréal, and Technion, Haifa, and
URI N. PELED, Israel Defence Forces and Technion, Haifa.

On the Maximization of a Pseudo-Boolean Function.

A B-B-B (Boolean Branch and Bound) type method is proposed for the maximization of real valued functions with variables taking only the values 0 and 1. The importance of the problem consists in the fact that numerous problems in operations research, graph theory, combinatorial mathematics, etc. can be brought to this form. The method has been tested on an IBM 360/50 with good results.

[1] P.L. HAMMER and URI N. PELED, On the Maximization of a Pseudo-Boolean Function. Technion, Mimeograph Series on Operations Research, Statistics and Economics, No. 69, 1970.

PIERRE HANSEN, Université Libre de Bruxelles, Bruxelles.

Non-linear 0-1 Programming by Implicit Enumeration.

An algorithm is proposed for the problem: minimize f(S) under the conditions $G(S) \geq 0$ and $S \in B_2^n$, where f(S) is a pseudo-boolean function, G(S) is a vector of pseudo-boolean functions and S a boolean vector. No requirements such as convexity or monotony are made on f(S) or the functions $g_i(S)$ of G(S), but if such properties do exist the algorithm can make use of them. A theorem on pseudo-boolean functions provides upper and lower bounds on the values taken by f(S) or $g_i(S)$ on subsets of the set of boolean vectors. This allows to construct for the general nonlinear 0-1 programming problem an implicit enumeration algorithm similar to those of BALAS, GLOVER, GEOFFRION for the linear case. Because of the nonlinearity of the functions, the ceiling tests and direct or conditional feasibility tests are more powerful than corresponding tests in those algorithms. Computer experience is favourable: all problems of a series with 40 terms and 10 to 80 variables were succesfully passed on an IBM 7040 with a maximum computing time of 626 seconds.

N.A.J. HASTINGS, University of Birmingham, Birmingham.

Optimization of Markov Decision Problems.

Methods for the optimization of Markov decision problems have been developed by several authors using linear and dynamic programming techniques. The dynamic programming methods include the value iteration algorithm developed by Bellman and modified by White and the policy iteration algorithm of Howard.

A recent development is the policy-value iteration method of Hastings which combines features of both earlier dynamic programming approaches. A discription of this method in its application to undiscounted, discounted, recurrent chain and transient Markov processess is given in this paper. A new and shorter proof of convergence of the policy-value iteration algorithm is also given. Bounds on the steady state gain of single recurrent processess are developed. These bounds provide a basis for terminating a computation when a current policy gives a gain which for practical purposes is sufficiently close to the optimum. Both the policy-value iteration method and the system of bounds extend to Markov renewal programming.

MICHAEL HELD, IBM Systems Research Institute, New York, and
RICHARD M. KARP, University of California, Berkeley.

A Combined Ascent and Branch-and-Bound Algorithm For the Traveling-Salesman Problem.

This paper presents an algorithm for the solution of symmetric traveling-salesman problems which embeds an ascent method within a branch and bound procedure. Central to this approach is the concept of a 1-tree which consists of a tree on the vertex set $\{2,3,\dots,n\}$ together with two distinct edges at vertex 1. The transformation on "intercity distances" $c_{ij} \to c_{ij} + \pi_i + \pi_j$ leaves the traveling-salesman problem invariant but changes the minimum 1-tree; also a tour is a 1-tree in which each vertex has degree 2.
Clearly,

$$C + 2\sum_i \pi_i \geq \min_k [c_k + \sum_i \pi_i d_{ik}]$$

where C is the weight of a minimum tour with respect to (c_{ij}), k is a generic index for 1-trees, c_k is the weight of the k-th 1-tree with respect to (c_{ij}), and d_{ik} is the degree of vertex i in the k-th 1-tree. Setting

$$w(\Pi) = \min_k [c_k + \sum_i \pi_i (d_{ik} - 2)],$$

we see that $C \geq w(\Pi)$; furthermore, the "best" such lower bound on the solution of the traveling-salesman problem is given by $\max_\Pi w(\Pi)$.

Various approaches at achieving an ascent method for the computation of this maximum have been tried. The one that has been most successful is given by the following scheme:

(1) Set π_i equal to some initial value $\bar{\pi}_i$

(2) Find a 1-tree k of minimum weight with respect to $(c_{ij} + \pi_i + \pi_j)$

(3) $\pi_i \leftarrow \pi_i + (d_{ik} - 2)$

(4) GO TO 2.

The iteration is terminated when the maximum obtained value of $w(\Pi)$ fails to improve after a certain number of steps. Although convergence may not be obtained, this iteration scheme seems to work well in practice. It resembles the method of ficticious play for the solution of max-min game problems; one player choosing values of Π and the other player choosing k(1-trees).

The combined ascent and branch and bound method is designed to produce an optimum tour in all cases, even when $C > \max_{\Pi} w(\Pi)$. It may be outlined as follows: Let X and Y be disjoint sets of edges. Let T(X,Y) be the set of 1-trees which include all the edges in X and none of the edges in Y. Define

$$w_{X,Y}(\Pi) = \min_{k \in T(X,Y)} [c_k + \sum_i \pi_i (d_{ik} - 2)].$$

The state of the computation is given by a list, each of whose entries has the form $(X,Y,\Pi,w_{X,Y}(\Pi))$; $w_{X,Y}(\Pi)$ is called the "bound" of the entry. Initially the list contains the single entry $(\phi,\phi,0,w(0))$. At a general step, an entry $(X,Y,\Pi,w_{X,Y}(\Pi))$ of least bound is considered. The ascent iteration is applied starting at the point Π in an attempt to increase $w_{X,Y}(\Pi)$. If the attempt is successful, a new entry $(X,Y,\Pi',w_{X,Y}(\Pi'))$ such that $w_{X,Y}(\Pi')$ exceeds $w_{X,Y}(\Pi)$ by an integral amount is obtained, and the old entry is deleted. Otherwise it is determined if a direction of ascent exists. If so, the process "branches", i.e., the old entry is replaced by a collection of new entries each determined by specifying sets X_i, Y_i such that $X_i \supset X$, $Y_i \supset Y$ and $T(X,Y) = \cup T(X_i, Y_i)$.

If no direction of ascent exists, then a search is conducted to determine if T(X,Y) includes a tour. If so, the process terminates, otherwise, branching is performed.

To date the method has been tested on standard 20, 25, 42, and 48-city problems appearing in the literature. The results have been outstanding. The 20 and 25-city problems were solved using the ascent method alone, without any branching required. With branching, the 42 and 48-city problems were easily solved. In each case the solution obtained was proved to be optimum. Computational experiments on larger problems are now being performed.

ELI HELLERMAN and DENNIS RARICK, CEIR Professional Services Division of CDC, Washington.

Reinversion with the Preassigned Pivot Procedure (P^3)

Mathematical programming computer systems using the product form of the inverse (PFI) must periodically resort to a reinversion with the current basis in order to reduce the amount of work to be done in the succeeding iterations.

In this paper, we show the consequences of column, pivot selection and sequence upon the transformation vector (eta) density and give an algorithm (P^3) which will tend to minimize eta density and work done per iteration.

The algorithm has been implemented and tested as a replacement for the previous inversion algorithm on the OPTIMA system for the CDC 6000 computers. In performance it shows itself to be from six to ten times faster than the previous algorithm with a proportional reduction in work.

L. HELLINCKX and M. RIJCKAERT, Katholieke Universiteit Leuven Nederlands, Leuven.

A Solution Procedure for Geometric Programming Problems with degrees of Difficulty.

Geometric Programming is a very suitable method to solve highly non-linear programming problems. Instead of solving the original problem, GP solves an auxiliary problem, which yields the same optimal solution. This solution can be found easily, when the degree of difficulty of the problem is zero. This means that $T = N + 1$, where T is the total number of terms in the objective function and in the constraints, as formulated in the original problem and where N is the number of variables in the same problem.
However, when the degree of difficulty is not zero, as in most practical cases, the solution is not uniquely determined by the linear orthogonality and normality conditions. Passy and Wilde (1), recently developed non-linear conditions, called equilibrium conditions, which when added to the regular linear conditions, determine the solution to the auxiliary problem uniquely. Although the problem was theoretically solved, in practice however, the solution is hard to be found, because of lack of convergence in the solution scheme. This paper presents an algorithm that converges to the optimal point starting from an arbitrary initial point. The algorithm consists of 2 major parts. The first one is an iterative solution scheme for the set of linear orthogonality and normality conditions and the non-linear equilibrium conditions. The second part deals with the convergence problem, mainly arising from auxiliary variables becoming negative during the solution procedure. To obtain convergence, a relaxation of the original equations is used together with a phase I procedure of linear programming.

Although the starting point of the algorithm may be chosen arbitrary, it is evident that a rational choice of the starting point will decrease the computer time for the algorithm. Therefore the choice of a starting point will be discussed too.

Finally the method will be demonstrated on a highly non-linear example, involving the optimal design of an oxygen production plant.

(1) U. Passy and D.J. Wilde, Mass Action and Polynomical Optimization. Journal of Engineering Mathematics, VOL. 3, No. 4, October 1969, Wolters-Noordhoff, Groningen, The Netherlands.

T.C. HU, University of Wisconsin, Madison.

Discrete Decision Tree.

Many problems in operations research reduce to the following model. A set Ω consisting of n objects is given. One object from Ω is chosen. The ith object has a known probability p_i of being chosen. A set of tests $T_j (j = 1,\dots,m)$ are used to identify the object. A test T_j can decide if the object belongs to one subset of Ω or to its complement. The problem is to find an optimum sequence of tests which minimize the expected cost.

When the cost c_j of performing T_j are the same, and $m = 2^{n-1}-1$ the problem becomes that of constructing an optimum variable-length binary code. This was solved by Huffman [1]. When c_j are the same, and m = n-1, the problem becomes that of constructing an optimum variable-length alphabetical code and was solved by Gilbert and Moore [2]. The present paper gives a better algorithm than [2] and considers some extensions.

[1] D.A. Huffman: "A Method for the Constructing of Minimum-Redundancy Codes", Proc. I.R.E., 40, Sept. 1952, p. 1098-1101.

[2] E.N. Gilbert and E.F. Moore, "Variable-length Binary Encodings", The Bell System Technical Journal, 38, July 1959, pp. 933-968.

WILLIAM S. JEWELL, University of California and TECHNEKRON Inc., Berkeley.

Branch-and-Bounding the Simplex.

In many respects, the ordinary linear program resembles a discrete optimization problem more than a continuous one. For example, the simplex method may be considered as a discrete decision problem in which a known number of activities are to be selected as basic, the remaining activities then being non-basic. Once this arbitration is accomplished, the actual levels of the activities in an optimal solution are uniquely determined.

This paper explores the use of implicit enumeration techniques, such as branch-and-bound (progressive separation and evaluation), backtracking, etc. to solve ordinary linear programs. The resulting solution procedures are not computationally competitive with modern computer-oriented algorithms, such as the revised simplex. However, this approach provides interesting pedagogical insights into the complementary nature of the primal and dual decisions, and demonstrates how "extra" information from the real problem can be advantageously used to influence and organize a loose and unstructured solution framework.

Among the freedoms open to the optimizer are adaptive choices of: which activities to arbitrate first; the order in which to price out resources; the amount of work to be expended in making the partial solution bounds (evaluators) tighter; and, the choice of when to accept an approximate solution. One can even "surrender" at any point, and switch to any ordinary simplex method.

This unusual approach to a well-solved problem also gives insights into the workings of implicit enumeration techniques, by demonstrating the additional leverage obtained through duality and orthogonality. This approach is also a natural starting point for discussing the inclusion of discrete variables into a mixed program.

After presenting the basic ideas of the B&B Simplex approach, various computational details on the separation and evaluation of partial solutions will be explored. The application of this framework to special problems in network theory will be shown, followed by a discussion of relationships with pivot theory and integer programming.

K. KAPUR, General Motors Research Laboratories, Warren, Mich., and
R.M. VAN SLYKE, Network Analysis Corporation, Glen Cove, N.Y.

A Cutting Plane Algorithm in Function Space with Application to Optimal Control Problems.

Cheney-Goldstein and Kelley independently proposed a cutting plane algorithm for convex programming. The problem they consider is

(1) $$\text{Minimize } \{c(u) \mid g(u) \leq 0,\ u \in S\}$$

where $u \in E^n$, $c(u)$ is a linear function of u. $S = \{u \mid A\,u \leq b\}$ is assumed to be a bounded polyhedral set, g is a continuous vector valued functional with uniformly bounded derivative. For this problem the cutting plane algorithm converges on at least a subsequence to an optimal solution (assuming the problem is feasible). The proof depends strongly on the compactness of S and the uniform bound on the derivative.

The algorithm and the convergence proof can be generalized directly to apply to the case where u belongs to Banach space U, and g(u) has a Frechet differential which is uniformly bounded on a compact subset S of U. However, in most applications of interest it is unrealistic to assume that S is strongly compact;usually weak compactness is the best one can hope for. Thus, we consider (1) in the situation where c(u) is a lower semi-continuous convex functional, g(u) a lower semi-continuous convex functional, and S is a convex, weakly sequentially compact subset.

Since in many applications a Frechet derivative is not available we use the more general sub-differential. $\partial g(u^0)$, the subgradient of g at u^0 is given by

$\partial g(u^0) = \{g^* \in U^* \mid g(u) \geq g(u^0) + < g^*, u - u^0 > \text{ for all } u \in U\}$.

The cutting plane algorithm for this problem is then

Step 0. Solve: Minimize $\{c(u) \mid u \in S\}$,

call the solution u^0; let k = 0. Go to Step 1.

Step 1: If $g(u^k) \leq 0$, u^k is a solution of (1). If not add the inequality

(2) $< g^{*k}, u > \leq < g^{*k}, u^k > - g(u^k)$ to (3) and go to

Step 2, where $g^{*k} \in \partial g(u^k)$.

Step 2:

Solve

(3) Minimize $\{c(u) \mid e \in S, < g^{*j}, u > \leq < g^{*j}, u^j > - g(u^j)$, $j = 0,\ldots,k\}$ obtaining as an optimal solution u^{k+1}, set k = k+1 and go to Step 1.

Note that the efficiency of the above algorithm depends on our ability to solve (3) significantly more easily than we can solve (1).

Theorem 1: Suppose in addition to the previous hypotheses for (1) that

$$\{g^{*k}\}_{k=1}^{\infty}$$

is relatively compact in the strong topology. Then the sequence $\{u^k\}$ of points generated by the algorithm contains a weakly convergent subsequence and for any weakly convergent subsequence $\{u^{k_i}\}$ converging to u, say, u solves (1).

A convex function c(u) defined on a convex subset K of a Banach space U is uniformly convex if there exists a monotone function $\delta(\tau)$ on $_1U(0,\infty)$ with $\delta_1(\tau) > 0$ for $\tau > 0$ such that for all $u^1 \in K$,

$u^2 \in K$ with $u^1 \neq u^2$ there exists $\lambda \in (0,1)$ such that

$$(1-\lambda)\ c(u^2) + \lambda\ c(u^2) \geq c((1-\lambda)u^1 + \lambda\ u^2) + \delta(||u^1 - u^2||).$$

Theorem 2: If in addition to the hypotheses of Theorem 1, $c(u)$ is uniformly convex then u^k converges strongly to u^0 which corresponds to the unique optimal solution of (1).

Another consequence of uniform continuity is that we do not need to keep all the added constraints in the algorithm. This is important for computational efficiency. This last result is a straight forward generalization of the corresponding result in finite dimensional spaces due to Topkis.

The cutting plane algorithm we have discussed is applied to a linear optimal control problem with state space constraints, where the problems solved in Step 0 and Step 2 of the algorithm become optimal control problems without state space constraints. Thus we have a method of solving state space constrained problems by solving a sequence of problems without state space constraints.

S. KARAMARDIAN, University of California, Irvine.

The Complementary Problem.

The general complementary problem is a problem of the following form: Given a map F from E^n into itself find a vector x such that $x \geq 0$, $F(x) \geq 0$, $x^T F(x) = 0$. A very important special case is the linear complementary problem where $F(x) = Mx + b$. The importance of these problems lies in the fact that they arise in a variety of fields such as mathematical programming, game theory, mechanics, and geometry.

This paper deals mainly with the question of the existence of solutions to both the linear and nonlinear complementary problem.

For the nonlinear case both nondifferentiable and differentiable F are considered. Several existence theorems are established which generalize previous results. In particular it is shown that the general complementary problem has a solution if the map $G(x) = F(x) - F(0)$ is continuous, positively homogeneous of some degree, and its Jacobian is a P-matrix i.e., all its principal minors are positive. For the linear case a new class of matrices called proper matrices is introduced. It is shown that if M is proper then the linear complementary problem has a solution for any b. The class of proper matrices includes as proper sub-classes all other classes for which it is known that the linear complementary problem has a solution for any b. In particular it includes the class of P-matrices, the class of strictly co-positive matrices, and the class of semi-monotone matrices. This class has the added two features:

(1) it can be characterized by its minors, and

(2) it may possess negative elements on its diagonal, a property which is lacked by all previously considered classes. Several examples of proper matrices are provided.

LASLO BELA KOVACS, Hungarian Academy of Sciences, Budapest.

An Algorithm for the Solution of an Integer Programming Problem by Dynamic Programming.

The problem is the following:

$$\min \underline{c}^T \underline{x}$$

$$A\underline{x} \geq \underline{b}$$

$$x_j = 0,1,2,\ldots \quad (j = 1,\ldots,n)$$

where A is an m×n matrix and $\underline{c} > \underline{0}, \underline{b}$ are vectors of corresponding size.

This problem is solved by an algorithm of dynamic programming type. In order to decrease the memory requirement the principle of branch and bound is also used.

For the case m=1 the requirement $\underline{c} > \underline{0}$ is not necessary. In this case the solution of the problem is given for any $\underline{b} > \underline{0}$. The solution of the general problem for any $\underline{b} > \underline{0}$ only if there exist a variable x_r and positive numbers k_j $(j = 1,\ldots,n)$ such that

$$k_j c_r \le c_j, \; k_j a_{ir} \ge a_{ij} \quad (i=1,\ldots,m; \; j=1,\ldots,n).$$

In order to avoid duplication of solutions a special grouping of solutions is introduced, which at the same time makes the use of branch and bound principle possible.

C.B. KRABEK, Control Data Sweden AB, Stockholm.

An Algorithm for the Solution of the Fixed Charge Network Flow Problem.

The problem treated here is that of finding the minimum total cost solution to a minimum cost network flow problem where some or all of the arcs may also have fixed charges associated with them.

The algorithm proposed is a combination of two implicit enumeration algorithms. The first, the "foreward" algorithm opens arcs one at a time and generates lower bounds for the total fixed cost of the optimum solution. The second algorithm, the "backward" algorithm closes arcs one at a time to generate lower bounds for the total variable cost of the optimum solution. The lower bounds on the variable cost developed by the "backward" algorithm are used by the "foreward" algorithm in branch trimming in the enumeration scheme and the lower bounds on the fixed charge developed in the "foreward" algorithm are used in the "backward" algorithm for branch trimming.

These and other branch trimming rules are explored in detail and a description of an existing large scale computer code based on the algorithm is given, along with computational results.

The algorithm has the advantage that no non integer solutions are obtained, that each potential feasible solution is examined at most once, and that a lower bound on the total cost is available at all times for use in selecting near optimal solutions.

J. KRARUP, The Technical University of Denmark, Lyngby.

The Relationship between Outerplanar Graphs and a Class of Discrete Optimization Problems.

This paper is a result of a discussion with professor Frank Harary, University of Michigan, Ann Arbor. We met in November 1969 and Hararay drew my attention to the concept of outerplanar graphs but knew of no "real life" applications. I missed at that time a way to characterize a certain class of graphs related to coordination of traffic signals. It turned out that my question answered his and vice versa.

Given a connected graph $G = (N,A)$ consisting of a finite set N of vertices together with a set A of unordered pairs of distinct elements of N. Each pair $(i,j) \in A$ constitutes an edge in G. Let $n = |N|$ and $a = |A|$ denote the cardinality (the number of distinct elements) of N and A respectively.

A set of variables, $x_i, \forall i \in N$, is associated with the vertices of G and $f_{ij}(x_i,x_j), \forall (i,j) \in A$ is a given set of real-valued functions associated with the edges of G and bounded below.

Problem: Choose $x = (x_1,x_2,\dots,x_n)$ so as to minimize $\sum_A f_{ij}(x_i,x_j)$

subject to $\forall i \in N$: $x_i \in Q$ where Q is a finite set with cardinality $q = |Q|$.

To formulate and solve this is terms of dynamic programming is straightforward. The number of computations, however, ranges from $(n-1)q^2$ if G is a tree to q^n if G is complete. It will be demonstrated,that the number of computations can be written as $c_1q^2 + c_2q^3$ where $c_1 \leq n-1$ and $c_2 \leq n-2$ if and only if G is homeomorphic with an outerplanar graph.

In the traffic signal coordination problem, Q is a set of consecutive integers and $f_{ij}(x_i,x_j)$ depends solely on (x_i-x_j). Thus, if G is homeomorphic with an outerplanar graph, the number of computations amounts to $c_1q + c_2q^2$.

P.D. KROLAK, W. FELTS and G. MARBLE, Vanderbilt University, Nashville, Tenn.

Heuristics for Solving the Traveling Salesman Problem and Various Related Scheduling Problems.

The traveling salesman problem is a well known scheduling problem. In this paper we will describe a heuristic procedure for solving large scale traveling salesman problems. This heuristic differs form other heuristics in that (1) it exploits the relationship of the traveling salesman problem and the related assignment problem, and (2) its performance does not decrease rapidly with an increase in problem size. This heuristic has been found to solve 100-200 city problems in a fraction of the time of currently reported procedures. When this heuristic is further combined into a real time computer system, the result is a highly reliable and economical method of solving very large problems. The paper will describe extensions of the above approach to solving several major related problems including the coin collector, (the salesman must return home after every so many stops), the bottleneck problem, and the traveling salesman in time.

The traveling salesman in time is a new generalization of the original problem. The salesman's reward is based not only on the cost of travel but also on how many clients are available to see him on the day of his arrival in town. This means that the circuit he makes will be closed in space but not in time. An exact algorithm for solving this problem will be given.

T.O.M. KRONSJO, University of Birmingham, Birmingham. send him my GBD Paper

Theory of Decomposition of a Large Nonlinear Convex Separable Programme in the Dual Direction.

A decomposition method is developed for the solution of the problem:

$$\min_{x_1, x_2} \{f^1(x_1) + f^2(x_2) \mid g^1(x_1) + g^2(x_2) \le 0,\ x_1 \in C_1\}$$

where super- and sub-script denote row and column vector (scalar), respectively, and $f^i(x_i)$ and $g^i(x_i)$, $(i = 1,2)$, denote convex scalar and vector functions, respectively.

The method describes the total cost of the problem as a function of the x_1 variables, for any x_1 value assuming the corresponding optimal x_2 values, i.e. by the function:

$$f^1(x_1) + \min_{x_2} \{f^2(x_2) \mid g^1(x_1) + g^2(x_2) \le 0\} \mid x_1 \in C_1\}$$

and then finds the value x_1 which minimizes the above function. Instead of using the exact function an approximating function is derived based upon the solution(s) of the subproblem of minimizing the function of x_2.

Upper and lower bounds upon the optimal value of the objective function, and a proof of convergence are derived.

The method may be seen as a nonlinear generalization of a decomposition method by J.F. Benders. It may be interpreted as a dialogue between two organisations. The first one minimizes an approximation of the total costs and informs the second one about the resulting requirements. The second one minimizes the cost of meeting these requirements and informs the first one about the actual cost and the marginal costs of each requirement. The first one uses this information to form an improved approximation of the total cost function.

H. KWAKERNAAK and R.C.W. STRIJBOS, Delft University of Technology, Delft.

Extremization of Functions with Equality Constraints.

The problem considered is that of minimizing the function $\alpha(\underline{x})$ subject to the constraint $\underline{g}(\underline{x}) = \underline{0}$ where $\underline{x} = \text{col}(x_1,\ldots,x_n)$ and $\underline{g} = \text{col}(g_1,\ldots,g_m)$ with $m < n$. By introducing the vector of Lagrange multipliers $\underline{\lambda} = \text{col}(\lambda_1,\ldots,\lambda_m)$ the necessary conditions for a stationary point of the constrained minimization problem may be expressed as

$$\begin{aligned} \underline{h}(\underline{x},\underline{\lambda}) &= \text{grad}[\alpha(\underline{x})] + \underline{J}_g^T(\underline{x})\underline{\lambda} = \underline{0} \\ \underline{g}(\underline{x}) &= \underline{0}\ . \end{aligned} \qquad (1)$$

Here $\underline{J}_g(\underline{x})$ is the Jacobian of the vector function $\underline{g}$. In this manner the solution of the minimization problem is converted to the solution of a set of (n+m) non-linear equations in the (n+m) unknowns $\underline{x}$ and $\underline{\lambda}$. In order to evaluate these equations the matrix $\underline{J}_g(\underline{x})$ and the vector $\text{grad}[\alpha(\underline{x})]$ are assumed to be explicity available for all $\underline{x}$.

To find the solution of the set of equations (1) by Newton's method the inverse of the Jacobian of these equations is required.

This Jacobian is given

$$\underline{J}(\underline{x},\underline{\lambda}) = \left(\begin{array}{c|c} \underline{G}(\underline{x},\underline{\lambda}) & \underline{J}_g^T(\underline{x}) \\ \hline \underline{J}_g(\underline{x}) & \underline{0} \end{array}\right)$$

where $\underline{G}(\underline{x},\underline{\lambda})$ is the Jacobian of $\underline{h}(\underline{x},\underline{\lambda})$ with respect to x. By using the formula for the inverse of a partitioned matrix it is possible to evaluate $\underline{J}^{-1}(\underline{x},\underline{\lambda})$ provided the $(n{\times}n)$ matrix $\underline{G}^{-1}(\underline{x},\underline{\lambda})$ is known. To avoid the repeated computation of $\underline{G}^{-1}$ an initial guess $\underline{G}^{-1}$ is made, which is updated during the course of the computations using compact and efficient formulas. The resulting method has the property that if the function $\alpha(\underline{x})$ is quadratic and the constraint functions $\underline{g}(\underline{x})$ are linear in $\underline{x}$, the exact solution of (1) is obtained at the end of n iterations.

Further considerations should point out whether the stationary point found is an extremum or not.

The proposed method has been tested on several examples and compared with a sequential unconstrained minimization technique proposed by Powell.

The paper also presents compact formulas for the solution of simultaneous nonlinear equations by the quasi-Newton method.

P.O. LINDBERG, The Royal Institute of Technology, Stockholm.

Planning of transportation with stochastic production and demand.

The usual operations research transportation problem is an approximation of the real world transportation problem. One simplifying assumption is that demand and production is known in advance.

When random fluctuations are present you run the risk of getting short of goods or not getting everything sold. To put a value to not used goods you have to solve the planning problem for several periods. In this way you run into a multistage stochastic decision problem. This latter problem can in principle be solved by dynamic programming techniques, but as usual you get a formidable state space which prohibites a practical solution.

I propose a method which lies between the deterministic and the multistage one:

For a given planning period you decide for a fixed transportation program. You evaluate the different programs by averaging over all possible outcomes during the planning period. The programs are made feasible for all outcomes by allowing backordering. Negative inventory is punished with a shortage cost. The problem of finding a program with optimal average performance is shown to be a network flow problem with convex cost in some arcs. This problem can be attacked by different algorithms.

When you have your optimal program you use the first period part of it as your action and then resolve the problem next period.

This problem was studied with transportation of empty containers and railway cars in mind. For these, ordering cost is negligable and shortage cost is much higher than transportation cost. By the time of the symposium computational results for problems with these characteristics will be available. For problems with very few places of production / consumption a comparison with the dynamic programming solution will have been made.

F.A. LOOTSMA, Philips Research Laboratories, Eindhoven.

Boundary Properties of Penalty Functions for Constrained Minimization.

Penalty-function techniques are designed to take into account the constraints of a minimization problem or, since almost none of the problems arising in practice have interior minima, to approach the boundary of the constraint set in a specifically controlled manner. The presentation starts therefore by a classification of penalty functions according to their behaviour in the neighbourhood of that boundary.

Appropriate convexity and differentiability conditions are imposed on the problem under consideration. Furthermore, certain uniqueness conditions involving the Jacobian matrix of the Kuhn--Tucker relations are satisfied by assumption. This implies that the problem has a unique minimum $\bar{x}$ with a unique vector $\bar{u}$ of associated Lagrangian multipliers.

Under these conditions the minimizing trajectory generated by a mixed penalty-function technique can be expanded in a Taylor series about $(\bar{x},\bar{u})$. This provides, as an important numerical application, a basis for extrapolation towards $(\bar{x},\bar{u})$. The series expansion is always one in terms of the controlling parameter, independently of the behaviour of the mixed penalty function at the boundary of the constraint set.

Next, there is the intriguing question of whether some penalty functions are easier or harder to minimize than other ones. Accordingly, the condition number of the principal Hessian matrix of a penalty function is studied. It comes out that the condition number varies with the inverse of the controlling parameter, independently of the behaviour of the mixed penalty function at the boundary of the constraint set.

The parametric penalty-function techniques just named can be modified into methods which do not explicitly operate with a controlling parameter. These "parameter-free versions", which are based on moving truncations of the constraint set, may be considered as penalty-function techniques adjusting the controlling parameter automatically. The crucial point is the efficiency of the adjustment. It is established how the rate of convergence depends on the vector $\bar{u}$ of Lagrangian multipliers associated with $\bar{x}$, on the boundary properties of a penalty function, on a weight factor p attached to the objective function, and on a relaxation factor ρ. The method of centres is a remarkable exception: its rate of convergence depends on the number of active constraints at $\bar{x}$, and on p and ρ.

HARRY R. LOVE and MICHAEL J.L. KIRBY, Dalhousie University, Halifax, N.S.

Extreme Point Mathematical Programming Models.

The paper considers a class of optimization problems. The problems are linear programming problems (maximize $\underline{c}\ \underline{x}$ subject to $A\underline{x} = \underline{b}$) with the additional constraint that $\underline{x}$ must also be an extreme point of a second convex polyhedron $D\underline{x} = \underline{d}$, $\underline{x} \geq \underline{0}$. A cutting-plane algorithm for solving such problems is presented. Two examples to demonstrate the applicability of the algorithm are included.

CHRISTOPH MAIER-ROTHE and MARTIN F. STANKARD, Jr., Arthur D. Little Inc., Cambridge, Mass.

A Linear Programming Approach to choosing between Multi-Objective Alternatives.

A common difficulty in decisionmaking is a choice between two multiobjective alternatives. Each alternative results in a vector of values, yet neither alternative surpasses the other on all criteria simultaneously. The decisionmaker wants to select only one alternative. This paper presents a procedure for collecting simple preference data from a decisionmaker and for using this data to reveal the decisionmaker's preference between the two multi-objective alternatives.

The procedure presents a series of hypothetical pairwise tradeoff questions to the decisionmaker. The answers to these questions imply linear constraints upon weights applied to the values of each objective. Two linear programming formulations test whether one alternative's weighted objectives has a larger value then the other's for all sets of weights allowed by the inequality constraints. If neither alternative is preferred to the other for all feasible weightings the procedure generates further questions so that fast convergence to a unique choice is achieved.

An analysis of the dual linear program produces these further questions so that their answers are most likely to force the choice of one of the alternatives. The dual formulation of the original linear programs have a very simple structure and the possibility of using a special simplified solution algorithm on the dual is discussed. A numerical example is also provided for illustration of the procedure.

ANTAL MAJTHAY, Hungarian Academy of Sciences, Budapest.

Some Remarks on Nonconvex Quadratic Programming.

The problem we are concerned with is that of finding the global minimum of a general quadratic function $c^Tx + x^TQx$ subject to the linear inequality constraints $Ax \geq b$, $x \geq 0$, or showing that none exists.

The only rigorous algorithm is Ritter's method [2], which works in three phases. Each of these phases can return many times during the procedure. In the first phase we are seeking a feasible point if there is any. In the second phase starting from this point we are looking for a local minimum or an unbounded solution, while in the third phase we construct a cutting plane which excludes the previously located local minimum without excluding the global minimum if it has not yet been found. After the cutting plane has been placed we apply the first phase to the augmented problem.

Ritter proved that if the set of feasible solutions is bounded, the algorithm cycles through the three phases only a finite number of times.

In their excellent paper it is shown by Cottle and Mylander [1] that there is a strong relationship between complementary pivot theory and Ritter's algorithm.

Following Cottle's presentation of the algorithm and using combinatorial methods only we show that a slight refinement of the algorithm is finite in the case of an unbounded feasible set as well.

[1] R.W. COTTLE and W.C. MYLANDER, Ritter's Cutting Plane Method for Nonconvex Quadratic Programming, Stanford TR 69-11.

[2] K. RITTER, A Method for Solving Maximum Problems with a Nonconcave Quadratic Objective Function, Z.Wahrscheinlichkeitstheorie verw.Geb. 4., /1966/, 340-351.

O.L. MANGASARIAN, University of Wisconsin, Madison, and
L.L. SCHUMAKER, University of Texas, Austin.

Discrete Splines via Mathematical Programming.

Mathematical programming is used to investigate constrained minimization problems in real Euclidean spaces which are the discrete analogs of spline problems posed as minimization problems in a Banach space. Existence, uniqueness, characterizing properties and computational methods are given.

B. MARTINET, Centre d'Etudes Nucléaires de Grenoble, Grenoble.

A General Algorithm for an Approximate Solution of a Game by using a Dual Method.

a) dual methods for the resolution of an optimisation problem with constraints, see [1] and [2].

We are dealing with a generalisation of cut-off and penalty functions methods. Let us consider the problem:

$$(1) \qquad \min_x \left(f(x) \mid g_j(x) \le 0, \forall_j \in J\right) = f(\overline{x}).$$

The proposed method is iterative: for instance, for each step n we have the unconstrained problem:

$$(2) \qquad \min_x \left(f(x) + \varphi_n(x)\right),$$

where φ_n is a partial constraint penalisation.

Let x^n be the point where the min (z) is reached. We go from n to (n+1) by adding one of "the most unsatisfied constraints"in x^n.

If the set in which we want to minimize is empty then:

$$\lim_{n\to\infty} f(x^n) + \varphi_n(x^n) = +\infty,$$

otherwise x^n converges to $\overline{x}$.

The interesting points of these methods are the following:

- a great mathematical generality,
- J may be any set (for instance infinite, which is important for the next application),
- if f is uniformly convex and g convex in x, we will be able to suppress the non-active constraints at each iteration.

a,a) a general method for the resolution of a strategic game, see [3]. Let us consider a min-max problem

$$\min_{x\in C} \max_{y\in D} J(x,y) = v = J(\overline{x},\overline{y}).$$

Let:

$E(a) = \{x \in C \mid J(x,y) \geq a; \forall y \in D\}$

$F(a) = \{y \in D \mid J(x,y) \leq a; \forall x \in C\}$

be two sets defined in general by an <u>infinity</u> of constraints.
We have:

$v = \max (a \mid F(a) \neq \emptyset) = \min (a \mid E(a) \neq \emptyset)$.

We are going to use a dual method in order to find if E(a) (or F(a)) is empty; if not we find an approximate element of the sets. Whence an iterative method which enables us to build $\{a^n\}$ such as $\lim_{n\to\infty} a^n = v$ and $\{(x^n,y^n)\}$: sequence of ε-strategies.
Under convenient hypotheses (x^n,y^n) converges to $(\overline{x},\overline{y})$.
We can apply the methods to very general cases such as: infinite games, differential games, n-person games, etc.

[1] P.J. LAURENT et B. MARTINET, Méthodes duales pour le calcul du minimum d'une fonction convexe sur une intersection de convexes, Colloque d'optimisation, Nice 1969.

[2] B. MARTINET, Méthodes duales pour la résolution approchée d'un problème d'optimisation, Seminaire d'analyse numérique, Faculté des Siences de Grenoble, décembre 1969.

[3] B. MARTINET, Algorithme de résolution approchée d'un jeu, Colloque d'analyse numérique de Super-Besse, France, juin 1970.

BELA MARTOS, Hungarian Academy of Sciences, Budapest.

Subdefinite Matrices and Quadratic Programming.

Many algortihms are known to solve the minimization of a pseudoconvex objective function subject to linear constraints. Within the problem of quadratic programming this fact directs our attention to quasiconvex and pseudoconvex quadratic functions. The notions introduced here enable us to characterize and recognize such functions.

A $n \times n$ real symmetric matrix C is called positive subdefinite, if for any $v \in E^n$, $v'Cv < 0$ implies that the n-vector Cv is either semipositive or seminegative. Positive semidefiniteness implies positive subdefiniteness but not conversely. C is positive subdefinite without being pos. semidefinite if and only if all its entries are nonpositive and the matrix has exactly one negative eigenvalue. The quadratic form $Q(x) = x'Cx$ is quasiconvex in the nonnegative orthant if and only if C is positive subdefinite. Q(x) is pseudoconvex in the semipositive orthant if and only if C is positive subdefinite and either positive semidefinite or free of 0-rows.

The quadratic function $\varphi(x) = \frac{1}{2} x'Cx + p'x$ is quasiconvex in the nonnegative orthant if and only if for any $v \in E^n$, $v'Cv < 0$ implies that the (n + 1)-vector $\begin{bmatrix} Cv \\ p'v \end{bmatrix}$ is semipositive or seminegative.

Otherwise: $\varphi(x)$ is quasiconvex in the nonnegative orthant if and only if C is positive subdefinite, and either it is positive semidefinite or the vector p satisfies the following two conditions: a) $p \leq 0$, b) there is some $q \in E^n$, such that $p = Cq$ and $p'q \leq 0$. If $\varphi(x)$ is quasiconvex in the nonnegative orthant and C is free of 0-rows, then $\varphi(x)$ is pseudoconvex in the semipositive orthant.

The above results enable us to find the minimum of a quasiconvex quadratic function in a polyhedral subset of the nonnegative orthant by any known method of general pseudoconvex programming (Frank-Wolfe, Rosen, Zoutendijk, etc.). None of the special convex quadratic programming algorithms, which guarantee finiteness, has been amenable to a similar extension as yet.

G. MENSCH, Tulane University, New Orleans.

Single-stage Linear Programming zero-one Solutions to some job-machine type Problems.

Scheduling problems can be formulated as mathematical programs with integrity requirement on some variables. In this paper some properties of linearly solvable mixed integer programs are discussed, and experiments are reported where integer solutions have been obtained by linear programming.

G. MITRA, S.I.A., London.

Designing Branch-and-Bound Algorithms for Mathematical Programming.

The theory of Branch-and-Bound technique has not to date been rigorously formalized: this is because the concept of splitting a problem into solvable and mutually exclusive subproblems has proved to be successful in varied contexts; the justification of the theory plays a less important role. An important property associated with the problems solved by Branch-and-Bound technique, however, should be stressed: that is the subproblems proposed and solved by this technique always possess global bounds and hence propose similar bounds on the original problem. In travelling salesman problem

this bound is obtained by canonical reduction [1]. In all the other known applications of this technique [2], [3], [4] the sub-problems are known convex mathematical programming problems.
The present paper is not just a survey of the work done by other workers. The advantage of computing penalties to obtain variables to branch on are questioned and the experience of using a priority scheme for choosing branching variables is described. Computational aspects of using dual and parametric RHS for the solution of sub-problems are outlined.
In general branching strategy is a combination of two level decisions:

1) which subproblem to solve next,
2) which variable to branch on within the subproblem.

Using combination of the above experiments of tree search for fixed charge problems are described. A technique of constructing feasible solution at every node of fixed charge problem is outlined.
Finally, a branch-and-bound method for solving mixed integer quadratic programs is described. Such a technique is easily designed on the basis of the properties of convex mathematical programming problems and finds use in solving Media Scheduling Problem.

[1] LITTLE, John, D.C., MURTY, KATTA G., et al, An Algorithm for the Travelling Salesman Problem. Opern. Res., (1963)(972-989).

[2] DAKIN, R.J., A Tree Search Algorithm for Mixed Integer Programming Problems, Computer Journal, 8 (1965), 250-255.

[3] BEALE, E.M.L., TOMLIN, J.A., Special Facilities in a General Mathematical Programming System for Non-Convex Problems using ordered sets of variables. Presented to International Conference in O.R.,VBNICR,1969.

[4] MITRA, G., A Dichotomizing Procedure for the Fixed Charge Problem, Presented to SIAM conference in Optimization, Toronto,1968.

HEINER MULLER-MERBACH, Universität Mainz, Mayence.

Approximation Methods for Integer Programming.

The application of exact integer programming (IP) methods are restricted (and seem to remain restricted) to small problems. Therefore, there is no chance to solve large IP problems by any other method than by approximation methods.

A survey on the existing approximation methods shall be given in the paper. The emphasis will be put on the basic principles of the algorithms, on the numerical experiences, and on the application to problems of different structures.

Method 1 (Method of "cautious approach" [3]):
In each iteration one small step (at least of the size 1) is carried out towards the direction of the "greatest change".

Method 2 [1]:
A different criterion is used to find a path of small steps from the origin to a near-optimal solution. Some special exploration is carried out in the neighbourhood of the solution.

Method 3 [2]:
Combinations of variables are considered in order to improve a solution.

The numerical experiences are encouraging.

[1] ROBERT E. ECHOLS, LEON COOPER, Solution of Integer Linear Programming Problems by Direct Search. Journal of the ACM, vol. 15 (1968) no. 1, pp. 75-84.

[2] HEINZ KREUZBERGER, Ein Näherungsverfahren zur Bestimmung ganzzahliger Lösungen bei linearen Optimierungsproblemen. Ablauf- und Planungsforschung, vol. 9 (1968) no. 3, pp. 137-152.

[3] HEINER MULLER-MERBACH, Das Verfahren der "vorsichtigen Annäherung" - Eine heuristische Methode zur Lösung gewisser Probleme der ganzzahligen Planungsrechnung. Elektronische Datenverarbeitung, vol. 11 (1969) no. 12, pp. 564-566.

HEINER MULLER-MERBACH, Universität Mainz, Mayence.

Separable Programming using the Upper Bounding Technique

In optimization problems there may occur separable nonlinear functions. HADLEY [1] and other authors have described two methods (the λ-form and the δ-form) by which these functions can be approximated by sets of linear functions. Each of these linear functions is valid within a certain interval. Thus the problem converts into a linear optimization problem and can be solved by means of a slightly modified simplex method.

A disadvantage of the λ-form and of the δ-form is the largely increased number of variables.

In this paper it will be shown that separable programming does not require additional variables at all, if the upper bounding technique is used to handle the borders of the intervals.

The main parts of the algorithm are:

1. Define the initial linear approximation of each function (including the border of the current interval as an upper bound).
2. Apply the simplex method (including the upper bounding technique). If any border is reached, → 3.
3. Redefine the approximations of the very functions (including the border) whose border is reached, → 2.

In addition to the saving of variables there is a second advantage of this method: The intervals of approximation of the single functions need not be established before the computation; they may be defined during the computation. This makes the method more flexible.

Numerical experiences [2] will be reported.

[1] G. HADLEY, Nonlinear and Dynamic Programming. Reading, Mass. 1964.

[2] H. MULLER-MERBACH, Separable Programming mit Hilfe der Upper-Bounding-Technique. Unternehmensforschung, vol. 14 (1970) no. 3, to appear.

W. MURRAY, National Physical Laboratory, Teddington.

An algorithm to find a local minimum of an indefinite quadratic program.

We present an algorithm to find a local minimum of the program

$$\min : F(x) = \tfrac{1}{2}x^TQx + f^Tx$$

$$\text{subject to } \bar{A}x \geq \bar{b}$$

where Q is not restricted to be either semi-positive definite or non-singular. The algorithm is based on the ability to recur from iteration to iteration the following matrices:

L : An m × m lower triangular matrix such that

$$A^T = [L \mid 0]\, P^T$$

where A is an n × m matrix of the coefficients of the currently active constraints and P is the product of m Householder transformation matrices.

Z : An $n \times (n-m)$ matrix such that

$$A^T Z = 0$$

$$Z^T Q Z = D \text{ a diagonal matrix.}$$

The algorithm has the following features.

i) It starts from a feasible approximation and all subsequent approximations are feasible.

ii) Provided F(x) is bounded below, the algorithm will converge in a finite number of iterations.

iii) $F(x_{k+1}) \leq F(x_k)$ where x_k is the k^{th} approximation.

iv) Each iteration takes $0(n^2 + m^2)$ operations where n is the number of variables and $m(\leq n)$ the number of active constraints.

v) In addition to the storage needed to define the problem we require $\frac{m^2}{2} + n^2 - nm$ locations.

K.G. MURTY, The University of Michigan, Ann Arbor.

A fundamental Problem in Linear Inequalities with application to the Travelling Salesman Problem.

All linear inequalities can be transformed into linear equations in nonnegative variables by introducing the necessary slack variables. Without any loss of generality we therefore consider a system of m linearly independent equality constraints in n nonnegative variables.

$$A\,x = b \tag{1}$$

$$x \geq 0. \tag{2}$$

The j-th column vector of the matrix A will be denoted by $A_{.j}$. The fundamental problem that we discuss is the following: suppose

we are given a set of r linearly independent column vectors of A, known as *the special column vectors*.
The problem is to develop an efficient algorithm to determine whether there exists a feasible basis for (1), (2) which contains all the special column vectors as basic column vectors and to find such a basis if one exists.

Such an algorithm has several applications in the area of mathematical programming. As an illustration, we show that the famous travelling salesman problem can be solved efficiently using this algorithm. Recent published work indicates that this algorithm has applications in integer linear programming.

An algorithm for this problem will be discussed in another paper to follow.

T.A.J. NICHOLSON and R.D. PULLEN, London Graduate School of Business Studies, London.

An inlet-control system for multi-stage sequencing.

Perhaps the most important area of application for sequencing techniques is in the preparation of production schedules. The production scheduling problem is to determine the times at which a number of jobs can be processed on a series of machines. Each job consists of a sequence of operations which must take place in turn, possibly on alternative machines, and the problem is to get the jobs completed by their due dates and keep down work in progress. If there are N jobs and y_j is the completion time of job j, t_j is its due time and e_j is the earliest time at which it could be completed, a useful lateness measure to be minimized which stresses the reduction of large lateness is

$$F(y_1, y_2 \dots y_N) = \sum_{j=1}^{N} \left(\frac{y_j - e_j}{t_j - e_j}\right)^2 .$$

The y_j quantities cannot themselves be controlled but depend on the series of start times of the successive operations and their durations, and the start times are restricted so that the total machine requirements do not exceed the capacities available. The capacity constraint prevents the use of linear or non-linear programming. The problem has previously been tackled by machine priority rules which give answers which are far from optimal or by permutation procedures which manipulate the schedule in total but require excessive computation.

Both these methods take the problem variables as the start times of all the operations of a job. However, as we are primarily concerned with minimizing the time a job spends in the system it may be adequate to consider only the times at which the first operations of the jobs start. Thereafter the multi-stage production system is simulated using a standard behavior pattern such as first-come-first-served to determine the timing of the other operations and the completion times of the jobs. This inlet control scheme greatly reduces the size of the optimization problem and is much easier to implement in practice.

Mathematically the task is to determine the inlet times $(x_1, x_2 \ldots x_N)$ of N jobs so as to minimize $F(y_1, y_2 \ldots y_N)$ where $\underline{x}$ and $\underline{y}$ are related through the simulation rules. When a schedule is created an interaction matrix can be calculated to express the tendency for the jobs to delay one another and this information can be used to give an estimate for the rate of change of F with x_j say $\frac{\Delta F}{\Delta x_j}$. These discrete derivates can be used to conduct a controlled direct search in terms of the x_j quantities to minimize $F(y_1, y_2 \ldots y_N)$. This direct search over the inlet times means that the method is independent of the peculiarities of individual application requirements which can be built into the simulator and it offers a practical method of control.

A. ORDEN, University of Chicago

On the Solution of Linear Equation/Inequality Systems.

Let the problem of finding a solution or showing that there is no solution to a linear equation/inequality system be written

(1.1) $\sum_{j=1}^{n} a_{ij} x_j \leq/=/\geq b_i, \quad i = 1 \dots m,$

(1.2) $x_j \geq 0$ for some subset of $x_1 \dots x_n$,

where: (a) $\leq/=/\geq$ means that for each i one of the three relationships $\leq$, $=$, or $\geq$ applies.

(b) Non-negativity of variables is set apart from other inequalities. Non-negative variables will be called "restricted", and the others "unrestricted".

(c) Systems of equations only (no inequalities) are not excluded.

A straightforward plan of solution is:

(2A) Insert non-negative slack variables into the inequalities in (1.1).

(2B) Apply Gaussian elimination by pivoting on the unrestricted variables.

(2C) If, after (2B), there remains a reduced system which involves restricted variables, apply phase I of the simplex method.

Conventionally, although the given problem (1.1), (1.2) has nothing to do with optimization, step (2C) calls for introduction of artificial variables and an artificial objective function. An alternative derivation of the simplex conputational procedure will be given which is based only on linear algebra arguments-inequality considerations are involved, but not artificial optimization.

A generalization of homogeneous linear systems, called "systems with conspicious solutions" will be used in place of the concept of basic solutions.

Definitions: Let $\sum_{j=1}^{n} a_{ij} x_j = b_i$, $i = 1 \ldots m$, be a linear system.

(3) The ith equation is in <u>reduced form</u> if either:

(a) it contains an isolated variable whose coefficient is +1 in that equation and 0 in all of the other equations,

or (b) the coefficients $a_{i1} \ldots a_{in}$ in that equation are all zero.

(4) A linear system of equations has a <u>conspicuous solution</u> if each equation either:

(a) is in reduced form,

or (b) has $b_i = 0$.

Homogeneous systems are the classic type of system with a conspicuous solution. More generally, the conspicuous solution of a system under (4) is given by:

If the ith equation is an equation in reduced form, set $x_{j_i} = b_i$ where x_{j_i} is the isolated variable in that equation. For the rest of the variables set $x_j = 0$.

Assume procedures (2A) and (2B) have been carried out. The problem to be solved under (2C) is the standard one of linear equation in non-negative variables:

(5.1) $\sum_{j=1}^{k} \alpha_{ij} x_j = \beta_i$, $i = 1 \ldots p$ (assume all $\beta_i \geq 0$),

(5.2) $x_j \geq 0$, $j = 1 \ldots k$.

(6) Let $\sum_{j=1}^{k} \gamma_j x_j = \delta$ be the equation which is obtained by summing the equations in (5.1) which are <u>not in reduced form</u>. Adjoin this (redundant) equation to (5.1). The matrix of the augmented system is

$$\begin{bmatrix} \alpha_{11} & \cdots & \alpha_{1k} & \beta_1 \\ \vdots & & & \vdots \\ \alpha_{p1} & \cdots & \alpha_{pk} & \beta_p \\ \gamma_1 & \cdots & \gamma_k & \delta \end{bmatrix}$$

A simple algebraic derivation shows that under a pivot transformation the last row of (7) always remains the sum of the rows which correspond to non-reduced equations. In particular, δ remains the sum of the β_i in the non-reduced rows. Using Max $\{\gamma_j > 0\}$ to select a pivot column, and the ratio technique of the simplex algorithm to determine a pivot row, pivoting produces a new system with all $\beta_i \geq 0$ and with a lower value of δ. In a finite sumber of stages either δ becomes 0,-- in which case we must have $\beta_i = 0$ in all non-reduced rows, and consequently we have a matrix which corresponds to a system which is equivalent to (5.1) and has a "conspicuous solution"; or all γ_j becomes < 0 and $\delta > 0$,-- in which case we have the matrix of a system equivalent to (5.1) which contains an infeasible equation, hence the system is infeasible. The argument is entirely in terms of linear algebra. Since there are no artificial variables, it serves no purpose to say that an objective function is involved.

Some properties of equation/inequality systems such as the Farkas theorem follow rather directly from the solution process. This has of course been done before via linear programming, but again the extraneous matter of optimization is avoided.

T.D. PARSONS, Pennsylvania State University, University Park, and A.W. TUCKER, Princeton University, Princeton.

Hybrid Programs: Linear and Least-Distance.

The convex quadratic program

(1) minimize $\varphi = \eta b + \frac{1}{2}\lambda\lambda^T$ for $\eta A + \lambda C \geq c,\ \eta \geq 0$

is a hybrid of the linear program to which it reduces when λ and C are vacuous, and the strictly convex quadratic program to which it reduces when η,b and A are vacuous. The latter is a "least-distance program" which seeks the point of the polyhedral set $\{\lambda | \lambda C \geq c\}$ at least distance from the origin. Solutions of (1) are hybrids of solutions of separate linear and least-distance subprograms that arise from projecting certain faces of the constraint set of (1) into the λ-subspace.

Let $P = C^T C$. Then (1) is dual to the concave quadratic program

(2) maximize $f = cx - \frac{1}{2}x^T Px$ for $Ax \leq b,\ x \geq 0$.

The key equation

$$\varphi - f = (\eta A + \lambda C - c)x + \eta(b - Ax) + \tfrac{1}{2}(\lambda^T - Cx)^T\ (\lambda^T - Cx)$$

yields $\varphi \geq f$ for feasible solutions of (1) and (2). These become optimal when $\varphi = f$. The hybrid nature of (1) sheds light on the geometry of this duality.

URY PASSY, Technion, Haifa.

Generalized Weighted Mean Programming.

The use of inequalities in mathematical programming is not new. The idea of Quasi-Linearization was initiated from Holder's inequality. The theory of Geometric Programming is intimately related to the Arithmetic-Geometric Mean inequality. However, use of inequalities in actually solving mathematical programs, is not widely practiced. In a recently published paper on Geometric Programmes, methods for solving such problems were described. This work shows that these methods can be viewed through a generalized relation existing within a certain class of programming problems, termed Generalized Weighted Mean Programming.

The main idea underlying the class of programming problems presently discussed is based on the well-known inequality relating the different "Weighted Means".

Let

$$(1) \qquad \underset{\sim}{y} \in (\mathbb{R}^n)^+ \text{ and } \underset{\sim}{q} \in (\mathbb{R}^n)^+$$

with

$$(2) \qquad \sum_{i=1}^{n} q_i = 1.$$

A generalized r-weighted mean denoted by $M_r(\underset{\sim}{y},\underset{\sim}{q})$ is given

$$(3) \qquad M_r(\underset{\sim}{y},\underset{\sim}{q}) = \left(\sum_{i=1}^{n} q_i y_i^{\,r} \right)^{1/r} .$$

Notice that if $r = 1$, $M_1(\underset{\sim}{y},\underset{\sim}{q})$ becomes the weighted arithmetic mean.

The relation between different weighted means is given by

$$(4) \qquad M_r(\underset{\sim}{y},\underset{\sim}{q}) < M_s(\underset{\sim}{y},\underset{\sim}{q}) \quad \text{if } r < s$$

with equality holding only when all $\underline{y_i}$ are equal.

<u>Definition. A generalized weighted mean programme (G.W.M.) is a minimization problem having the following form</u>

$$(5) \qquad \text{Min } f_o(\underset{\sim}{x})$$

subject to

$$(6) \qquad \underset{\sim}{x} \in S \quad \mathbb{R}^m$$

and

$$(7) \qquad g_i(\underset{\sim}{x})\, M_{r(i)}^{(i)}\, (F_i(\underset{\sim}{x}),\underset{\sim}{E}) < 1 \;, \quad i = 1,\ldots,P \,,$$

where

$$(8) \qquad \underset{\sim}{R} = [r(1), r(2), \ldots, r(P)]$$

is a given vector.

(9) $F_i(\underset{\sim}{x}) = [f_{i1}(\underset{\sim}{x}), f_{i2}(\underset{\sim}{x}), \ldots, f_{iI(i)}(\underset{\sim}{x})]$ $i = 1,\ldots,P$

(10) $\underset{\sim}{E}_i(\underset{\sim}{x}) = [\varepsilon_{i1}, \varepsilon_{i2}, \ldots, \varepsilon_{iI(i)}]$ $i = 1,\ldots,P.$

The components of E_i are given positive numbers satisfying

(11) $\sum_{j=1}^{I(i)} \varepsilon_{ij} = 1 \qquad \varepsilon_{ij} > 0 \qquad j = 1,\ldots,I(i)$
$i = 1,\ldots,P.$

(12) $I(1), I(2), \ldots, I(P)$ are given positive integers.

The functions $f_o, f_{11}, \ldots, f_{1I(i)}, \ldots, f_{PI(P)}, g_1, \ldots, g_P$ are continuous differentiable non—negative ones on the domain defined by the constraints Eqs. (6) and (7).

Various relations between such programmes are proved, and algorithms with solved examples are included.

The first algorithm is the Exterior Algorithm, which resembles the cutting planes algorithms, but a hypersurface cut replaces the linear cut. Thus, the programme need not be convex. In this algorithm the solution is approached from outside.

The second algorithm called the Interior Algorithm, is similar to polygonal approximation methods. When using this algorithm, the solution is improved after each iteration.

JOHN D. PEARSON and ALBERT C. DE KONINCK, N.V. Philips Gloeilampenfabrieken, Eindhoven.

Interchange algorithms versus multi-pass restricted optimization procedures, for a large scheduling problem.

This paper is concerned with a large schedule optimization problem for a group of M different two-sided asymmetrical machines which can process two jobs at a time, one on each side, from start to completion.

The problem is to rearrange 2 × m × T jobs into M sequances of T pairs of jobs, one for each machine, such that the sum of the sequence-dependent changeover and pairing costs is minimized, subject to constraints on pairing, due dates, labour requirements and production facilities. An initial partially feasible arrangement is provided by a prior capacity allocation phase, and an important simplification is that the orders are all multiples of the expected output from a job of standard duration.
In a practical situation 400 jobs are to be scheduled into 40 machines, 2 at a time, in a sequence of 5 pairs per machine. This is about an order of magnitude beyond any known optimizing algorithm. Most practical solutions to problems of this kind operate by interchanging blocks of jobs between machines or machine-sides, to achieve an improvement. Such an interchange algorithm has been implemented and it proves to be fast and effective.
The question investigated here is whether a better solution can be obtained more quickly, by optimizing overlapping portions of the scheduling problem in an iterative fashion.
If one or two machines are optimized at a time, the problem is one of quadratic assignment, which has been solved directly by a branch-and-bound algorithm, indirectly by a double route travelling salesman algorithm and heuristically by an extended but simplified form of the 1-opt, 2-opt idea of Lin.
If the jobs in non-adjacent schedule positions are considered, the optimization problem is one of integer assignment with side conditions, for which a branch-and-bound routine has been written.
Using these algorithms suitable for restricted parts of the problem, a variety of scheduling procedures can be devized, which make several passes to optimize the bad features detected in the schedule at each stage. Two implementations of this philosophy have been tested.

The paper describes the algorithms, the ideas behind the form of the implementations and gives comparisons based on random schedules generated from a practical situation.

ELMOR L. PETERSON, Northwestern University, Evanston, Ill.

Symmetric Duality in Convex Programming and its Economic Interpretation

The first completely symmetric formulation of duality for all convex programs with explicit constraints is given by generalizing and suitably modifying the original formulation of duality in geometric programming. The generalization consists of: (1) replacing the special convex functions in the original primal program by arbitrary convex functions, and (2) replacing the arbitrary pair of orthogonal complementary subspaces in the original primal and dual programs by an arbitrary pair of closed convex polar cones. The modification consists of adding convex functions of a particular type to the resulting primal objective function while simultaneously enlarging the class of permissible dual constraints. The resulting primal and dual "geometric programs" have the same form, and the "geometric dual" of the dual geometric program is the primal geometric program.

The geometric dual of such a program is constructed from: (1) the conjugate transforms of the convex functions appearing in the objective function, (2) the conjugate transforms of the convex functions appearing in the constraints, and (3) the polar of the closed convex cone appearing in the program.

Some rather elementary observations about conjugate convex functions and certain bases for orthogonal complementary subspaces show that this formulation can be specialized to give either Rockafellar's recent formulation or Fenchel's original formulation. Moreover, a different specialization followed by a suboptimization produces an extension of Wolfe's formulation.

In addition, complete economic interpretations of the various formulations of duality are provided by new duality theorems that are part of a new closed-form solution to a special, but economically interesting, class of convex programs. In particular, the well-known economic interpretation of dual optimal solutions as "shadow price" vectors is complemented by giving comparable economic interpretations of the dual non-optimal solutions and the dual objective function. It seems that these results are new even in linear programming.

REFERENCES

[1] PETERSON, E.L., "An Economic Interpretation of Duality in Linear Programming", to appear in Jour. Math. Anal. Appls., but presently available as Report # 953 from the Mathematics Research Center, University of Wisconsin, Madison 53706, U.S.A.

[2] —————, "Symmetric Duality for Generalized Unconstrained Geometric Programming", to appear in SIAM Jour. Appl. Math., but presently available as Report # 991 from the Mathematics Research Center, University of Wisconsin, Madison 53706, U.S.A.

[3] —————, "Generalization and Symmetrization of Duality in Geometric Programming", in preparation.

JOHAN PHILIP, Stanford University, Stanford, and The Royal Institute of Technology, Stockholm

Algorithms for the Vector Maximization Problem

We consider a convex set S in R^n described as the intersection of halfspaces $a_i^T x \leq b_i$, $(i \in I)$ and a set of linear objective functions $f_j = c_j^T x$, $(j \in J)$. The index sets I and J are allowed to be infinite in some of the algorithms. We give various characerizations of the efficient points of S (also called functionally efficient or Pareto optimal points). With the aid of the characterizations we give algorithms that solve the following problems:

I. To decide if a given point in S is efficient.

II. To find an efficient point in S.

III. To decide if a given efficient point in S is unique or if there are other efficient points.

IV. If a given efficient point is not unique, to find directions to other efficient points.

V. The solutions of the above problems do not depend on the absolute magnitude of the c_j. They only describe the relative importance of the different activities $x_1,\dots,x_n$. Therefore we also consider the problem

$$\max_{x \text{ efficient}} g(x)$$

for some function g. An algorithm that solves this problem for linear g is given.

H.E. PICKETT, Aerospace Corporation, San Bernardino, Cal.

A Contribution to the Thaumaturgy of Nonlinear Programming

This paper reports on a method for solving nonlinear optimization problems in which there is continuous gradient information about both the objective function and the constraint functions. The algorithm may be divided naturally into two parts. The first, which we call the basic set of constraints algorithm, deals with inequality constraints by posing equality constrained optimization problems for the second part of the algorithm to solve. It is a combinatorial, or global, algorithm, as distinct from the local methods such as the method of gradient projections. The second part of the algorithm is a method for optimizing functions subject to equality constraints. This method makes use of the Davidon-Fletcher-Powell algorithm for unconstrained optimization, and, like that method, converges from any starting point and has eventual quadratic convergence for quadratic functions.

In the paper, we first heuristically motivate and develop the basic set of constraints algorithm. Also we prove (Theorem 1) that if the algorithm does not cycle, then it obtains a local optimum of the function. A matrix method used to implement a portion of the algorithm is presented. The second method, which is essentially an algorithmic procedure for eliminating variables, is developed next. Because of its intrinsic simplicity, we hesitate to call it novel, but we know of no place where it has been presented in its present form.

In addition to the algorithm, we present a set of sufficient conditions for the existence of a unique global optimum (Theorem 3) which was suggested by the algorithm. The theorem is combinatorial in nature and is not based upon the usual convexity assumptions. We also consider the question of cycling of the basic set of constraints algorithm. Since cycling can occur, we have preferred to refer to this algorithm as part of the "magic arts" of nonlinear optimization. However, we conjecture that under conditions only slightly stronger than those of Theorem 3, the algorithm does not cycle. We adduce supportive evidence for this conjecture and prove (Theorem 4) certain special instances of it.

We conlcude with several illustrative examples.

JEAN-MARIE PLA, S.N.C.F., Paris

An "out-of-kilter" Algorithm for solving Minimal Cost Potential Problems

Most of network flow problems can be efficiently and elegantly solved by means of the so-called "out-of-kilter" algorithm published by D.R. FULKERSON in 1961. The present paper aims at producing a similar algorithm suited to network potential problems.

This algorithm, being almost exactly the dual of "out-of-kilter", has the same attractive features as the latter:
- the progress towards optimum is monotone for each arc of the network;
- the strating solution may be infeasible;

- finally, all or part of the data can be altered during or after the calculation, keeping the current solution as a starting one for the new problem.

Like "out-of-kilter", the algorithm to be described includes three routines:

- a labeling process, which leads either to "breakthrough" or to "non-breakthrough";
- a potential altering process (if non-breakthrough);
- a flow altering process (if breakthrough).

It allows detecting infeasibility, but also the lack of finite optimum - what "out-of-kilter" likewise could do for slight modifications - and terminates in a finite number of steps, provided the data are integers or rationals.

M.A. POLLATSCHEK and B.AVI-ITZHAK, Technion, Haifa

A Deep-cutting Procedure for Integer Programming

A cutting procedure is developed for the 0-1 linear program:

$$\sum_{j=1}^{n} a_{ij}x_j \le b_i, \; i = 1,2,\dots,m; \quad 0 \le x_j \le 1, \quad j = 1,2,\dots,n;$$

$$\sum_{j=1}^{n} c_j x_j \to \max, \tag{1a}$$

$$x_j \text{ is integer,} \tag{1b}$$

where c_j is integer and $0 \le c_1 \le c_2 \le \dots \le c_n$.

Supposing that z is an upper bound of the maximum, we have:

$$\sum_{j=1}^{n} c_j x_j \le z; \quad x_j \text{ is 0 or 1, } j = 1,2,\dots,n. \tag{2}$$

Considering x as a real vector, the convex hull of the solutions to (2) can be written as:

$$\sum_{j=1}^{n} \alpha_{k,j}x_j \leq 1, \quad k = 1,2,\ldots, K; \quad 0 \leq x_j \leq 1, \quad j = 1,2,\ldots, n, \tag{3}$$

where n-vector $\underset{\sim}{\alpha}_k$ and scalar K should be determined.

<u>Theorem.</u> $\underset{\sim}{\alpha}_k$ are the basic feasible solutions to the system in $\underset{\sim}{\beta}$:

$$\sum_{j=1}^{n} y_{\ell,j}\beta_j \leq 1, \quad \ell = 1,2,\ldots, L; \quad \beta_j \geq 0, \quad j = 1,2,\ldots, n \tag{4}$$

where y_ℓ, $\ell = 1,2,\ldots, L$ are all the solutions to (2).

Given a point u, the "farthest" inequality,

$$\sum_{j=1}^{n} \alpha_j x_j \leq 1,$$

among the inequalities of system (3) from point u in norm L_1, approximately, is given by the solution of

$$\max \sum_{j=1}^{n} u_j\beta_j$$

subject to (4).

Thus,

$$\sum_{j=1}^{n} \bar{\alpha}_j x_j \leq 1$$

is the deepest cut in L_1 norm, approximately, from point u, using inequality (2).

The basic cutting procedure consists of the following steps:

1. Find a solution to (1a). Let $x^{(1)}$ be the solution and $z^{(1)}$ the value of the objective function, k := 1.
2. If $x^{(k)}$ has 0-1 entries only, stop. It is a solution to (1a-1b). Otherwise proceed.
3. Find:

$$\alpha_o = \text{Max}\ [\Sigma x_j^{(k)}\ \beta_j \mid \forall_\ell : \Sigma y_{\ell,j}\beta_j \leq 1,\ \beta_j \geq 0] = \Sigma x_j^{(k)}\ \bar{\alpha}_j.$$

4. If $\alpha_o > 1$, the constraint $\Sigma x_j \bar{\alpha}_j \leq 1$ is added to the system, reoptimize, the solution is $x^{(k+1)}$ with value $z^{(k+1)}$, $k := k+1$ and return to step 2. Otherwise proceed.
5. Find whether any 0-1 solution to $\Sigma c_j x_j = [z^k]$ satisfies the original problem. In case of a positive answer, stop: the solution is found. Otherwise proceed to step 6.
6. $z^{(k+1)} = z^{(k)} - 1$; $x^{(k+1)} := x^{(k)}$; $k := k+1$, return to step 3.

The procedure converges in a finite number of steps to the solution of (1a) - (1b).

The same ideas can be applied to general integer linear programs.

ANDRAS PREKOPA, Hungarian Academy of Sciences, Budapest

Programming under probabilistic constraints and programming under constraints involving conditional expectations

1. The following fundamental theorem holds: Let $Q(\underline{x})$ be a convex function in R and $g(z)$ be a decreasing and differentiable function in the range of values of $Q(\underline{x})$. We suppose that $g(z) \geqslant 0$, $-g'(z)$ is logconcave, $\int_{R^n} g(Q(\underline{x}))\, d\underline{x} < \infty$(logconcavity of a function $h(\underline{x})$ is defined by the inequality

$$h(\lambda \underline{x}_1 + (1-\lambda)\, \underline{x}_2) \geq [h(\underline{x}_1)]^{\lambda}\, [h(\underline{x}_2)]^{1-\lambda}, \quad 0 < \lambda < 1).$$

We consider the integral $\int_{A+t} g(Q(\underline{x}))\, d\underline{x}$ which is a function of the variable vector $\underline{t}$. A is a convex subset of R^n and A+$\underline{t}$ means a translation by $\underline{t}$. Statement: $f(\underline{t})$ is logconcave in R^n. The function $g(Q(\underline{x}))$ will be used as the probability density of a random vector. Logconcave densities satisfy the above assumptions because they are of the form $e^{-Q(\underline{x})}$ and $g(z) = e^{-z}$. In this category we find among others the multivariate normal distribution (with arbitrary parameters), the Wishart, the Dirichlet and the multivariate beta distributions (with suitable parameters). If the density is logconvex then the probability of the set $A+\underline{t}$ is a logconvex function of $\underline{t}$. This last statement is far more immediate then the fundamental theorem. Example for a logconvex probability measure is the multivariate Pareto distribution. The fundamental theorem implies in particular that $\log \Phi(t_1,\ldots,t_n)$ is a concave function in the entire n-dimensional space where $\Phi(t_1,\ldots,t_n)$ is the multivariate normal probability distribution function (with arbitrary parameters).

2. Applications to stochastic programming models.

a) Let $\underline{\beta}_1$, $\underline{\beta}_2$ be random vectors where the components have a joint normal distribution. Constraints of the type $P(\underline{\beta}_1 \le A\underline{x} \le \underline{\beta}_2) \ge p$ or $P(\underline{h}(\underline{x}) \ge \underline{\beta}_1) \ge p$ will be considered where $\underline{h}(\underline{x})$ is a concave vector-valued function and p is an arbitrary probability between 0 and 1. In view of the fundamental theorem the functions on the left hand sides are logconcave in $\underline{x}$ thus such constraints can be parts of convex or quasi-convex programming problems. Semi-infinite stochastic programming problems can also be formulated by introducing constraints e.g. of the following type: $P(\beta_1(t) \le A(t,\underline{x}) \le \beta_2(t),\ t \in T) \ge p$, where T is an interval, $\beta_1(t)$, $\beta_2(t)$ are stochastic processes with Gaussian finite dimensional distributions and $A(t,\underline{x})$ is a linear function in $\underline{x} \in R^n$ for every fixed $t \in T$.

b) The Dantzig-Madansky-model (two stage programming under uncertainty) is transformed by introducing a new condition containing a prescribed lower bound for the probability of solvability of the second stage problem (the second stage problem is not supposed to be solvable for all first stage decision vectors $\underline{x}$). The objective function is also modified and the new model is shown to be a convex programming problem.

3. Consider the problem: $g_i(\underline{x}) \ge \beta_i$, $i = 1,\ldots, m$, $\underline{x} \ge \underline{0}$, $\min g(\underline{x})$, where the functions $g_1(\underline{x}),\ldots, g_m(\underline{x})$ are concave. If $\beta_1,\ldots,\beta_m$ are

random variables then this problem looses its meaning. Our problem is now formulated as:

$$E(\beta_i - g_i(\underline{x}) \mid \beta_i - g_i(\underline{x}) > 0) \leq f_i, \quad i = 1,\ldots,m, \ \underline{x} \geq 0, \ \min g(\underline{x});$$

here $f_1,\ldots,f_m$ are arbitrary positive constants and on the left hand sides there stand conditional expectations. If $\beta_1,\ldots,\beta_m$ have normal distributions, then this last problem can be shown to be a convex programming problem (provided, of course, $g(\underline{x})$ is also convex).

4. Algorithmic solutions. Algorithms for the solution of models 2.a) and 3.) will be given and the solution possibility of the model 2.b) will be discussed.

L.D. PYLE and G.J. MCWILLIAMS, University of Texas, Austin

On the Use of the Projected Gradient in place of Ward's Transformation in the Approximate Solution of Two-Dimensional Transportation Problems

In a paper in Psychometrika [1] J.H. Ward discusses a transformation similar to that for computing the projected gradient [2]. When applied to the cost vector of a two-dimensional transportation problem, Ward's transformation yields a vector which is then used in determining a feasible solution often found to be optimal or "near optimal".

This paper presents the results of a series of numerical experiments being conducted using the projected gradient in place of the transformed vector given by Ward. Implications for general linear programming problems are discussed, together with some of the attendant difficulties, such as the computation of projected gradients for large, sparse linear systems [3].

[1] WARD, J.H., The Counseling Assignment Problem, Psychometrika, Vol. 23, nr. 1, March 1958.

[2] CLINE, R.E., and L.D. PYLE, The Generalized Inverse in Linear Programming - An Intersection Projection Method and the Solution of a Class of Structured Linear Programming Problems, TNN - 103, Computation Center, The University of Texas at Austin, Austin, Texas (1970).

[3] SMITH, D.K., A Dynamic Component Suppression Algorithm for the Acceleration of Vector Sequences, Doctoral thesis, Purdue University, Lafayette, Indiana (1969).

REZA RAZANI, Pahlavi University, Shiraz, Iran

Application of Mathematical Programming to Optimum Design of Engineering Structures; a Survey and Recent Advances.

Optimum design of engineering structures can be formulated in the form of the following mathematical programming problem:

$$\min_{x}\{f(x) \mid \Phi(x,u) = 0,\ g(x,u) \geq 0,\ h(x) \geq 0\}$$

where: x represents the design variables such as the size of the structural members,

u represents the behavior variables such as stresses, displacements, etc.,

f(x) is the objective function representing the weight or cost of the structure,

$\Phi(x,u) = 0$ represents the set of analysis equations generally consisting of the equilibrium and compatability systems of equations,

$g(x,u) \geq 0$ represents the set of behavior constraints guarding the structure against various failure modes such as excessive strains, etc.,

$h(x) \geq 0$ represents constraints on the design of structural elements due to codes and specifications such as minimum specified plate thickness, etc.

In the traditional method of structural design, the designer tries to obtain feasible designs satisfying analysis equations and constraints. He takes into consideration the design objectives using design principles, intuition, experience, and numerous cycles of trial analysis and redesign. For certain classes of structures the designer finds that designs based on the Fully-Stressed principle or the Simultaneous Mode of Failure principle leads to minimum weight design. Unfortunately, the traditional designer often extrapolates these principles beyond the range of their application resulting in inefficient structures.

By considering the structural design as a mathematical programming problem, it is shown that following the traditional design principles leads to a particular vertex of the constraint set $g(x,u) \geq 0$. For some nonlinear cases this vertex is not the optimum point. The Kuhn-Tucker Optimality Condition is used to verify the optimality of the designs based on traditional principles.

Linear programming has been used for the limit analysis and design of structures. In this case the minimum-weight design problem is obtained using the Static formulation or its dual, the Kinematic formulation. Recently, linear programming has been used in the minimum-weight designs of trusses to find not only the areas of the members, but the topology of the optimum truss as well.

In general, most design optimization problems are non-linear; therefore, non-linear programming methods have been used for the optimum design of many types of elastic structures. For the solution of the non-linear problem, various methods of feasible directions or methods of unconstrained minimization using penalty function techniques or Cutting Plane methods using a sequence of linear programming problems are used.

Problems of minimum-cost designs of structures have led to the generation of many types of Integer Programming and Fixed-Charge Problems. In the solution of some of these problems, a Dynamic Programming technique is used.

Optimum design of structures on the basis of reliability is another important area of research. Here, the problem of the optimum design reduces to that of mathematical programming under uncertainty.

By using either potential or complementary energy formulation, many conservative structural analysis problems can be treated as unconstrained minimization problems. Methods of unconstrained minimization in conjunction with energy formulation is also used for the solution of combined structural analysis and design optimization problems.

Under certain conditions the mathematical models of determination of the plastic strain rates in plastic and elastoplastic structures results in the minimization of a quadratic function, subject to linear inequalities. The solution of this quadratic programming problem leads to a minimum theorem for plastic strain rates. Quadratic Programming techniques are also used for the solution of the matrix method of stationary creep analysis of structures and many other design-analysis problems using energy formulation.

Many other applications of mathematical programming techniques to the analysis and optimum design of structures together with the problem areas, future trends and unsolved problems are also reported.

PIERRE ROBILLARD and MICHAEL FLORIAN, Université de Montreal, Montreal

(0,1)-Hyperbolic Programming Problems

The problem of (0,1)-hyperbolic programming introduced by Hammer and Rudeanu (Boolean methods in Operations Research) can be formulated as follows:

minimize the function $F(X) = (a_0 + \sum_{i=1}^{n} a_i x_i)/(b_0 + \sum_{i=1}^{n} b_i x_i)$

subject to the constraints $H_j(X) \leq d_j$, $j = 1,2,\ldots, m$ where the H_j are pseudo-Boolean functions and $x_i = \{0,1\}$, $i = 1,2,\ldots, n$. We denote this problem by P and X^* is an optimal solution of P if minimizes F(X) and satisfies the set of constraints.

The solution X^* can be obtained by solving a finite number of problems where a linear function is minimized under the original constraints. When the functions H_j are linear the solution X^* is obtained by solving a finite and usually small number of (0,1) linear programming problems. When no constraints of the form $H_j(X) \leq d_j$ are imposed the method of solution reduces to the one described by Hammer and Rudeanu.

The algorithm introduced above is illustrated by few numerical exemples and a series of computer trials with a suitable code.

J.B. ROSEN, University of Wisconsin, Madison

Solution and Error Bound for Partial Differential Equations by Linear Programming

A general method for obtaining an approximate solution, together with an error bound, for certain types of boundary value problems will be described. An approximate solution is assumed as a linear combination of selected basis functions. The coefficients of this linear combination are determined using linear programming so as to minimize the error bound on the approximate solution.

DAVID S. RUBIN and ROBERT L. GRAVES, University of Chicago, Chicago

The Modified Dantzig Cuts for Integer Programming

We consider the integer program

$$\begin{aligned} \max\ c'x &= z \\ \text{subject to}\quad Ax &= b \\ x &\geq 0 \text{ and integer} \end{aligned} \qquad \text{(IP)}$$

where c is (m+n) x 1, b is m x 1, A is (m+n) x n, and all three have all integer components.

Let B be the optimal basis for IP taken as a linear program, and let A be partitioned as (B,N), c as $\binom{c_B}{c_N}$, and x as $\binom{x_B}{x_N}$.

Then Tucker's linear programming optimal tableau is

$$\begin{array}{rl|c} & & -x_N \\ \hline z & = \gamma_0 & \gamma' \\ x_B & = B^{-1}b & B^{-1}N \\ x_N & = 0 & -I \end{array}$$

where $\gamma_0 = c_B B^{-1} b$ and $\gamma' = c_B B^{-1} N - c_N'$. We denote the elements of x_B as $x_{B1}, \ldots, x_{Bm}$, and those of x_N as $x_{N1}, \ldots, x_{Nn}$.

If $B^{-1}b$ is not an all integer vector, we can proceed by adding additional constraints, or cuts, to the problem. In 1959 [1], Dantzig proposed the cut

$$\sum_{j=1}^{n} x_{Nj} \geq 1$$

to be used for integer programming, but did not give a proof that an algorithm based solely on this cut (the "Dantzig cut") would converge to the IP optimum. In 1963 [2], Gomory and Hoffman proved that such an algorithm would not converge in general. In this paper we present some slightly modified versions of the Dantzig cut which can be shown to converge in all cases.

We denote the matrix in the tableau by Y, and we number its rows from 0 to m+n and its columns form 0 to n. Write Y = W+F where $f_{ij} = y_{ij} - [y_{ij}]$, the positive fractional part of y_{ij}. Choose some index i such that $f_{i0} \neq 0$. We call row i the "source row" for the cut. Let $\hat{\delta}_{ij} = \begin{cases} 1 \text{ if } f_{ij} \neq 0 \\ 0 \text{ if } f_{ij} = 0 \end{cases}$. We define the first modified Dantzig cut (the "MD1 cut") by

$$\sum_{j=1}^{n} \delta_{ij} x_{Nj} \geq 1$$

and propose the MD1 algorithm:

1) Use the simplex method to solve IP as a linear program, finishing with a lexico-dual feasible tableau. (We assume that the problem is bounded.)

A. If infeasible, then IP is infeasible.

B. If feasible, and the solution is all integer, it is optimal in IP.

C. If feasible, and the solution is not all integer, go to step (2).

2) Let i^* be the least i such that $f_{i0} \neq 0$, and let row i^* be the source of an MD1 cut. (If all $f_{i0} = 0$, $i = 0,\ldots,m+n$, then the tableau is optimal for IP.) Let $\eta = \begin{bmatrix} y_{00} \\ \vdots \\ y_{i^*0} \end{bmatrix} = \begin{pmatrix} y_{00} \\ \vdots \\ y_{i^*0} \end{pmatrix}$.

Add the cut to IP, pivot once, and to to step (3).

3) If the new tableau (whose matrix we denote by Y') is

A. primal feasible, go to step (2);

B. primal infeasible, and if

1. $\begin{pmatrix} y'_{00} \\ \vdots \\ y'_{i^*0} \end{pmatrix} \leq \eta$, pivot until primal feasibility is restored, and go to step (2). If feasibility cannot be restored, then IP is infeasible.

2. $\begin{pmatrix} y'_{00} \\ \vdots \\ y'_{i^*0} \end{pmatrix} > \eta$, add another MD1 cut derived from the same row, pivot once, and to to step (3).

Lemma: The MD1 algorithm cycles in step (3) until either

a. $y'_{i*0} \leq \eta_{i^*}$ or

b. $y'_{k0} < \eta_k$ for some $k = 0,1,\ldots,i^* - 1$, or

c. it determines that IP is infeasible.

One of these cases must occur after a finite length of time.

Theorem: The MD1 algorithm converges to the solution of IP or shows that

IP is infeasible in a finite length of time.

Although convergent, the algorithm can be shown to be quite slow. The paper continues by investigating ways to strenghten the MD1 cuts. We show that in very general circumstances, the right hand side of the cut can be changed to a two. We call this the MD2 cut. It also turns out to be quite weak. For a final strenghtening of the cut, we use its coefficients as an objective to Gomory's asymptotic algorithm [3, 5], and let that procedure determine the right hand side of the cut, frequently an integer greater than two. We call this the MDk cut.

We conclude with a discussion of the computational aspects of the MDk cuts, and a suggestion for an algorithm which combines this cut with the method of integer forms cut [4].

[1] DANTZIG, G.B., Note on Solving Linear Programs in Integers. Naval Research Logistics Quarterly, VI (1959), 75-76.

[2] GOMORY, R.E. and HOFFMAN, A.J., On the Convergence of an Integer Programming Process, Naval Research Logistics Quarterly, X (1963), 121-123.

[3] GOMORY, R.E., Some Polyhedra Related to Combinatorial Problems. IBM Report RC2145, July 1968.

[4] GOMORY, R.E., An Algorithm for Integer Solutions to Linear Programs, in R.L. Graves and P. Wolfe (eds.), Recent Advances in Mathematical Programming. New York: McGraw-Hill, 1963.

[5] RUBIN, D.S., The Neighboring Vertex Cut and Other Cuts Derived with Gomory's Asymptotic Algorithm. Unpublished doctoral dissertation, Graduate School of Business, University of Chicago, June 1970.

R.W.H. SARGENT and B.A. MURTAGH, Imperial College, London

Projection Methods for Non-Linear Programming

The paper describes several projection algorithms for non-linear equality constraints. These are based on the "rank-one" quasi-Newton minimization technique combined with variable metric projection, which in the best algorithm fits the constraints to second order. To achieve convergence for non-linear constraints it was found necessary to limit the extent of constraint violation at each step, either by use of a penalty function, or by a correction procedure. Several correction procedures are examined and compared with the use of a penalty function.

Non-linear inequality constraints are effectively dealt with by converting them to equality constraints using slack variables, and linear inequalities by using Rosen's strategy for constructing a sub-set of active constraints.

Numerical results are given for the set of problems published by Colville, showing that the algorithm as finally developed converges satisfactorily to the correct solution for all the problems, in each case requiring fewer function evaluations than the best of the algorithms previously recorded.

ROLAND SCHINZINGER, University of California, Irvine

On the Reduction of the Continuous Variable Component in Mixed Integer Linear Programming

There exists for every mixed integer linear programming problem an optimum at which the independent continuous variables assume the value zero. These are surface optima, not necessarily unique, but more easily located. More specifically, the locus of such an optimum is the intersection of g planes which in turn are defined by g zero-valued continuous variables; g is the original number of continuous variables, i.e. excluding continuous slack variables.

As a reasonable starting point in the search for the optimum one may therefore select g likely candidates from among the continuous variables; both design variables and slack variables may be chosen as long as they are of the continuous variety. These g continuous variables would then be set equal to zero, leaving one with m (or less) dependent continuous variables as well as the set of integer variables. Here m is the number of constraints. There is no assurance that the original set of g candidates was the optimum set. Other combinations must therefore be explored, but the search is truncated by stringent tests. The problem is then referred to any enumerative type search, such as the Shrinking Boundary Algorithm *) which may be modified to advantage to handle continuous variables among the dependent variables, or to the conversion algorithm which transforms a mixed integer problem into a pure integer problem. The latter has not been described elsewhere; therefore it will be sketched briefly in this paper.

*) "A Shrinking Boundary Algorithm for Discrete Systems Models" by R.M. Saunders en R. Schinzinger, IEEE Trans. on Systems Science and Cybernetics, May 1970.

B.M.E. DE SILVA, University of Technology, Loughborough, Leicestershire

Optimal Vibrational Modes of a Turbine Disc

This investigation is a continuation of a research program into computational procedures based on the methods of mathematical programming for solving structural optimization problems in the presence of design constraints. Such procedures were successfully developed for obtaining minimum weight solutions |1,2,3| to a turbine disc of variable thickness in the presence of constraints on the stresses, natural frequencies of vibration. The problem was formulated as a general problem in optimal control theory in the presence of inequality constraints on the state and control variables.

Numerical solutions were obtained by transforming the variational formulation into a discrete nonlinear programming formulation using a piecewise linear representation for the control variables.

In the present investigation an attempt is made to further generalise these procedures and to extend their scope to include more complex structural optimization problems.

For this purpose, the paper considers the problem of maximising some linear combination of the frequencies of vibration of a turbine disc subject to constraints on the dimensions and tolerances of the disc and its total weight. The problem is again formulated as a general optimal control problem with the frequencies as control parameters. The design requirements are represented by state and control inequality constraints, the control and state variables being given by functions describing the variations in thickness and deformation fields. Significant progress has been made in solving the problem using purely analytical techniques based on the (restricted) maximum principle of Pontryagin. (These include the analytical solutions of systems of ordinary differential equations using perturbation techniques and analytical solutions of fourth order differential equations using WKB expansions.) These transform the problem into a nonlinear programming problem which is then solved numerically using the Heaviside penalty function transformation in conjunction with Rosenbrock's hill-climbing procedures.

Available computational experience indicates that these procedures provide powerful tools for handling complex structural optimization problems.

[1] B.M.E. DE SILVA, The application of nonlinear programming to the automated minimum weight design of rotating discs: Optimization

(editor R. Fletcher), Academic Press (1969), pp 115-150.

[2] B.M.E. DE SILVA, The minimum weight design of discs using a frequency constraint: Transactions of the American Society of Mechanical Engineers, Journal of Engineering for Industry, November (1969), pp 1091-1099.

[3] B.M.E. DE SILVA, The application of Pontryagin's Principle to a minimum weight design problem: to appear in Transactions of the American Society of Mechanical Engineers, Journal of Basic Engineering.

MATTHEW J. SOBEL, C.O.R.E., Heverlee, Belgium

A finite Algorithm for Equilibrium Points of Games

This paper presents an algorithm for computing, in a finite number of steps, an equilibrium point in any finite noncooperative game. Thus the algorithm is also a constructive (algebraic) proof of John Nash's theorem that such a game has an equilibrium point (EP).

Let $\Omega = \{1,\ldots,N\}$ be a set of players and A_i be the set of actions available to player $i \in \Omega$. The game is finite if $K \equiv \Sigma |A_i| < \infty$ and none of the A_i is empty. The payoff to player i is $r_i(a)$ after the players take their actions $a \in A_i$. If randomized rules are used, the players' utilities are their expected rewards. Let $x^i = (x_{ik})$ be a randomized rule for player i, $k \in A_i$. Then $x^i \in \pi_i$ which is the $|A_i| - 1$ dimensional unit simplex. Let $\pi = \boldsymbol{X}\pi_i$. The utilities are

$$U_i(x) = \Sigma_{a_1 \in A_1} \cdots \Sigma_{a_N \in A_N} x_{1a_1} \cdots x_{Na_N} \, r_i(a_1,\ldots,a_N).$$

For $x = (x^1,\ldots,x^N)$, let $x_{-i} = (x^1,\ldots,x^{i-1},x^{i+1},\ldots,x^N)$ and write $x = (x^i,x_{-i})$. Then $\bar{x} \in \pi$ is said to be an <u>equilibrium point</u> (EP) if

$$U_i(\bar{x}) \geq U_i(x^i,\bar{x}_{-i}), \quad \text{all } x^i \in \pi_i, \quad \text{all } i \in \Omega.$$

It can be shown that an equivalent condition is that there be an associated $v \epsilon E^N$ and $\alpha \epsilon E^k$ such that

(1) $v_i = U_i(e_k, x_{-i}) + \alpha_{ik}, \qquad k \epsilon A_i, \quad i \epsilon \Omega$

(2) $x \epsilon \pi, \ \alpha \geq 0$

(3) $\alpha x = 0.$

Robert Wilson has recently used a system equivalent to (1)-(3) to prove Nash's theorem algebraically. Our algorithm is closely related to Lemke and Howson's algorithm for the case N = 2 (bimatrix games) and to Wilson's proof.

Our algorithm begins by constructing (by induction on N) a starting point that satisfies (1) and (2); it also satisfies (3) except possibly for one index (i,k) so $\alpha x = \alpha_{ik} x_{ik} \geq 0$. Such a point is an almost complementary solution (AC). If a solution is AC and $\alpha x = 0$, it is an EP.

The algorithm consists of examining a sequence of AC solutions until an EP is reached. The pivot from one AC solution to the next one is accomplished by solving a linear programming problem.

I.M. STANCU-MINASIAN, Academy of the Socialist Republic of Romania, Bucharest

The Solution of a Transportation Network in the Case of Multiple Criteria

In the paper we consider three problems, that appear in connection with a transportation network, problems that are differently treated in the literature. We prove that in fact we have one problem to which distinct efficiency functions are associated.

Suppose that in m production centres A_1, A_2,... A_m are $a_1, a_2, \ldots, a_m$ quantities of a certain material that is asked by n centres of consumption $B_1, B_2, \ldots, B_n$, each asking the quantities $b_1, b_2, \ldots, b_n$. For each route (A_i, B_j) we denote by d_{ij} the greatest quantity that can be transported, by c_{ij} the transportation cost of a single unity of the considered material, and by t_{ij} the time of transportation.

Problem 1. It is required to organise the dispatch process to that within the availability limit and irrespective of the transportation costs and times, the quantity to be transported should be maximum.

If we consider a fictitious centre A_0 that concentrates the whole available quantity of material, and a fictitious centre B_{n+1} that concentrates the whole necessary quantity of material, we are confronted with a maximal flow problem in a transportation network. We denote by ξ_{ij} the quantity to be transported. The mathematical model of the problem is the following:

$$\sum_{j=1}^{n} \xi_{ij} = \xi_{oi} \qquad i = 1,2,\dots,m \qquad (1)$$

$$\sum_{i=1}^{m} \xi_{ij} = \xi_{jn+1} \qquad j = 1,2,\dots,n \qquad (2)$$

$$0 \le \xi_{oi} \le a_i \qquad i = 1,2,\dots,m \qquad (3)$$

$$0 \le \xi_{jm+1} \le b_j \qquad j = 1,2,\dots,n \qquad (4)$$

$$\text{Max } F_1 = \sum_{j=1}^{n} \xi_{jn+1} \qquad (5)$$

Problem 2. In the same conditions it is required to organise the dispatch so that the total cost of transportation be minimal. This is a classical transportation problem. It is proved that the model of the transportation problem can be expressed by the constraints (1) - (5).

The efficiency function of the problem will be:

$$\text{Min } F_2 = \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij}\, \xi_{ij} \qquad (6)$$

Problem 3. The data being the ones in the problem 1, organise the dispatch so that the total time be minimal.

Similarly to the problem 2, the constraints are (1) - (5), and the efficiency function is:

$$\text{Min } F_3 = \sum_{i=1}^{m} \sum_{j=1}^{n} t_{ij} \xi_{ij} \tag{7}$$

In order to take into account all the three efficiency functions, they are transformed into utility functions:

$$F_1' = \sum_{j=1}^{n} a_1 \xi_{jn+1} + b_1,$$

$$F_2' = \sum_{i=1}^{m} \sum_{j=1}^{n} a_2 c_{ij} \xi_{ij} + b_2$$

$$F_3' = \sum_{i=1}^{m} \sum_{j=1}^{n} a_3 t_{ij} \xi_{ij} + b_3$$

that are summed up; thus we obtain a synthesis function

$$F^* = K_1 F_1' + K_2 F_2' + K_3 F_3',$$

where K_1, K_2, K_3 are importance coefficients for each function, and a_1, a_2, a_3, b_1, b_2, b_3 are some transformation coefficients obtained by the application of the von Neumann-Morgenstern's definition of the utility function.
At the end of the paper, an example is considered.

JANOS STAHL, INFELOR, Budapest

On the two-stage Stochastic Linear Programming Problem

We deal with the two-stage stochastic LP problem

(1) $\inf \{E(cx + \inf(dy \mid A_2x + By = \xi;\ y \geq 0)) \mid A_1x = b;\ x \geq 0\}$

where A_1, A_2, B, c and d are given matrices and vectors of appropriate size, ξ is a random vector with known distribution function and E denotes the expectation operator.

As far as we know computationally feasible procedures for solving (1) are known in the case $B = [I, -I]$ and in cases equivalent to the previous one. (I denotes unit matrix.) But in this case the model is not too satisfactory from a statistical viewpoint, since the solution depends only on the boundary distribution of ξ.

It is known that the set of feasible solutions, i.e. that subset of $\{x \mid A_1x = b;\ x \geq 0\}$ for which $\{y \mid By = \xi - A_2x;\ y \geq 0\} \neq \emptyset$ for every realization of ξ, is a convex polyhedral set. In the first part of the paper algorithms for determining this set will be considered.

If ξ has a discrete distribution, (1) is a LP problem. Under mild assumptions it can be proven that the optimal value of the LP problem belonging to a discrete $\overline{\xi}$ converges to the value of (1) if $\overline{\xi}$ converges (weakly) to ξ. A procedure based on appropriate approximation of this type will be discussed.

Finally we consider some extensions involving further stochastic elements among the parameters.

R.B. STANFIELD, ESSO Mathematics & Systems Inc., Florham Park, N.J.

Nonlinear Programming in Large Models

The use of large models today usually involves large application oriented matrix generator systems. In addition, a number of these simulations are limited by our ability to adequately represent nonlinear business situations. Most classical nonlinear algorithms are unsuitable for the massive data processing and systems design problems that occur in these large generator systems. Iterative linear programming techniques are sufficient, however, to represent and solve these nonlinear models.

These paper discusses a solution through the use of the modular generator system, each module representing a generic business activity. The algorithms, linear and nonlinear, for the specific technological areas of the model would be coded within these modules. The coding would determine the existence, function, and arguments to compute each element to represent an activity.

In the applications examined, a major fraction of the elements are nonlinear. To recompute a linearized model based on new solution values, one wishes to avoid the input interpretation, file searching and sorting required in matrix generation. The concept of retaining the functional representation of an element until as late in data processing as possible is proposed.

Each nonlinear element would then exist in a virtual form on a master work file. The actual work file would be produced by passing the master work file against an edit program and solution values. Such regenerations can be extremely fast, as they will not involve input interpretation or sorting.

A middle distillates blending model involving the nonlinear pooling problem was chosen for a demonstration of the concepts. The model involved 200 rows and 2500 nonzero elements. An algorithm for pooling was implemented using iterative LP techniques. A system was written to demonstrate both the system efficiently as well as the stability and convergence of the pooling algorithm. The results have been encouraging.

In general, considerable analysis will be required of a given technology to generate general and reliable nonlinear codes. However, the inclusion of such algorithms in general generator systems now seems feasible.

DANIEL TABAK, Rensselaer Polytechnic Institute of Connecticut, Inc., East Windsor Hill, Conn.

Optimal Control by Mathematical Programming

Direct applicability of mathematical programming techniques in the design of optimal control systems is discussed [1,2] .

Various case studies of actual implementation of mathematical programming algorithms in various problems of practical applications are presented.

The types of control systems discussed, include linear, nonlinear, continuous and discrete (time systems, deterministic and stochastic as well as distributed) parameter systems. The applicability of ideas derived from mathematical programming algorithms to the solution of nonzero-sum, constrained difference games, applied to problems of economic competition is also touched upon.

The areas of application include aerospace trajectory optimization and rendezvous problems, computer control of processes, nuclear reactors and constrained estimation problems.

[1] D. TABAK, Application of Mathematical Programming in the Design of Optimal Control Systems, Ph.D. Thesis, University of Illinois, Urbana, Ill., 1967.

[2] D. TABAK, B.C. KUO, Optimal Control by Mathematical Programming, Prentice-Hall Inc., Englewood Cliffs, N.J. (to appear in 1970).

GERALD L. THOMPSON, Carnegie-Mellon University, Pittsburgh, and ROMAN L. WEIL, University of Chicago, Chicago.

Optimizing λ (or x) subject to $Ax = \lambda Bx$: the Generalized Eigenvalue Problem

For square A, $Ax = \lambda x$ is the standard eigenvalue-eigenvector (hereafter, eigensystem) problem. We have studied the problem $Ax = \lambda Bx$ for $m \times n$ matrices A and B. We show how the problem of finding such (λ, x) can be reduced by a recursive application of elementary row and column operations to a standard eigensystem problem.

If A and B are square and B is nonsingular, then the eigensystem of $Ax = \lambda Bx$ is that of $B^{-1}Ax = \lambda x$, a standard eigensystem problem.

When A is singular or, more generally, when A and B are rectangular, more detailed analysis is required to isolate the standard eigensystem whose solution is that of the generalized eigensystem.

We believe the result to be fundamental and of wide potential application. As of this writing the only application we have seen is to optimal control problems of the form

$$x = Ax + Bu$$
$$\text{s.t.} \quad Cx = f(t).$$

LEONARD TORNHEIM, Chevron Research Company, Richmond, Cal.

A Linear Programming Algorithm for Maximum Change at each Iteration

This algorithm selects a pivot at each iteration which maximizes the change in the objective function and yet requires about the same amount of computation. Since experiments [1] indicate that fewer iterations are usually required, this method would be more economical than the other common procedures.

Such a procedure was given by M.A. Efroymson, but only an abstract was published [2]. Also, G.B. Dantzig described one in an oral communication but did not provide for degeneracy. It is not indicated in the abstract of Efroymson that he allowed for degeneracy. This is done here, with a small increase in time.

Also, the algorithm is extended to include upper bounding, which requires appreciably more time to handle than in the usual simplex method.

The normal form used for the equations $\sum^{n} a_{ij} x_j = b_i$ $(i = 1,\ldots,m)$, $\sum c_j x_j = z + z_o$ (minimize) is described as follows. At each iteration, the indices $1,\ldots,m$ of the equations are separated into two sets, I_P and I_Z, which correspond to the positive x_j and the zero x_j in the present basis $(x_{j_1},\ldots, x_{j_m})$. We can assume $I_P \neq \emptyset$.

For a certain index $h \in I_P$, $b_h = 1$, $b_i = 0$ $(i \neq h)$; $c_{j_i} = 0$; $a_{ij_i} > 0$; all for $i = 1,\ldots,m$. Also, for $i \neq h$, $a_{gj_i} = 0$ $(g \neq i)$, $a_{ij_h} = -a_{hj_h}$ $(i \in I_P)$, and $a_{ij_h} = 0$ $(i \in I_Z)$. Hence, the present solution is $x_{j_i} = 1/a_{ij_i}$ $(i \in I_P)$ and $x_{j_i} = 0$ $(i \in I_Z)$.

The choice of the next pivot is as follows. For each column j having $c_j < 0$, let $a_j = \infty$ if there is an $a_{ij} > 0$ with $i \in I_Z$. Otherwise, let $a_{j'} = \max a_{ij}$ for $i \neq h$, $i \in I_P$, $a_{ij} > 0$; and $a_j = \max (a_{hj}, a_{hj} + a_{j'})$. If $a_j < 0$, then the minimum of z does not exist. Otherwise, find the maximum of $|c_j|/a_j$. This gives the column s of the pivot, the row r having been determined when finding a_j.

The formulas for pivoting are similar to the usual ones but depend upon whether r is in I_P or I_Z and whether i equals h or is in I_P or I_Z.

[1] P. WOLFE and L. CUTTER, Experiments in Linear Programming, in R.L. Graves and P. Wolfe, Recent Advances in Mathematical Programming, 1963, p. 188.

[2] M.A. EFROYMSON, Some New Algorithms for Linear Programming (Abstract), in R.L. Graves and P. Wolfe, ibid., p. 219.

G.E. VERESS, I.M. PALLAI and G.A. ALMASY, Hungarian Academy of Sciences, Budapest

Application of Dynamic Programming to an Industrial Problem

The problem is to find the optimum control for a chemical plant. The objective of the optimization is to maximize the output of the product. The plant is assumed to operate in quasi-steady-state and its operation is characterized, for a given input, by a certain number of manipulated and disturbing variables.

Because of the considerable number of variables, the process should be divided into a sequence of the different units, where the flows linking the consecutive stages are described by a limited number of variables. Having partitioned the system, the optimization can be carried out by means of dynamic programming.

The dynamic programming principle can be applied to systems consisting of units of different type, provided that appropriate objective functions of these units are available. Since the aim of the optimization is to maximize the output of the product, terminal optimization is concerned. In terminal optimization, the overall objective function can always be described as a sum of certain suitably chosen stage objective functions. In order to construct such suitable functions, we introduce the concept of the "potential product", defining the stage objective functions as the increase of the potential product.

The objective function of the n^{th} stage, G_n, depends on the vectors of the input flow state variables $\underline{x}_n$, the stage control variables $\underline{y}_n$, and the stage disturbing variables $\underline{z}_n$, that is:

$$G_n = G_n(\underline{x}_n, \underline{y}_n, \underline{z}_n) \qquad n = 1,\dots,N.$$

With the knowledge of the stage objective functions, the overall objective function of the plant can be stated in accordance with the dynamic programming principle. That is, if h_n denotes the quantity of the potential product over n stages using an optimal policy, then

$$h_n(\underline{x}_n,\underline{z}_1,\dots,\underline{z}_n) = \max_{\underline{y}_n} \; G_n(\underline{x}_n,\underline{y}_n,\underline{z}_n) + h_{n-1}(\underline{x}_{n-1},\underline{z}_1,\dots,\underline{z}_{n-1}) \qquad n = 1,\dots,N$$

provided that

$$\underline{x}_{n-1} = \underline{x}_{n-1}(\underline{x}_n,\underline{y}_n,\underline{z}_n)$$

$$\underline{K}_n(\underline{x}_n,\underline{y}_n,\underline{z}_n) \geq \underline{K}_n^* .$$

The value of the function h_N is, for given disturbing variable vectors and using an optimal policy, the maximum output of the product.

This method is being applied to an ammonia production problem. Here the potential product is hydrogen (with the assumption of a constant hydrogen to nitrogen ratio). The unit models are considered blackbox models with quadratic objective functions and linear constraints.

MILAN VLACH, Delft University of Technology, Delft

On Conditions of Optimality in Linear Spaces

For the formulation of optimisation problems the following scheme is adopted. Let G be a set, let Ω be a subset of G and let R be a reflexive and transitive relation on G. An element of G is called optimal with respect to Ω and R if (a) $x \in \Omega$; (b) $y \in \Omega$ and $y R x \Rightarrow x R y$. Evidently if $x \in \Omega$, then x is optimal if and only if $G_x^R \cap \Omega = \emptyset$, where $G_x^R = \{y \in G \mid y R x$ and $\neg\ x R y$. Depending on the additional properties of G, Ω and R various "approximations" to the sets G_x^R and Ω at the point x can be developed. If these approximations have the property that $G_x^R \cap \Omega = \emptyset$ implies (or even, is equivalent to) emptiness of the intersection of the corresponding "approximations", then we obtain necessary (or even, necessary and sufficient) conditions of optimality. It is intended to present several realizations of this idea in real linear spaces and in some classes of real linear topological spaces and to demonstrate relations to other, both classical and nonclassical, conditions of optimality.

WILLIAM F. WALKER Jr. and FRANCIS A. MLYNARCZYK Jr., First National City Bank, New York

Solution to Nonlinear Least Squares Problems by Convex Programming

A convex programming procedure is applied to the nonlinear least squares problem involving power transformations and to some generalizations and extensions.

Consider a set of m observations $y_1, y_2, \ldots, y_m$ which are available at m set of conditions $x_{i1}, x_{i2}, \ldots, x_{in}, z_{i1}, z_{i2}, \ldots, z_{ik}$; $i \in I_1 = \{1, 2, \ldots, m\}$.

Let $$E(y_i) = \eta_i \tag{1}$$

and $$E(y_i - \eta_i)(y_j - \eta_j) = \{\begin{matrix} \sigma^2 & i = j \\ 0 & i = j \end{matrix} \tag{2}$$

where the functions η_i, $i \in I_1$, can be represented

$$\eta_i = \sum_{j=1}^{n} \beta_j x_{ij} + \sum_{j=1}^{k} \gamma_j z_{ij}{}^{\alpha_j} \tag{3}$$

also let

$$f_i(b,c,a,e) = \sum_{j=1}^{n} b_j x_{ij} + \sum_{j=1}^{k} c_j z_{ij}^{aj}, \quad i \in I_1 \tag{4}$$

$$g_i(a) = \sum_{j=1}^{k} d_{ij} a_j, \quad i \in I_2 = \{1, 2, \ldots, p\} \tag{5}$$

$$h_j(a) = a_j, \quad j \in J_1 = \{1, 2, \ldots, q\} \tag{6}$$

$$F(b,c,a,e) = \sum_{i=m}^{n} e_i^2 \tag{7}$$

In this paper we describe a procedure to obtain estimates $b_1, b_2, \ldots, b_n, c_1, c_2, \ldots, c_k, a_1, a_2, \ldots, a_k$ of the parameters $\beta_1, \beta_2, \ldots, \beta_n, \gamma_1, \gamma_2, \ldots, \gamma_k, \alpha_1, \alpha_2, \ldots, \alpha_k$ for the nonlinear least squares problem where the estimates $b_1, b_2, \ldots, b_n, c_1, c_2, \ldots, c_k$, $a_1, a_2, \ldots, a_k$ are not constrained by linear constraints i.e.

I. $\text{MIN } \{F(b,c,a,e) \mid f_i(b,c,a,e) = y_i, i \in I_1,\ l_j \le h_j(a) \le u_j, j \in J_1\}$

and for the nonlinear least squares problem where the estimates $b_1,b_2,\ldots,b_n,c_1,c_2,\ldots,c_k,a_1,a_2,\ldots,a_k$ are constrained by linear constraints i.e.

II. $\text{MIN } \{F(b,c,a,e) \mid f_i(b,c,a,e) = y_i, i \in I_1, l_j \le h_j(a) \le u_j,$

$$j \in J_1,\ g_i(a) = r_i,\ i \in I_2\}.$$

Computational experience and statistical implications are also discussed.

ROMAN L. WEIL and PAUL C. KETTLER, University of Chicago, Chicago

Rearranging Matrices to Block-Angular Form for Decomposition (and other) Algorithms

The reader is perhaps most familiar with the exploitation of block-angular structures in the context of mathematical programming. For example, if the rows and columns of the coefficient matrix of a mathematical program can be arranged so that the matrix has form

$$\begin{bmatrix} A_{01} & A_{02} & \cdots & A_{0n} \\ A_{11} & 0 & \cdots & 0 \\ 0 & A_{22} & \cdots & 0 \\ \cdot & \cdot & \cdots & \cdot \\ 0 & 0 & \cdots & A_{nn} \end{bmatrix}$$

then the well-known time-saving decomposition algorithms can be used. For other numerical calculations the discovery and exploitation of block angularity can be just as useful. We describe a method for finding block-angular structures in matrices. We specify the problem, outline the algorithm for solving it while illustrating with an example, detail the steps of the algorithm, and relate some of our computing experience. The techniques for the most part are not new. Their combination is. We have borrowed freely from directed and bipartite graph theory.

DOUGLAS J. WILDE, Stanford University, Stanford, and
J.M. SANCHEZ-ANTON, IBM Corporation, Madrid

Discrete Optimization on a Multivariable Boolean Lattice

Consider the problem of finding the minimum value of a scalar objective function whose arguments are the n components of 2n vector elements partially ordered as a Boolean lattice. If the function is strictly decreasing along any shortest path from the minimum point to its logical complement, then the minimum can be located precisely after sequential measurement of the objective function at $n + 1$ points. This result suggests a new line of research on discrete optimization problems.

A.C. WILLIAMS and CARL KALLINA, Mobil Research and Development Corp., Princeton, N.J.

Generalized Linear Programming

Our purpose is to give a new, elementary, self-contained account of the theory of linear programming in infinite dimensional spaces. The theory is developed from the appropriate generalization of the Farkas Lemma, thereby providing an infinite dimensional theory which closely parallels the finite dimensional theory.

The extensions of the theory of finite linear programs to the case of linear programs in more general linear spaces was initiated by Duffin in his fundamental paper in 1956 [1]. Duffin introduced the essential ingredients of a theory of programming in linear spaces; namely, (1) the replacement of constraint equalities and inequalities by the requirement that the indicated quantities lie in given closed convex cones, (2) the notion of subconsistency of the constraint set, and (3) the notion that the only two requirements that need to be specified for the topologies of the linear spaces involved are that they be locally convex, so that the separating hyperplane theorem holds, and also that they be reflexive so that primal and dual programs are symmetrically related.

Ben Israel, Charnes, Kortanek [2] have made a significant contribution by implicitly pointing out that to study programs in quite general linear spaces it suffices to study finite linear programs modified only by the replacement of constraint equalities and inequalities by general closed convex cones. Their paper, however, contains an error and is not complete, since their theory does not provide a one-to-one relationship between primal and dual properties. The present paper provides this completion. We shall show, also, how this leads immediately to the results of R.T. Rockafellar [3], in which existence of optimal solutions and absence of duality gaps are related to the notions of stability. Finally, the interpretation of the dual solution as a measure of the rate of change of the primal optimal value with the inhomogeneous term is extended to the more general case.

[1] Ann. of Math. Studies, no. 38, Princeton Univ. Press. 1956, pp. 157-170.

[2] Bull. Am. Math. Soc., 75 (1969), pp. 318-324.

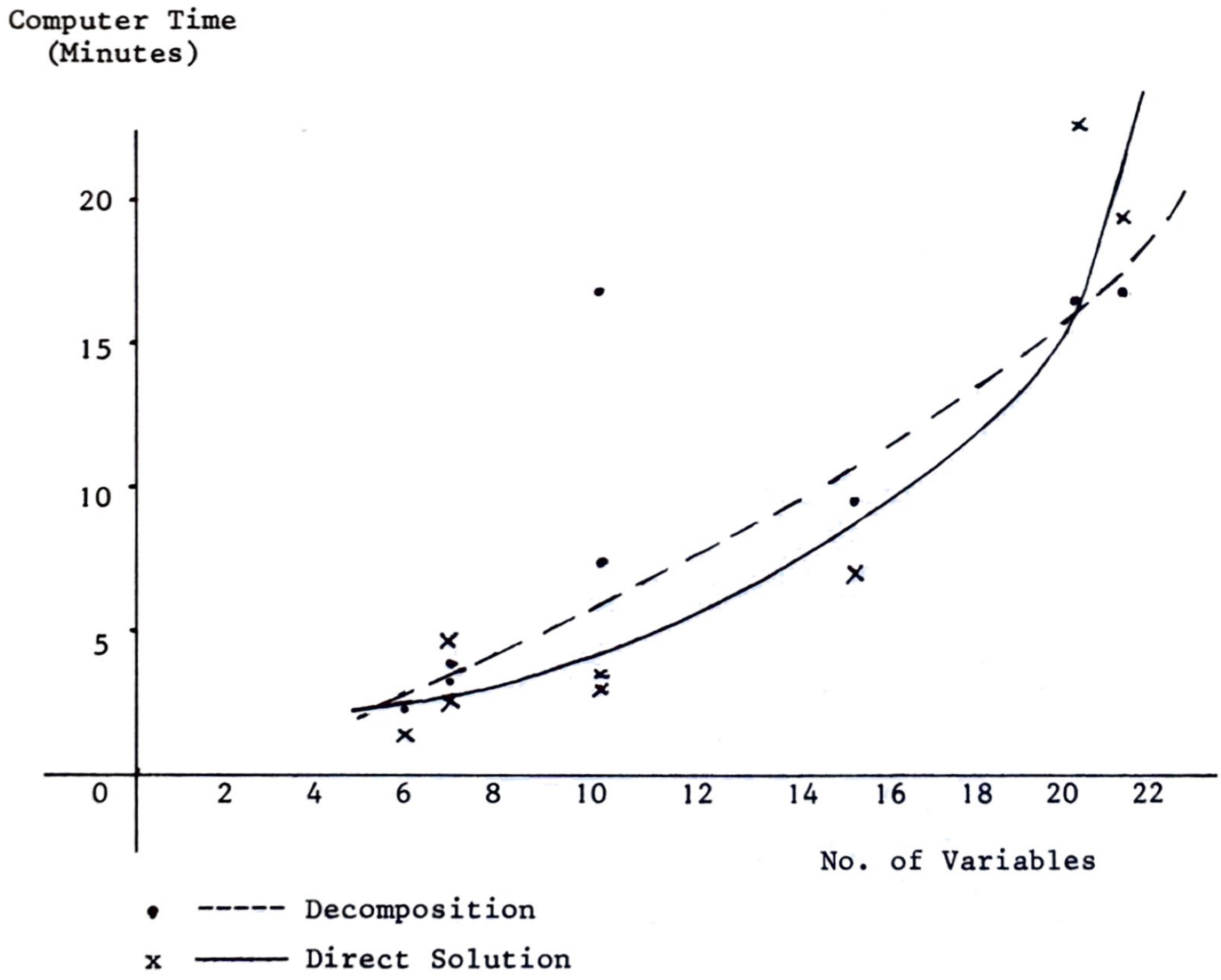
[3] Pac. J. of Math., 21, (1967), pp. 167-187.

K.P. WONG, University of Birmingham, Birmingham

Computer Implementation of Decomposition of Nonlinear Convex Separable Programmes in the Dual Direction

This will be a report on preliminary testings of Professor

T.O.M. Kronsjö's decompositional scheme. At the time of writing eight problems involving six to twenty-one variables and two to seventeen constraints, grouped into two to nine subproblems, have been solved directly and by decomposition. For the type of problem considered, the computer time required to solve a problem directly is generally less than that required by decomposition when the number of variables involved is fifteen or less. However, the former tends to rise steeply as the number of variables increases beyond fifteen. On the other hand, the latter rises at a relatively slow rate as the size of the problem increases. The accompanying diagram provides a bird's-eye view of the performance of decomposition vis-a-vis the direct method of solution. Further investigations are under way, the outcome of which will be presented at the symposium.



R.E.D. WOOLSEY, Colorado School of Mines, Golden

On telling it like it is in Integer Programming

The author will review the state of the art of actually attempting to solve (as opposed to devising new algorithms for) integer programming problems. This review will be based wholly on the computational experience of the author and others, using available codes and common sense. Various special techniques in formulation, and reformulation, will be pointed out which have materially aided progress on various types of codes and problems.
No panaceas will be presented, but rather the accumulated results of some years of bitter experience in attempting to run problems of type X on code Y.

W.W.G. YEH, A.J. ASKEW and W.A. HALL, University of California, Los Angeles

Optimal Planning and Operation of a Multiple Purpose Reservoir System

A method is developed for the determination of optimal contract levels and optimal operating policies of a multiple-purpose water resources system. The system is capable of producing firm power and firm water with flood control and mandatory releases for water quality, fish and wildlife and navigation as parametric constraints. The method used in obtaining the optimal solution involves the combination of dynamic programming and a modified gradient technique. Dynamic programming optimizes the sequential decision making process while the gradient technique forces the system to achieve its optimality by constraints which recursively prevent nonoptimal behavior. The input to the system is the streamflow records. The objective function is the returns obtained from the sale of firm

power and firm water for all planning periods. The state variable is the storage level in the reservoir while all the decisions are imbeded in a single decision variable, i.e., the release policy. The result of the analysis is a unique set of optimal releases for each period of the planning horizon such that the objective function is maximized. Due to the stochastic nature of the input, a first order Markov-chain model is utilized to generate equally likely hydrographs. The generated hydrographs maintain the first three moments of the historical records and would have more extreme events. These generated hydrographs are then used to determine a set of long-term firm contract levels. The mean, standard deviation and frequency distribution are determined. Upon this information an optimum policy risk relationship is derived. The method is applied to Shasta Reservoir in Northern California, U.S.A.

FRIEDA F. GRANOT, Université de Montréal, Montréal, and
PETER L. HAMMER, Université de Montréal and Technion, Haifa.

On the Use of Boolean Equations in Bivalent Programming

The "resolvant" $R(x_1, \ldots, x_n)$ of a(system) of linear or non-linear inequalities in 0-1 variables is defined as being a Boolean function with the property that the given system holds for those and only those values of the variables, for which $R(x_1, \ldots, x_n) = 0$. An efficient way of constructing the resolvant is given. It is shown, that the resolvant can be used for the construction of a sequence of Boolean functions $R_k(x_1, \ldots, x_n)$ $(k = 0,1,2,\ldots)$ such that the first k for which the equation $R_k(x_1,\ldots,x_n) = 0$ becomes infeasible (and there is a simple test of feasibility), determines the optimal value of the objective function under the given constraints; once this value is known, the determination of all the optimizing points can be carried out either by using the sim-

plex method (in the linear case), or by solving a Boolean equation.

Finally, it is shown that any linear or nonlinear system of constraints in 0-1 variables, is equivalent to the system of constraints of a generalized covering problem (AX ≥ b, where all the elements of the matrix A are equal to 0, +1 or -1), and a B-B-B-method is deviced for solving such problems.

[1] FRIEDA GRANOT and PETER L. HAMMER, On the Use of Boolean equations in Bivalent Programming. Technion, Mimeograph Series on Operations Research, Statistics and Economics, nr. 73, 1970.

AUTHOR INDEX