NOTICE: When government or other drawings, specifications or other data are used for any purpose other than in connection with a definitely related government procurement operation, the U. S. Government thereby incurs no responsibility, nor any obligation whatsoever; and the fact that the Government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data is not to be regarded by implication or otherwise as in any manner licensing the holder or any other person or corporation, or conveying any rights or permission to manufacture, use or sell any patented invention that may in any way be related thereto.
LINEAR PROGRAMMING IN A MARKOV CHAIN

Notes on Linear Programming and Extensions – Part 59

G. B. Dantzig and Philip Wolfe
LINEAR PROGRAMMING IN A MARKOV CHAIN
Notes on Linear Programming and Extensions – Part 59
G. B. Dantzig and Philip Wolfe
PREFACE

Part of the Project RAND research program consists of basic studies in mathematics, including linear programming. While the present Memorandum will be of particular interest to mathematical analysts and programmers concerned with inventory problems, it will also be of general interest to most mathematicians and computer scientists.
SUMMARY

This Memorandum concerns an infinite Markov process with a finite number of states in which the transition probabilities for each stage range independently over sets that either are finite or are convex polyhedra. A finite computational procedure is given for choosing those transition probabilities which minimize appropriate functions of the resulting equilibrium probabilities.
## CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>PREFACE</td>
<td>iii</td>
</tr>
<tr>
<td>SUMMARY</td>
<td>v</td>
</tr>
<tr>
<td>Section</td>
<td></td>
</tr>
<tr>
<td>1. INTRODUCTION</td>
<td>1</td>
</tr>
<tr>
<td>2. THE PROBLEM</td>
<td>3</td>
</tr>
<tr>
<td>3. FORMULATION AS A LINEAR PROGRAMMING PROBLEM</td>
<td>6</td>
</tr>
<tr>
<td>4. COMPUTATIONAL ALGORITHM—THE MASTER PROBLEM</td>
<td>9</td>
</tr>
<tr>
<td>The Iterative Step</td>
<td>10</td>
</tr>
<tr>
<td>Phase One</td>
<td>11</td>
</tr>
<tr>
<td>Phase Two</td>
<td>12</td>
</tr>
<tr>
<td>5. THE SUBPROBLEM AND PROOF OF TERMINATION</td>
<td>13</td>
</tr>
<tr>
<td>REFERENCES</td>
<td>17</td>
</tr>
<tr>
<td>LIST OF RAND NOTES ON LINEAR PROGRAMMING AND EXTENSIONS</td>
<td>19</td>
</tr>
</tbody>
</table>
LINEAR PROGRAMMING IN A MARKOV CHAIN

1. INTRODUCTION

Recent studies, cited below, have indicated considerable interest in optimization problems of a type formulable as the problem of choosing a set of distributions, constituting the transition probabilities of a finite Markov process, in such a way as to minimize certain "costs" associated with the process.

The following inventory problem is a typical example of this class: Let the \( n \) attainable levels of the inventory of an item constitute the \( n \) states of a Markov process. Transition from one state to another will occur at the end of each of an infinite sequence of time periods. Owing to the uncertain nature of supply and demand for the item (only its distributions are assumed known), the effect of a given inventory policy must be described as a distribution. For any inventory policy the probability \( p_{ij} \) of transition from state \( i \) to state \( j \) in one time period is known, as well as the cost \( c_{ij} \), dependent on the policy, which will be incurred if that transition is made. Under any policy the time series of inventory levels constitutes a Markov process described by the given probabilities. When an initial state for the first period has been given, the long-run probabilities \( \bar{p}_{ij} \) are then determined. Intuitively, \( \bar{p}_{ij} \) is the probability that, at a typical time period in the indefinite future, the transition from state \( i \) to state \( j \) will take
place. The "long-range expected cost" of using the particular policy is then defined as $\Sigma_{i,j} c_{i,j} \bar{p}_{i,j}$. Then the computational problem is that of minimizing this expected cost over all available inventory policies.

The formulation of such a problem as one of linear programming has been done by Manne [5], d'Epenoux [3], and Oliver [6] for problems in which the transition probabilities $p_{i,j}$ can be chosen, for each $i$ separately, from a given finite set of distributions. The same assumption concerning the available distributions is made by Howard [4] in his "dynamic programming" treatment of this class of problems. When the problem is formulated as a linear program, however, it can be efficiently attacked by means of a specialization of the decomposition algorithm for linear programming [2]. This makes it possible to broaden considerably the class of problems that can be handled, by permitting other descriptions of the sets of available alternatives. In the sequel two extreme cases are considered: on the one hand, the case described above; on the other, the case in which the distributions that may be used are restricted only in that they must satisfy certain linear inequalities. Since these two extreme cases are handled by essentially the same method, any intermediate cases of practical interest can readily be treated by the same technique.
2. THE PROBLEM

Throughout this paper \( n \) is a fixed integer. By distribution we shall mean an \( n \)-vector \( x = (x_1, \ldots, x_n) \) such that \( x_i \geq 0 \) (all \( i \)) and \( \Sigma x_i = 1 \). A Markov process is defined by \( n \) distributions \( P_i = (p_{i1}, \ldots, p_{in}) \), for \( i = 1, \ldots, n \), where \( p_{ij} \) is the probability of transition from state \( i \) of the process to state \( j \).

In the problem studied here, a particular Markov process is defined by a choice of distributions from certain sets (which will be assumed finite in this section and the next):

For each \( i = 1, \ldots, n \), let \( S_i \) be a finite set of distributions.

In addition, a "cost" \( c_i(P) \) is associated with each distribution \( P \) in \( S_i \):

For each \( i = 1, \ldots, n \), let \( c_i \) be a real-valued function on \( S_i \).

For \( P \) in \( S_i \), \( c_i(P) \) is thought of as a fee to be paid for the use of the distribution \( P \) when passing through state \( i \).

The particular manner in which \( S_i \) and \( c_i \) are described is not of great importance in the discussion which follows, but it does play an important role in the computational algorithm of Secs. 4 and 5. The more extensive discussion of

---

*The symbol "\( \Sigma \)" is used throughout as an abbreviation of \( \sum_{i=1}^{n} \).
Sec. 5 can be anticipated by the observation that the algorithm is aimed at handling either of the following two extremes: (a) $S_1$ is given as an arbitrary finite set, and $c_1$ as an arbitrary function on $S_1$; (b) a finite set of linear inequalities in $n + 1$ variables is given, defining an $(n + 1)$-dimensional polyhedron in such a way that the first $n$ coordinates of any point of this polyhedron form a distribution. The first $n$ coordinates of any extreme point of this polyhedron constitute a member $P$ of $S_1$, with $c_1(P)$ defined as the minimal $(n + 1)$-st coordinate of all extreme points whose first $n$ coordinates constitute $P$.*

If now particular $P_i$ in $S_1$ are chosen for each $i$, then a Markov process is defined. Let $x$ be an equilibrium distribution for this process—that is, a distribution satisfying relation (2.2) below. The "expected cost" of the process per stage, when the equilibrium $x$ obtains, is then

\[(2.1) \quad \Sigma c_1(P_i) x_i.\]

The Markov programming problem is that of choosing the $P_i$ in such a way that this expected cost is minimized.

*It will be seen from the discussion of case (b) in Sec. 5 that the restriction of $S_1$ to extreme points of the polyhedron is unnecessary, since even if all points were admitted, only extreme points would appear in the solution of the problem. This restriction is made because of the convenience of assuming $S_1$ to be finite.
Formally stated, the problem is as follows:

Determine \( P_1 \) in \( S_i \) \((i = 1, \ldots, n)\) such that (2.1) is minimized for all \( x \) for which

\[
(2.2) \quad x_1 \geq 0, \quad \sum x_1 = 1, \quad \text{and} \quad \sum x_1 P_1 = x.
\]

It will be convenient for the sequel to restate this problem in such a way that the equations (2.2) have constant right-hand sides.

For each \( i \), let \( T_1 \) be the set of all \( n \)-vectors

\[
(2.3) \quad Q_1 = (p_{11}, \ldots, p_{11} - 1, \ldots, p_{1n})
\]

for which \((p_{11}, \ldots, p_{1n}) = P_i\) is in \( S_i\), and define \( c_1 \) on \( T_1 \) by \( c_1(Q_1) = c_1(P_i)\), using the correspondence given. The problem may then be stated:

Determine \( Q_1 \) in \( T_1 \) \((i = 1, \ldots, n)\) such that

\[
(2.4) \quad \sum c_1(Q_1) x_1
\]

is minimized for all \( x \) for which

\[
(2.5) \quad x_1 \geq 0, \quad \sum x_1 = 1, \quad \text{and} \quad \sum x_1 Q_1 = 0.
\]

It is clear that any solution \( x; P_1, \ldots, P_n \) of the problem stated by (2.1) and (2.2) gives a solution \( x; Q_1, \ldots, Q_n \) of the problem (2.4, 2.5), and vice versa.
3. FORMULATION AS A LINEAR PROGRAMMING PROBLEM

The problem (2.4, 2.5) will be solved with the devices developed for the "decomposition" of linear programming problems of special structure [2], specialized to the case at hand. The central idea of this approach is the formulation of the problem as one of linear programming in which the data consist primarily of the coordinates of points of the set $T_1$. This will be done in this section. For each $i$, let the $K_i$ points of $T_1$ be $Q_i^k$ for $k = 1, \ldots, K_i$. As an abbreviation, let $c_{ik} = c_i(Q_i^k)$ for all $i, k$. Consider the linear programming problem:

Minimize

$$\sum_{k=1}^{K_i} c_{ik} y_{1k}$$

under the constraints

$$y_{1k} \geq 0, \quad \sum_{k=1}^{K_i} y_{1k} = 1, \quad \sum_{k=1}^{K_i} y_{1k} Q_i^k = 0.$$  \hspace{1cm} (3.2)

Theorem 1 below will show that this problem is equivalent to the problem of the previous section. In general, replacing a discrete problem by a continuous one in this manner can lead to a solution that is not discrete. The Lemma below shows, however, that for the problem studied here the solution of the continuous problem is itself sufficiently "discrete" to ensure equivalence: For each $i$, only a single $Q_i^k$ is actually involved in the solution of the problem (3.1, 3.2).
Lemma. There is a solution of the problem (3.1, 3.2) with the property that for each $i$ there is at most one $k$ for which $y_{ik} > 0$.

Proof. The coefficients and right-hand side of the linear programming problem (3.1, 3.2) are displayed in the table below, headed by their variables $y_{ik}$, where $p_{1j}^k$ denotes the appropriate component of the distribution $P$ corresponding to $Q_1^k$.

Table 1

COEFFICIENTS OF THE LINEAR PROGRAMMING PROBLEM

<table>
<thead>
<tr>
<th>$y_{11}$</th>
<th>$y_{12}$</th>
<th>$y_{21}$</th>
<th>$y_{22}$</th>
<th>$\cdots$</th>
<th>$\cdots$</th>
<th>$\cdots$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>$\cdots$</td>
<td>$\cdots$</td>
<td>$\cdots$</td>
</tr>
<tr>
<td>$p_{11}^1$</td>
<td>$p_{11}^2$</td>
<td>$p_{21}^1$</td>
<td>$p_{21}^2$</td>
<td>$\cdots$</td>
<td>$\cdots$</td>
<td>$\cdots$</td>
</tr>
<tr>
<td>$p_{12}^1$</td>
<td>$p_{12}^2$</td>
<td>$\cdots$</td>
<td>$p_{22}^1$</td>
<td>$p_{22}^2$</td>
<td>$\cdots$</td>
<td>$\cdots$</td>
</tr>
<tr>
<td>$\vdots$</td>
<td>$\vdots$</td>
<td>$\vdots$</td>
<td>$\vdots$</td>
<td>$\vdots$</td>
<td>$\vdots$</td>
<td>$\vdots$</td>
</tr>
<tr>
<td>$p_{1n}^1$</td>
<td>$p_{1n}^2$</td>
<td>$p_{2n}^1$</td>
<td>$p_{2n}^2$</td>
<td>$\cdots$</td>
<td>$\cdots$</td>
<td>$\cdots$</td>
</tr>
</tbody>
</table>

It is a basic property of linear programming problems [1] that, when a solution exists, there is a solution having exactly $r$, say, positive components for which the submatrix consisting of those columns of the coefficient matrix associated with the positive components has rank $r$. For this problem, denote by $B$ the $(n + 1) \times r$ submatrix of
Given by that property; the associated solution will be
the one whose existence the lemma asserts. (As a matter of
fact, the simplex method solution of this linear programming
problem will yield a solution of just this type.)

Let \( s \) be the number of rows of \( B \) in which may be
found an entry of the form \( p_i^k - 1 \). Excluding the first
row, the other \( n - s \) rows have only nonnegative entries;
since their right-hand sides are zero and their variables
\( y_{1k} \) positive, these rows must in fact vanish. Hence \( B \) has
just \( s + 1 \) nonvanishing rows. The nonvanishing rows are,
however, linearly dependent (the sum of all rows but the
first is zero), whence the rank of \( B \) is at most \( s \), that
is, \( s \geq r \). Since \( B \) has just \( r \) columns, it follows that
at most one entry of the form \( p_i^k - 1 \) can be found in any
row of \( B \), so that at most one column of (3.3) can be found
in \( B \) for \( j \) given, which proves the lemma.

Theorem. The programming problems (2.4, 2.5) and (3.1,
3.2) are equivalent, with solutions related in the following
way:

Given \( y_{1k} \) solving (3.1, 3.2) and satisfying the
conclusion of the lemma, for each \( i \) let

\[
\begin{align*}
x_1 &= y_{1k} \\
Q_i &= Q_{1k}^k \\
x_i &= 0 \\
Q_i & \text{ arbitrary in } S_i
\end{align*}
\]

where \( y_{1k} > 0 \) for some \( k \), if \( y_{1k} = 0 \) for all \( k \).
On the other hand, given $x_1, Q_1$ solving (2.4, 2.5), let

$$y_{1k} = \begin{cases} x_1 & \text{for } k \text{ such that } Q_1^k = Q_1, \\ 0 & \text{otherwise}. \end{cases}$$

The proof is obvious.

4. COMPUTATIONAL ALGORITHM—THE MASTER PROBLEM

The linear programming problem formulated in the last section has only $n + 1$ equations, but it has $\Sigma K_i$ variables, a number which may be very large, and in fact not even known for problems whose data are given implicitly. The revised simplex method [1] is particularly advantageous for problems having many more variables than constraints. The decomposition algorithm uses this efficiency of the revised simplex method by clearly separating the considerations involving the constraints alone from those connected with the handling of the variables. That part of the problem involving the constraints is called the "master problem," and its handling is set forth in this section. That part of the problem involving the variables, called the "subproblem," is dealt with in the next section. It will be seen that the work of treating the master problem consists of little more than the application of the revised simplex method to the Markov programming problem as formulated in Sec. 3. The general iterative step is given below, followed by the procedures for initiating the iterative process and for passing from the
determination of an initial feasible point (Phase One) to the
determination of the solution of the problem (Phase Two).
(The phenomenon of degeneracy plays the same role in this
algorithm as in any linear programming problem, and it will
be supposed that standard methods [1] may be relied upon when
necessary.)

The Iterative Step

At any step in the course of the solution of the problem
(3.1, 3.2) by the revised simplex method, there will be at
hand some \( n + 1 \) column vectors \( \overline{Q}^1, ..., \overline{Q}^{n+1} \) (of length
\( n + 1 \)) constituting a "feasible basis"; that is, they are
linearly independent, and the right-hand side of the equations
(3.3) may be expressed as a nonnegative linear combination of
them. (The weights in this linear combination, which of
course constitute a solution of equations similar to (3.3)
deriving their coefficients from the \( \overline{Q}^1 \), are called
collectively a "basic feasible point.")

Let the "cost" \( \overline{c}^i \) be associated with the column \( \overline{Q}^i \),
for \( i = 1, ..., n + 1 \). The "prices," assumed known,
associated with this basis are defined to be the components of
the \( (n + 1) \)-vector \( \overline{\pi} = (\pi_1, ..., \pi_{n+1}) \) satisfying the relation-
ships \( \pi \overline{Q}^i = \overline{c}^i (i = 1, ..., n + 1) \).

One iteration of the simplex method consists of the
following steps:

(1) Find a column \( Q \) of the matrix (3.3) which, with
its associated cost \( c \), satisfies the relation
\[ (4.1) \quad c - \pi q < 0. \]

Commonly, the column chosen is that for which \( c - \pi q \) is minimal. (This is the only point in the revised simplex method at which all the columns—i.e., all the variables—in the problem come into play. This step forms the "subproblem," which will be discussed in Sec. 5.)

(2) If no column satisfying (4.1) can be found, then the current basis is "optimal," and the solutions of the equations (3.3) solve the linear programming problem.

(3) Otherwise, add the column found in Step 1 to the current basis, and remove one column in such a way (given by the rules of the simplex method) that the remainder still forms a feasible basis. Calculate the new prices, and begin again.

Phase One

The algorithm can be started with precisely the same device, called Phase One, used for the general linear programming problem. This device consists in augmenting the problem with \( m + 1 \) "artificial" variables in terms of which an initial feasible basis and the prices associated with the corresponding initial feasible basis are readily given. The algorithm can then be applied to the problem of removing the artificial variables. After this has been done, the required starting conditions for the ordinary application of the algorithm are automatically met.
For \( i = 1, \ldots, n + 1 \), let \( y_i \) be a nonnegative variable; let \( I_i \) be the \( i \)-th column of the \((n + 1)\)-order identity matrix; and let \( c_i = 1 \) be the cost associated with the variable \( y_i \). For this phase, replace all the costs \( c_{ik} \) of the original problem with zeroes.

Designating \( I_1, \ldots, I_{n+1} \) as the initial feasible basis, employ the iterative step outlined above until the linear form \( \sum_{i=1}^{n+1} y_i \) has been minimized. (Note that the initial feasible point is \((y_1, \ldots, y_{n+1}) = (1,0,\ldots,0)\) and that the initial prices are \( \pi = (1,1,\ldots,1) \).

The above process will reduce the form \( \sum_{i=1}^{n} y_i \), and hence each \( y_i \) separately, to zero. (If it did not, then the equations (3.2) would have no solution, which is impossible.) Owing to the linear dependence of the equations (3.2), some of the starting columns \( I_i \) will remain in the feasible basis at the end of Phase One; this can be shown to cause no difficulty in the ensuing process [1].

**Phase Two**

When Phase One is finished, restore the deleted costs \( c_{ik} \) to the columns \( Q^k \), using these costs from now on in the determination of the prices \( \pi \). Repeat the iterative step until it terminates in its part (2).

At termination, associated with each \( Q^k \) in the final feasible basis is a component of the "feasible point," the weight given \( Q^k \) in expressing the right-hand side of the equations as a linear combination of the columns of the basis.
For $i = 1, \ldots, n$, according to the Theorem of Sec. 3, there can be no more than one $Q_i^k$ in $T_i$ in the basis having positive weight; thus let

$$x_i = \begin{cases} 
\text{weight for } Q_i^k, & \text{if positive}, \\
0, & \text{otherwise}.
\end{cases}$$

The resulting $(x_1, \ldots, x_n)$ is the solution of the problem (2.2).

5. THE SUBPROBLEM AND PROOF OF TERMINATION

As mentioned previously, a detailed discussion of part (1) of the iterative step of Sec. 4, the "pricing out" operation in the ordinary revised simplex method, will be given in this section. Given the quantities $\pi$, the problem is to determine some column $Q$ and its associated cost $c$ for which

$$(5.1) \quad c - \pi Q < 0,$$

where $Q$ may come from any of the sets $T_i$. How this is done depends on the nature of the description of the original sets $S_i$ from which the $T_i$ were obtained. It is evident that the problem of satisfying (5.1) from among the union of all the $T_i$ may be "decomposed" into $n$ problems, the $i$-th one of which $(i = 1, \ldots, n)$ is that of satisfying (5.1) for $Q$ in $T_i$. If all of these "subproblems" can be solved, then the stated problem has been solved.
For each \( i = 1, \ldots, n \), one of the two "extreme" cases (a) or (b) mentioned in Sec. 2 may obtain. (Some "intermediate" case might also be considered, but this will not be done here.)

(a) \( S_1 \) is given directly as a finite set of distributions, a cost \( c_1(P) \) being associated with each member \( P \) of \( S_1 \).

(b) There is given a finite set of linear relations

\[
(5.2) \quad g_j(z) \geq 0, \quad j = 1, \ldots, m,
\]

in the \( n + 1 \) variables \( (z_1, \ldots, z_{n+1}) = z \), such that if \( z \) satisfies \( (5.2) \), then \( (z_1, \ldots, z_n) \) is a distribution; \( S_1 \) is defined to be the set of all \( P = (z_1, \ldots, z_n) \) such that for some \( z_{n+1} \), \( z = (P; z_{n+1}) \) is an extreme point of the set of all \( z \) satisfying \( (5.2) \); and for \( P \) in \( S_1 \), \( c_1(P) \) is defined to be the smallest value of \( z_{n+1} \) for which \( (P; z_{n+1}) \) is such an extreme point. (The index \( i \) has been omitted above; of course, the relations \( (5.2) \) may be different for each \( i \), or even absent.)

For the case (a), there is not much to be said. Phrased via the definition \( (2.3) \) in terms of \( S_1 \), relation \( (5.1) \) urges the selection of \( P \) in \( S_1 \) for which

\[
(5.3) \quad c_1(P) - \pi P + \pi_1 < 0.
\]

Such a \( P \) will yield through \( (2.3) \) a column \( Q \) satisfying \( (5.1) \).

Case (b) is more interesting, in view of the fact that the extreme points of the polyhedron defined by \( (5.2) \) have not
been assumed to be available in advance. Replacing \( P \) and 
\[ c_1(P) \] in (5.3) by their definitions in this case, it is 
desired to choose 
\[ z = (z_1, \ldots, z_{n+1}) \] under the constraints 
(5.2) in such a way that 
\[
(5.4) \quad z_{n+1} - \sum_{j=1}^{n} \pi_j z_j + \pi_1 < 0.
\]
This is nearly a linear programming problem. If the 
customary procedure for the simplex method—that of making 
the left-hand side of (5.3) as small as possible—is 
followed, then the task is precisely a linear programming 
problem: Under the constraints (5.2), minimize the left-hand 
side of (5.4). Once performed, if the result is not negative, 
this minimization is of no interest; but if it is negative, 
then the column 
\[ Q = (z_1, \ldots, z_1 - 1, \ldots, z_n) \] and its cost 
\[ c = z_{n+1} \] constructed from the solution of the problem 
satisfy equation (5.1). Furthermore, \( Q \) will be an extreme 
point of the polyhedron.

The complete solution of the subproblem then goes as 
follows: For each \( i = 1, \ldots, n \), attempt to satisfy (5.1) 
from \( T_i \)—or equivalently, attempt to satisfy (5.3), or 
(5.4), from \( S_i \). If this can be done for any \( i \), part (1) 
of the iterative step of Sec. 4 can be accomplished. (It is 
indifferent to the fact of the convergence of the procedure, 
although probably not to its rate, whether or not the \( i \) for 
which (5.1) is minimized is chosen.) If, on the other hand, 
(5.1) cannot be accomplished for any \( i \), then part (2) of the 
iterative step obtains, and the procedure has terminated.
It remains only to show that the algorithm is finite. This follows immediately, however, from the finiteness of the simplex algorithm for linear programming [1], for as described in Sec. 4, this algorithm is precisely the simplex method applied to the linear programming problem (3.1, 3.2). Whether the sets $T_1$ of columns are described in the manner (a) or (b) above, they are finite in number, and the proof is complete.
REFERENCES


### LIST OF RAND NOTES ON LINEAR PROGRAMMING

| RM-1266 | Part 3: Computational Algorithm of the Simplex Method by G. B. Dantzig, October 26, 1953. (ASTIA No. AD 114136) |


RM-1375 Part 18: Status of Solution of Large-scale Linear Programming Problems, by C. B. Dantzig, November 30, 1954. (ASTIA No. AD 6396)


RM-1452 Part 25: The Elimination Form of the Inverse and Its Application to Linear Programming, by H. M. Markowitz, April 9, 1955. (ASTIA No. AD 86956)


RM-1736 Part 32: Solving the Transportation Problem, by L. R. Ford and D. R. Fulkerson, June 20, 1956. Published in Management Science, Vol. 3, No. 1, October, 1956, pp. 24-32. (ASTIA No. AD 111816)


Part 35: Discrete-variable Extremum Problems, by G. B. Dantzig, December 6, 1956. Published in Operations Research, April, 1957. (ASTIA No. AD 112411)


Part 37: Concerning Multicommodity Networks, by J. T. Robacker, September 26, 1956. (ASTIA No. AD 112392)


Part 42: Linear Programming and Structural Design, by W. Prager, December 3, 1957. (ASTIA No. AD 150661)

RM-2158  Part 44: Transient Flows in Networks, by D. Gale, April 11, 1958. (ASTIA No. AD 150686)


Part 54: An Algorithm for the Mixed Integer Problem, by Ralph Gomory, July 7, 1960. (ASTIA No. AD 243212)

Part 55: On the Solution of Two-Stage Linear Programs under Uncertainty, by George Dantzig and Albert Madansky, July 1961. (ASTIA No. AD 263219)

Part 56: Methods of Solution of Linear Programs under Uncertainty, by Albert Madansky, April 1961. (ASTIA No. 257816)

Part 57: The Decomposition Algorithm for Linear Programming, by George Dantzig and Philip Wolfe, August 1961. (ASTIA No. AD 263628)