

UNCLASSIFIED

**A
D
8
6
3
9
6**

Armed Services Technical Information Agency

Reproduced by

DOCUMENT SERVICE CENTER

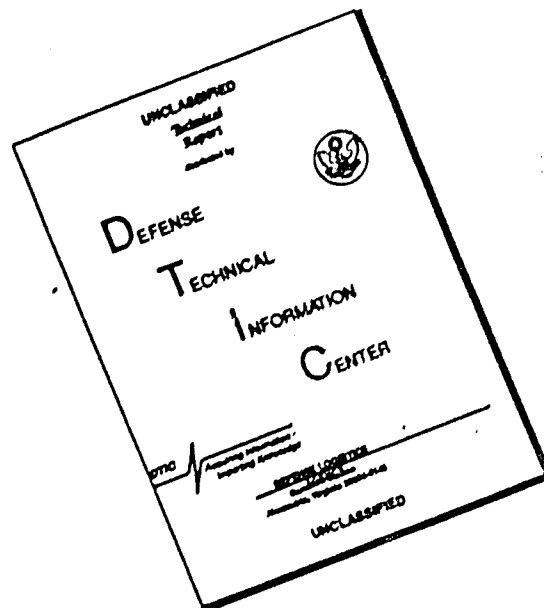
KNOTT BUILDING, DAYTON, 2, OHIO

This document is the property of the United States Government. It is furnished for the duration of the contract and shall be returned when no longer required, or upon recall by ASTIA to the following address: Armed Services Technical Information Agency, Document Service Center, Knott Building, Dayton 2, Ohio.

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.

UNCLASSIFIED

DISCLAIMER NOTICE



THIS DOCUMENT IS BEST QUALITY AVAILABLE. THE COPY FURNISHED TO DTIC CONTAINED A SIGNIFICANT NUMBER OF PAGES WHICH DO NOT REPRODUCE LEGIBLY.

AD No. 86396

ASTIA FILE COPY

U.S. AIR FORCE
Project
RESEARCH

RAND FC
MEMORANDUM

This is a working paper. It may be expanded, modified, or withdrawn at any time. The views, conclusions, and recommendations expressed herein do not necessarily reflect the official views or policies of the U. S. Air Force.

The RAND Corporation
SANTA MONICA • CALIFORNIA

U S AIR FORCE

PROJECT RAND

RESEARCH MEMORANDUM

Notes on Linear Programming: Part XVIII
STATUS OF SOLUTION OF LARGE SCALE
LINEAR PROGRAMMING PROBLEMS

Talk before
Institute of Management Sciences
Pittsburgh, Pennsylvania
October 22, 1954

George B. Dantzig

RM-1375

30 November 1954

Assigned to _____

ASTIA (1)

This is a working paper. It may be expanded, modified, or withdrawn at any time. The views, conclusions, and recommendations expressed herein do not necessarily reflect the official views or policies of the United States Air Force.

The RAND Corporation

1700 MAIN ST • SANTA MONICA • CALIFORNIA

SUMMARY

Unless techniques are developed to solve special classes of linear programming problems, it is likely that only models involving between 100 and 200 equations will be successfully computed by general techniques now available. This paper discusses the need to compute large scale systems and points out some of the common characteristics of many practical models which promise to lead to short cut procedures.

STATUS OF SOLUTION OF LARGE SCALE
LINEAR PROGRAMMING PROBLEMS

Talk before
Institute of Management Sciences
Pittsburgh, Pennsylvania
October 22, 1954

by

George B. Dantzig

The purpose of my talk is to discuss with you the possibilities for solving large scale linear programming problems. I shall sound both an optimistic and a pessimistic note. The pessimistic note concerns the ability of the problem formulator, either amateur or professional, to develop models that are large scale. The pessimistic note also concerns the inability of the problem solver to compute models by general techniques when they are large scale. If this is so, is not the great promise that the linear programming approach will solve scheduling and long range planning problems with substantial savings to the organizations adopting these methods but an illusion and a snare? Are the big problems going to be solved as they have always been solved — by a detailed system of on-the-spot somewhat natural set of priorities that resolve every possible alternative as it arises?

Let us consider a modest planner who is concerned with the expansion of motor production — let us say a special type motor that requires a special type of steel and must use tools fabricated from this steel and the tools which fabricate these tools also use this steel. The tools that fabricate steel we will call steel

model above is at best only useful as an over-all type of guide, but hardly detailed in a realistic sense.

Let me cite an example from another area — the problem of routing cargo aircraft. Let the variable x_{1jk} represent the number of aircraft of type k routed between city i and j .

Let us distinguish between six types of aircraft, ten time periods, and twenty cities. In addition, consider a second set of variables $y_{1j\ell}$ which is the tons of cargo shipped between city i and j on the way to ℓ . Our equations become

(2)

$$\text{Aircraft in} = \text{Aircraft out: } \sum_j x_{ojk} = \sum_i x_{iok} \quad (k=1, \dots, 6)(o=1, \dots, 20)$$

$$\text{Cargo in} = \text{Cargo out: } a_{c\ell} + \sum_j y_{ojk} = \sum_i y_{iok} + b_{c\ell} \quad (k=1, \dots, 20)(o=1, \dots, 20)$$

$$\text{Tonage Cap.} \geq \text{Tonage Req.: } \sum_k \lambda_{ij} x_{ijk} = \sum_\ell y_{1j\ell} \quad (i=1, \dots, 20)(j=1, \dots, 20)$$

$$\text{Plane Months Available: } \sum_i \sum_j \mu_{ij} x_{ijk} = P_k$$

As we see again such a system involving only a few cities, type aircraft, and cargo destinations generate easily a system in a 1000 equations in 10,000 unknowns. Superficially, a very discouraging situation.

For a number of years research has been going on, now centered at RAND, specifically directed toward the solution of large scale systems and my talk to you today is in a nature of a progress report. At RAND we have a simplex code which can successfully solve a system of 100 equations in any number of unknowns in about five hours. By successful I mean it has done so on a number of occasions and has produced accurate usable results.

Theoretically the machine could do 200 equations but it would take so long that, with machine failures, restarts, etc., it does not appear practical to do so. We also have a special code for a special type of problem called the "machine processing" model. Mathematically, this may be described as a slightly generalized type of transportation model. A transportation model as developed by Hitchcock and later independently by T. C. Koopmans is of the form

$$(3) \quad \begin{cases} \sum_j x_{1j} = a_1 & i=1,2,\dots,m \\ \sum_i x_{ij} = b_j & j=1,2,\dots,n \end{cases}$$

$$\sum x_{ij} c_{ij} = \text{Min}$$

We can slightly generalize it by replacing the coefficient of x_{ij} in the first equation by λ_{ij} .

The special RAND code which only works for a particular choice of c_{ij} can work out solutions to a 300-equation system of this

type in about an hour.

Basically, our approach has been to study a large number of models as they arise in practice to discover if they have any characteristic that can be taken advantage of computationally. When we have developed a theoretical approach we often take a watered-down version of the model that can be solved directly by the general code and compare this with the new method. Often these new methods are powerful enough to do by hand what cannot be achieved by machine. Of course, situations of this sort are not new. A transportation problem of the Hitchcock-Koopmans variety involving, say, a hundred rows and columns combined or one involving 10 rows and hundreds of columns can be nicely handled by clerks. Recently, Cooper and Charnes and others have reported that they were successful in solving certain very large systems by hand methods. All of this gives encouragement to believe that many large scale systems will be successfully solved in the future.

Except in highly specialized models of the transportation type or others where unusual characteristics can be taken advantage of, greater progress will be made by developing in the first phase consolidated versions of the model. There are several reasons for this. In the first place this effort results in a model which is often very useful in itself. Secondly, it provides an excellent dry run for methodology. In a word, it is better for administrative and technical reasons to keep the model initially small.

Inevitably the size of models does increase. I wish to discuss some devices that can greatly reduce the amount of computation. In the first place there appears to be a number of important characteristics commonly found in practical models.

- (a) Matrix of coefficients is composed of blocks of zeros.
- (b) Within non-zero blocks most elements are zero.
- (c) Matrix is often block triangular.
- (d) Many variables have simple upper bounds or satisfy a system of secondary constraints, most of which are non-active in any given problem.

Block triangularity is one of the most promising characteristics to exploit. The basis in the simplex method consists in a selected subset of the columns of the coefficients. It is square, non-singular matrix which from one cycle to the next is modified by one column. Its inverse or equivalent is needed on each iteration. If the basis is of this form (1), and let us suppose the diagonal submatrices designated A (but not necessarily the same) are square and non-singular. If it has this property it will be referred to as "square block triangular." This form is ideal because it is necessary only to have the inverses of the diagonal submatrices. Furthermore, from one iteration to the next it becomes only necessary to modify one of these smaller inverses. Strictly speaking most bases taken from block triangular systems do not conform to this ideal situation but are very close to being of the required form. In this case, it is possible by transforming the matrix slightly to get into the required square block triangular form.

When a matrix is composed largely of zeros where the zeros are in no obvious pattern, it is often practical to solve directly for the prices and the representations of the vectors entering the basis rather than to solve for them by means of the inverses of successive bases. The transportation model is a classical case where this approach has paid off.

When many variables have simple upper bounds, it is no longer necessary to add one more variable and equation for each such restraint. Instead, it is possible to slightly modify the original simplex algorithm and apply it to the system excluding the upper bounds. Essentially, one replaces a variable by $\bar{x} = b - x$ where b is an upper bound for x whenever in the simplex process x reaches its upper bound.

In many problems there are equations that may be considered as forming a set of side conditions; for example, conditions that capacity of certain machines is never exceeded or the characteristics of certain products; e.g., its viscosity is within specifications, etc. In most problems only a small subset of these "secondary" constraints are likely to be active, i.e., at their critical value—the others being well within specifications or capacities. In such cases it is recommended that the linear programming problem be first solved without regard to these secondary constraints. Then the system is enlarged to include the secondary constraints and an initial basis is obtained by augmenting the final basis of the smaller problem. This will result in a basis

in which not all variables associated with the secondary constraints are positive. However, in this form the dual simplex procedure of Lemke may be employed. Often in practical cases only a few iterations are needed to clean up the negative variables and obtain an optimum solution.

Details of the three procedures outlined above may be found in RAND "Notes on Linear Programming: Parts VIII, IX, X — Upper Bounds, Secondary Constraints, and Block Triangularity in Linear Programming," dated 4 October 1954.