Reprinted from:
MATHEMATICAL PROGRAMMING

ACADEMIC PRESS, INC., NEW YORK AND LONDON

# On the Need for a System Optimization Laboratory

*GEORGE B. DANTZIG, R. W. COTTLE,*
*B. C. EAVES, F. S. HILLIER, A. S. MANNE,*
*G. H. GOLUB, D. J. WILDE, AND R. B. WILSON*[1]

## Need.

From its very inception, it was envisioned that linear programming would be applied to very large, detailed models of economic and logistical systems [Wood and Dantzig (1947)]. Kantorovich's 1939 proposals, which were before the advent of the electronic computer, mentioned such possibilities. In the intervening 25 or so years, electronic computers have become increasingly more powerful, permitting general techniques for solving linear programs to be applied to larger and larger practical problems. In the author's opinion, however, additional steps are necessary if there is to be significant progress in solving certain pressing problems that face the world today.

The conference on Large-Scale Resource Allocation Problems held at Elsinore, Denmark, July 5-9, 1971 represents an historic first because it demonstrates that optimization of very large-scale planning problems can be achieved on significant problems.[2] I cite some examples from the conference:

Arthur Geoffrion's paper "Optimal Distribution System Design" is of interest because (1) it described the successful solution of a large-scale problem from commerce, (2) it involved discrete variables (representing the integer number of warehouses to be built or closed down), (3) it successfully

---

[1, 2]See footnotes following references.

combined a variety of advanced techniques in a single computer program.

Leon Lasdon's paper "Uses of Generalized Upper Bounding Methods in Production Scheduling" is of interest because (1) it not only described a successful large-scale application (this time to a rubber factory), (2) made use of advanced techniques, but also, (3) because it showed it was possible to automatically schedule day-to-day operations consistent with the long-term goals, i.e., it successfully combined short and long-term planning goals of an enterprise.

The papers by several authors (for example those of Abadie, Buzby, Huard) are particularly noteworthy because they described the successful solution of real problems (Electric Energy Production and Olefin Production) that were essentially non-linear and large-scale in nature.

Society would benefit greatly if certain total systems can be modeled and successfully solved. For example, crude economic planning models of many developing countries indicate a potential growth rate of GNP of 10% to 15% per year. To implement such a growth (aside from political difficulties) requires a carefully worked out detailed model and the availability of computer programs that can solve the resulting large-scale systems. The world is currently faced with difficult problems related to population growth, availability of natural resources, ecological evaluation and control, urban redesign, design of large-scale engineering systems (e.g., atomic energy, and recycling systems), and the modeling of man's physiological system for the purpose of diagnosis and treatment. These problems are complex, are urgent and can only be solved if viewed as total systems. If not, then only patchwork, piecemeal solutions will be developed (as it has been in the past) and the world will continue to be plagued by one crisis after another caused by poor planning techniques. For solutions, these problems require total system planning, modeling and optimization.

It is my belief that it is necessary at this time to create several system optimization laboratories where enough critical mass would exist that representative large-scale models (of the type referred to above) could be practically modeled and numerically solved. Solving large-scale systems





cannot be approached piecemeal or by publishing a few theoretical papers. It is a complex art requiring the development of a whole arsenal of special tools.

## Background.

The optimization of large-scale systems is technically an extremely difficult subject. Historically, starting with U. S. Air Force problems in 1947, linear programs were formulated to solve just such systems. These problems involved systems of interlocking relations involving many planning periods, combat units, types of personnel and supply. It led to thousands of equations in many thousands of unknowns. This was beyond computational capabilities. It was necessary to severely restrict the class of practical problems to be solved. Starting around 1954 a series of purely theoretical papers began to appear on how to efficiently solve large systems and by 1970 they numbered about 200. There was little in the way of implementation. Exceptions were the out-of-kilter algorithms for network flow problems proposed by Ford and Fulkerson [1958] and the "decomposition principle" of Philip Wolfe and myself which had been tried but with variable results [1960]. On the other hand a more modest proposal of Richard Van Slyke and myself (generalized-upper bounds) has been very successful [1967]. Apparently a great deal in the way of empirical testing of ideas is necessary and this has not been easy to do because the test models have to be complex to be pertinent and cost a great deal of money to program and solve. Therefore progress has been slow up to the time of the Elsinore meeting.

Since its origins in the development of transport allocation methods in the early 1940's, and especially since the introduction of the Simplex Method of linear programming in 1947, the power of the methods of mathematical programming, and the range of effectiveness of its applications, have grown enormously. In the intervening decades the methodology has been extended to include non-linear and integer programming, dynamic programming and optimal control, and a host of other types of optimization problems.

The range of applications has been extended from simple allocation problems to an enormous variety of problems in intertemporal allocation and investment planning, engineering design and optimization, and scientific studies of physical, biological, and ecological systems. There is, in fact, no end foreseeable to the applications of mathematical programming to a number of important (and crucial) optimization problems.

Some Examples of Important Applications.

A. INVESTMENT PLANNING (INTERTEMPORAL ALLOCATION): Problems of aggregate economic planning for a (developing) country, present an exploitable special structure that has been studied intensively and has great potential. Related structures occur in problems of dynamic programming and optimal control. Related but more complicated structures arise, for example, in problems of plant location and time-phasing, and in investment planning in general in the firm.

B. DECENTRALIZED ALLOCATION: The origin of the modern methods of decomposition, and still one of the major areas of application, is the class of decentralized allocation problems, in which scarce resources are to be allocated among several otherwise independent enterprises or "divisions". Closely related is the class of problems of two-stage allocation under uncertainty, for which in the linear case it is known that the dual problem is one of decentralized allocation. It is of particular importance to realize that the "divisional subproblems" may themselves be of a special structure (e.g., a transportation problem) which can be exploited.

C. ENGINEERING DESIGN AND OPTIMIZATION: A variety of engineering design and process optimization problems present specially-structured mathematical programs for which the structural features are highly dependent on the process being studied. Problems of this type illustrate the need for a flexible and comprehensive software package from





which components can be drawn to build up models of very complex systems.

D. PHYSICAL, BIOLOGICAL, AND ECOLOGICAL SYSTEMS: A number of problems in the physical sciences (e.g., X-ray crystallography) and biological sciences (e.g., models of body processes) present specially-structured mathematical programming problems. An extreme example are models of ecological systems in which the many and varied relationships among the components again require a flexible and comprehensive software package.

E. URBAN PLANNING: Coordinated planning of the many component subsystems (e.g., transport, recreation, education, etc.) of an urban environment presents a complex systems optimization problem for which ordinarily the most powerful and flexible methods are required.

F. LOGISTICS: Coordinated logistical support for any large industrial (e.g., warehousing and transport) or government (military) activity normally presents a system optimization problem of considerable size and complexity, but with exploitable structural features.

G. TRANSPORTATION SYSTEMS: Various problems concerning the design of transportation systems can be formulated as network optimization models of a combinatorial nature. These models typically have very special mathematical-programming structures for which highly efficient algorithms can be devised.

The Functions of a Systems Optimization Laboratory.

The purpose of such a laboratory would be to support the development of computational methods and associated computer routines for numerical analysis and optimization of large-scale systems. The ultimate objective of the development effort would be to provide an integrated set of computer routines for systems optimization which:

I. is freely and publicly available to users of government, science, and industry,
II. is thoroughly practical and widely useful in applications to diverse kinds of large-scale systems optimization problems,
III. embodies the most powerful techniques of mathematical programming and numerical analysis, and
IV. has been thoroughly tested for efficiency and effectiveness on representative systems optimization problems arising in practice.

The development effort of such a laboratory in its initial stages would consist of three basic activities: [1] research in mathematical programming, including particularly the analysis, development, and testing of special computational methods for certain specially structured optimization problems that occur frequently in systems optimization, or as subproblems of larger systems, [2] collection of representative systems optimization problems arising in practice in government, science, and industry, in order both to study their mathematical structure and to use them as test problems for studies of efficiency, and [3] development of an integrated set of computer routines, and an associated macro-language to enable its flexible use, which implements the most powerful of existing methods for systems optimization.

The creation of such a laboratory would be a concerted effort to break a bottleneck which is currently constricting the applications of mathematical programming to many of the most important systems optimization problems. This bottleneck is the lack of an integrated collection of compatible computer routines, preferably organized and callable via a macro-language, which can be employed efficiently and flexibly in a wide diversity of practical applications.

The origins and nature of the bottleneck can be described as follows. The existing methods of mathematical programming exploit either general structure or special structure. Those that exploit general structure take advantage of the fact that in a particular problem, the functional forms involved are linear, or quadratic, or convex, separable, etc.





Methods of this kind ordinarily are limited in their applications by the size and speed of the computing equipment available according to some power (often the third or fourth) of the number of variables and/or constraints. Those that exploit special structure take advantage of further particular features of a problem. For example, in the case of linear programming, which is the most highly developed in this respect, there are methods which exploit the special structures of (1) network problems arising in transport planning, (2) "block-diagonal" problems arising in decentralized allocation problems, (3) "staircase" problems arising in dynamic investment planning, economic growth models, and optimal control, (4) problems amenable to "column generation" arising in production scheduling and elsewhere, (5) general problems with "sparse" matrices etc. Moreover, there is a substantial and powerful theory of how to <u>decompose</u> large and complicated systems into their component sybsystems and from analyses of these components to derive solutions to the original system. Methods that exploit special structure are not limited in the range of their applicability in the way that ordinary general-structure methods are; indeed, with present methods and computing equipment it is practical in certain cases to solve systems with close to a million of variables and constraints. (For example the National Biscuit Company problem solved by Mathematica.)

It is the nature of human activity, and in large part of the physical world as well, that large and complicated endeavors are organized as systems of interrelated parts, and indeed, as systematic hierarchies of interrelated subsystems. Such systems typically exhibit special mathematical structures. These special structures permit numerical analysis and optimization via methods that exploit the special structure, whereas general-structure methods would be infeasible if the problem is of the size normally encountered in practice. The extension of the range of applications of mathematical programming is, therefore, most promising for pressing world problems involving total system optimization discussed earlier since they exhibit special structures.

## Nature of the Bottleneck.

The bottleneck, however, is that presently there is not available any collection of decomposition methods and special-structure methods implemented in freely available, efficient, tested, flexible computer routines which can be applied easily, cheaply, and with confidence to practical problems as they arise. The result has been, and will continue (if development work does not proceed), that in each potential application it would be necessary to develop computer routines especially for the project. Because this is so costly in expense or time, it is generally not done and the valuable potential application to the system optimization is foregone.

There are three reasons for this unfortunate state of affairs. One is that in the past researchers on decomposition methods and special-structure methods have not had a viable way of enabling their work to contribute directly to the construction of such a collection of computer routines. Either there was no incentive to complement their research results with practically useful computer routines; or, if they did do it, there was no way that the routines could be documented, tested, and ultimately incorporated into a larger collection of established routines; indeed, there has been almost a complete absence of standard documentation procedures, standard test problems, and standard compatibility requirements for callability, data input, and output. The consequence has been that research, implementation, and applications of systems optimization have been uncoordinated and disconnected, to everyone's detriment.

The second reason is that the incentives to development work have operated at cross purposes with the ultimate goals. As mentioned, in a particular application it is usually too costly or time-consuming to undertake the development of the needed computer routines, or just as likely, the organization faced with the tasks lacks the expert competence among its staff to complete the job successfully. On the other hand, occasional development work has been undertaken by private software firms. Indeed, five or ten years ago one would have





had great hopes that this approach would succeed. In fact, however, the incentive to private firms has in nearly every instance been to keep their routines proprietary, expensive to use, and noncallable. For the most part, private firms have responded to the natural incentive to appropriate the public know-how into a privately saleable commodity.

The third reason is that there has not been support for a coordinated development effort, one that assembles expert competence in theory, numerical methods, and computer science, and that ensures the permanence of its work through a thorough program of experimentation, testing, documentation, and enforced compatibility requirements.

A Systems Optimization Laboratory could be carefully designed to overcome these impediments to progress in the field. It could bring together the various kinds of expert competence that are needed, and it could implement the development effort in a coordinated program of research, programming, experimentation, testing, and documentation, with the results to be made freely and widely available for diverse applications in a flexible and easily used form.

The major research activities of a System Optimization Laboratory can be classified broadly as follows: (1) basic research related to optimization theory, (2) development of computational methodology for mathematical programming, including general-structure methods, decomposition methods, and special-structure methods, and (3) construction and evaluation of algorithms.

## Software Development.

A major activity of System Optimization Laboratory would be the development of software packages for systems optimization. This development effort could proceed on two different levels. First, a major activity would be the completion of a macro-language for organizing and calling routines in the software package. Mainly this could be an extension of the macro-language Mathematical Programming Language [MPL] under development by the author. The second major activity could be the programming, testing, and documentation of algorithms for decomposition and special structures, including experimentation with alternative algorithms, and testing of algorithms on practical problems. Computer routines would be thoroughly documented, tested on standard problems, and written in a format compatible with and callable by the macro-language.

External Affairs.

Three important activities of the Laboratory fall under this heading. First, members of the technical staff could undertake the collection and study of examples of systems optimization problems arising in government, science, and industry, for use both as test problems and as indicators of the types of systems and specially-structured problems of major importance in practice. Many examples are already known, but further empirical data is considered desirable to ensure the ultimate usefulness of the Laboratory's work. Second, other researchers in the field could be solicited to obtain algorithms, computer routines, and test problems for inclusion in the Laboratory's studies. Also, the Laboratory could disseminate information to potential contributions on the requirements for computer routines to be compatible with the Laboratory's software package. Third, when the Laboratory's software package is reasonably complete, it could undertake to make it available to users--this being, of course, the ultimate purpose of the Laboratory.

Research Projects of a System Optimization Laboratory.

A major goal of the Systems Optimization Laboratory would be to provide standardized computer routines for systems optimization. The types of research activities that would be needed to support this effort are outlined below. Particular areas of research that might be planned for the initial project period will be described first:

A. Decomposition Methods: The chief requirement in the construction of numerical methods for optimizing large





systems is that the algorithm exploit the special structure of the system. The body of theory and techniques which addresses this requirement are generally called decomposition methods. The range of decomposition methods is quite diverse, however, since of necessity a particular algorithm must reflect the special structure of the class of problems to which it is applicable.

One preliminary task in the development of decomposition methods would be the construction of an efficient taxonomy for system structures. This task is only partially complete. The major taxonomic features that are well understood can be described briefly as follows. First, there is a large and important class of problems whose special structure permits the design of an efficient algorithm based directly on this structure. Usually, duality and compact representation schemes play a key role in the design of the network problems, problems with upper and lower bound constraints, and a number of nonlinear problems (geometric programming, fractional programming, variable-factor programming, etc.). Often problems with these special structures occur as subproblems in larger systems and it is therefore important to have available efficient, tested, and documented routines for these problems which are easily callable.

A second major class of problems are those which, in the linear case, are characterized by sparce matrices. (Hence the numerical structure is quite general except for the known presence of many zeros.) Compact representation schemes for sparse matrices play the major role in the development of algorithms for these problems [Dantzig (1963)].

A third class of problems are those which are amenable to generation techniques. The major examples from this class are the column generating techniques of Gilmore and Gomory (1961, 1965) for "cutting-stock" and related problems, and the row and column generating techniques of Wilson (1972) for 2-person games in extensive forms, both of which use dynamic programming as the means of generating data explicitly that is otherwise embodied implicitly in the problem formulation. A generating technique of much greater generality is the method of generalized programming in which it is required only that the data be generated from a convex set using duality information from a master coordinating problem.

The generalized programming method of Wolfe (see Chapter 22 in Dantzig (1963)) is actually a generalization of the decomposition method for linear programs with block-angular structures [Dantzig and Wolfe (1962)], which represent a fourth major class of system structures--those which (in either primal or dual form, including multi-stage programming under uncertainty [cf. Dantzig and Madansky (1961) and a variety of other dynamic programming problems] represent a problem of allocating scarce resources to otherwise independent subproblems. Zschau's (1967) primal decomposition algorithm also applies to this class of problems, which are of prime importance in applications.

Wolfe's generalized programming approach is also applicable to a fifth major class of problems which is closely related to the previously mentioned class, namely the class of multi-stage allocation problems represented by dynamic investment problems and optimal control problems. Another example is the linear control problem which can be solved using generalized programming [see Dantzig (1966)].

Both of these last two classes are instance of a general class, which can be called nearly decomposable problems. In this general class one finds a macro-structure which would be perfectly decomposable into independent subproblems except for the presence of a relatively few connections (and therefore interdependencies) among the subproblems. The development of efficient algorithms for nearly decomposable problems is a major area for research and one for which the range of applications is enormous. Its successful conclusion may require the development of general methods for highly connected systems, such as have been recently proposed by Douglass Wilde (unpublished). One form of such a method is presently available in Benders' decomposition method (1962).

Surveys of the major decomposition methods are given by Geoffrion (1970) and Lasdon (1971).

In the area of decomposition methods, the Systems Optimization Laboratory would pursue essentially three





research and development activities. First, a major effort could be to program, test and document existing decomposition methods as part of the development of the macro-language MPL [Dantzig et al. (1970)]. This development effort is aimed at creating a useful software package for many of the most important systems optimization problems which arise in practice. Second, a part of the research effort could be devoted to the construction of new algorithms for general nearly-decomposable problems and for tightly connected systems.

The third part of the research program would reflect the important role of structural taxonomy in the development of decomposition methods. In connection with the Laboratory's empirical studies of some of the major systems optimization problems encountered in practice, a structural taxonomy could be developed and comparative studies made of the relative efficiencies of alternative methods of optimizing systems of similar structures. There are, moreover, a number of systems optimization problems of known structure, and of great practical importance, for which an intensive development effort could be devoted to the construction of efficient algorithms. First on this list is the class of "staircase" problems represented by dynamic investment models in economics and business and optimal control problems arising in (among other contexts) ecological models.

In general, the Laboratory's work on decomposition methods would provide a synthesizing focus for its entire spectrum of studies on systems optimization. The primary objective would be to provide an unified body of theory, methods, and computer routines for the efficient and practical numerical analysis of large systems.

B. Mathematical Programming, Matrix Decomposition and Sparse Matrix Techniques: (The comments of this section are due to Gene Golub.) For many algorithms in mathematical programming it is necessary to compute a sequence of matrix decompositions. For example, in the classical simplex algorithms for solving linear programming problems it is necessary to solve two or three systems of linear algebraic equations at each iteration. There are many ways of solving these systems, but a particularly effective numerical algorithm is to use some form of the LU decomposition of a matrix. At each stage of the simplex algorithm the coefficient matrix is changed by one column so that one is concerned with techniques of updating the matrix decomposition in an efficient and stable manner, especially when the data matrix is very sparse.

In general, suppose that a matrix A and some factorization of A are given, e.g., $A = PTQ^T$, where P and Q are orthogonal matrices and T is a triangular matrix. The problem then is to compute the factorization of $A + \sigma \underset{\sim}{u} \underset{\sim}{v}^T$ where $\underset{\sim}{u}$ and $\underset{\sim}{v}$ are given vectors and $\sigma$ is scalar quantity, or a factorization of A when A is changed by one column.

The three basic considerations in computing the new factorization are the following: (1) The updating should be performed in as few operations as possible. This is especially true when handling large masses of data where continual updating is needed. (2) The numerical procedure should be stable. Some procedures which have been recommended in the literature can easily become numerically unstable. This is especially true for the Cholesky factorization of a matrix when $\sigma = -1$. (3) The updating procedure should preserve sparsity. Quite often the original matrix factorization will be sparse, and it is desirable to preserve the sparsity by possibly rearranging the rows and columns of the original data matrix.

The problem of updating occurs in many other contexts, e.g., statistics and control theory. For this reason, it is especially important to have methods which yield fast, accurate, and sparse factorizations.

Therefore, a study would be made of various factorizations and how they may be used in large scale programming problems, especially when the data matrix is structured. The sparse-matrix techniques are especially useful as an alternative when the decomposition principle is applicable. Furthermore, the matrix-decomposition methods would be most useful when the complementarity methods for solving mathematical programming problems are applicable. Some study has already been made in this direction [Tomlin (1971)].





C. Complementarity Methods: The development of complementarity methods is the major advance in the theory and technique of mathematical programming in recent years. The application of this approach to decomposition and special-structure methods remains largely undeveloped, however. There is a prospect, moreover, that further development of the present general-structure complementarity methods will lead to substantial improvements in their efficiency and range of applications. Due to the probable importance of complementarity methods in the development of new algorithms, the research program could pursue several major topics in this area.

[1] Linear Complementarity. Linear complementarity problems arise in linear and quadratic programming and in 2-person games, and they are a basic component of nonlinear programs and n-person games (see Cottle (1964), (1967), (1968a, b, c), (1970), (1971a, b, c), Eaves (1971a, b), and Lemke (1964), (1965)). In this area the research program could concentrate on the development and testing of methods which exploit the special structure of quadratic programs, especially ones of the large size and structure arising in major-system optimization problems [Beale (1967)].

[2] Nonlinear Complementarity. Nonlinear complementarity problems (see Cottle (1966), Eaves (1971d), Karamardian (1966), (1971) and Lemke (1970)) arise in general nonlinear programs and n-person games ($n \geq 3$). Normally such problems are most efficiently handled via linear or quadratic approximations. However, there is a variety of important nonlinear problems arising in practice whose special structure can be exploited to obtain more efficient procedures. The principal devices here are (a) the use of duality theory to obtain simpler dual problems or to pre-optimize subproblems of a larger system, and (b) the design of special complementarity algorithms which take advantage of the special structure. Both of these approaches could be pursued in the research program. A major class of practical problems which would be investigated are the pooling or the pre-processor problems. A major goal would be to convert systems of allocation and pooling problems into equivalent systems all of

one type or the other.

[3] Computation of Equilibria and Fixed-Points. One of the major outgrowths of complementarity methods has been the development for the first time of practical numerical methods for the computation of systems equilibria and fixed-points of mappings. (See Eaves (1970), (1971c,e,f), (1972), Freidenfelds (1971), Kuhn (1968), Scarf (1967a,b ), (1969), (1972), Rosenmuller (1971), and Wilson (1971), (1972).) The advent of these methods opens the possibility of computing directly the equilibria of chemical, biological, and physical systems, and equilibria n-person games, rather than via the awkward approximation methods normally used. Moreover, it raises the possibility of a unified body of theory and computational methods (since, for example, convex programming problems can be shown to be equivalent to finding the fixed points of certain related mappings, and system equilibria are normally characterized either via the fixed-points of the equations of disequilibria or in terms of minimizing a measure of the loss from disequilibrium). The research program could pursue the further development of complementarity methods (including methods based on primitive sets and simplical subdivisions) for such problems with particular emphasis on the development of practical methods for computing the equilibria of large systems.

D. Combinatorial Problems and Integer Programming with Special Structure. The fundamental feature of many systems optimization problems is their combinatorial character. This may occur either because the problem has a special network structure or because it has discrete decision variables, so that a huge number of combinations must be considered. As Fulkerson (1966) discusses, such combinatorial problems arise in a wide variety of contexts. These problems sometimes can be solved, of course, but only by developing clever algorithms which exploit their special structure. Therefore, algorithmic development in this area will be one of the major research activities of a System Optimization Laboratory.





Probably the most important combinatorial problem for systems optimization is the integer programming problem [Gomory (1963)]. One reason is that so many optimization problems would be linear programs except that the decision variables make sense only with integer values (e.g., see Cushing (1969)), and so they become integer linear programs instead. In addition, it is possible to reformulate a number of important but difficult (indeed seemingly impossible) problems of a nonlinear, nonconvex and combinatorial character as mixed integer linear programming problems (see Dantzig (1963)). Another important reason is that many large-scale mathematical programming systems include subproblems which are integer programs, so that decomposition methods for such systems (e.g., see Benders (1962)) would need an integer programming algorithm as a subroutine.

Because of these considerations particular emphasis will be placed on algorithmic development for integer programming. This has been an area of substantial research for over a decade, and significant progress is being made (e.g., see Hillier (1969a), Balinski and Spielberg (1969), and Geoffrion and Marsten (1972)). Unfortunately, the problem is very difficult, and the efficiency of the available algorithms does not remotely approach that of the simplex method for the linear programming problem. Therefore, the main thrust of this research could be the development of special-purpose algorithms for important classes of integer programming problems in order to exploit special structure. Thorough testing and evaluation could be conducted, which would necessitate a major programming effort, so the resources and long-range continuity of the Systems Optimization Laboratory would play a vital role in carrying out this development beyond an initial stage. Decomposition methods for mixed integer programming systems could also be investigated. Another part of this research program would involve developing special-purpose heuristic procedures (see Hillier (1969b)) for obtaining good approximate solutions for large-scale integer programming systems having various common kinds of special structure.

E. Further Development of a Macro-Language for Mathematical Programming. Commerical codes for solving mathematical programming problems typically involve about 200,000 assembly-language instructions. One can anticipate that efficient commerical programs for solving structured systems optimization problems will be an order of magnitude more complex. In order for such programs to be developed and maintained, the language in which they are written must be highly readable and easy to modify. This is the purpose of the new MPL [Mathematical Programming Language] now under development. The continuation of this work could be one of the major projects of a Laboratory.

MPL is a high-level user-oriented programming language intended particularly for developing, testing, and communicating mathematical algorithms (see Dantzig, et al. (1970)). It is being developed to provide a language for mathematical algorithms that will be easier to write, to read, and to modify than currently available computer languages such as FORTRAN, ALGOL, PL/1, APL .

The need for a highly readable mathematically-based computer language has been apparent for some time. Generally speaking, standard mathematical notation in a suitably algorithm-like structure appears best for this purpose, since most researchers are familiar with the language of mathematics. Therefore, MPL closely parallels the vernacular of applied mathematics. An important area of application of MPL is for the development and testing of algorithms for systems optimization problems. To date, many methods have been proposed for solving such problems, but few have been experimentally tested and compared because of the high cost and the long time it takes to program them, and because it is difficult to debug and to modify them quickly after they are written. It is believed that highly readable programs would greatly facilitate experimentation with these proposed methods and would shorten the time until they can be used in practice. Thus, the development of a sophisticated version of MPL will provide a vital tool for the Systems Optimization Laboratory, as well as for other researchers.

As pointed out by William Orchard-Hays many other





special purpose languages beside MPL would be required as basic research tools. There is a need to have special language for Job Control, Computer Control, Matrix Generation, Procedure Programming (e.g., MPL or APL); languages for File Mechanisms; languages for organizing the entire system of computation.

. . . . . . . . . . .

To summarize: Large-Scale Optimization requires laboratories where a large number of test models, computer programs, and special "tools" to aid in developing variants of existing techniques, are assembled in a systematic way. Only this way can one hope to model and solve the host of pressing total system problems that the world faces today.

## REFERENCES

1. Abadie, J. M. and A. C. Williams (1963), "Dual and Parametric Methods in Decomposition", in Recent Advances in Math. Prog., R. Graves and P. Wolfe (eds.), McGraw-Hill, New York, pp. 149-158.

2. Aronofsky, Julius S. (ed.) (1969), Progress in Operations Research (Relationship Between Operations Research and the Computer), Vol. III, Wiley, New York.

3. Bartels, R. H., (1971), "A Stabilization of the Simplex Method", Numerische Mathematik, Vol. 16, pp. 414-434.

4. Bartels, R. H. and G. H. Golub (1969), "The simplex method of linear programming using LU decomposition", Comm. ACM, Vol. 12, pp. 266-268.

5. Balas, Egon, (1966), "An Infeasibility - Pricing Decomposition Method for Linear Programs", Operations Research, Vol. 14, pp. 843-873.

6. Balinski, M. L. and K. Spielberg, (1969), "Methods for Integer Programming: Algebraic, Combinatorial, and Enumerative", Ch. 7 , pp. 195-292, in Progress in Operations Research (Relationship Between Operations Research and the Computer), J. S. Aronofsky (ed.), Wiley, New York.

7. Balinski, M. L. (1966), "On Some Decomposition Approaches in Linear Programming", and "Integer Programming", The University of Michigan Engineering Summer Conferences.

8. Baumol, W. J. and T. Fabian, (1964), "Decomposition, Pricing for Decentralization and External Economies", Management Science, Vol. 11, No. 1 , pp. 1-32.

9. Beale, E. M. L., (ed.)(1970), Applications of Mathematical Programming Techniques, American Elsevier, New York.

10. Beale, E. M. L. (1967), "Decomposition and Partitioning Methods for Nonlinear Programming", in Non-Linear Programming, J. Abadie (ed.), North-Holland, Amsterdam.

11. Beale, E. M. L., (1963), "The Simplex Method Using Pseudo-Basic Variables for Structured Linear Programming Problems", from Recent Advances in Math. Prog., R. Graves and P. Wolfe (eds.) McGraw-Hill, New York, pp. 133-148.

12. Benders, J. F., (1962), Partitioning Procedures for Solving Mixed-Variables Programming Problems", Numerische Mathematik, Vol. 4, pp. 238-252.

13. Bennett, J. M. (1966), "An Approach to Some Structured Linear Programming Problems", Operations Research, 14, 4 (July-August), pp. 636-645.





14. Broise, P., P. Huard, and J. Sentenac, (1968), Décomposition des Programmes Mathématiques, Monographies de Recherche Opérationnelle, Dunod, Paris.

15. Cottle, R. W. and J. Krarup (eds.), (1971), Optimization Methods for Resource-Allocation, Proceedings of the NATO Conference, Elsinore, Denmark. English University Press, London, 1973.

16. Cottle, R. W. (1964), "Note on a Fundamental Theorem in Quadratic Programming", SIAM J. Appl. Math., 12, pp. 663-665.

17. Cottle, R. W. (1966), "Nonlinear Programs with Positively Bounded Jacobians", SIAM J. Appl. Math., 14, 1, pp. 147-158.

18. Cottle, R. W. and G. B. Dantzig (1967), "Positive (Semi-) Definite Programming", in Nonlinear Programming, North-Holland, J. Abadie (ed.), pp. 55-73, Amsterdam.

19. Cottle, R. W. and G. B. Dantzig (1968a), "Complementary Pivot Theory of Mathematical Programming", Lin. Alg. and Its Applns., 1, pp. 103-125.

20. Cottle, R. W. (1968b), "The Principal Pivoting Method of Quadratic Programming", in Mathematics of the Decision Sciences, Part I, American Mathematical Society, G. B. Dantzig and A. F. Veinott, Jr. (eds.), Providence, R. I., pp. 144-162.

21. Cottle, R. W. (1968c), "On a Problem in Linear Inequalities", J. London Math. Soc., 42, pp. 378-384.

22. Cottle, R. W. and W. C. Mylander (1970), "Ritter's Cutting Plane Method for Non-convex Quadratic Programming", in Integer and Nonlinear Programming, North-Holland, J. Abadie (ed.), Amsterdam, pp. 257-283.

23. Cottle, R. W. and A. F. Veinott, Jr. (1971a), "Polyhedral Sets Having a Least Element", Math Prog. Vol. 3, pp. 238-249.

24. Cottle, R. W. (1971b), "Solution to G. Maier's Problem on Parametric Linear Complementarity Problems", submitted to SIAM Review.

25. Cottle, R. W. (1971c), "Monotone Solutions of the Parametric Linear Complementarity Problem", Technical Report No. 71-19, Department of Operations Research, Stanford University, submitted to Math. Prog.

26. Cushing, B. E., (1969), "The Application Potential of Integer Programming", The Journal of Business, Vol. 42, pp. 457-467.

27. Dantzig, G. B., (1963), Linear Programming and Extensions, Princeton University Press, Princeton, N. J.

28. Dantzig, G. B., (1960), "On the Significance of Solving Linear Programming Problems with Some Integer Variables", Econometrica, Vol. 28, pp. 30-44.

29. Dantzig, G. B. and P. Wolfe, (1960), "Decomposition Principle for Linear Programming Problems", Operations Research, Vol. 8, pp. 101-110.





30. Dantzig, G. B. and R. M. Van Slyke, (1964), "Generalized Upper Bounded Techniques for Linear Programming - I", Proceedings IBM Scientific Computing Symposium, Combinatorial Problems, March 16-18, pp. 249-261.

31. Dantzig, G. B. and R. M. Van Slyke, (1967), "Generalized Upper Bounded Techniques for Linear Programming - II", J. Computer and System Sciences, Vol. 1, pp. 213-226.

32. Dantzig, G. B., S. Eisenstat, T. Magnanti, S. Maier, M. McGrath, V. Nicholson, C. Riedl, (1970), "MPL-Mathematical Programming Language-Specification Manual", Stanford University Computer Science Department Technical Report, STAN-DS-70-187.

33. Dantzig, G. B., (1963), "Compact basis triangularization for the simplex method" in R. Graves and P. Wolfe (eds.), Recent Advances in Math. Prog., McGraw-Hill, New York, pp. 125-132.

34. Dantzig, G. B., (1955), "Upper Bounds, Secondary Constraints and Block Triangularity in Linear Programming", Econometrica, Vol. 23, No. 2, pp. 174-183.

35. Dantzig, G. B., (1955), "Optimal Solution of a Dynamic Leontief Model with Substitution", Econometrica, Vol. 23, No. 3, pp. 295-307.

36. Dantzig, G. B., (1955), "Linear Programming Under Uncertainty", Management Science, Vol. 1, pp. 197-206.

37. Dantzig, G. B., (1966), "Linear Control Processes and Mathematical Programming", SIAM J. Control, Vol. 4, No. 1, pp. 56-60.

38. Dantzig, G. B., D. R. Fulkerson, and S. Johnson, (1954), "Solution of a Large Scale Traveling Salesman Problem", Operations Research, Vol. 2, No. 4, pp. 393-410.

39. Dantzig, G. B., R. Harvey, R. McKnight, (1964), "Updating the Product Form of the Inverse for the Revised Simplex Method", Operations Research Center, University of California, Berkeley.

40. Dantzig, G. B. and R. M. Van Slyke (1967), "Generalized Upper Bounding Techniques", J. Computer System Science, Vol. 1, pp. 213-226.

41. Dzielinski, P. and R. E. Gomory (1963) (abstract), "Lot size Programming and the Decomposition Principle", Econometrica, Vol. 31, p. 595.

42. Eaves, B. C. (1970), "An Odd Theorem", Proc. of Amer. Math. Soc., 26, 3, pp. 509-513.

43. Eaves, B. C. (1971a), "The Linear Complementarity Problem", Management Science, 17, 9, pp. 612-634.

44. Eaves, B. C., (1971b), "On Quadratic Programming", Management Science, 17, 11, pp. 698-711.

45. Eaves, B. C., (1971c), "Computing Kakutani Fixed Points", SIAM J. Appl. Math., 21, 2, pp. 236-244.

46. Eaves, B. C., (1971d), "On the Basic Theorem of Complementarity", Math. Prog., 1, 1, pp. 68-75.

47. Eaves, B. C., (1971e), "Homotopies for Computation of Fixed Points", Math. Prog. Vol. 3, pp. 1-22.





48. Eaves, B. C., and R. Saigal (1971f), "Homotopies for Computation of Fixed Points on Unbounded Regions", Math. Prog. Vol. 3, pp. 225-237.

49. Eaves, B. C., (1972), "Polymatrix Games with Joint Constraints", Department of Operations Research, Stanford University, submitted to SIAM J. Appl. Math.

50. Electricité de France, Direction des études et recherches, "Programmation Linéaire, Méthode de Dantzig et Wolfe, Programme Expérimental", Paris.

51. Elmaghraby, S. E., (1970), The Theory of Networks and Management Science, Parts I, II," Management Science, Vol. 17, pp. 1-34, B-54-71.

52. Ford, L., Jr., and D. R. Fulkerson, (1958), "Suggested Computation for Maximal Multi-Commodity Network Flows", Management Science, Vol. 5, No. 1, pp. 97-101.

53. Forrest, J. J. H., and J. A. Tomlin, "Updating triangular factors of the basis to maintain sparsity in the product form simplex method", in Cottle and Krarup (forthcoming).

54. Freidenfelds, J., (1971), "A Fixed-Point Algorithm and Almost-Complementary Sets", Technical Report No. 71-3, Department of Operations Research, Stanford University. In addition, "Fixed-Point Algorithms and Almost-Complementary Sets", Ph. D. Dissertation, Department of Operations Research, Stanford University.

55. Frisch, R., (1962), "Tentative Formulation of the Multiplex Method for the Case of a Large Number of Basic Variables", Institute of Economics, University of Oslo.

56. Fulkerson, D. R., (1966), "Flow Networks and Combinatorial Operations Research", American Mathematical Monthly, Vol. 73, pp. 115-138.

57. Gale, D., (1966), "On Optimal Development in a Multi-sector Economy", Operations Research Center, 1966-11, University of California, Berkeley.

58. Gass, S. I., (1966), "The Dualplex Method for Large-Scale Linear Programs", Operations Research Center, 1966-15, University of California, Berkeley, Ph. D. Thesis.

59. Geoffrion, A. M., (1970), "Elements of Large-Scale Mathematical Programming", Management Science, Vol. 16, pp. 652-691.

60. Geoffrion, A. M., (1971), "Optimal Distribution System Design", Proceedings of the NATO Conference, Optimization Methods for Resource Allocation, R. W. Cottle and J. Krarup, (eds), Elsinore, Denmark, English University Press, London, 1973.

61. Geoffrion, A. M. and R. E. Marsten, (1972), "Integer Programming Algorithms: A Framework and State-of-the-Art Survey", Management Science, Vol. 18, (forthcoming).

62. Gill, P. E. and W. Murray, (1971), "A Numerically Stable Form of the Simplex Algorithm", Technical report No. Maths 87, National Physical Laboratory, Teddington.

63. Gill, P. E., G. H. Golub, W. Murray and M. A. Saunders, "Methods for Modifying Matrix Factorizations", to be published.

64. Gilmore, P. C., and R. E. Gomory, (1965), "Multistage Cutting Stock Problems of Two and More Dimensions", Operations Research, Vol. 13, pp. 94-120.





65. Gilmore, P. C., and R. E. Gomory, (1961), "A Linear Programming Approach to the Cutting Stock Problem", Operations Research, Vol. 9, pp. 849-859.

66. Glassey, C. R., (1971), "Dynamic Linear Programs for Production Scheduling", Operations Research, Vol. 19, pp. 45-56.

67. Gomory, R. E., (1963), "Large and Non-Convex Problems in L. P.", Proc. Sympos. Appl. Math. 15, pp. 125-139.

68. Gomory, R. E., and T. C. Hu, (1962), "An Application of Generalized Linear Programming to Network Flows", SIAM Journal, Vol. 10, No. 2, pp. 260-283.

69. Graves, R. L., P. Wolfe, (eds.) (1963), Recent Advances in Mathematical Programming, McGraw-Hill, New York

70. Haley, R. B., (1966), "A General Method of Solution for Special Structure Linear Programmes", Operational Research Quarterly, Vol. 17, No. 1, pp. 83-90.

71. Heesterman, A. R. G., and J. Sandea, (1965), "Special Simplex Algorithm for Linked Problems", Management Science, 11, 3, pp. 420-428.

72. Hellerman, E., (1965), "The Dantzig-Wolfe Decomposition Algorithm as Implemented on a Large-Scale (Systems Engineering) Computer", presented at Modern Techniques in the Analysis of Large-Scale Engineering Systems.

73. Hu, T. C., (1963), "Multi-Commodity Network Flows", Operations Research, Vol. 11, pp. 344-360.

74. Hillier, F. S., (1969a), "A Bound-and-Scan Algorithm for Pure Integer Linear Programming with General Variables", Operations Research, Vol. 17, pp. 638-679.

75. Hillier, F. S., (1969b), "Efficient Heuristic Procedure for Integer Linear Programming with an Interior", Operations Research, Vol. 17, pp. 600-637.

76. Kutcher, G. P., "On Decomposing Price-Endogeneous Models", in L. M. Gorens and A. S Manne (eds.), Multi-level Planning: Case Studies in Mexico, North Holland, Amsterdam, forthcoming, Ch. V. 2.

77. Karamardian, S., (1966), "Duality in Mathematical Programming", Operations Research Center, University of California, Berkeley, or Parts 1 and 2 of "The Nonlinear Complementarity Problem with Applications", JOTA, 4, 2, and 3 (1969), pp. 87-98, pp. 167-181.

78. Karamardian, S., (1971), "The Complementarity Problem", Graduate School of Administration and Department of Mathematics, University of California, Irvine, to appear in Math. Prog.

79. Kuhn, H., (1968), "Simplicial Approximation of Fixed Points", Proc. Nat. Acad. Sci. U.S.A., 61, pp. 1238-1242.

80. Kron, G., (1963), Diakoptics, The Piecewise Solution of Large-Scale Systems, Macdonald and Co., London.

81. Lasdon, L. S., (1970), "Optimization Theory for Large Systems", The Macmillan Company, New York.





82. Lasdon, L. S., (1971, "Uses of Generalized Upper Bounding Methods in Production Scheduling", in R. W. Cottle and J. Krarup (eds), Optimization Methods for Resource-Allocation, Proceedings of the NATO Conference, Elsinore, Denmark, English University Press, London, 1973.

83. Lemke, C. E., (1965), "Bimatrix Equilibrium Points and Mathematical Programming", Management Science 11, 7, pp. 681-689.

84. Lemke, C. E., (1970), "Recent Results in Complementarity Problems", in Nonlinear Programming, J. B. Rosen, O. Mangasarian and K. Ritter, (eds.), Academic Press, New York, pp. 349-384.

85. Lemke, C. E. and J. T. Howson, Jr., (1964), "Equilibrium Points of Bimatrix Games", SIAM J. Appl. Math., 12, pp. 413-423.

86. Reid, J. K., (ed.) (1971), "Large Sparse Sets of Linear Equations", Academic Press, New York.

87. Ritter, K., (1967), "A Decomposition Method for Linear Programming Problems with Coupling Constraints and Variables", Math. Research Center, The University of Wisconsin #739. Published with other works as: M. D. Grigoriadis and K. Ritter, "A decomposition method for structured linear and non-linear programs", J. Comput. Systems Sci., 3 (1969), pp. 335-360.

88. Rosen, J. B., (1964), "Primal Partition Programming for Block Diagonal Matrices", Numerische Math., 6, 3, pp. 250-264.

89. Rosenmüller, J., (1971), "On a Generalization of the Lemke-Howson Algorithm to Noncooperative N-Person Games", SIAM J. Appl. Math., 21, 1, pp. 73-79.

90. Saunders, M. A., (1972), "Large-scale linear programming using the Cholesky factorization", Technical Report No. CS-252, Computer Science Department, Stanford University.

91. Scarf, H., (1967a), "The Core of an N-Person Game", Econometrica, 35, pp. 50-69.

92. Scarf, H., (1967b), "The approximation of Fixed Points of a Continuous Mapping", SIAM J. Appl. Math., 15, pp. 1328-1343.

93. Scarf, H., and T. Hansen, (1969), "On the Applications of a Recent Combinatorial Algorithm", Discussion Paper No. 272, Cowles Foundation, Yale University.

94. Scarf, H., in collaboration with T. Hansen (1972), The Computation of Economic Equilibria, Cowles Foundation, Yale University, to be published.

95. Tomlin, J. A., (1971), "Modifying triangular factors of the basis in the simplex method", presented at the Symposium on Sparse Matrices and their Applications, IBM Thomas J. Watson Research Center, Yorktown Heights, New York.

96. Van Slyke, R. M. and R. Wets, (1969), "L-Shaped Linear Programs with Applications to Optimal Control and Stochastic Programming", SIAM J. Appl. Math., 17, pp. 638-663.

97. Van Slyke, R. and R. Wets, (1966), "Programming Under Uncertainty and Stochastic Optimal Control", SIAM J. Control, Vol. 14, No. 1, pp. 179-193.

98. Varaiya, P., (1966), "Decomposition of Large-Scale Systems", SIAM J. Control, Vol. 4, No. 1, pp. 173-179.





99. Wagner, H. M., (1957), "A Linear Programming Solution to Dynamic Leontief Type Models", Management Science, Vol. 3, No. 3, pp. 234-254.

100. Wilde, D. J., (1963), "Production Planning of Large Systems", Chemical Engineering Progress, Vol. 59, pp. 46-51.

101. Williams, J. D., (1960), "The Method of Continuous Coefficients, Parts I and II", Report No. ECC 60.3, Socony.

102. Williams, A. C., (1962), "A Treatment of Transportation Problems by Decomposition", SIAM Journal, Vol. 10, No. 1, pp. 35-48.

103. Willoughby, R. A., (ed.) (1969), Proceedings of the Symposium on Sparse Matrices and Their Applications, IBM Thomas J. Watson Research Center, Yorktown Heights, New York.

104. Wilson, R. B., (1971), "Computing Equilibria of N-Person Games", SIAM J. Appl. Math., 21, 1, pp. 80-87.

105. Wilson, R. B., (1972), "Computation of Equilibria of Two-Person Games in Extensive Form", Graduate School of Business, Stanford University, to appear in Management Science.

106. Wismer, D. A., (ed.) (1971) "Optimization Methods for Large-Scale Systems... With Applications", McGraw-Hill, New York.

107. Wolfe, P., (1961) "Accelerating the Cutting Plane Method for Non-linear Programming", SIAM Journal, Vol. 9, No. 3, pp. 481-488.

108. Wolfe, P., and G. B. Dantzig, (1962), "Linear Programming in a Markov Chain", Operations Research, 10, pp. 702-710.

109. Wood, M. K., and G. B. Dantzig (1947), "The Programming of Interdependent Activities", in T. C. Koopmans, Activity Analysis of Production and Allocation, Wiley, New York, 1951; Also Econometrica Vol. 17, Nos. 3 and 4, July-October, 1949, pp. 193-199.

110. Zschau, E. V. W., (1967), "A Primal Decomposition Algorithm for Linear Programming", Graduate School of Business, Stanford University.

## FOOTNOTES

1. Parts of the material in this paper were drawn from a draft of a proposal to establish such a laboratory at Stanford University, prepared by R. Cottle, B. C. Eaves, G. H. Golub, F. S. Hillier, A. S. Manne, D. J. Wilde, R. B. Wilson and the author.

2. This paper will also appear in Optimization Methods for Resource Allocation, English University Press, London, 1973.

Department of Operations Research
Stanford University
Stanford, California. 94305