METHODOLOGIES FOR WATER RESOURCES PLANNING:

DDDP AND MLOM(TLOM)

Ven Te Chow
Project Director

FINAL REPORT
Project No. B-030-ILL

The work upon which this publication is based was supported by funds provided by the U.S. Department of the Interior as authorized under the Water Resources Research Act of 1964, P.L. 88-379
Agreement No. 14-01-0001-1899

UNIVERSITY OF ILLINOIS
WATER RESOURCES CENTER
2535 Civil Engineering Building
Urbana, Illinois 61801

October, 1971
ABSTRACT

METHODOLOGIES FOR WATER RESOURCES PLANNING: DDDP AND MLOM(TLOM)

This is the completion report for the first phase of a research program on advanced methodologies for water resources planning. A number of advanced concepts of water resources planning are investigated in order to develop practical methodologies for optimization of water resources systems. As a result, two new methodologies are developed; namely, the discrete differential dynamic programming (DDDP) and the multi-level optimization model (MLOM). The DDDP is a mathematical tool which can overcome the multi-dimensional difficulty that is often involved in the optimization of a complex water resources system, and moreover it can greatly save the cost of analysis by reducing the required computer storage capacity to several hundredths of and the required computer time to tenths of those required by the conventional dynamic programming technique. The MLOM is a novel scheme to decompose a complicated water resources system into a form that can be optimized at several levels for a general solution. Because of their practical usefulness, these two methodologies are now being introduced to actual water resources planning processes. The report describes the principles and procedures of DDDP and MLOM including examples: an optimal operation of a reservoir network for hydropower and irrigation, and a two-level optimization of farm irrigation systems (TLOM). The report also summarizes other accomplishments.

Chow, Ven Te
METHODOLOGIES FOR WATER RESOURCES PLANNING: DDDP AND MLOM(TLOM)
University of Illinois Water Resources Center Report No. 47. Oct 1971
KEYWORDS--dynamic programming/ *hydroeconomic analysis/ optimization/ *systems analysis/ *water resources planning
CONTENTS

Abstract ........................................................................................................... 1
List of Figures .................................................................................................. 3
List of Tables .................................................................................................... 4
I. INTRODUCTION .......................................................................................... 5
   1. Objective of the Study .............................................................................. 5
   2. Scope of the Study .................................................................................. 6
   3. Summary of Accomplishments ............................................................... 8
   4. Acknowledgements .................................................................................. 11

II. THE DISCRETE DIFFERENTIAL DYNAMIC PROGRAMMING (DDDP) ...... 13
   1. The Method ............................................................................................ 13
   2. The Procedure ....................................................................................... 16
   3. An Example ............................................................................................ 21

III. THE TWO-LEVEL OPTIMIZATION MODEL (TLOM) ............................. 27
   1. The Mathematical Model ....................................................................... 27
   2. First Level of Optimization .................................................................... 29
   3. Second Level of Optimization ................................................................ 36
   4. An Example ............................................................................................ 39

IV. DISCUSSIONS AND CONCLUSIONS ..................................................... 44
   1. On Discrete Differential Dynamic Programming .................................... 44
   2. On Two-Level Optimization Model ....................................................... 46

V. REFERENCES ............................................................................................... 49
LIST OF FIGURES

Figure 1. Examples of state subdomains at stage $n$.

Figure 2. Schematic representation of a trial trajectory, the boundaries defining corridor $C_k$, and optimal trajectory in $C_k$ of $k$-th iteration for a system with $m = 1$ and $T = 3$.

Figure 3. Flow chart showing steps of the DDDP approach.

Figure 4. Reservoir network of a simplified system.

Figure 5. Flow chart of main computer program.
LIST OF TABLES

Table 1. Irrigation Policy for a Typical Activity.

Table 2. Optimal Farm Plans for the Selected Levels of Irrigation Water Supply, Production Capital, and Farm Labor.

Table 3. Computer (IBM360/75) Time Requirements of the Proposed Method for the Solution of the System in Figure 4.
I. INTRODUCTION

1. Objective of the Study

This is the completion report of Phase I of a research program on advanced methodologies for water resources planning. The overall objective of the entire research program is to investigate a number of advanced concepts in water resources planning which are of basic importance but have not been generally introduced into practice, and to develop practical methodologies of applying such concepts to optimization of water resources systems. Modern concept of water resources planning is to formulate water resources problems as hydroeconomic systems and then to optimize the systems by operations research techniques.

Phase I of the research program has already produced new optimization tools for planning, including DDDP (Discrete Differential Dynamic Programming) and MLOM (Multi-Level Optimization Model). This report covers essentially a description of DDDP and the two-level optimization model (TLOM) as an example for MLOM. Because of their usefulness for practical purposes, these new tools are now being considered for actual water resources planning by several planning agencies. For research purposes, these tools could be further refined in order to enhance their effectiveness, they could be extended to produce additional needed tools for water resources planning, and they could be used for analysis of water resources problems to test hydrologic, economic, urban and other environmental aspects of the problems. Such possibilities are being investigated in a continuing Phase II of the research program.
2. **Scope of the Study**

The problem of water resources planning is to decide what water resources system to build. Since there are usually numerous alternatives to solve this problem, a sound rational procedure to select the best alternatives is needed. The advanced methodologies for water resources planning investigated in this research program are aimed to formulate such procedures for practical applications. Modern water resources systems are often complex beyond the conception of planner's judgment. But the tangible part of the systems can be simulated by mathematical models that are subject to programming techniques and computer solution for optimization, thus providing a sound basis for planning.

The DDDP and MLOM developed in Phase I of this research program are the advanced methodologies particularly suitable for the above-mentioned purpose.

The operating policies of multiple-unit and multiple-purpose water resources systems are commonly optimized by traditional dynamic programming with the use of high-speed digital computers. However, this method generally encounters two great difficulties that restrict its general application; they are the unmanageable large computer memory requirement and the excessive computer time requirement. The DDDP approach can ease these difficulties considerably. Whereas the traditional dynamic programming can usually deal with two, or three at most, state variables, the DDDP can easily handle seven, or up to eight, state variables. The DDDP procedure is an iterative method in which the recursive equation of dynamic programming [Chow, 1964] is
used to search for an improved trajectory among the discrete states in the neighborhood of a trial trajectory. The procedure starts with the trial trajectory in the state-stage domain to satisfy a specific set of initial and final conditions of the system and then applies Bellman's recursive equation [Bellman, 1957] in the neighborhood, or called "corridor," of this trajectory. At the end of each iteration step a locally improved trajectory is obtained and then used as the trial trajectory in the next step. The step is repeated until a near optimal trajectory that determines the practically optimal operating policy of the system is found. Instead of searching for the optimal operating policy over the entire state-stage domain as in the case of the traditional dynamic programming, the DDDP narrows down successively to the optimal rule, thus saving a considerable amount of computer time. In the DDDP procedure, optimization is limited only to few lattice points around a trial trajectory, and the computer memory requirements are curbed substantially. Hence, the state variables that can be handled by this method can be increased. It is believed that further improvement of DDDP is very possible and is now being investigated in Phase II of the research program.

The MLOM can be illustrated by the two-level optimization model (TLOM) which is to be described in this report. The (TLOM) is applied to farm irrigation schemes in humid areas such as in Illinois. This two-level optimization should be extended to a three-level optimization for the purpose of resources allocation. For this purpose, the continuing Phase II research is to formulate a mathematical model which is capable of determining the optimal scale of development for a multiple-
unit system, or, taking irrigation for example, the optimal scale of irrigation development for a multi-farm scheme.

For the irrigation scheme, as an example, the physical system would consist of a number of ground water and surface water sources and a number of farms, or demand regions, each having different soil characteristics and each capable of producing a range of crops, subject to resource limitations on the farms. The model is designed to select the optimal allocation of the seasonal irrigation water supply to individual farms. In addition, it selects the irrigation systems and the level and extent of irrigation development which best suit the needs of individual farms, and which yield maximum expected profit for the entire system.

The basic problem involves a large number of variables and is complicated by the fact that the return from the system is a function not only of the design variables but also of the system operation. This suggests the need for a multi-level optimization solution technique. By this means the entire system is decomposed into separate components, or subsystems, each one of which is optimized independently of the others. The subsystems are then combined into larger systems, knowing that if the adding is done optimally there is no need to revise the earlier allocations. This process may be repeated in steps for higher levels of optimization until the complete system is optimized.

3. **Summary of Accomplishments**

Phase I of the research program has achieved the following major accomplishments:

(1) Completion of two publications on annotated bibliography and
review of programming techniques for water resources systems analysis:


(2) Completion of two publications on annotated bibliography and review of stochastic processes for water resources systems analysis:


(3) Development of a new discrete differential dynamic programming (DDDP) technique for practical application to optimize operating rules of complex water resources systems. The technique applies to systems involving up to seven or more state variables, thus providing a breakthrough in optimization technology and outmoding the conventional dynamic programming which can handle only two or three state variables at the most. For this development, a paper was presented at the 1970 Fall Meeting of the American Geophysical Union:
Heidari, M., Chow, V. T., Meredith, D. D., a discrete differential dynamic programming approach to water resources systems analysis, paper presented at the 1970 Fall Meeting of American Geophysical Union, San Francisco.

and also the following report and paper are published:


(4) Development of a new concept of multi-level optimization model (MLOM) technique for resources allocation, providing a joint use of linear and dynamic programmings. A two-level optimization model for irrigation schemes particularly suitable to Illinois has been formulated for practical application. For this development, a paper was presented at the 1971 Annual Meeting of the American Geophysical Union:


and also the following report and paper are published:


(5) Investigation of the theory of network analysis and geometric programming for application to a demand-and-supply water resources system.
(6) Investigation of mathematical models for water quality and pollution control.

(7) Investigation of mathematical models for flood control.

(8) Investigation of mathematical models for recreation purposes.

(9) The following three doctoral theses were completed:


(10) A general report on water resources systems planning was presented at an international conference:


This report will further describe the DDDP and MLOM(TLOM) as mentioned in the above items (3) and (4), whereas investigations mentioned in items (5), (6), (7) and (8) will not be further discussed as they are in progress in Phase II of the research program.

4. Acknowledgements

The research program reported here is directed by Ven Te Chow. Other investigators engaged in this program at one time or another include the following: Manoutchehr Heidari, Sotirios J. Kareliotis, Peter V. Kokotović, Dale D. Meredith, and John E. Schaufelberger. The research assistants involved in this investigation at one time or another were: Eugene S. Cetwinski, Gonzalo Cortes, Gabriel Escallon, Freddy Isambert, James S. Windsor, and Chang-Lung Yin.
Several meetings discussing problems of the research were held with outside consultants including: Dr. W. A. Hall, Professor of Agricultural Engineering, University of California at Riverside, California; Mr. Arden O. Weiss, Director of Systems Engineering, Texas Water Development Board, Austin, Texas; Dr. W. L. Meier, Professor of Industrial Engineering, Texas A & M University, College Station, Texas; Professor C. Schreck of Technical University of Berlin, West Germany; Dr. T. Dracos, Director of Institute of Hydromechanics and Water Resources, Swiss Federal Technical University, Zurich, Switzerland; Dr. N. A. Kartvelishvili, Professor of Geography, Moscow State University, Moscow, U.S.S.R.; Dr. Earl R. Swanson, Professor of Agricultural Engineering, University of Illinois at Urbana-Champaign, Urbana, Illinois; and Mr. L. A. Beard, Hydrologic Engineering Center, U.S. Army Corps of Engineers, Davis, California.
II. THE DISCRETE DIFFERENTIAL DYNAMIC PROGRAMMING (DDDP)

1. The Method

The DDDP is an iterative technique in which the recursive equation of dynamic programming is used to search for an improved trajectory among the discrete states in the neighborhood of a trial trajectory.

Consider the dynamic system whose state equation is

$$s(n) = \phi[s(n - 1), u(n - 1), n - 1]$$  \hspace{1cm} (1)

$$n = 1, 2, ..., N$$

where \(n\) is an index specifying a stage (beginning of a time increment), \(N\) is the total number of time increments into which the time horizon has been divided, \(s(n)\) is an \(m\)-dimensional state vector at stage \(n\) (\(m\) being the number of state variables), \(u(n - 1)\) is a \(q\)-dimensional decision vector at stage \(n - 1\) (\(q\) being the number of decision variables), and

$$s(n) \in S(n) \quad u(n) \in U(n)$$  \hspace{1cm} (2)

where \(S(n)\) is the admissible domain in the state space at stage \(n\), and \(U(n)\) is the admissible domain in the decision space at stage \(n\). In water resources systems, for example in a network of reservoirs, state refers to storage, and decision refers to release from storage. The objective function to be maximized is

$$F = \sum_{n=1}^{N} R[s(n - 1), u(n - 1), n - 1]$$  \hspace{1cm} (3)
where $F$ is the sum of returns from the system over the time horizon and $R[s(n - 1), u(n - 1), n - 1]$ is the return obtained as a result of a decision $u(n - 1)$ that is made at stage $n - 1$ with the system in state $s(n - 1)$ and that lasts until stage $n$.

The forward algorithm of dynamic programming may be used to optimize (3) over $n$ stages as follows:

$$F^*[s(n), n] = \max_{u(n-1) \in U(n-1)} \{R[s(n - 1), u(n-1), n - 1] + F^*[s(n - 1), n - 1]\}$$

where $F^*[s(n), n]$ is the maximum total of the returns from stage 0 to stage $n$ when the state at stage $n$ is $s(n)$. Let us solve (1) for $s(n - 1)$,

$$s(n - 1) = \theta[s(n), u(n - 1), n - 1]$$

Substituting (5) into (4), we obtain the following recursive equation:

$$F^*[s(n), n] = \max_{u(n-1) \in U(n-1)} \{R[\theta, u(n - 1), n - 1] + F^*[s(n - 1), n - 1]\}$$

which may be solved for every $s(n)$, as a function of $u(n - 1)$ only.

Solution of (6) for a specific state in (2) provides an optimum $u(n - 1)$; i.e., the optimum decision that should be made for some state at stage $n - 1$ to bring the system to the specific state at stage $n$. 

14
Let us assume that the objective function (3) for the system of (1) is to be optimized subject to (2) and that the m-dimensional state vectors at the initial and final stages are specified such that

\[ s(0) = a(0) \quad s(N) = a(N) \]  \hspace{1cm} (7)

In the proposed DDDP approach a trial sequence of admissible decision vectors, \( u'(n), n = 0, 1, \ldots, N - 1 \), called the trial policy, that satisfies (2) is assumed, and the state vectors at different stages are then determined. The sequence of values of the state vector satisfying (2) and (7) is called the trial trajectory and is designated by \( s'(n), n = 0, 1, \ldots, N \). For invertible systems, which will be defined later, it is possible first to assume an admissible trial trajectory \( s'(n), n = 0, 1, \ldots, N \), and then to use it to calculate the trial policy \( u'(n), n = 0, 1, \ldots, N - 1 \).

Introducing \( u'(n) \) and \( s'(n) \) into (3) we obtain \( F' \) or

\[ F' = \sum_{n=1}^{N} R[s'(n - 1), u'(n - 1), n - 1] \]  \hspace{1cm} (8)

where \( F' \) is the total return due to the trial trajectory and policy over the entire time horizon. \( F' \) may not be the optimum return.

Now, consider a set of incremental m-dimensional vectors

\[
\Delta s_i(n) = \begin{bmatrix}
\delta s_{i1}(n) \\
\delta s_{i2}(n) \\
\vdots \\
\delta s_{ij}(n) \\
\vdots \\
\delta s_{im}(n)
\end{bmatrix}
\]

with \( n = 0, 1, \ldots, N \) and \( i = 1, 2, \ldots, T^m \)  \hspace{1cm} (9)
whose j-th component $\delta s_{ij}(n)$, $j = 1, 2, ..., m$, can take any one value $\sigma_t$, $t = 1, 2, ..., T$, from a set of assumed incremental values of the state domain. The value $\sigma_t$ is the t-th assumed increment from the state domain and T is the total number of assumed increments from the state domain. Thus the total number of $\Delta s_i(n)$ vectors at stage n is $T^m$. When added to the trial trajectory at a stage, these vectors form an m-dimensional subdomain designated by $D(n)$.

$$s'(n) + \Delta s_i(n) \quad i = 1, 2, ..., T^m \quad (10)$$

Note that one value of $\sigma_t$ must be zero since the trial trajectory is always in the subdomain. In Figure 1 two such subdomains for $m = 2$, $T = 4$ and $m = 3$, $T = 3$ are presented. All $D(n)$, $n = 0, 1, ..., N$, together are called a 'corridor' and designated by C as shown in Figure 2 by the space between two solid lines for a system with $m = 1$, $T = 3$, and $n = 10$.

2. The Procedure

In DDDP a corridor C is used as a set of admissible states, and the optimization constrained to these states is performed by means of the recursive relation (6). The value of return $F$ obtained is at least equal to or greater than $F'$ in (8). If $F$ is greater than $F'$, the corresponding trajectory and policy obtained from corridor C are used in the next iteration step as the trial trajectory and trial policy. Thus the k-th iteration step is as follows:

1. Use the results $[s^*(n)]_{k-1}$ and $[u^*(n)]_{k-1}$ of the $(k - 1)$th iteration step as the trial trajectory and policy for the k-th iteration step, i.e.,
A state sub-domain $D(n)$ defined by 16 lattice points in the neighborhood of $s(n)$ for a 2-dimensional state vector and $T = 4(\sigma_{j,1} = +2.0, \sigma_{j,2} = +1.0, \sigma_{j,3} = 0, \sigma_{j,4} = -1.0$ for $j = 1,2$).

A state sub-domain $D(n)$ defined by 27 lattice points in the neighborhood of $s(n)$ for a 3-dimensional state vector and $T = 3(\sigma_{j,1} = +1.0, \sigma_{j,2} = 0, \sigma_{j,3} = -1.0$ for $j = 1,2,3$).

FIGURE 1. Examples of state sub-domains at stage $n$
FIGURE 2. Schematic representation of a trial trajectory, the boundaries defining corridor $C_k$, and optimal trajectory in $C_k$ of $k$-th iteration for a system with $m = 1$ and $T = 3$.
2. Select $[\sigma_1^k, [\sigma_2^k, \ldots, [\sigma_T^k]$ to define the k-th corridor $C_k$, and use (6) to maximize $F$ subject to $s(n) \in C_k$.

3. Among the optimum trajectories in corridor $C_k$, trace the optimum trajectory satisfying boundary conditions (7) $[s^*(n)]_k$, and the corresponding optimum policy $[u^*(n)]_k$.

4. Determine $F_k^*$; if $F_k^* - F_{k-1}^* \leq \varepsilon$ where $\varepsilon$ is some prespecified constant, stop the iteration; otherwise go to step 1.

Figure 3 shows the flow chart of this procedure.

Since the boundary conditions (7) must be satisfied, one may exclude from the analysis all the states in the subdomain at stage $n = 0$ except $s'(0) = a(0)$. If in step 3 the trajectory having a final state $a(N)$ is traced, the preservation of boundary conditions (7) is guaranteed.

Note that in the course of the iteration process, the corridor size may be varied gradually by choosing different $[\sigma_t^k], t = 1, 2, \ldots, T$, in step 2. If the corridor size is kept constant for every iteration and little or no improvement can be achieved after the k-th iteration, it is suggested that $[\sigma_t^k], t = 1, 2, \ldots, T$, then be reduced starting at the $(k + 1)$th iteration and that the process be continued with the new corridor size until another iteration that behaves like the k-th iteration is reached. Then the corridor size is further reduced starting at the next iteration, and the procedure is repeated until the condition in step 4 is
FIGURE 3. Flow chart showing steps of the DDDP approach
satisfied. Note that it is also possible to assume a different set of $\sigma_t$ increments for each state variable.

3. An Example

The following simplified system, which was formulated and solved by Larson [1968] by linear programming and successive approximation dynamic programming, was solved by means of the proposed approach.

The operating policy of the four-dimensional ($m = 4$) reservoir network presented in Figure 4 is to be optimized over 12 operating periods ($N = 12$). The inflows into reservoirs 1 and 2 during any operating period are $y_1$ and $y_2$, respectively. The outflows or releases (decisions) $u_i(n)$, $i = 1, 2, 3, 4$, and $n = 0, 1, \ldots, 11$, from the reservoirs are used to generate hydro-power, and $u_4(n)$ after passing through the turbines is diverted toward an irrigation project. The storages of the four reservoirs represent a four-dimensional state vector whose constraints during any operating period were set as

$$
0 \leq s_1(n) \leq 10 \quad 0 \leq s_2(n) \leq 10
$$

$$
0 \leq s_3(n) \leq 10 \quad 0 \leq s_4(n) \leq 15
$$

$$
n = 0, 1, \ldots, 12
$$

The constraints on decisions during any operating period are

$$
0 \leq u_1(n) \leq 3 \quad 0 \leq u_2(n) \leq 4
$$

$$
0 \leq u_3(n) \leq 4 \quad 0 \leq u_4(n) \leq 7
$$

$$
n = 0, 1, \ldots, 11
$$

21
FIGURE 4. Reservoir network of the simplified system
The system equations expressing the dynamic behavior of each component at any stage \( n \) are

\[
\begin{align*}
    s_1(n) &= s_1(n - 1) + y_1 - u_1(n - 1) \\
    s_2(n) &= s_2(n - 1) + y_2 - u_2(n - 1) \\
    s_3(n) &= s_3(n - 1) + u_2(n - 1) - u_3(n - 1) \\
    s_4(n) &= s_4(n - 1) + u_3(n - 1) + u_4(n - 1) - u_4(n - 1)
\end{align*}
\]

(14)

\( n = 1, 2, \ldots, 12 \)

The inflows were set at

\[
\begin{align*}
    y_1 &= 2 \\
    y_2 &= 3
\end{align*}
\]

(15)

for all time increments. All the preceding variables and constants have units of volume.

The performance criterion to be maximized is the sum of the returns due to power generated by the four power plants and the return from the diversion of \( u_4(n) \) to the irrigation project.

\[
F = \sum_{n=0}^{11} \sum_{i=1}^{4} b_i(n)u_i(n) + \sum_{n=0}^{11} b_5(n)u_4(n)
\]

\[
+ \sum_{i=1}^{4} g_i[s_i(N), a_i(N)]
\]

(16)

where \( F \) is the total return from the system for the 12 time periods, \( b_i(n) \) is the unit return due to activity \( i, i = 1, \ldots, 5 \), during a period
starting at stage \( n \) and lasting until stage \( n + 1 \), and \( g_i[s_i(N), a_i(N)] \) is a function that assesses a penalty to the system when the final state of the \( i \)-th component of the system at stage \( N \) is \( s_i(N) \) instead of the desired state \( a_i(N) \), \( i = 1, 2, 3, 4 \). Such a penalty function is necessary for traditional dynamic programming for which boundary conditions may not be satisfied.

The penalty function in (16) was assumed to be

\[
g_i[s_i(N), a_i(N)] = -40(s_i(N) - a_i(N))^2
\]

\[s_i(N) \leq a_i(N)\] (17)

\[g_i[s_i(N), a_i(N)] = 0, \text{ otherwise}\]

The desired state vectors of the initial and final stages for \( i = 1, 2, 3, 4 \) were assumed to be

\[
a(0) = \begin{bmatrix} 5 \\ 5 \\ 5 \\ 5 \end{bmatrix}, \quad a(N) = \begin{bmatrix} 5 \\ 5 \\ 5 \\ 7 \end{bmatrix}
\] (18)

There are a total of five activities in the above criterion: four hydro-power generation activities and one irrigation activity.

Larson's [1968] solution to this problem (solved by successive approximation dynamic programming and checked by linear programming), using \( \sigma_1 = +1.0, \sigma_2 = 0.0, \sigma_3 = -1.0 \), gives the optimum return as 401.3. (The
optimal trajectory presented in table 12.11 of Larson [1968] is slightly in error, as noted through private communication with R. E. Larson, 1969.)

Application of the proposed approach to this system, which is invertible, starts with the assumption of a trial trajectory \( s'(n) \), \( n = 0, 1, ..., 12 \), satisfying (12) and (18). When substituted in (14) with constants in (15), the trial trajectory will produce a trial policy \( u'(n) \), \( n = 0, 1, ..., 11 \), which should be checked for constraints (13). It is considerably easier to treat this problem as a free end point problem, i.e., not to satisfy either the initial or the final boundary condition. However, the simplicity of the system equations in this example makes it possible to satisfy both boundary conditions in (18). The penalty function (17) is therefore not needed in the DDDP, since boundary conditions (18) are always satisfied. Three such trial trajectories are calculated. Next, three values of \( \sigma_t \) are assumed:

\[
\sigma_1 = 1.0 \quad \sigma_2 = 0 \quad \sigma_3 = -1.0 \quad (19)
\]

and a set of \( T^m = 3^4 \) incremental vectors is formed that when added to the trial trajectory produces a subdomain consisting of 81 lattice points at each stage.

The problem is solved three times. Each time the calculations start with one of the trial trajectories 1, 2, or 3. All three solutions converge to the optimal trajectory. The number of iterations required for convergence is 7, 12, and 7 for trial trajectories 1, 2, and 3, respectively.
After the required iterations for the three trial trajectories the reduction of $\sigma_t$, $t = 1, 2, 3$, does not produce any improvement in the return. This result may be attributed to three factors: (1) the optimal trajectory of this system follows full integer states; (2) the trial trajectories are chosen so that they follow full integer states; and (3) the values of $\sigma_t$, $t = 1, 2, 3$, for all stages are set at full integers. In a separate try, trial trajectory 1 is subjected to the iteration process with

$$
\sigma_1 = 1.3 \quad \sigma_2 = 0 \quad \sigma_3 = -1.3
$$

for all stages starting with iteration 1, and the idea of reducing $\sigma_t$, $t = 1, 2, 3$, is employed. A total of 18 iterations in four corridors produces a return of 399.06 as compared to the optimal return of 401.3. Thus one concludes that when the optimal values of $\sigma_t$ are unknown, the result may be considered only an approximation to optimum.
III. THE TWO-LEVEL OPTIMIZATION MODEL (TLOM)

1. The Mathematical Model

As an illustration of the concept of the MLOM, a two-level optimization model for farm irrigation systems is described here. The basic problem considered is the formulation of a mathematical model for the design and analysis of a multi-crop, multi-soil, farm irrigation system. Fixed resources, or limitations, in the model consist of land, labor, capital, and seasonal irrigation water supply. A two-level optimization technique is used to break the system down into a number of manageable subsystems. The most logical decomposition in this case is to treat the individual crop-soil combinations as acre units and to optimize their seasonal outputs separately and independently.

Dynamic programming is selected for the first level of optimization since it is ideally suited to the analysis of multi-stage decision processes, and may easily be structured to handle the uncertain nature of weather elements. Several levels of irrigation development are considered for each crop-soil combination and each irrigation system, by treating the irrigation interval and application amounts as system parameters. At the next level of optimization a linear programming model is used to select the optimal crop mix, the level and extent of irrigation development, and the type of irrigation system which maximize the expected farm profit without violating any of the farm constraints. The flow chart of the model is presented in Figure 5. It should be noted that in order to improve the efficiency of optimization, the dynamic programming may be replaced by DDDP. However, this is not necessary in the present discussion of the model.
FIGURE 5. Flow chart of the main computer program
2. **First Level of Optimization**

Dynamic programming considers a physical system which is to be operated over a number of consecutive stages, or time periods, in order to optimize a certain objective function. At the beginning of each stage the system is in a certain state which may be described by a state vector if several states are involved. During any one stage the system is subjected to changes in state which are either deterministic or probabilistic. A process in which a choice of transformation exists is generally referred to as a decision process. If the decisions are made during successive stages it is termed a sequential decision process. The term policy is used to define a particular sequence of decisions.

In the case of an irrigation system the transformation equation may be represented by the following water balance equation:

\[
S_{k-1} = \Delta S_k + \min\{F_k; \max\{S_k + d_k + \rho R_k - E_{ak}; 0\}\}
\] (21)

where \(S_k\) and \(S_{k-1}\) are the states of the soil reservoir at the start of time periods \((k)\) and \((k-1)\), respectively; \(\Delta S_k\) is the available moisture added to the soil reservoir by extension of the root system during time period \((k)\); \(F_k\) is the field capacity of the effective root zone at the start of time period \((k)\); \(d_k\) is the decision variable which represents the net irrigation input to the soil reservoir at the start of time period \((k)\); \(\rho\) is a dimensionless coefficient which converts total rainfall to effective rainfall, and is assumed equal to unity in this study; \(E_{ak}\) is the actual evapotranspiration during time period \((k)\); \(R_k\) is the total rainfall during time period \((k)\). The inner maximization is required.
to ensure that the content of the soil reservoir does not fall below permanent wilting point. The outer minimization is used to prevent the soil moisture content exceeding field capacity.

Equation (21) shows that the state of the soil reservoir at the start of time period \((k-1)\) depends not only on the state and decision variables at the start of time period \((k)\) but also on the extension of the crop root system during time period \((k)\) and the random weather variables. The transition from one state to the next is therefore not given exactly, but in terms of the joint probability of occurrence of the random weather variables \(E_k\) and \(R_k\), i.e., \(P(E_k, R_k)\).

Assume now that the system is being operated on the basis of an n-day irrigation cycle so that decisions are only permitted every n-th day of the irrigation period. Assume also that the total irrigation period extends over \(K\) days. Then the serial optimization problem is to maximize the total expected profit \(p_K\) over the set of decision variables \(d_1, d_2, \ldots, d_K\).

Denoting \(f_K(S_K)\) as the maximum expected return for the entire period, the functional, or recursive, equation may be derived by defining the return for the \(k\)-th period as

\[
r_k(S_k, d_k, E_k) = B_k(S_k, d_k, E_k) - C_k(d_k)
\]

where \(k = 1, 2, \ldots, K\); \(B_k(S_k, d_k, E_k)\) is the incremental benefit in dollars for the \(k\)-th time period; \(S_k\) is the state of the soil moisture in the crop root zone at the start of the \(k\)-th time period; \(E_k\) is the average daily atmospheric demand for the \(k\)-th time period; and \(C_k\) is the cost of the irrigation decision \(d_k\) for the \(k\)-th time period. The irrigation
decision is defined as

\[ d_k = 0 \]  

(23a)

if \( k \) does not coincide with a decision stage, i.e., a point in time when a decision is made whether or not to irrigate, and

\[ d_k = 0 \text{ or } \min\{F_k - S_k; \max \text{ irrigation application}\} \]  

(23b)

if \( k \) coincides with a decision stage. In (23b) the outer minimization is required to ensure that the soil reservoir is not filled beyond field capacity \( F_k \).

The total return for the \( K \) time periods is given by the sum of the individual daily returns:

\[ p_K(S_k, \ldots, S_1; d_k, \ldots, d_1; E_k, \ldots, E_1) = \sum_{k=1}^{K} r_k(S_k, d_k, E_k) \]

subject to (21)

(24)

In order to determine the optimal operating policy and the system output, knowledge is required concerning climatic variations at the site of the irrigation project. One method to determine hydrologic record for design use is to define the random nature of the rainfall and evaporation by some specific probability distributions, and then use the theory of probabilistic dynamic programming in the solution procedure.

Denoting the joint probability distribution of daily rainfall and evaporation on the \( k \)-th day by \( P_k(E_k, R_k) \), and the corresponding marginal
probability distribution of daily evaporation by $P_k(E_k)$, the objective now is to maximize the expected value of the return over the $K$-day period; thus

$$f_K(S_k) = \max_{d_k, \ldots, d_1} E[p_k]$$  \hspace{1cm} (25)

Equation (25) cannot be solved directly. However, by embedding it in the following equation, a solution is readily obtainable:

$$f_K(S_k) = \max_{d_k} \left[ \sum_{E_k} P_k(E_k) R_k(S_k, d_k, E_k) + \sum_{E_k} \sum_{R_k} P_k(E_k \cap R_k) f_{K-1}(S_{K-1}) \right]$$

subject to (21) \hspace{1cm} (26)

This relationship is the mathematical formulation of Bellman's principle of optimality [Bellman, 1957]. It shows that the optimal return $f_K(S_k)$ can be obtained from the optimal return $f_{K-1}(S_{K-1})$ by a one state, one decision, optimization problem. The functional equation therefore allows us to solve a sequence of related problems one at a time starting at the end stage and progressing backwards in time to the initial stage.

Adopting a plant-water production function which is based on the stress-day concept [Denmead and Shaw, 1962], (26) may be rewritten as follows:

$$f_k(S_k) = \max_{d_k} \left[ \sum_{E_k} P_k(E_k) \psi \left( \theta_k - \theta_{TL} \right) \Delta g_k P - C_k(d_k) \right]$$

$$+ \sum_{E_k} \sum_{R_k} P_k(E_k \cap R_k) f_{K-1}(S_{K-1})$$ \hspace{1cm} (27)
where $\theta_K$ is the percentage of the available soil moisture content at the start of the $K$-th time period and is defined as

$$\theta_K = \frac{S_K - \theta_{TL}}{F_K} \quad (28a)$$

if the decision is not to irrigate, and

$$\theta_K = \frac{(S_K + d_K) - \theta_{TL}}{F_K} \quad (28b)$$

if the decision is to irrigate. The term $S_K$ has a finite set of values in the range of available storage between permanent wilting point and field capacity at the start of period $K$, $\theta_{TL}$ is the turgor loss point related to crop potential evapotranspiration, $\Delta G_K$ is the harvestable portion of the plant potential growth increment during period $K$, and $P$ is the market value per unit growth increment.

In accordance with experimental evidence, if the available soil moisture content $\theta_K$ equals, or exceeds, the turgor loss function $\theta_{TL}$ on a particular day, the plant maintains full turgor and hence grows at the potential rate. On the other hand, if the available soil moisture content is less than the turgor loss function the plant becomes stressed and effectively loses a daily growth increment. Or expressed mathematically:

$$\psi(\theta_K - \theta_{TL}) = 1.0 \text{ in (27), if } \theta_K - \theta_{TL} > 0 \quad (29a)$$

$$\psi(\theta_K - \theta_{TL}) = 0 \text{ in (27), if } \theta_K - \theta_{TL} < 0 \quad (29b)$$
Equation (27) is therefore solved at each decision stage by comparing the expected net profits with, and without, irrigation at each of the feasible points in state space and then selecting the decision which yields maximum return. During intermediate time periods when the pipe system has been moved to alternate positions in the field, irrigation decisions are no longer possible for the unit area of crop under consideration, and only the total expected returns are computed.

By similar reasoning it is possible to obtain the corresponding equations for estimating the expected irrigation labor and applications for each month of the irrigation season. The total expected irrigation labor per acre at time \( (k) \) for a given initial state \( (S_k) \) is

\[
L_k(S_k) = \phi(d_k^*) H + \sum_{E_k} \sum_{R_k} P_k(E_k \cap R_k) L_{k-1}(S_{k-1})
\]

(30)

where \( k \) represents the number of days in the month; \( H \) represents the man-hours of labor per irrigation per acre; and \( \phi(d_k^*) \) is a step function which is defined as

\[
\phi(d_k^*) = 0
\]

(31a)

if the optimal decision is not to irrigate, and

\[
\phi(d_k^*) = 1
\]

(31b)

if the optimal decision is to irrigate. Similarly, the total expected irrigation application per acre at time \( (k) \) for a given initial state \( (S_k) \) is
where $d_k^*$ is the optimal decision at the $k$-th time period.

The results of this probabilistic analysis are the expected seasonal profits, and the expected monthly labor and irrigation requirements for each feasible state of the system at the start of the respective time periods. The next step in the analysis is to determine the probability of being in a specific state at the start of each month given the state of the system at the start of the irrigation period. This is accomplished by means of a forward algorithm.

Thus, let the state at time ($k$) be denoted by $S_k$, and the probability of transition from one state at time ($k$) to another state $S_{k-1}$ at time ($k-1$) be denoted by $P_k(S_k \cap E_k \cap R_k \cap S_{k-1})$. Then the probability of being in state $S_{k-1}$ at time ($k-1$), denoted by $\Pi_{k-1}(S_{k-1})$, is obtained by multiplying the probability of being in state $S_k$ at time ($k$), $\Pi_k(S_k)$, by the transition probability $P_k(S_k \cap E_k \cap R_k \cap S_{k-1})$, and then summing over all the feasible states $S_k$ and discrete values of $E_k$ and $R_k$ in the joint probability matrix at time ($k$). Thus

$$\Pi_{k-1}(S_{k-1}) = \sum_{S_k} \sum_{E_k} \sum_{R_k} P_k(S_k \cap E_k \cap R_k \cap S_{k-1}) \Pi_k(S_k)$$ \hspace{1cm} (33)

subject to (21)

and

$$\sum_{S_{k-1}} \Pi_{k-1}(S_{k-1}) = 1.0$$ \hspace{1cm} (34)
Equation (33) may be solved for successive time periods in terms of the initial state probabilities knowing the joint probability distribution of daily pan evaporation and rainfall and the vector of optimal decisions for each stage of the system. The end result is the vector of state probabilities at the start of each month.

Assuming now that the appropriate values of the expected monthly irrigation quantities \( I_k(S_k) \) and labor requirements \( L_k(S_k) \) have been stored at the corresponding points on the matrix using the backward algorithm, then the expected monthly irrigation quantity irrespective of state is

\[
I_k = \sum_{S_k} \Pi_k(S_k) I_k(S_k)
\]

(35)

And, the expected monthly irrigation labor is

\[
L_k = \sum_{S_k} \Pi_k(S_k) L_k(S_k)
\]

(36)

where \( k \) represents the first day of the month, and \( S_k \) is the set of feasible states.

3. Second Level of Optimization

Linear programming is a method whereby the optimal farm plan can be selected from among the multitude of choices open to the farmer. In the case of crop production the linear programming model not only selects the types of crops to be grown but also specifies the number of acres of land to be allocated to each crop and the optimal method of crop production.
Although a number of attempts have been made to apply this technique to the analysis of irrigation systems, these have not been entirely successful due to inherent limitations. This is due to the fact that the irrigation problem is basically a stochastic multi-stage decision process which involves both space and time allocations of the irrigation inputs and therefore is not suited to standard linear programming analysis. In addition, the random variations in soil moisture content and other environmental factors not only affect crop production but are also intimately connected with the problem of system design and operation. It would appear, therefore, that a need exists for a more general model which combines the properties of dynamic and linear programming models.

Linear programming differs from many other mathematical programming techniques in that the mathematical model is stated in terms of linear relationships. The complete mathematical statement of the crop production process includes a set of linear inequalities which represent the constraints on the problem and a linear function which describes the objective. The linear constraints form a convex polygon in n-dimensional space. Only points in this set satisfy the linear constraints and are regarded as feasible solutions. The corner points of this convex set of solutions are termed basic feasible solutions, and if there is an optimal solution to the problem at least one basic feasible solution will be optimal.

Assume now that the linear programming model is structured to consider N crops, each crop may be grown in J fields at L levels of irrigation intensity, a choice exists between M irrigation systems, and the model is constrained by I resources. Then the production possibilities of the farm
will consist of \( J \times L \times M \times N \) separate activities and these may be represented symbolically by the following matrix:

\[
\begin{align*}
& a_{i1111} x_{1111} + a_{i1211} x_{2111} + \ldots + a_{i1jnm} x_{jlmn} + \ldots + a_{i1jLMN} x_{jLMN} \leq b_i \\
& a_{i1111} x_{1111} + a_{i1211} x_{2111} + \ldots + a_{i1jmn} x_{j1mn} + \ldots + a_{i1jLMN} x_{jLMN} \leq b_i \quad (37)
\end{align*}
\]

where \( i = 1, 2, \ldots, I; j = 1, 2, \ldots, J; l = 1, 2, \ldots, L; m = 1, 2, \ldots, M; \) and \( n = 1, 2, \ldots, N. \)

An additional constraint is that the \( x_{j1mn} \)'s should not be negative, i.e.,

\[
x_{j1mn} \geq 0 \quad (38)
\]

The columns of the matrix \( S_{j1mn} \) represent the individual activities which may be included in the production process. The coefficients of these columns \( a_{ij1mn} \) describe the resource inputs per unit level of each activity. Each row of the matrix \( B_1 \) corresponds to some economic input, or resource level which may be limitational and is held constant during the analysis. The specific problem is to select the unknown levels of the column vectors which maximize the total net profits, i.e.,

\[
\text{Maximize } \sum_{j=1}^{J} \sum_{l=1}^{L} \sum_{m=1}^{M} \sum_{n=1}^{N} C_{j1mn} x_{j1mn} \quad (39)
\]

where the \( C \)'s are the gross unit revenues for each activity minus the total production costs.
In many problems the linear programming technique can be applied to situations in which the mathematical expressions for the system are nonlinear by solving the problem with linear approximations, or with a piecewise solution using linear portions of the actual expression. A special form of nonlinearity arises if there are joint interactions between some of the activities regarding either total usage of some resource or total measure of effectiveness.

The constraint data comprising acres in each field, labor available each month and acreage restrictions for crops should be readily available for most farms. However, expected yield data and expected irrigation labor and water requirements for each crop, each field, and each level of irrigation are not likely to be available, since these depend to a large extent on the random weather variations. To circumvent this difficulty a dynamic programming model is used at the first level of optimization to estimate the crop yields and additional resource requirements involved in the conversion from dryland to irrigated farming. The inclusion of cycle time and application amount at this level enables the model to consider different levels of irrigation intensity, and hence generate a variety of outputs as the water availability changes over time.

4. An Example

In order to show how the TLOM can be set up on a computer to handle a farm irrigation problem a Fortran IV program was developed as part of this study. This was used to predict the optimal scale of irrigation development for a hypothetical farm situation consisting of two soil types, or fields, each 150 acres in extent, and each capable of producing two crops: soybeans and corn.
For expository purposes it was assumed that a choice existed between only two types of irrigation equipment, the tow-line and self-propelled sprinkler systems, and that these could be characterised by the design parameters, cycle time, application amount per irrigation, and system efficiency. The alternative production activities incorporated in the model included four dry farming and 32 irrigated farming activities.

In order to indicate the farm adjustments to a change in resource levels and the effect of these adjustments on farm income, eight resource combinations were analyzed in the model. These consisted of two levels of monthly labor supply, seasonal irrigation water supply, and annual production capital.

Proceeding in the manner described in the previous section the optimal policy was determined for each crop activity. Table 1 shows a typical irrigation policy $IP(KS, I)$ in matrix form for one of the crop activities. Here $IP(KS, I) = 1$ is used to denote the decision to irrigate, and $IP(KS, I) = 0$ signifies the decision not to irrigate. The matrix therefore indicates for each stage $KS$ the level to which the available soil moisture is allowed to deplete before it becomes profitable to irrigate. The decision to irrigate at lower levels towards the beginning and end of the irrigation season reflects the lower possibility of soil moisture depletion.

Other results obtained from the dynamic programming model include the expected seasonal profit, the expected monthly irrigation labor, and the expected monthly irrigation water requirements per acre of each crop activity. This information constituted part of the basic input to the linear programming model. Other input data included the labor and non-
<table>
<thead>
<tr>
<th>IS</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
<th>16</th>
<th>17</th>
<th>18</th>
<th>19</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>6</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>7</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>8</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>9</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

*1st irrigation activity: tow-line system; 5-day cycle; 1/2-inch irrigations; field 1; corn*
labor input-output coefficients for the dry farming situation. These were selected from a Farm Management Manual [1969].

In formalizing the physical system varying degrees of reality can be achieved by the assumptions built into the model. A case in point arises in the selection of irrigation equipment. Self-propelled systems, for example, are manufactured only in a limited number of standard sizes. Another characteristic of these systems is that ideally they should only be used to irrigate one field crop at a time. In addition economies of size are inherent in their design. These factors tend to complicate the linear programming analysis, but nevertheless should be recognized in a more complete representation of the practical situation.

The solution of the farm model is summarized in Table 2. This shows that under the selected price and cost conditions the production of soybeans is not economic. These results also indicate significant differences in farm profits with changes in the level of resource availability. Further, as capital is substituted for labor, the model indicates that it becomes more profitable to select the more expensive self-propelled system in preference to the tow-line system with its higher labor requirements.
### TABLE 2

Optimal Farm Plans and Resource Allocations for the Selected Levels of Irrigation Water Supply, Production Capital, and Farm Labor

<table>
<thead>
<tr>
<th>Model Run</th>
<th>Resource quantities available</th>
<th>Field No.</th>
<th>Crop Selected</th>
<th>Crop acreage</th>
<th>Irrigated farming</th>
<th>Self-propelled system</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Capital $</td>
<td>Labor man-hours</td>
<td>Water supply acre ft.</td>
<td>Dry Farming</td>
<td>5-day</td>
<td>10-day</td>
</tr>
<tr>
<td>1</td>
<td>10,000</td>
<td>300</td>
<td>3000</td>
<td>100</td>
<td>1 Corn 41</td>
<td>109</td>
</tr>
<tr>
<td>2</td>
<td>10,000</td>
<td>300</td>
<td>3000</td>
<td>200</td>
<td>1 Corn 41, 2 Corn 150</td>
<td>109</td>
</tr>
<tr>
<td>3</td>
<td>20,000</td>
<td>300</td>
<td>3000</td>
<td>100</td>
<td>1 Corn 57</td>
<td>93</td>
</tr>
<tr>
<td>4</td>
<td>20,000</td>
<td>300</td>
<td>3000</td>
<td>200</td>
<td>1 Corn 88</td>
<td>62</td>
</tr>
<tr>
<td>5</td>
<td>10,000</td>
<td>260</td>
<td>2600</td>
<td>100</td>
<td>1 Corn 114</td>
<td>36</td>
</tr>
<tr>
<td>6</td>
<td>10,000</td>
<td>260</td>
<td>2600</td>
<td>200</td>
<td>1 Corn 114</td>
<td>36</td>
</tr>
<tr>
<td>7</td>
<td>20,000</td>
<td>260</td>
<td>2600</td>
<td>100</td>
<td>1 Corn 114</td>
<td>36</td>
</tr>
<tr>
<td>8</td>
<td>20,000</td>
<td>260</td>
<td>2600</td>
<td>200</td>
<td>1 Corn 36</td>
<td>114</td>
</tr>
</tbody>
</table>
IV. DISCUSSIONS AND CONCLUSIONS

1. On Discrete Differential Dynamic Programming

The major factors that inspired the DDDP approach were the inherent drawbacks of traditional dynamic programming, namely, memory capacity and computer time requirements. By limiting optimization to a few lattice points around a trial trajectory, the memory requirements appear to have been curbed substantially. To illustrate this point numerically, consider the memory requirements of the example.

The problem has four state variables, whose admissible ranges are given in (17). With the values of $\sigma_t$ given in (19) the DDDP requires 243 words of computer memory, whereas traditional dynamic programming using the same grid size would require 63,888 words.

Another major difficulty in applying traditional dynamic programming is the computer time required because of the number of computations and comparisons that must be performed at each lattice point. At each stage of the example there are 21,296 lattice points. If the domain of the decisions given in (13) is divided into lattice points with $\Delta u = 1$ unit, a total of $4 \times 5 \times 5 \times 8 = 800$ combinations of decisions must be tested at each state lattice point of each stage. By limiting the optimization to the neighborhood of a trial trajectory, the number of lattice points is reduced, and therefore fewer tests will have to be made per state of each stage. Furthermore, if the system is invertible, even greater efficiency may be achieved. For example, if $T = 3$ at each stage, then for a four-dimensional invertible problem there are only $3^4 = 81$ possibilities that states at stage $n - 1$ may lead to a particular state
at stage \( n \). Therefore at a particular state of stage \( n \) only 81 tests instead of 800 will need to be made.

Table 3 summarizes the processing time of IBM 360/75 required to solve the example by means of the proposed approach. The number of iterations in this table is one more than that needed to arrive at the optimum results. The last iteration is required to confirm that optimum results have been reached in the previous iteration.

If the values of \( \sigma_t \) are not chosen properly, it is possible for the procedure to converge to a local minimum or maximum. Jacobson and Mayne [1970] and the results of the present study indicate that it may be advisable to calculate the values of \( \sigma_t \) as a function of the stage either at the beginning of each iteration or when the results of two successive iterations show little or no improvement in the return.

TABLE 3. Computer (IBM 360/75) Time Requirements of the Proposed Method for the Solution of the System in Figure 4.

<table>
<thead>
<tr>
<th>Nominal Trajectories</th>
<th>Operating Periods</th>
<th>No. of Interations</th>
<th>Total Processing Time, sec</th>
<th>Processing Time per Iteration, sec</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>12</td>
<td>8</td>
<td>35.32</td>
<td>4.42</td>
</tr>
<tr>
<td>2</td>
<td>12</td>
<td>13</td>
<td>48.39</td>
<td>3.72</td>
</tr>
<tr>
<td>3</td>
<td>12</td>
<td>8</td>
<td>31.04</td>
<td>3.88</td>
</tr>
</tbody>
</table>

One must realize that the difficulties encountered in the choice of the trial trajectory and in the determination of the state subdomain are by no means limited to the DDDDP approach. Other iterative optimization techniques, such as the gradient methods and the second variation methods, also face the same problems.
2. **On Two-Level Optimization Model**

Although the proposed TLOM for irrigation systems considers many of the system variables affecting plant growth, it is recognized that it is still a highly simplified version of the physical system and that there is still much need for further refinement. One assumption is that water supply is the important variable which controls plant growth and development. Climatic influences, however, go far beyond the obvious limiting effects of drought. Air temperature, for example, not only affects evaporation losses but may also depress crop yields if it is far removed from the optimal value for crop growth and development.

Another assumption is that daily potential growth increments during the irrigation season are constant and independent of each other, and that they may be summed to obtain a measure of crop yield at the end of the growing season. Different varieties of crops, however, often exhibit marked differences in response to water stress, and these differences vary with the stage of plant development. It has also been observed in certain instances, that the effect of moisture stress incurred in the early stages of plant development may affect growth in later periods. It would appear, therefore, that some additional effort is required to determine the effect of inter-period dependencies on crop growth and development, and to find how the daily growth increments should be weighted to reflect growth potential at each stage of plant development. Both additive and multiplicative relations may be required to define more adequately the crop systems in the more complete model.

Furthermore, maximization of expected profit has been used as the criterion function in this analysis, but it is not always the primary
objective of farm management. The farmer is also influenced to some extent by the variability in production and demand, with the result that risk plays an important role in the farmer's decision making. Under normal circumstances irrigation should result in less variability in crop production through better control of the environment. Some assessment of this reduced variability should, therefore, be included in the model analysis to obtain a more complete evaluation of the returns from irrigation development.

Yet another extension which may be introduced into the model to increase its usefulness relates to the water balance relationship in the effective root zone of the plants. In general it cannot be assumed that all the precipitation falling on the land can be regarded as effective rainfall. On the contrary, a large portion of the rainfall occurs as intensive short duration storms which compact and seal the soil causing some of the water to be lost as surface runoff. The actual processes involved in the soil moisture budget usually depend on a number of factors which vary over time and space. A knowledge of the general form of these relationships and their space and time variations is therefore essential in a more complete representation of the physical system.

Despite the fact that the model presented in this study does not completely describe the entire physical system, it does provide a systematic procedure for the design and analysis of irrigation systems. Under the present status of imperfect knowledge on the physical characteristics of the system, particularly in the area of plant growth and development, and deficiencies in the input data, the proposed model can
nevertheless serve as a good guide for rational planning of irrigation systems in humid areas.
REFERENCES


