THE USE OF A DYNAMIC DIGRAPH STRUCTURE IN A POPULATION SIMULATION MODEL FOR GRAIN SORGHUM

By

JESS WALTER CURRY, JR.

B.S. Oklahoma State University
Stillwater, 1970

A MASTER'S REPORT

submitted in partial fulfillment of the
requirements for the degree

MASTER OF SCIENCE

Department of Computer Science
KANSAS STATE UNIVERSITY
Manhattan, Kansas
1977

Approved by:

Major Professor

ACKNOWLEDGEMENTS

The project detailed in this report was funded by the Kansas State University Agricultural Experimental Station, Floyd W. Smith, Director.

I would like to thank Dr. Richard L. Vanderlip who spent many hours directing this project and Dr. Linda G. Shapiro who advised and directed me throughout.

TABLE OF CONTENTS

LIST OF FIGURES AND TABLES

The Use of a Dynamic Digraph Structure in a Population Simulation Model for Grain Sorghum

I. INTRODUCTION

Sorghum (often referred to as milo) has, through successful research efforts, grown to be one of the leading world crops both in production area and in food value. As of 1969, sorghum ranked fifth in acreage of crops in the world, being exceeded only by wheat, rice, corn, and barley. Perhaps of greater importance is the fact that it is a major food crop in the most heavily populated regions of the world. The three top producers of sorghum in 1970 were the United States, Mainland China, and India.

Because of its resistance to drought and heat, sorghum has been a prominent crop in the southern half of the Great Plains area since the early 19th Century. It was a highly successful livestock food for the early pioneer farmers especially in the semi-arid regions. Also, due primarily to research, sorghum production grew seven-fold between 1940 and 1968 alone, and as a result, it has become an important cash crop for the American farmer. The development of early maturing varieties allowed expansion to the higher altitudes to the west and cooler areas to the north. Shorter varieties allowed automation of harvesting while development of insect and disease resistant hybrids have increased per acre production considerably.

Kansas ranks second only to Texas in total sorghum production and has played a key role in the continued improvement of sorghum. Significant contributions have been made and are presently being made by the agricultural experiment stations of Texas and Kansas. One such contribution was the development in 1975 of the first computer simulation model for grain sorghum by Drs. G. F. Arkin and J. T. Ritchie of the Blacklands Conservation Research Center, Temple, Texas and Dr. R. L. Vanderlip of the Agronomy Department at Kansas State University <1>. This model was developed primarily as a tool for improving management decisions for increased sorghum crop production.

The Arkin, Ritchie, Vanderlip model was based upon simulating a single sorghum plant from emergence to maturity under the weather and soil conditions given for the sorghum field under consideration. A per acre production figure was derived by multiplying the production of the single plant by the number of sorghum plants per acre. This model, which will be referred to as the Single Plant Model, was a fairly sophisticated model with a good degree of accuracy. However, a detailed study of the Single Plant Model showed a complete lack of program structuring and internal documentation. More importantly, the study raised the question as to the accuracy of modeling an entire field based upon the output of one single plant. For example, in the original simulation model, no consideration was given to field variables such as soil variability, seed distribution, emergence date variance, and plant leaf area and tillering.

The current project was instigated to design a field level simulation model which: 1) would have a modular structure, 2) would have sufficient internal documentation, and 3) would expand the model developed by Drs. Arkin, Ritchie, and Vanderlip to include simulation of the variability within a field of sorghum. The new simulation model developed during this project will be referred to as the Field Level Model. The basis of this model is a data structure representing a directed graph (digraph). Each node in the data structure will represent a portion of the field which has some significant difference from the rest of the model. The digraph is the result of plotting the nodes from day to day as they diverge and converge (see figure 3). For example, if half of the field holds water very well while the remainder can not, then the model would be split (the graph will diverge) into two equal parts (or nodes) each of which will be simulated based upon its separate characteristics. The nodes will also be allowed to merge (or converge) if the difference between the two parts is eliminated. If in the above example it rains enough to allow sufficient water for the entire field then the nodes would converge. The converging and diverging of nodes will be based upon internal policies programmed into the model at a later date. It is the objective of the current project to develop the data structure and prove that it is applicable and useful.

II. The Single Plant Model

It is difficult to present the workings of the Single Plant Model because of the initial lack of modularity and program structure. To overcome this problem the model will be presented in a more logical format with proper modularization and program flow. This represents the state of the model after it was reorganized for use in the Field Level Model but before any new modules were added.

Figure 1. gives a high level diagram for the Single Plant Model showing the various subroutines in the order that they are called. Figure 2. shows a hierarchy diagram of the model before modularization was undertaken. The main routine initializes the critical variables to zero and reads the remaining constant variables from cards. These variables represent plant data, soil data, planting data and location data, all of which remain unchanged throughout the simulation. Table 1 gives a breakdown of the information used in each data type.

After the constant data has been read, up to one year's climatic data is read. This data gives the maximum and minumum temperatures, solar radiation, and rainfall. As each day's climatic data is read, the day's evaporation is calculated by the subroutine EVAP. This value is subtracted from the available water content of the soil by a second subroutine called SOLWAT. SOLWAT also adds rainfall amounts to the soil contents when applicable and then calculates the moisture limiting factor for photosynthesis. Both of these

TABLE

Plant Data

Leaf number -- total number of leaves produced

Leaf area -- maximum area of each individual leaf, cm2

Planting Data

Planting date

Plant population

Row width

Row direction

Climatic Data (daily from planting to maturity)

Maximum temperature, C

Minimum temperature, C

Solar radiation, langleys per day

Rainfall, cm

Location Data

Extractable soil water capacity, cm

Initial extractable soil water content, cm

Latitude

Table 1. Input data required.

The Single Plant Model
(After Modularization)

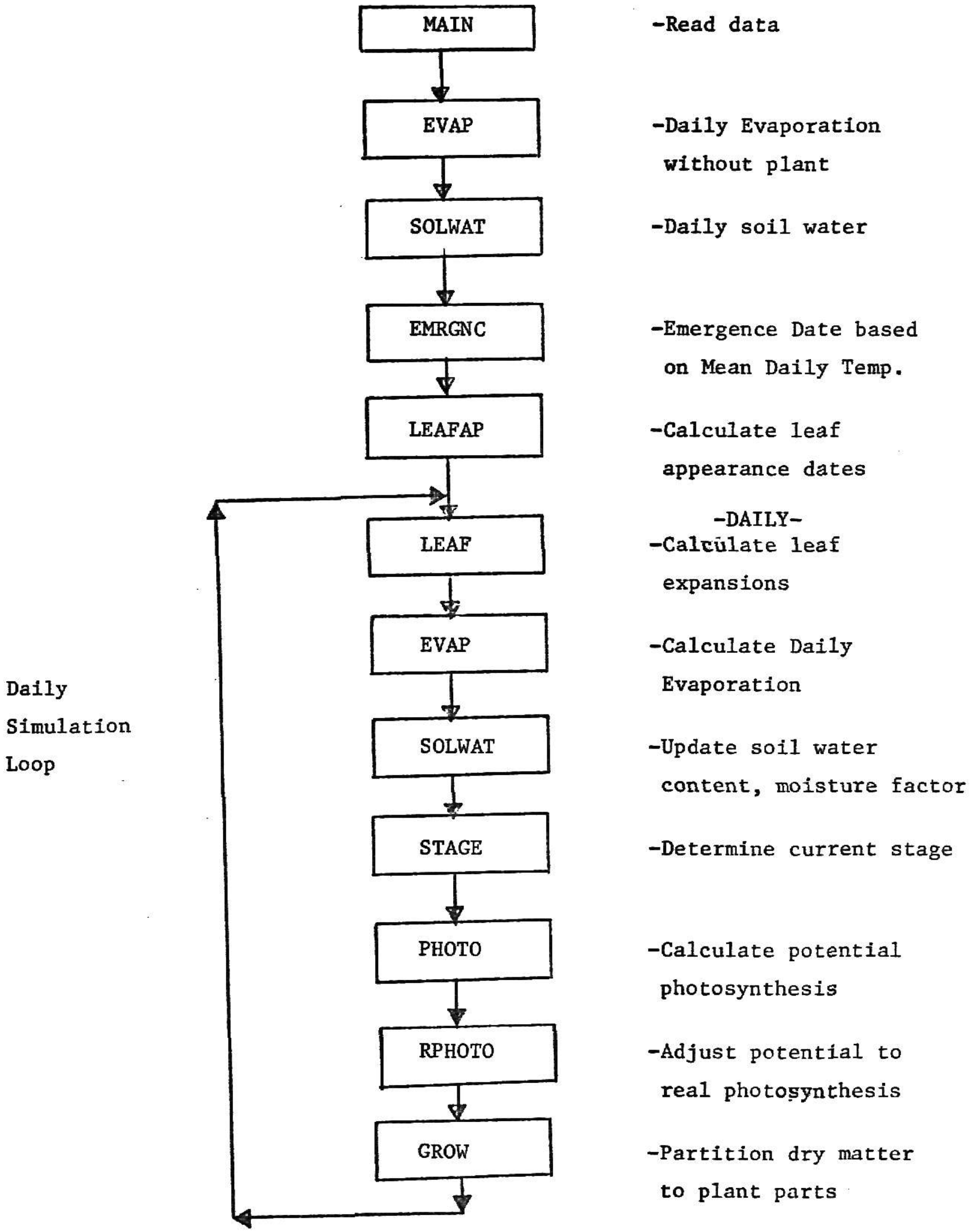

FIGURE 1.

Single Plant Model's Hierarchy Diagram*

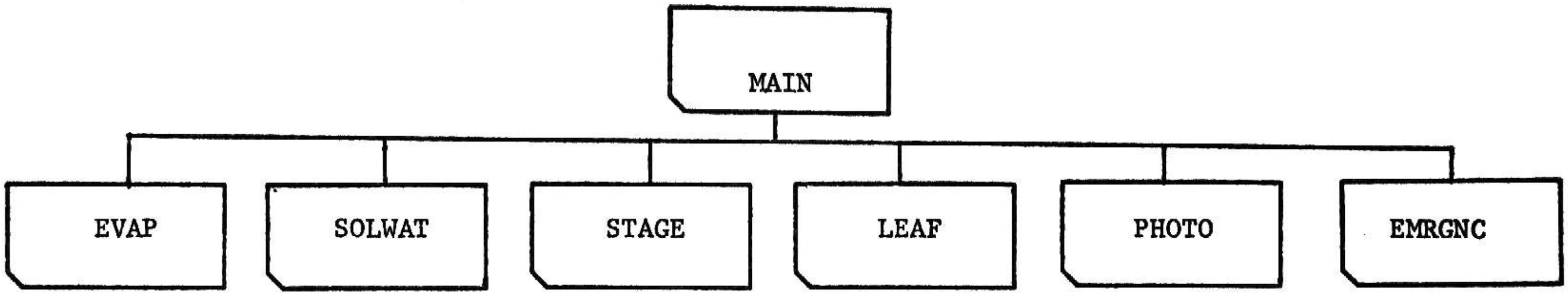

FIGURE 2.

subroutines will be used again when growth simulation begins. After all the climatic data has been read, the emergence module (EMRGNC) is called to calculate emergence date of the single plant based upon mean daily temperature.

Each sorghum hybrid has a maximum number of leaves which will develop and a maximum leaf area for each leaf. The model is heavily based upon the amount of light intercepted by the plant. Therefore, the leaf area per plant must be known. Field and phytotron studies have shown that the rate leaves appear on grain sorghum is directly related to mean daily temperature. The next step is to call the leaf appearance module to calculate the date each leaf appears (subroutine LEAFAP) based on mean daily temperature.

These first three stages are completed in preparation for the actual daily growth simulation. A large loop is established beginning on the date of emergence and ending either at plant maturity or the last date climatic data was supplied, whichever comes first. Within this loop there are seven stages or modules (1).

The first stage calls the module LEAF which calculates the expansion of each leaf and adds this amount to the area of that particular leaf. This figure will then be used to

---

(1) This loop with some modifications will later be referred to as the Daily Simulation Control Routine or DSCR.

calculate the amount of light intercepted by the plant. Leaf senescence (death due to aging) is accounted for by subtracting the maximum leaf area of the first leaf from the total leaf area at the time the twelfth leaf has reached its maximum area. This process then continues for each successive leaf which appears after the twelfth leaf. Once all the leaves have appeared the modeling of senescence continues by subtracting 0.4 percent of the total leaf area from the total area each day until maturity.

In the second and third stages, the EVAP and SOLWAT routines are called again to give the daily water evaporation and to update the soil water content, respectively. During initialization these routines were called to establish the soil condition for each day under the assumption that no plant was present. This data is now used as input on a daily basis and new figures are calculated based upon the presence of the sorghum plant. Although this seems to be a fine point on the differences between the initialization calls to EVAP and SOLWAT and the simulation calls, it is an important point. This is especially true for SOLWAT which takes the evaporation figures from EVAP and separates them into soil evaporation and plant evaporation both of which are taken from the soil content figure. The division is important from two standpoints; first in the initial calls no plant evaporation occurs, thus affecting the soil content in a different manner, and secondly in the simulation calls the moisture limiting factor for photosynthesis which is derived in SOLWAT is a major factor on plant growth.

The fourth stage is for the simulation model to determine the stage of plant development. There are three important stages in the development of grain sorghum and the model must identify the current stage in order to apply certain important growth factors. The subroutine STAGE does this based upon the date that certain leaves reach their maximum leaf area which is an output from the LEAF routine. STAGE sets a common variable representing the plants development stage which is then referenced by the final three stages.

Stage five of the plant simulation is the most sophisticated and technically based portion of the model. In this stage the PHOTO subroutine is called to calculate the potential photosynthesis of the plant under each day's conditions. Since light interception is the most critical element in the model, the shading of leaves both from the plant itself and from neighboring plants becomes a critical factor. Complicating the problem is the fact that shading within the plant's canopy is dynamic, that is, it changes with the sun's altitude and azimuth and with plant's size. To account for these dynamic interactions the PHOTO routine applies a mathematical model developed by Arkin and Ritchie <2> to simulate the light interception in the grain sorghum plant canopy.

Using this model, PHOTO estimates the net potential photosynthesis in terms of the amount of carbon dioxide fixed during daylight hours under certain common conditions. The calculations are made and summed for each hour of the daylight portion of the day. The result is a daily figure representing potential photosysthesis.

In stage six the potential photosynthesis is passed to the RPHOTO routine where a series of efficiency functions are applied to the data. These functions modify the potential photosynthesis to reflect a more realistic value based upon non-optimum temperatures and soil water conditions. There is an efficiency parameter for each environmental constraint on the photosynthetic rate. Examples of these would be night-time respiration loss, temperature conditions and soil water conditions. As an example of their effect, all photosynthesis stops below the temperature of minus five degrees C; thus all potential photosynthesis calculated below that temperature would be subtracted to give a zero rate for each hour the temperature was below minus five degrees C. Similarly, reduction in photosynthesis due to limiting soil moisture has been shown to be proportionate to the reduction in plant evaporation resulting from the limited water availability <7>. There is a threshold of extractable soil moisture below which plant evaporation is affected. This threshold is dependent upon the particular soil and enters the model as a single soil water holding capacity variable. The extractable soil water is determined daily by the model using a modified soil water balance routine developed by Ritchie <6>. Once the water available has been reduced to 25% of the soil holding capability an effect upon net photosynthesis is seen.

The seventh and final stage of the Single Plant Model is the GROW routine. It is within this routine that all dry matter is partitioned between the various plant parts (root, leaf,culm, head, and grain). The basis of the distribution is the stage of development which is an output from the STAGE routine. The plant's growth is broken into three major phases and each phase gives a different dry matter distribution. Each part's weight is kept on an acculmulative basis from day to day until maturity.

After the GROW routine has calculated the day's growth factors and they have been added to the plant's development, the model loops back to stage one and begins simulating the next day's growth. Final maturity is based upon 1.6 times the number of days to half-bloom (anthesis) which, in turn, is based upon leaf development.

In summary, the model goes through seven stages each day with each stage applying critical factors progressively until the final stage where the day's growth estimates are partitioned into the various plant parts. The plant's maturity date is calculated as a function of leaf development. These steps are summarized in Figures 1 and 2.

III. OBJECTIVES AND APPROACHES

The following are the four main objectives of this project.

1. To add internal documentation to the main routines of the Single Plant Model.
2. To modularize the model into logical units in preparation for later updates and replacements.
3. To develop an acceptable approach to extending the Single Plant Model to a field population model.
4. To prove the validity of the approach by implementing divergence due to variations in emergence dates.

The first two objectives were to represent only about five percent of the project effort. The development of the field model was expected to account for approximately 40% of the project effort, while the remaining 55% was dedicated to converting the Single Plant Model into a field model in accordance with objective four.

A limited literature search was undertaken to study some of the more widely used plant simulation models. Included in the study was a cotton model developed at

Missippippi State <5>, a vegetative plant growth model <8>, a sugar beet model <4>, and a distribution model of organism Development times <3>. Although several of the models studied presented alternate approaches, the result of the study was that no new approach would be undertaken and the project would be oriented toward the Single Plant Model because of the large amount of local time and effort already committed to the Single Plant Model and the successes seen in its development.

Because of the already large size of the program that simulates the Single Plant Model (86K-bytes), the problem was to develop an approach which would not greatly increase the model's core requirements or run time.

The first serious approach considered was to partition the sorghum field into a variable number of segments, each of which would represent a specified combination of field variables. For example, a field consisting of two basic soil types, 60% of one and 40% of another would be split into two partitions representing 60% and 40% of the field, respectively. These partitions would then be sub-partitioned based upon a second variable, for example, there might be three separate emergence dates representing 30% emergence on day one, 40% on day two, and 30% on day three (2). This results in six partitions representing 18%,

---

(2) That is 30% of all the plants which will emerge do so on the same day the first plant emerges while 40% emerge on the next day and all remaining plants emerge on day three.

12%, 24%, 16%, 18%, and 12% of the field. The process would continue for each field variable appliciable to the model. The Single Plant Model would then be called to simulate each partition and a total field production would be estimated by weighting each partition based upon the percent of the entire field it represented.

There were several major difficulties with this approach. The number of simulations, and thus the run time required, appeared to grow exponentially with the number of field variables added to the model. Secondly, several of the crucial field variables do not become critical until later into the simulation itself. Since these variables could not be ignored, a modification was considered which would allow the model to dynamically add additional partitions during the simulation of any partition. These partitions would then be added to the list of partitions to be simulated. Although, the new approach cut down the initial number of partitions required, it allowed each partition to create new partitions which in turn could create still more partitions, etc. The approach was theorically desirable but since there appeared to be no limit to the number of partitions which could be created, it was not considered realistically applicable.

An alternative solution was to apply the same theoretical approach, but in a more manageable structure. The idea was to build a dynamic tree structure which would allow the concurrent simulation of the various partitions on a daily basis. The tree would grow dynamically each day with each leaf (3) representing a field partition. When a new field variable becomes critical at one of the leaves then that leaf would diverge into one or more additional leaves in preparation for the next day's simulation.

At first, a tree search routine was considered for limiting the number of nodes. With this approach partitioning or branching would be unlimited. At simulation time each branch would be searched to see if its simulation would be critical based upon policies which would have to be written into the program. Figure 4. demonstrates how this approach would have worked. In the beginning each branch is considered and only those nodes considered critical would be simulated (nodes 1,2,4, and 6) . On the next day only those branches and their potential diverging (sons) branches would be considered. At day two nodes 1, 2, 7, 8, and 6 are simulated. This procedure would continue until all paths reach maturity.

Theoretically this approach also seemed acceptable because there are several known algorithms for tree searching and dynamic divergence would be allowed. However, some major difficulties arose. First allowing the creation of branches which may never be used increases storage requirements unnecessarily. Even if such unused branches were ignored and never stored in memory there would be no

---

(3) The word 'leaf' or 'leaves' here represents leaves in the tree structure not leaves on the Sorghum plant.

limitations on the number of branch nodes which would have to be kept. The second major problem was that tree searching is a very time consuming process. It was felt that as much time would be used searching as simulating and if possible this should be eliminated. Therefore a modification was considered which would limit the number of partitions the tree structure would be allowed to diverge into by requiring that any new partition represent five percent or more of the entire field. Once a leaf matured its daily simulation would be discontinued, but the remaining leaves would be simulated on a daily basis until all leaves reached maturity.

This approach seemed to overcome all of the major objections to the previous alternatives considered. The maximum number of leaves which would have to be simulated on any given day would be twenty (4). This limited run time to an acceptable level. Secondly, the critical variables would be able to enter the system when they became critical without requiring any form of catch-up simulation time. However, if a variable which became critical represented less than 5% of the field, then it would not be considered crucial to the model as a whole and divergence would not be allowed. The only remaining concern was the amount of core that would be required to simultaneously simulate twenty plants. Since this approach had generated a considerable

---

(4) There are at most twenty nodes representing 5% or more of a field. However, as will be seen later a twenty-first node is used for data accumulation after maturity.

amount of excitement, this concern was ruled secondary to the potential of the proposed data structure. It was hoped that the tree structure would lend itself to better memory management and that the core requirements could be reduced in the Single Plant Model itself.

Although the tree structure approach had several desirable characteristics, one additional modification was made to take advantage of the fact that with grain sorghum, it is quite common for a field variable which has become critical and has caused a slow-down in the plant's growth, to be eliminated before it causes too much damage. In this case, the plant could 'catch up' to its normal growth level. For example, a lack of available soil water in one area of the field might cause a slow-down in the growth rate in that partition (simulated by a divergence at that point). However, if rain comes, the plants in the partition may speed up growth until they reach the growth level of the rest of the field. To allow for this possibility the proposed data structure was modified from a tree structure with only divergence to a directed graph structure with divergence and convergence.

Convergence in the model is a unique concept that is expected to increase the accuracy of the model. Once a node matures it will be possible to merge the associated data and to free the node space for other uses. This concept of convergence add tremendous potential to the model from the stand point of the Agronomist, and it is perhaps as unique as the proposed data structure itself.

The final conclusion was that a graph structure which allowed dynamic divergence and convergence of nodes would best serve the objectives of the project.

IV. THE PROPOSED DATA STRUCTURE

The core of the Field Level Model is its data structure. Figure 3 illustrates the gradual development of the model's digraph structure. The Source Node represents the field prior to entering the simulation model. The Sink Node at the bottom of the graph is a special node, where all simulation nodes are merged at maturity. It is represented in the program as the twenty-first element of each of the dimensioned variables in the NODBLK COMMON block. Node 1 represents the field as initialized from the reading of global and field variables; that is, one full set of data is read in and stored in the first of the twenty available nodes. At this point the weighting factor for the node is 100%. Immediately following the input of both the global and field data, the emergence routine is called to calculate the days and percent per day that emergence occurs. The Emergence Routine in turn calls the CREATE routine for each additional day emergence occurs. Each call to the CREATE routine represents a divergence of the node presently being processed.

For the example shown in Figure 3, four emergence days are shown (June 1,3,4, and 5). Therefore, CREATE must be called three times. The weighting factor for each node is calculated by the Emergence Routine; for our example, 15%, 35%, 35%, and 15% respectively. (Note that node one simply has had its emergence date established and its weighting factor changed.) After the necessary divergence has occurred

The Digraph Structure

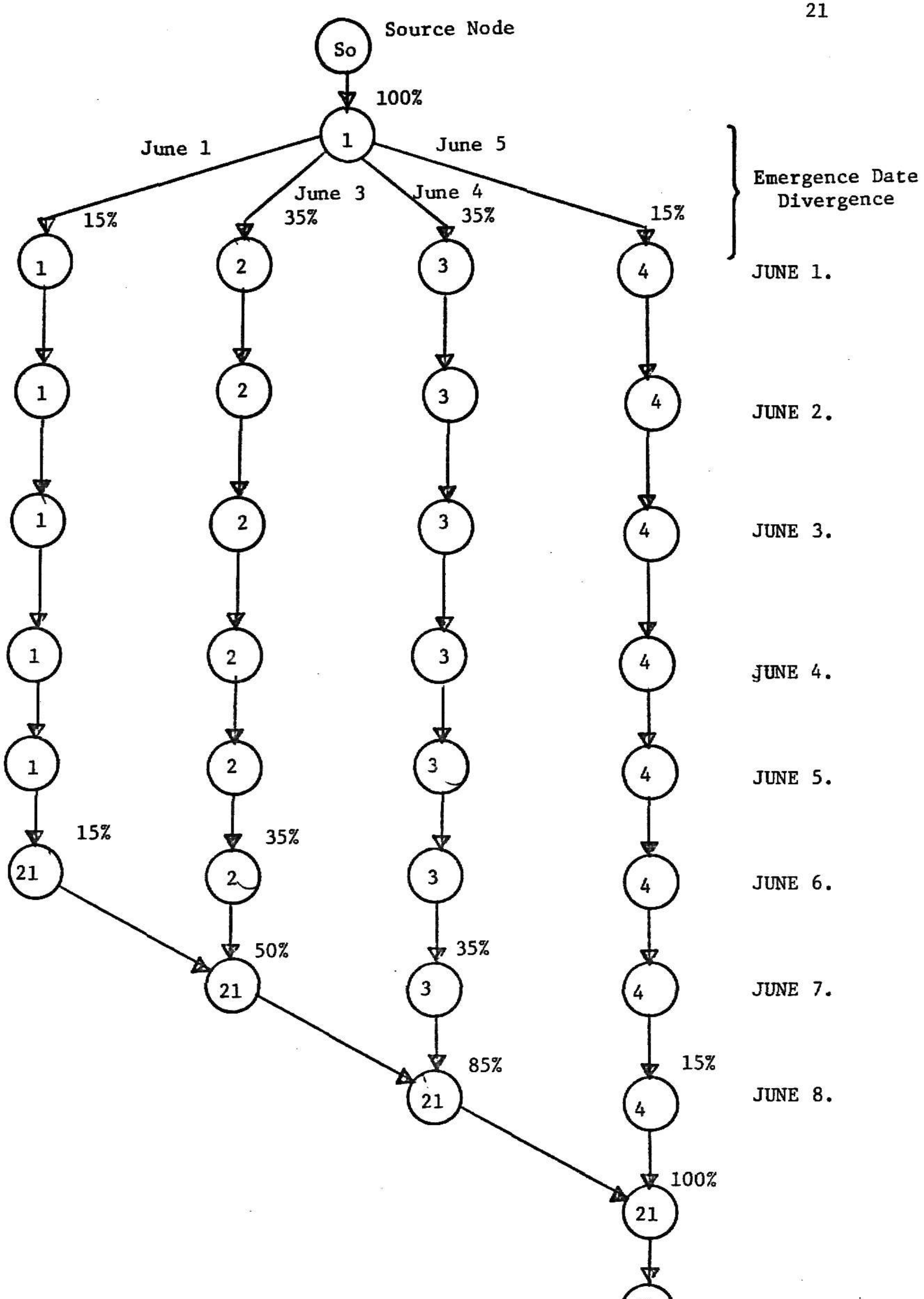

FIGURE 3.