T-H-A-T-S

-TIMBER
-HARVESTING
-AND
-TRANSPORT
-SIMULATOR

WITH SUBROUTINES FOR APPALACHIAN LOGGING

BY A. JEFF MARTIN

USDA FOREST SERVICE RESEARCH PAPER NE-316
1975

NORTHEASTERN FOREST EXPERIMENT STATION
FOREST SERVICE, U.S. DEPARTMENT OF AGRICULTURE
6816 MARKET STREET, UPPER DARBY, PA. 19082
F. BRYAN CLARK, STATION DIRECTOR

THE AUTHOR

A. JEFF MARTIN attended Michigan State University, receiving a degree in forestry in 1965, an M.S. degree in forest management in 1966, and a Ph.D. degree in forest management in 1969. He joined the Staff of the Northeastern Forest Experiment Station at the Forest Products Marketing Laboratory, Princeton, West Virginia, in 1969. He is conducting research in the marketing of primary wood products, the application of system-analysis procedures to the transportation of Appalachian roundwood products, and logging residue.

MANUSCRIPT RECEIVED FOR PUBLICATION 13 MAY 1974

ABSTRACT

A computer program for simulating harvesting operations is presented. Written in FORTRAN IV, the program contains subroutines that were developed for Appalachian logging conditions. However, with appropriate modifications, the simulator would be applicable for most logging operations and locations. The details of model development and its methodology are presented, along with generalized flow charts. Each component of the simulator (felling, skidding, etc.) is discussed individually. Input requirements and program output are presented in detail with the aid of information for an example logging operation.

T-H-A-T-S

-TIMBER
-HARVESTING
-AND
-TRANSPORT
-SIMULATOR

WITH SUBROUTINES FOR APPALACHIAN LOGGING

BY A. JEFF MARTIN

USDA FOREST SERVICE RESEARCH PAPER NE-316
1975

NORTHEASTERN FOREST EXPERIMENT STATION
FOREST SERVICE, U.S. DEPARTMENT OF AGRICULTURE
6816 MARKET STREET, UPPER DARBY, PA. 19082
F. BRYAN CLARK, STATION DIRECTOR

CONTEMPORARY TIMBER harvesting is a complex enterprise requiring many managerial decisions. In addition to the age-old concerns for profit margin, timber supply, and labor, the manager today faces new challenges on an increasingly broadened front. Machinery costs are higher than ever, and a proliferation of designs makes purchasing decisions more and more difficult. Greater attention to the environmental aspects of logging demands sharpened managerial skills.

Consequently, logging operators should use all the modern decision-making tools that they can. One of these, computer simulation, is a low-cost means of assessing the outcomes of decisions before they are actually carried out. In this report, a computer program for simulating the harvesting process is described in detail, along with instructions for its use. We call this program T-H-A-T-S, for Timber-Harvesting-And-Transport-Simulation.

Computer usage is no longer a luxury enjoyed by only the big companies. Today the computer is fast becoming a necessity; and even to the smaller operators, it is no longer a financial burden. With the advent of remote terminals, linked to large computers, access is available to virtually anyone.

Of course, simulation should not be thought of as a panacea; it is only as good as the quality of data used with it. Simulation is not a replacement for sound judgment and experience; rather, it is a managerial supplement for today's complex decision-making.

USES OF T-H-A-T-S

In its present form, T-H-A-T-S is best suited for investigations related to existing logging operations. We have quantified some of the many relationships contained within a logging system, but the job is not complete. Also, we have used data exclusively from the Appalachian Region.

Input for T-H-A-T-S includes time and motion data appropriate to the system being studied. For those functions in which operating or delay time has not been related quantitatively to system or stand parameters, the input consists of expected values (and their standard deviations). Therefore, to evaluate proposed systems, or to estimate costs and production for proposed sales, considerable care must be exercised. Data obtained from other sales could be used initially; however, these should be verified as soon as possible by observing the operation under the new conditions. Updated information should then be used in the simulation studies.

The basic simulator should be applicable for virtually any logging system. However, for areas outside the Appalachian Region, certain modifications would undoubtedly be needed. Most of these changes would involve only the subroutines, because the main program handles the logic of harvesting, utilizing data from the subroutines. Consequently, even such exotic systems as balloon logging can be analyzed by T-H-A-T-S if appropriate data and relationships are available.

As the adequacy of data improves, we will be better equipped for determining the relationships between system variables. This will greatly improve the simulator's capability for prediction and will permit better analyses of proposed systems and contemplated timber sales.

The uses of T-H-A-T-S are about as varied as the investigator's imagination. Some of the more obvious uses are listed below (*Martin 1973*):

Cost comparisons.—The existing system's economic picture can be compared to what would result if more men or machines were added, or if certain parts were modified or even eliminated. Many possibilities can be examined at a fraction of the cost involved with field-testing.

Production comparisons.—These can be made both with the system and its possible modifications, and between the system components.

Time analysis.—How is the operating and delay time distributed? How are operating times distributed within a cycle? What is the variability of the time factors?

Trouble-shooting.—Locating the limiting factors in the system. Why doesn't actual production keep pace with potential?

Predicting sale parameters.—Daily production and costs; total production and costs; time required for logging the sale; when various machines can be expected to be available for other areas, etc.

STRUCTURE OF T-H-A-T-S

T-H-A-T-S consists of a main program and 23 subroutines. The main program is composed of eight components (felling, bunching, skidding, bucking, loading, hauling, roadbuilding, and cost accounting) and a "clock." The clock records the passage of time for "real-world" similarity (fig. 1).

Each component utilizes one or more subroutines, and some subroutines are used by more than one component. The subroutines and where they are used will be explained in greater detail.

METHODOLOGY OF T-H-A-T-S

There are two broad classes of simulation models: those that are continuous and those that are discrete. Briefly, the continuous model updates the clock by some fixed time increment, and then proceeds through each step of the system before advancing the clock again. The discrete model first examines all components of the system and then advances the clock to the time when the next event will occur.

T-H-A-T-S is actually a meld of both concepts. It is continuous in the sense that time is incremented minute by minute; and it is discrete in the sense that only a few steps of the system are encountered during each increment. This is accomplished by a series of status checks that cause the system to bypass components that are performing a task. Only when the task has been completed will the system return to the component so that values (time, volume, etc.) for the next task can be generated. This approach causes the computer program to leapfrog through the system and hence greatly reduces the computer time and charges (see appendix C).

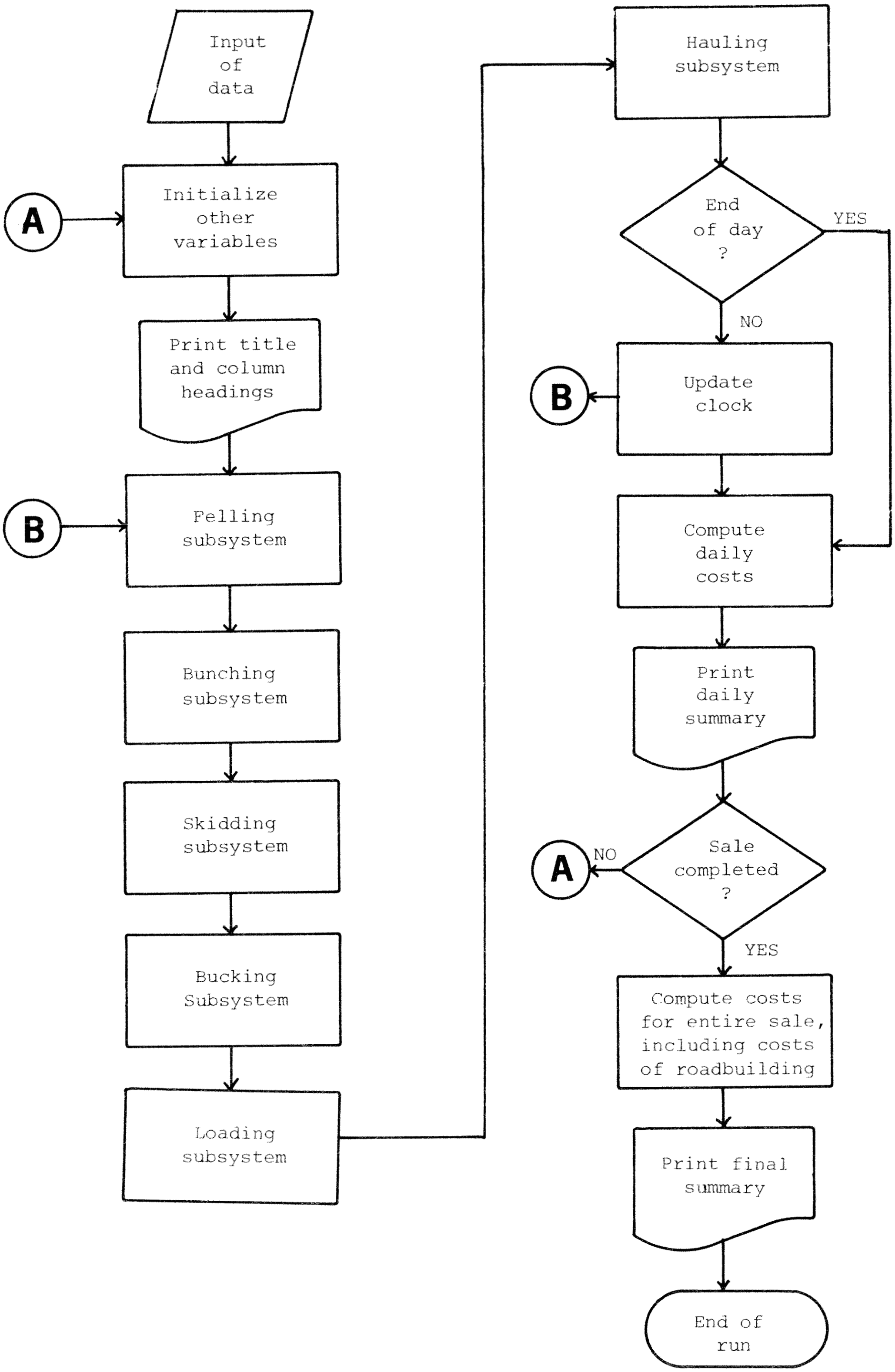
Control of the leapfrogging is maintained by a set of surplus-time variables, one for each system component. For example, if 6 minutes have been generated as the time required for a skidder to travel from landing to the felling area, then the surplus time for this skidder would be set initially at 6 minutes. Then, each time the clock is incremented by 1 minute, the surplus time would be decremented by 1 minute. When surplus time equals zero, it will be time to start another task—in this case, hooking up the logs.

The simulated time for performing each task is generated in one of two ways. Either the time is based on an average value obtained from the operation being studied and is part of the input data; or it is estimated by means of an equation obtained from regression analysis of data collected on several operations. Most of the subroutines are designed so that either option may be used. However, the equations were developed from data collected in the Appalachian Region, hence their use outside this area is not recommended unless they are checked for local accuracy.

The average or predicted time is then combined with the appropriate standard deviation or standard error to generate a random variable (the simulated time), using either a normal or lognormal generator (see appendix B). The simulated time for each task within each system component is generated separately and independently for both the operating and delay time. The total cycle time for a component is the sum of all individual time elements. For example, suppose the following times were generated for a felling cycle:

Task	*Operating time (min.)*	*Delay time (min.)*
Walking between trees	2	2
Felling	3	0
Limbing and topping	2	1
Bucking	0	0
Total	7	3

Figure 1.—Flow chart showing the components and major operations of T-H-A-T-S.



Total cycle time in this instance is 10 minutes.

The generation of operating time for a particular task requires only an average value (input or predicted by an equation) and its standard error; generating delay times requires the frequency of occurrence as well, with frequency defined as follows:

$$f_{ij} = E\ (N_{ij})/100$$

where f_{ij} = Frequency of delay for the ith task in component j.

$E\ (N_{ij})$ = Expected number of delays per 100 cycles for the ith task in component j.

Delay frequencies, averages, and standard deviations must be obtained from the operation being studied or, if care is used, from past operations.

In addition to generated delay times, there are various points in the system at which the interaction of two or more components may give rise to additional delays. For example, delay results when one truck must wait at the landing until another truck finishes loading. Or both trucks and the loader may have to wait until a skidder brings more logs (or trees) to the landing.

All the generated operating times are in integer minutes equal to or greater than 1. All delay times generated by T-H-A-T-S are integer values equal to or greater than zero.

When a cycle has been completed, all variables pertaining to that cycle are reset at zero. However, before this, all the values are added to cumulative arrays so that daily costs, times, and production can be summarized.

There are several queuing points within the system to accumulate logs or trees until they are moved to another point. There are queues for logs (or trees) waiting to be bunched, waiting to be skidded, waiting to be bucked (if trees are bucked at the landing), and waiting to be loaded. Another queue keeps track of trucks waiting at the landing and "releases" them for loading on a first-in-first-out basis.

T-H-A-T-S also provides a half-hour lunch break during each day. Of course, this can be deleted or changed to suit the program user (see statements 910 + 8, 919, and 919 + 1 in the program listing). Copies of the program listing, input instructions, sample output, etc. are available on request from the author.

The system shuts down at the end of each day in a staggered manner. In other words, each machine, man, or crew finishes the day near the specified quitting time; but it may be sooner for some and later for others. To better understand this procedure, let's define A = quitting time—10 minutes. The program calls for all components (except hauling) to finish a cycle if it was started before A: but not to begin another cycle after A. For example, if a felling crew starts to fell a tree at 4:45 p.m. and the cycle time is 10 minutes, the crew will quit at 4:55 p.m. (if the nominal quitting time is 5:00 p.m.).

The quitting rule for hauling is a bit more complex because an algorithm is used to simulate the driver's decision-making as to whether he thinks another trip can be completed in the time remaining. The estimated time for the next trip is assumed to be equal to the time used for the last trip less one-half of its delay time. If this estimated time is less than the time remaining in the day (which is based on the specified quitting time plus 1 hour), the driver will return to the landing. Of course, these assumptions may be changed by the user (statements 845 + 2 and 850 + 1). The program also makes provision for a truck waiting overnight at the landing, either empty or partially loaded. This would occur if the loader should quit before a truck arrives or if the log supply should run out before a truck is loaded at quitting time.

There are also checks for each system component to determine when they have completed their work on the current timber sale. When the fellers have cut all of the available volume (number of acres times volume per acre), they are finished. When the skidders have moved this volume to the landing, they are finished; and so forth for all components. It is possible that not enough volume remains at the end of a day to warrant having all machines in a component working on the next day. For example, suppose a system has two skidders in operation, but on the day when the felling crews finish cutting, it seems that only one skidder will be needed to finish up on the next day. The program is designed to make and implement these decisions for all of the components.

On the following pages, the components are discussed in some detail, and a flow chart is provided for each. The flow charts depict the workings of the simulator, but do not necessarily conform to the way in which the program is written-notably the bucking, loading, and hauling portions. The program listing shows that these three components are quite interwoven with respect to the statements. The flow charts separate them so that the operation is easier to understand.

FELLING COMPONENT

The felling component comprises four separate tasks: walking between trees, felling the trees, limbing and topping, and bucking (optional). Completing all four tasks constitutes one felling cycle, and generally one tree has been prepared for transporting to the landing. When a cycle is over, the next one is begun (fig. 2). This process continues until it is time to quit for the day, whereupon a daily summary of operating and delay times, costs, and production is prepared for each felling crew.

Certain subroutines are required for each task. Subroutine WALK requires an average value (average walking time between trees) as input. Subroutines FELL and LIMB permit the use of an average value (as input), but also provide equations (based on 517 trees from 7 operations) for estimating an average based on characteristics of the current tree. The equation in subroutine FELL is:

$$AT = -1.21 + 0.157\ DBH + 0.4\ MOP\ (-1.21 + 0.157\ DBH)$$

$$R^2 = .55 \qquad S_{y.x} = .64$$

Where AT = Average time to fell the tree (minutes).

DBH = Diameter of the tree at breast height (inches) from subroutine TREEV.

MOP = Method of payment for the felling crew = 0 if they are paid per thousand, = 1 if they are paid on an hourly basis.

AT and the standard error about the regression (or standard deviation if the value of AT is part of the input data) are then utilized by subroutine LOGNOR to generate the simulated felling time for the current tree. LOGNOR provides a random variable from a lognormal distribution (see appendix B) with the aid of RANDU, a random number generator maintained in the computer library at Virginia Polytechnic Institute and State University, Blacksburg, Virginia, for use with the IBM system/370 computer (IBM Corp., 1968).

The procedure that has been outlined above is essentially the same for all operating times in the harvesting system. Differences between subroutines will generally be in the following:

- The equation(s) will be unavailable for some subroutines.
- The equation(s) themselves will, of course, be different.
- Some tasks are based on the lognormal distribution (subroutine LOGNOR) and others on the normal distribution (subroutine NORMAL).

For subroutine LIMB the equation is:

$$AT = -1.825 + 0.161\ DBH + 0.0295\ HT$$

$$R^2 = .34 \qquad S_{y.x} = 1.25$$

Where:

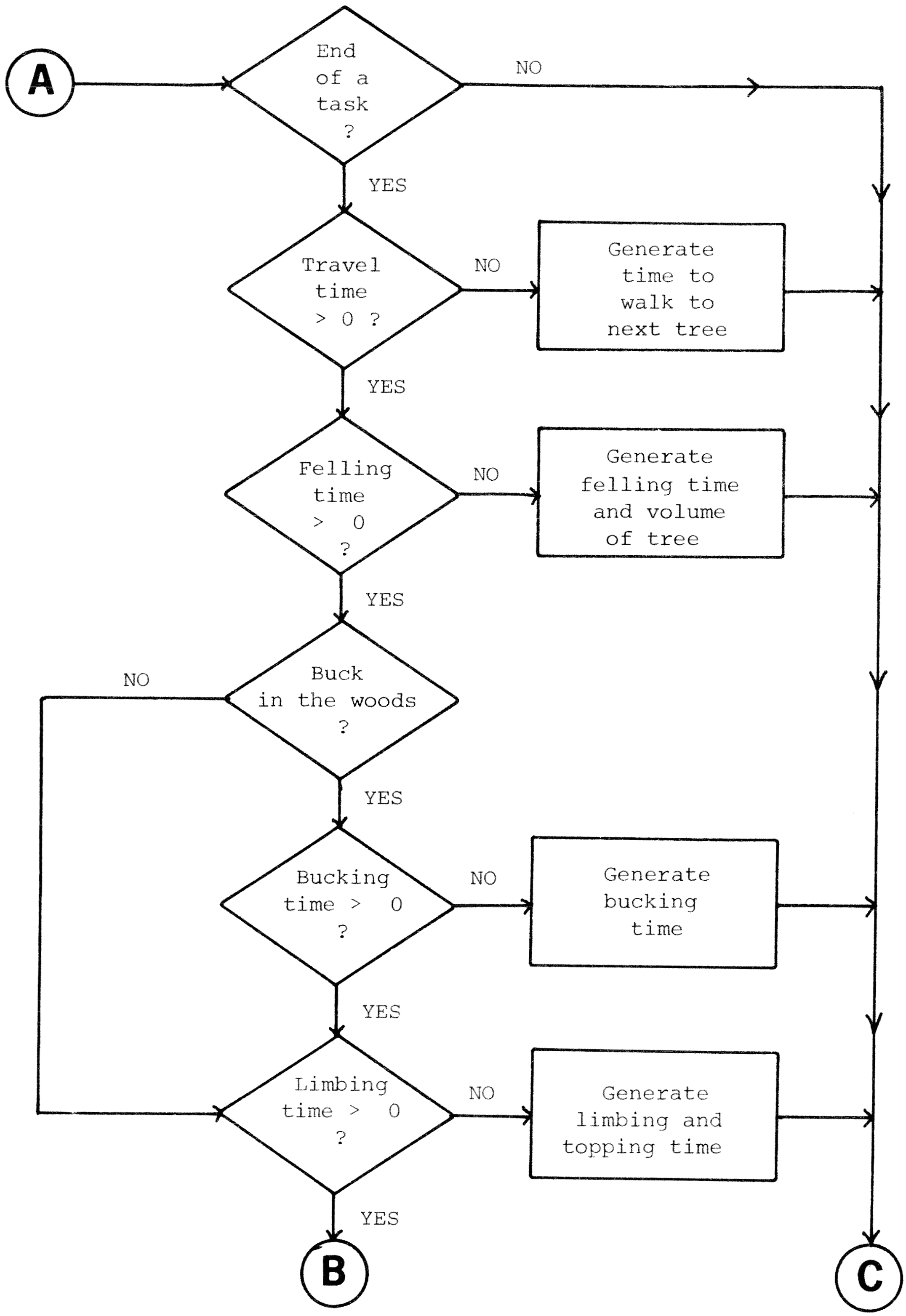
AT = Average time to limb and top the tree (minutes).

HT = Merchantable height of the tree (in feet), from subroutine TREEV.

As noted in table 1, bucking is optional; it may be done in the woods as part of the felling component, or it may be a separate operation performed at the landing (when tree-length skidding is done) (See Input Instructions, card 14.) In either case, the subroutine (BUCK) is the same. The bucking component would not be used at all if tree-length hauling is done.

Subroutine TREEV generates the DBH, merchantable height (HT), and volume (NTV) of the current tree. DBH is based on an empirical distribution for the stand of timber being harvested (see Input Instructions, Card 20). Simulated tree height is based on average merchantable height (and its standard deviation) for the timber stand; both are input variables. When DBH and height have been determined for the current tree, a volume

Figure 2.—Flow chart of felling component.



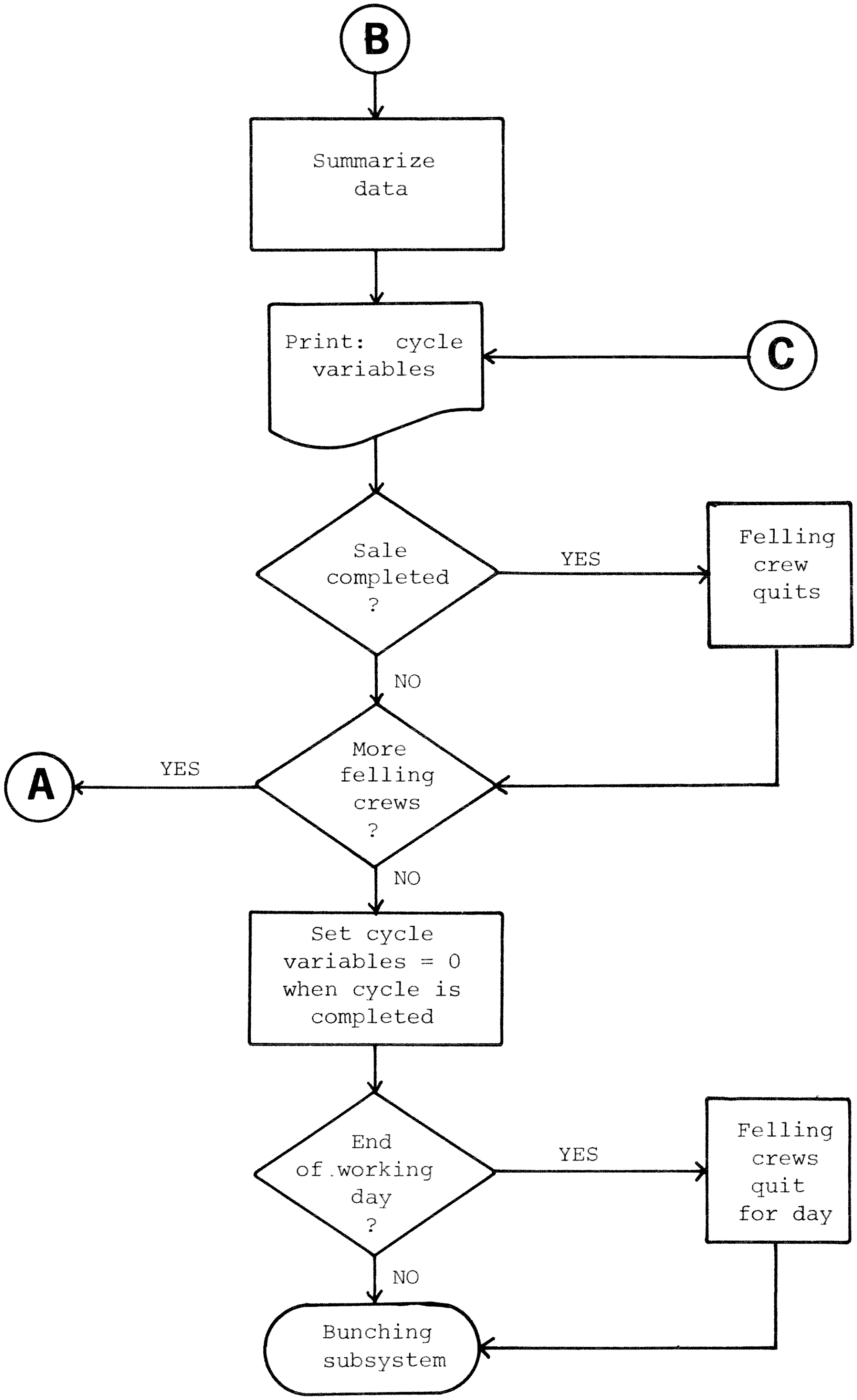

B
Summarize data
Print: cycle variables
C
Sale completed ?
YES
Felling crew quits
NO
More felling crews ?
YES
A
NO
Set cycle variables = 0 when cycle is completed
End of working day ?
YES
Felling crews quit for day
NO
Bunching subsystem

Table 1.—Expected time equations for subroutine empty

Road type	Equation	Standard error of estimate	R^2
2-lane paved	$TE_1 = 5.26 + 1.41\ DIST_1$	6.2	0.91
1-lane paved	$TE_2 = 1.00 + 2.28\ DIST_2$	2.9	.77
Gravel	$TE_3 = 3.58 + 3.41\ DIST_3$	4.7	.83
Woods road	$TE_4 = 2.22 + 8.40\ DIST_4$	5.1	.83

TE_i = Travel time in minutes.
$DIST_i$ = Hauling distance in miles.

table (also part of the input) is scanned, and the proper board foot volume is obtained.

Delay times for each task in the felling component (and all other tasks in the system) are generated by subroutine DELAY. Inputs to this subroutine are the average delay time for the task, its standard deviation, and its frequency of occurrence. DELAY calls RANDU and receives a random number between 0 and 1. If this number is less than the frequency of occurrence, a delay time will be generated, using subroutine LOGNOR; if it is greater than the frequency, the simulated delay will be set at 0. No waiting delays occur in the felling component because this is the first part of the harvesting system.

When a cycle has been completed, all operating and delay times for each task are totaled for the cycle time. All time elements and the volume are then placed in cumulative arrays for eventual conversion to daily summaries. The cycle variables are then set at zero in preparation for the next felling cycle.

The program is designed so that the felling crews may start working in the sale area a few days before the start of skidding, hauling, etc. To initiate this condition, the user merely enters the proper number of lead days on Card 16 (Input Instructions).

BUNCHING COMPONENT

The bunching component is an optional feature of T-H-A-T-S. Some logging operations employ skidders or tractors to prebunch logs (trees) before skidding them to the landing. When the system being simulated is of this configuration, the bunching component is called into use by proper coding of the input data (See Input Instructions, Cards No. 4, 28, 29, and 56).

The bunching phase has not been subdivided into tasks, hence only one operating time and one delay time are generated during each cycle (fig. 3). Subroutine BUNCH generates the simulated operating time for a bunching cycle, using a normal random process. BUNCH does not contain a predictive equation, hence an average time must be included with the input data.

In addition to generated delay, there is a possibility that waiting delay will arise if the felling crew(s) can't keep pace with the buncher(s).

The volume bunched during any one cycle is generated by subroutine SKIVOL. SKIVOL initially determines the number of trees (or logs) to be bunched in the current cycle (using subroutine NORMAL), and then generates the volume of each, using LOGVOL if the trees are bucked in the woods, or TREEV if they are bucked elsewhere.

The procedures followed by subroutine LOGVOL are similar to those used by TREEV. First a diameter and length are generated for the log. The diameter is based on an empirical frequency distribution of log diameter by length class. This distribution is part of the input deck (see Input Instructions Card 45) and should be obtained from the sale area or from past sale information of similar timber types. The simulated log length is based on a length distribution that is also part of the input. A series of equations are then used to determine log volume, using the generated diameter and length. These are presently set up for the International ¼-inch log rule; however, they may be changed to suit the user. LOGVOL is currently designed for 35 diameter classes from 6 inches to 40 inches, and 5 length classes (8-, 10-, 12-, 14-, and 16-foot logs). However, this can be easily altered by changing three statements (set off by COMMENT cards) in the subroutine. The statements:

MINL = 8
INC = 2

refer to the length parameters; MINL = 8 = length of shortest log, INC = 2 = the increment (in feet) between log sizes. The statement:

MIND = 6

establishes the minimum log diameter (in this case 6 inches) to be considered in the simulation.

When a bunching cycle is completed, the variables are reset at zero; and when the day has ended, a cost, time, and production summary is printed.

When there are two or more felling crews and/or two or more bunchers, the interaction between the two components is a bit sticky. Consequently, a decision rule was programmed into the simulator that may be somewhat arbitrary, but nevertheless is fairly realistic. If the number of bunchers is equal to the number of felling crews, they will be matched up, feller No. 1 with buncher No. 1, feller No. 2 with buncher No. 2, etc. If the numbers are unequal, then the buncher(s) will alternate with the felling crew(s) in a systematic manner throughout the day.

SKIDDING COMPONENT

The skidding component is divided into five tasks: outbound or outturn time (landing to woods), hookup time, skidding or inturn time (woods to landing), unhooking time, and positioning time (decking logs or trees at the landing). Completion of the cycle means that one turn of logs (trees) has been brought from the felling area to the landing (fig. 4).

Subroutine RETURN permits the generation of travel time by using either an average time or an equation. The equation, based on 195 observations from seven operators, is as follows:

$$AT = 0.558 + 0.0022\ ASD - 1.1535\ SLOPE$$
$$R^2 = .77 \qquad S_{y.x} = 1.09$$

Where:

AT = The average outturn time in minutes.
ASD = The skidding distance in feet.
SLOPE = An indicator of the aspect of the skidroad grade. If the trip to the woods is on level ground or downhill, SLOPE = 1, if uphill, SLOPE = 0.

The skidding distance (ASD) varies with the amount of timber removed. To permit this, the simulator must first estimate the maximum volume that will be skidded to an individual landing. The area serviced by a landing is assumed to be one-eighth of the area defined by a circle with radius equal to the maximum expected skidding distance. Of course, this assumption can be easily altered to one-fourth or one-half, etc., by changing the denominator in statement 13 + 6 (this value equals 8.0 in the present version). This area-per-landing (in acres) is simply multiplied by the average volume per acre (VPA) to arrive at the maximum volume skidded to a given landing (MAXVOL).

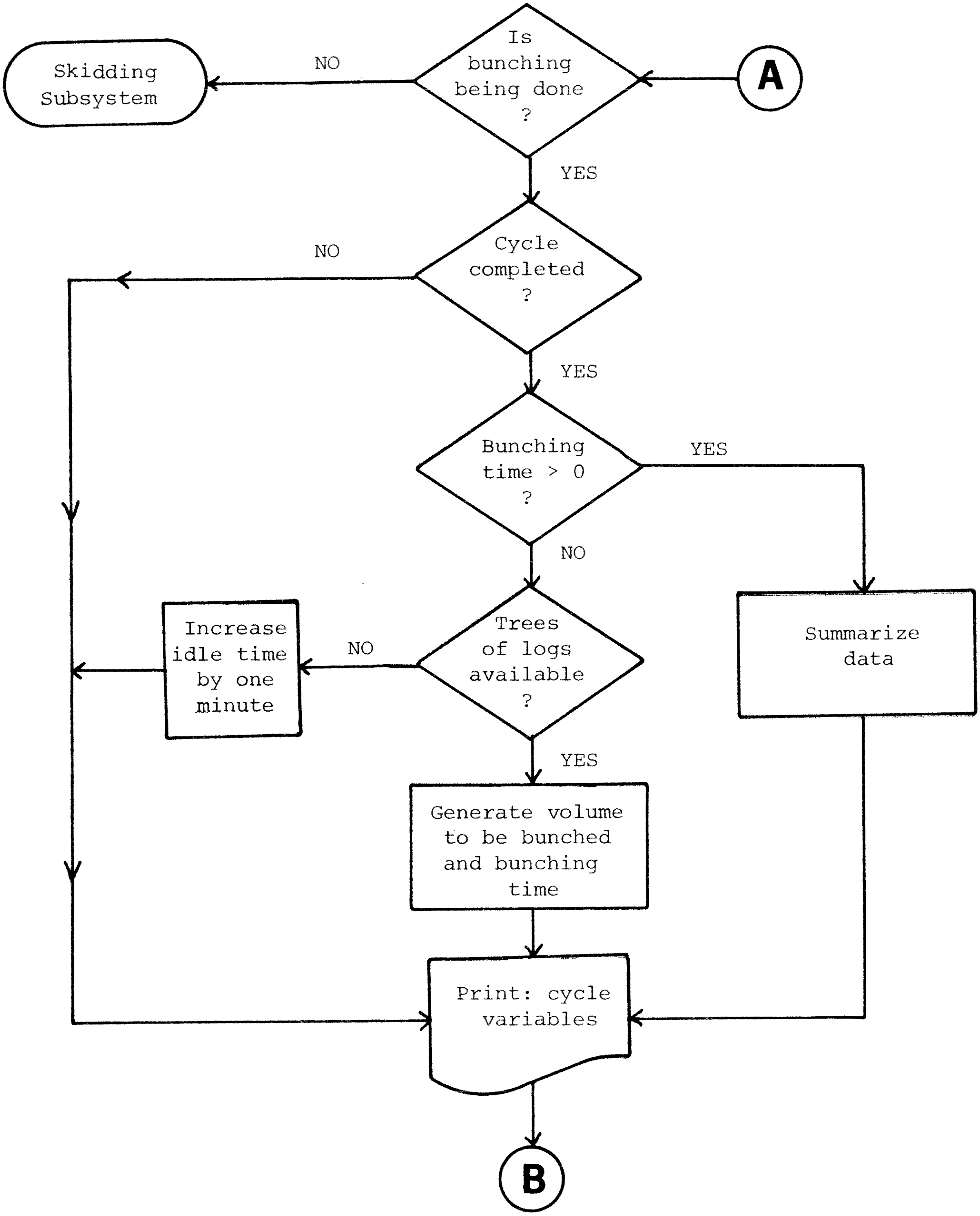
When the accumulated volume (IVMAX) that has been skidded to a landing equals or exceeds MAXVOL, it becomes time to move to another landing. This is explained further in the loading component section; however, it was necessary to introduce IVMAX at this point in the discussion. IVMAX is used in the following equation to determine the skidding distance for each trip:

$$ASD = \sqrt{\frac{(43{,}560)\ (8)\ (IVMAX)}{(\pi)\ (VPA)}}$$

This is simply the equation used for calculating MAXVOL, rewritten in terms of ASD.

Subroutine SKIVOL is used to generate the volume of a turn, and this has been discussed in the section describing the bunching component. Subroutine HOOKUP is used to gener-

Figure 3.—Flow chart of bunching component.



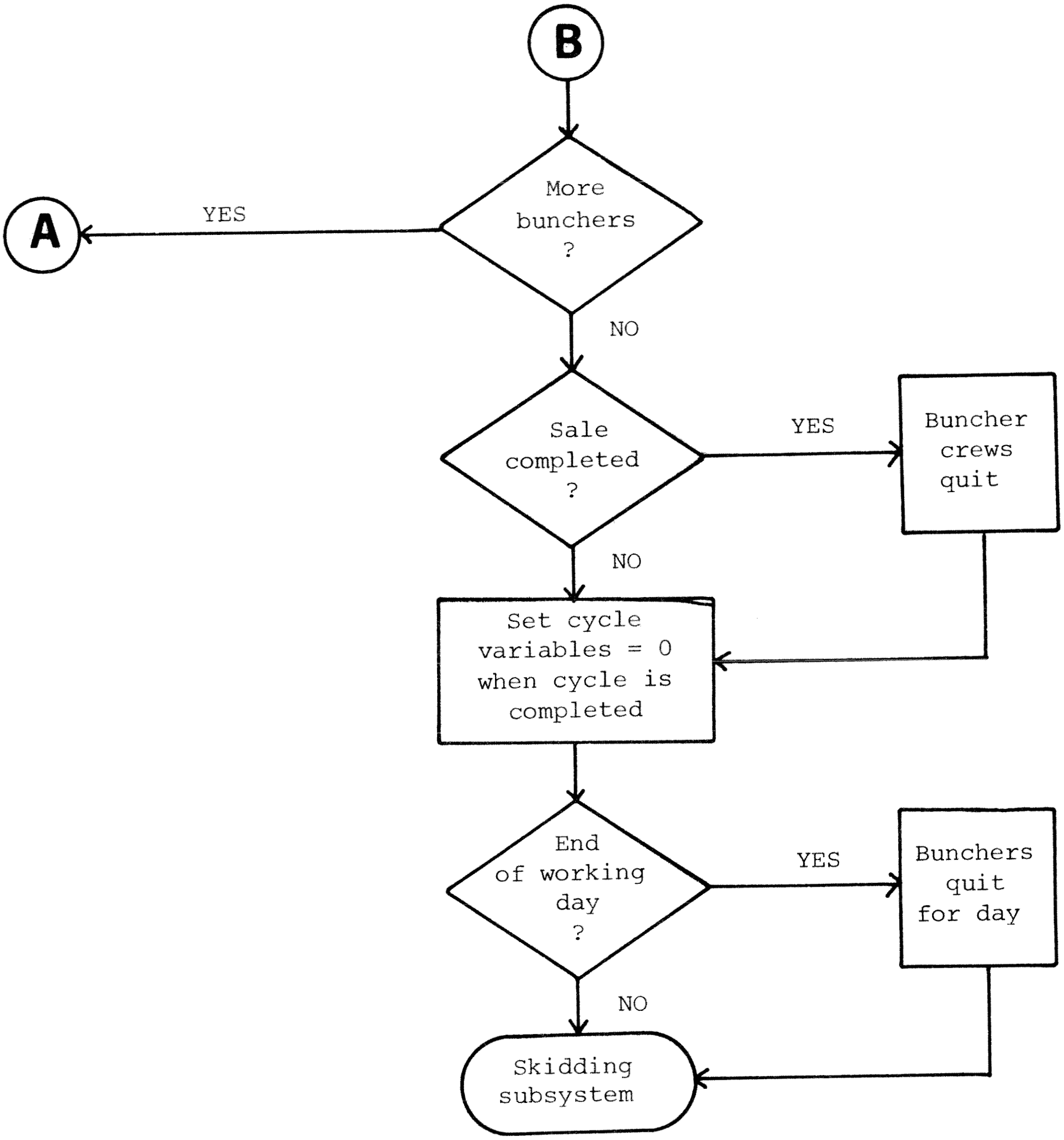
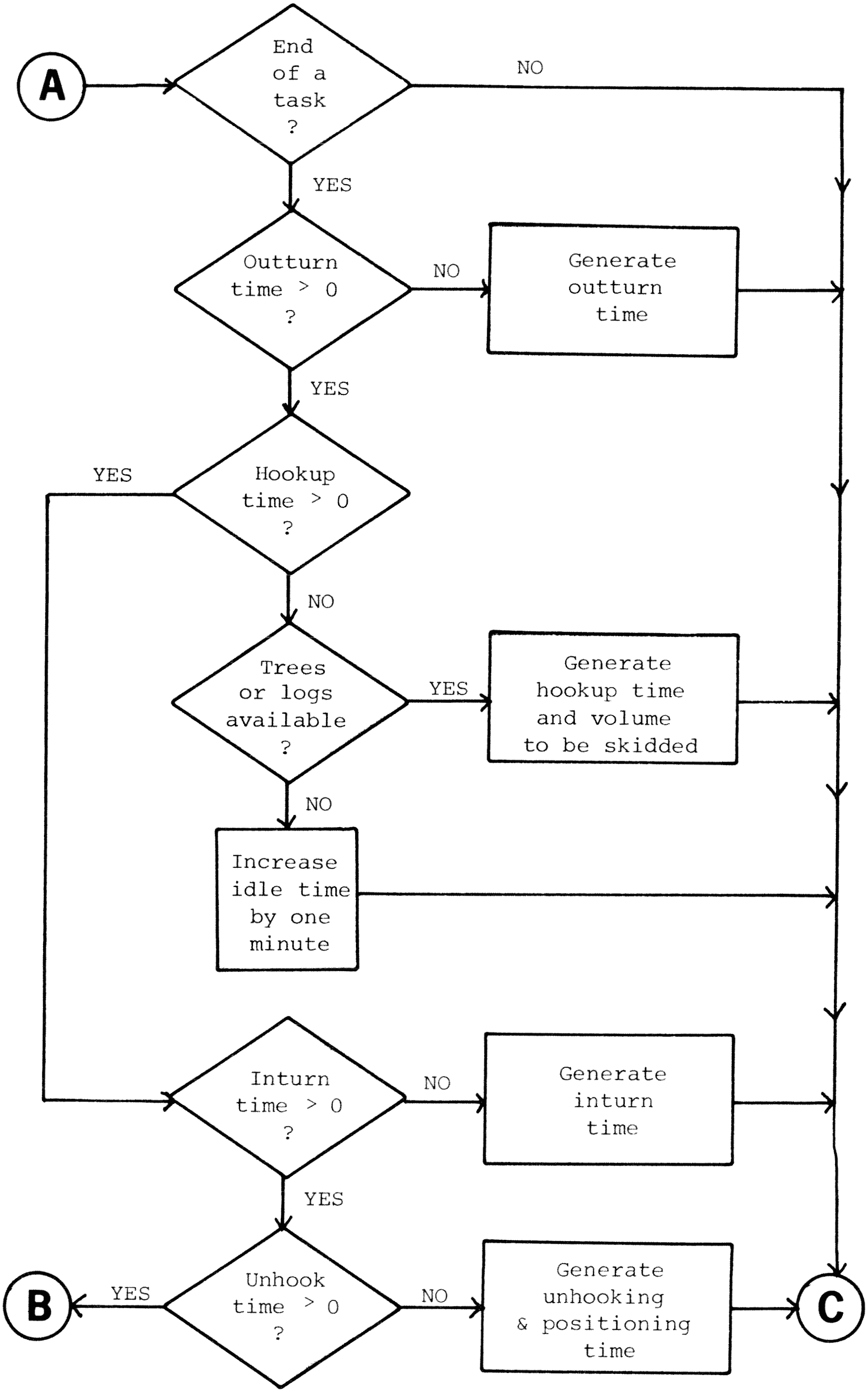
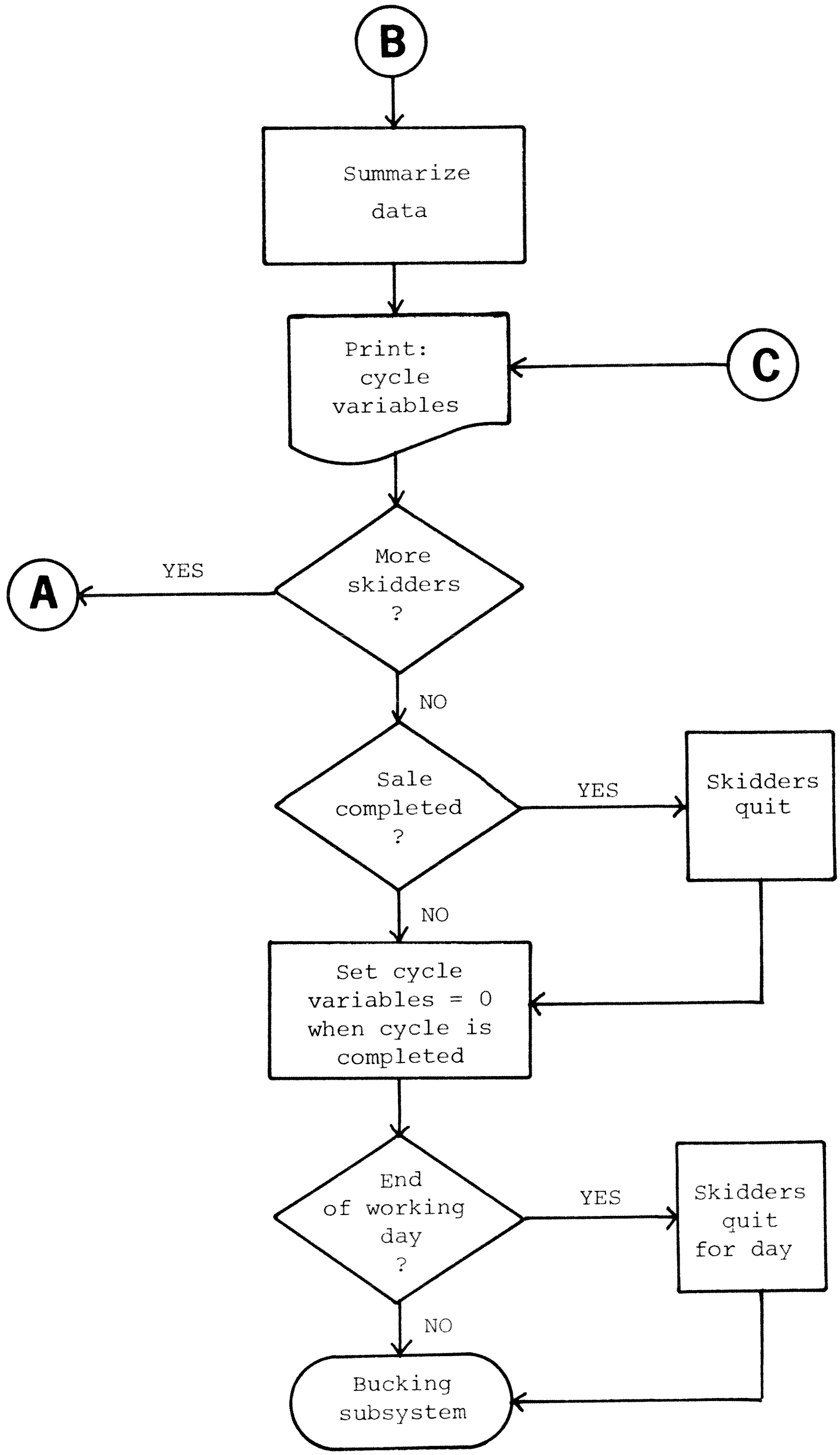

B
More bunchers ?
YES
A
NO
Sale completed ?
YES
Buncher crews quit
NO
Set cycle variables = 0 when cycle is completed
End of working day ?
YES
Bunchers quit for day
NO
Skidding subsystem

Figure 4.—Flow chart of skidding component.




B
Summarize data
Print: cycle variables
C
More skidders ?
YES
A
NO
Sale completed ?
YES
Skidders quit
NO
Set cycle variables = 0 when cycle is completed
End of working day ?
YES
Skidders quit for day
NO
Bucking subsystem

ate the time required for hooking up the turn of logs (trees). There is no equation available; however, the subroutine permits the use of an average fixed time in conjunction with an average variable time (per tree or log) if the user wishes. These options are explained in the instructions for preparing the input data.

Generation of the simulated skidding time (from woods to landing) relies on the use of subroutine INTURN. This subroutine allows the user to employ an average (expected) time (read as input) or to compute this figure by means of the following equation:

$$AT = 4.37665 + 0.00142\ IV + .00082\ ASD + 2.59108\ SLOP1 + 2.13999\ SLOP2 - 3.15239\ MACH$$

$$R^2 = .51 \qquad S_{y.x} = 1.81$$

Where:

- AT = The average inturn time in minutes.
- IV = Volume skidded during the current turn (board feet) from subroutine SKIVOL.
- ASD = Distance that the turn is skidded (feet).
- SLOP1 = 1 if the turn is skidded uphill to the landing, = 0 if on level ground or downhill.
- SLOP2 = 1 if the turn is skidded over level terrain to the landing, = 0 if uphill or downhill.
- MACH = 1 if the skidder is rubber-tired with articulated frame, = 0 otherwise.

Unhooking the turn at the landing involves subroutine UNHOOK, and this is simply a procedure for using an average value (with its standard deviation) from the input data.

Occasionally the skidder will have to position some of the logs (trees) at the landing. If this occurs within a particular cycle, subroutine POSITN is used to generate the time involved. The inputs to this routine are an average time, its standard deviation, and an expected frequency of occurrence (used as in subroutine DELAY).

The generated delay times are obtained by subroutine DELAY, as discussed in the felling component section. Waiting delays that may arise can be attributed to two sources. The skidder may have to wait for logs (trees) in the woods and this may be caused by a felling crew, a buncher, or both.

When a cycle has been completed, the trees go into a bucking queue if bucking is done at the landing; if trees are bucked elsewhere the skidded logs (trees) are placed in a loading queue (the deck).

As with bunching, there is a considerable amount of interaction between the skidder(s) and the preceding function (either bunching or felling). This is handled in the same manner and with the same decision rules as used in the bunching component.

BUCKING COMPONENT

The simulator determines first whether bucking is performed in the woods, at the landing, or at the mill yard. If in the woods, program control transfers to the loading component; if at the landing, the bucking component is utilized; if at the mill yard, bucking is not considered.

The bucking component is composed of the bucking and marking time elements (generated as one value) and the delay times. Completion of a bucking cycle means that one (or sometimes two) trees have been bucked and the volume added to that available for loading (fig. 5).

Three subroutines are required in the bucking component. Subroutine TREEV generates the volume, diameter, and length of the tree to be bucked (as discussed previously). The simulated time for marking and bucking the tree is provided by subroutine BUCK. The average, or expected, time is obtained from the input deck or from the following equation (based on 317 observations from 7 operations):

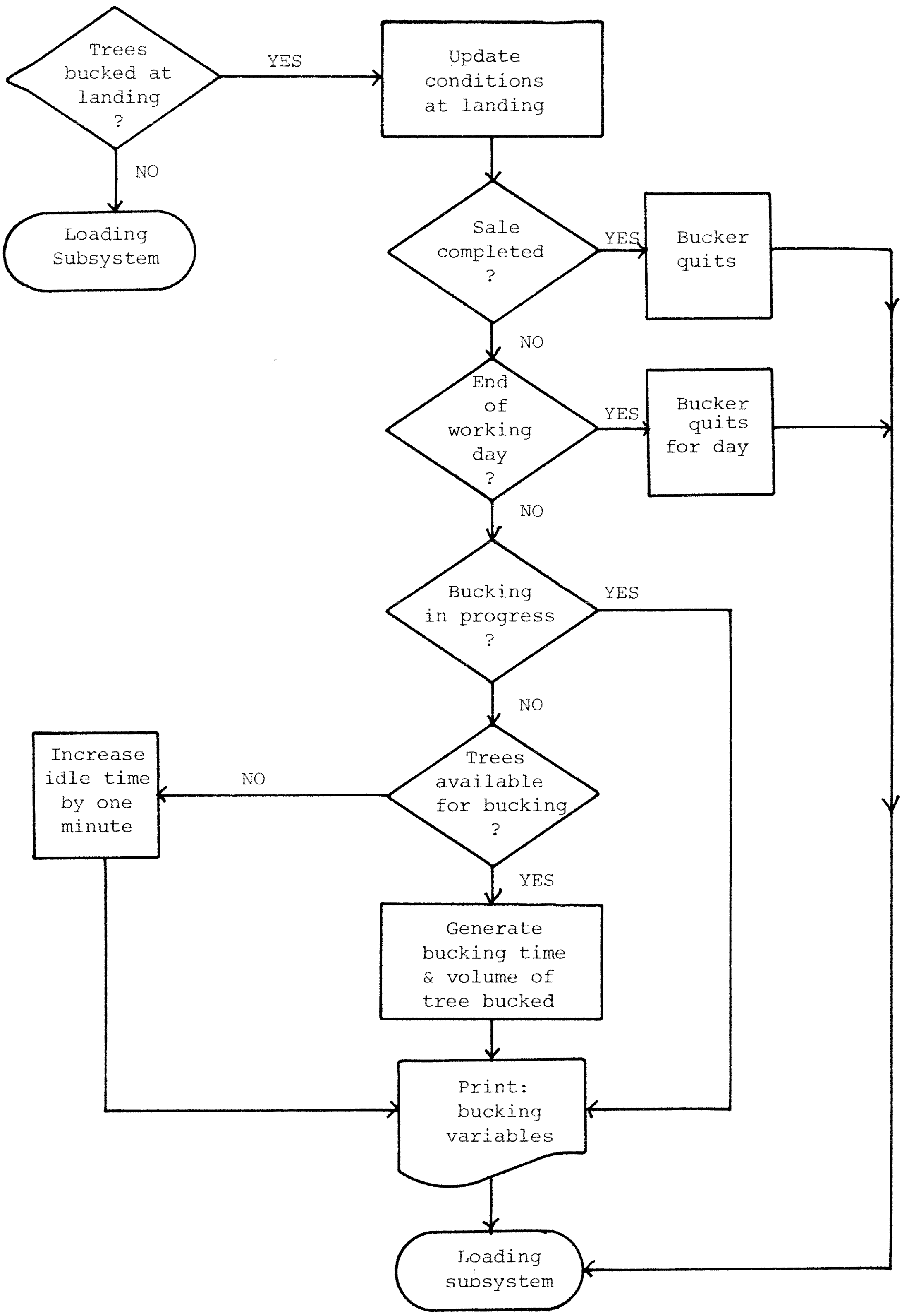
$$AT = -5.244 + 0.244\ DBH + 0.107\ HT$$

$$R^2 = .56 \qquad S_{y.x} = 1.01$$

Where:

- AT = The average marking and bucking time in minutes.
- DBH = Diameter (breast height) of the tree being bucked, from subroutine TREEV.

Figure 5.—Flow chart of bucking component.



HT = The height (length) of the tree being bucked, from subroutine TREEV.

When the tree has been bucked, this volume becomes available for loading, and it is added to the loading queue.

Delay time may be generated during the bucking of a tree and/or it may arise if the bucking crew has to wait for trees to arrive at the landing. This wait may now be attributable to any one (or all) of three preceding operations (felling, bunching, and skidding).

LOADING COMPONENT

Before any logs can be loaded, the simulator must first determine whether both logs and a truck are available. If either condition is not met, then waiting delay time will accrue. The loading section also contains the procedures for selecting the truck that will be loaded next. When trucks arrive at the landing, they are placed in a waiting line if another truck is currently being loaded. When loading is completed, the truck that has been waiting the longest time is selected to be loaded next.

The procedures involved in loading are detailed in figure 6. A loading cycle ends when a log has been positioned on the truck, and the elements of this cycle are simply the operating time and any delay that may arise. Subroutine LOADER generates these two time elements as well as calling for subroutine LOGVOL (or TREEV if tree-length hauling is done) to generate the log's volume. LOGVOL and TREEV have been explained in discussing the bunching and felling components, and their use in loading is the same. The operating time is obtained quite simply from subroutine LOGNOR and is based on an expected value from the input data (card 49).

Delay time is either generated (caused by the loader itself) or caused by waiting for logs or trucks. There are four possible sources of delay (felling, bunching, skidding, and bucking) that would cause the loader to set idle if logs weren't available. And if there were no trucks at the landing, the loader would be idle until one arrived.

After each log is placed on the truck, the program checks to see whether the truck is loaded. This is done by comparing volume on the truck to the expected load volume of the vehicle. Expected load volumes are read in as part of the input data (card 51).

After the truck is loaded, the times and volume are added to cumulative arrays; and at the end of the day a cost, time, and production summary is printed.

Provision is also made for moving the landing as cutting progresses through the stand. The decision rule for determining when to make the move has been discussed in the skidding section. When the time to move has been reached, the skidder(s) start to skid to the new landing area. If bucking is done at the landing, the bucking crew (if different from the loading crew) moves to the new location when they have completed their task at the old landing. When all logs at the old landing have been loaded, the loader is prepared for moving to the new site. To generate the time required for the move, subroutine BUNCH is again utilized. Since this is a simple routine having no specific equations, it is used instead of adding another subroutine designed for moving only. The generated time is based on an expected value (and its standard deviation) from input card 33.

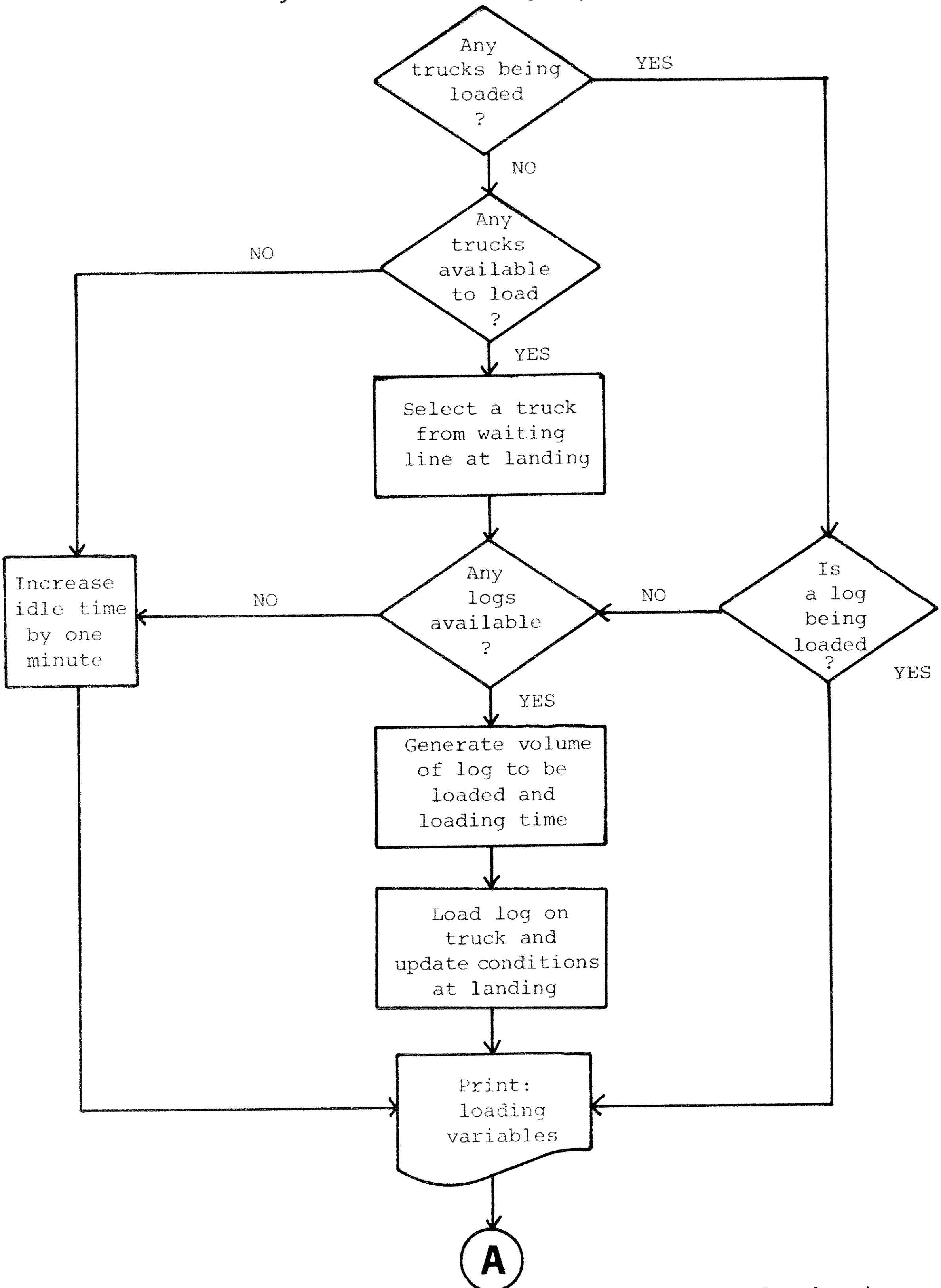
HAULING COMPONENT

Hauling consists of four tasks: the empty travel from a mill to the woods, the loading time, the loaded travel from woods to mill, and the unloading time at the mill yard. When a hauling cycle is finished, a truckload of logs has been delivered to the mill (fig. 7).

Most of the subroutines used in the hauling segment of T-H-A-T-S have been discussed before, hence only those introduced here will be explained.

Subroutine EMPTY generates the travel time from the mill to a woods landing. It incorporates four equations: one for each of four road types (table 1). The mechanics of this subroutine are best illustrated by considering an example. Suppose the one-way hauling distance is 29.7 miles, 24.7 miles of 2-lane paved roads, 1.4 miles of 1-lane paved roads, 1.3 miles of gravel roads, and 2.3 miles of woods

Figure 6.—Flow chart of loading component.



Continued

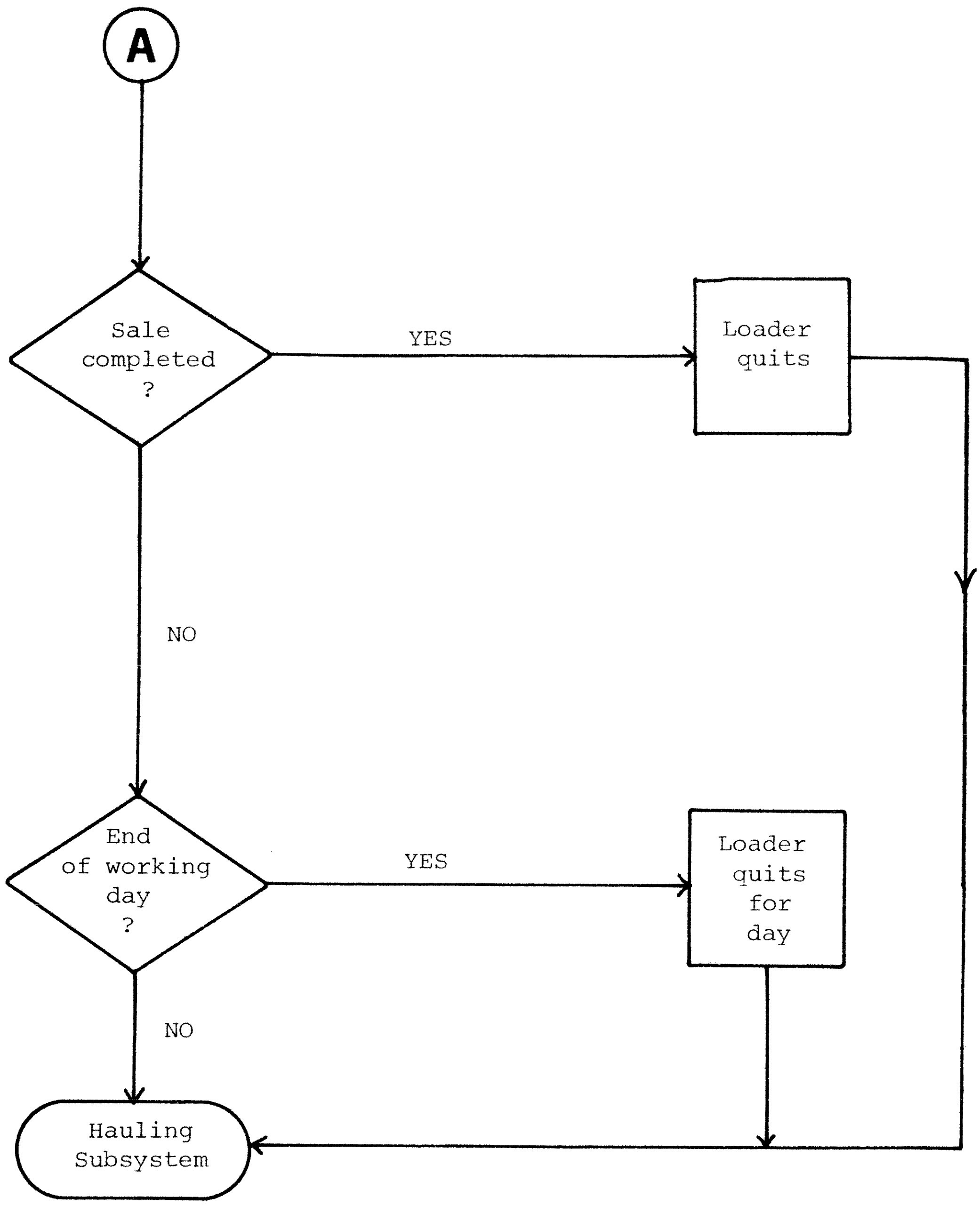
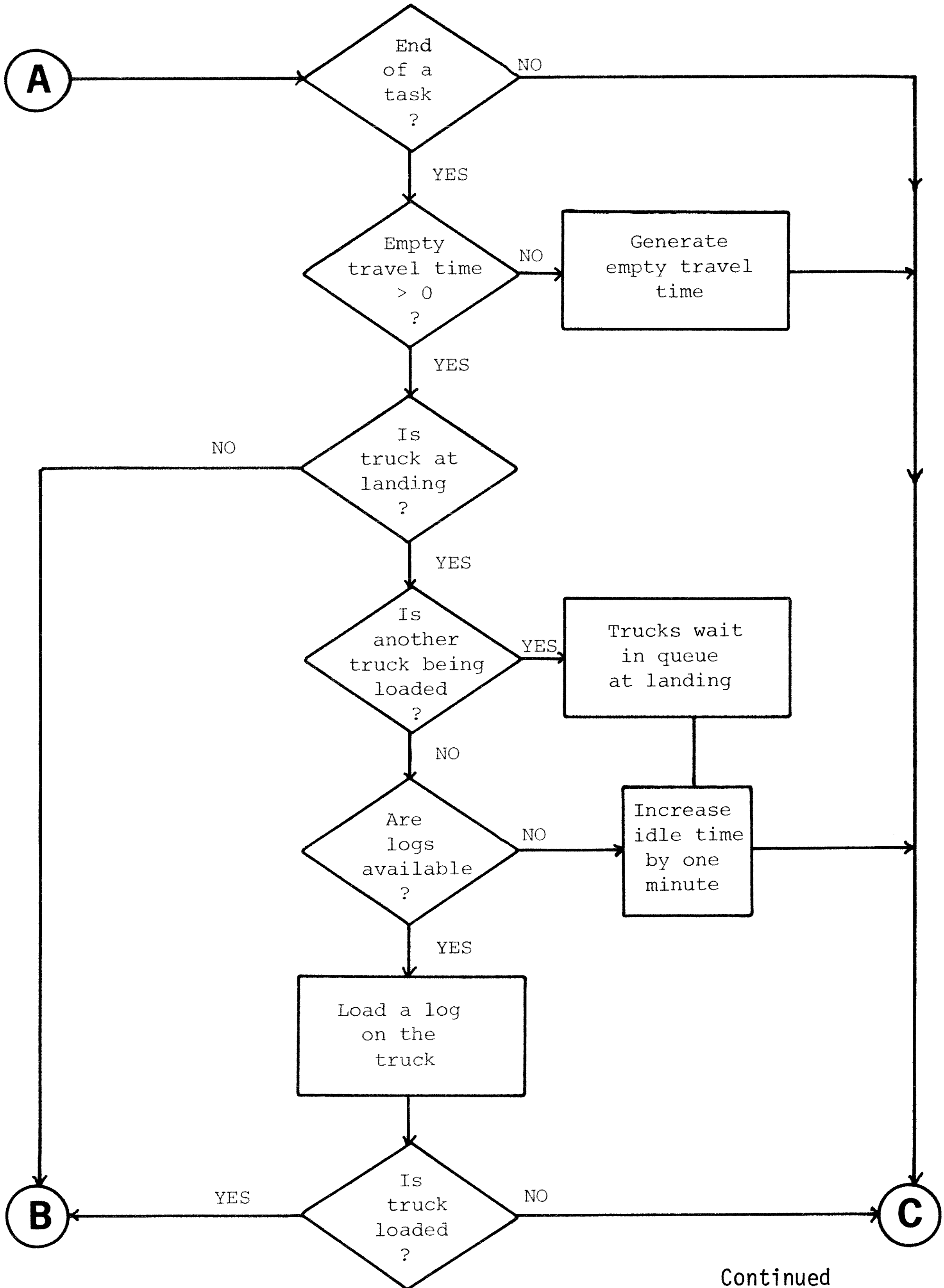
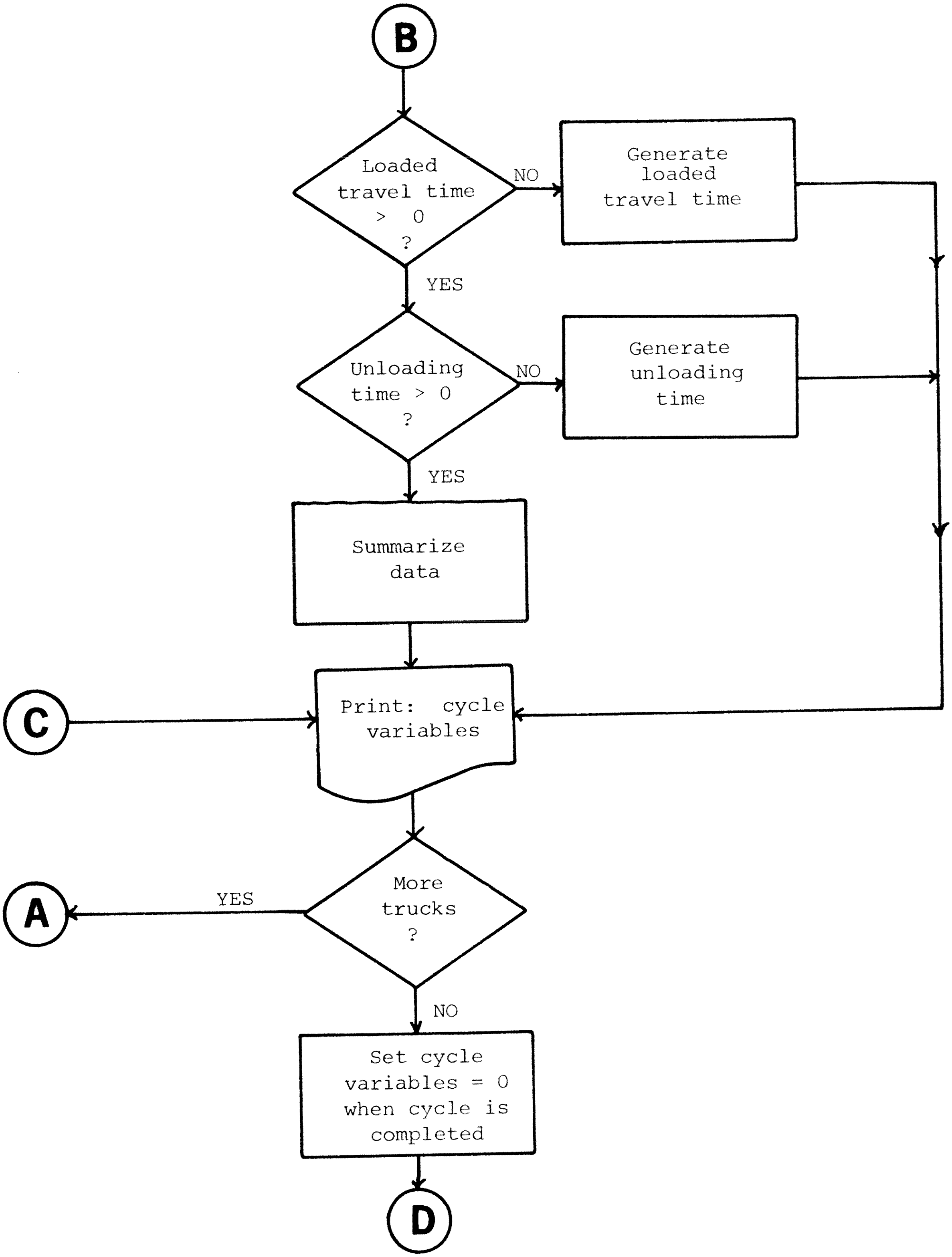
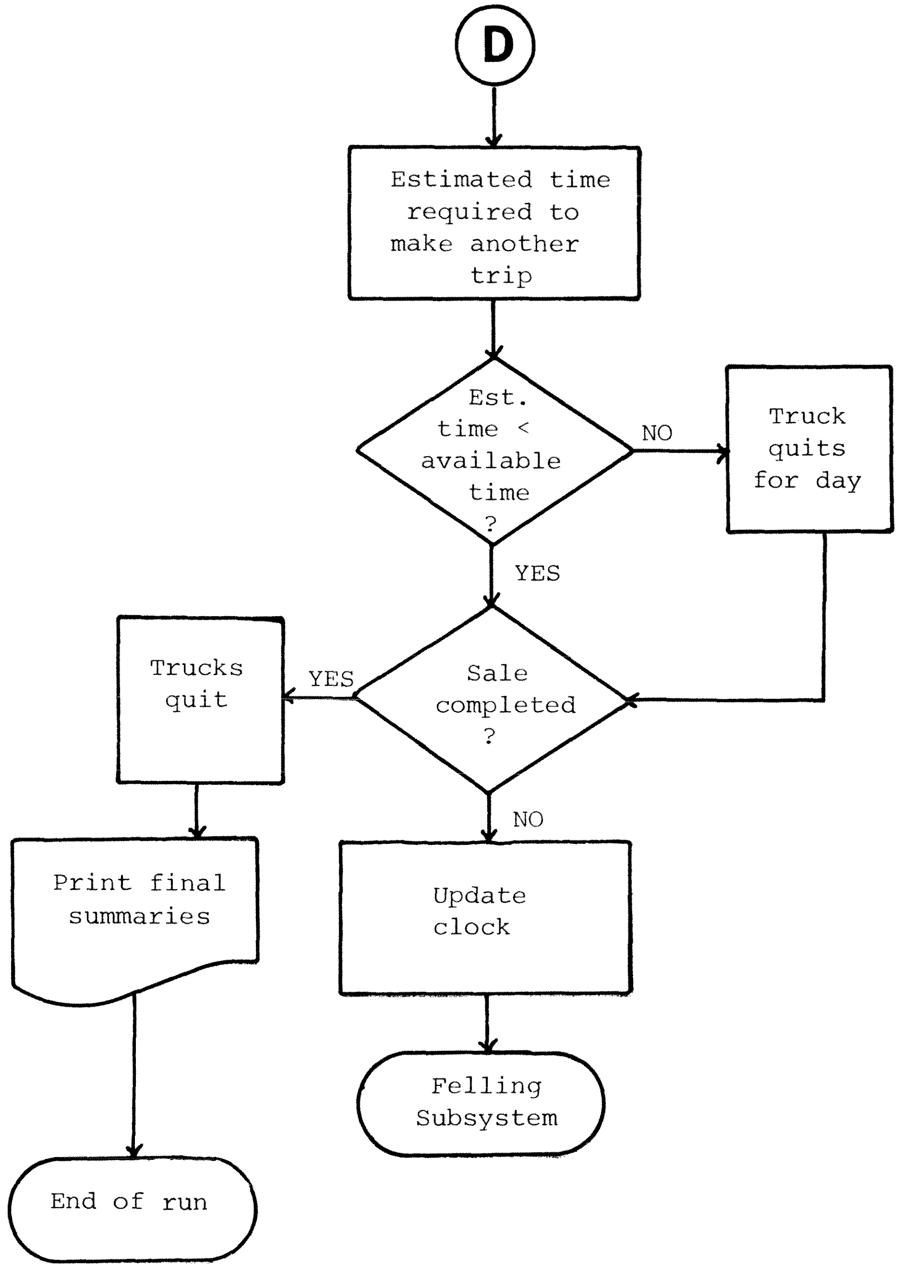

A
Sale completed ?
YES
Loader quits
NO
End of working day ?
YES
Loader quits for day
NO
Hauling Subsystem

Figure 7.—Flow chart of hauling component.



Continued


B
Loaded travel time > 0 ?
NO
Generate loaded travel time
YES
Unloading time > 0 ?
NO
Generate unloading time
YES
Summarize data
C
Print: cycle variables
A
YES
More trucks ?
NO
Set cycle variables = 0 when cycle is completed
D



roads. Substituting these values in the appropriate equations in table 1, we find the expected travel times for each segment are 40.1, 4.2, 8.0, and 21.5 minutes, respectively. These expected times, along with the appropriate standard error of estimate values (table 1), are then used in subroutine NORMAL to generate simulated times for each road segment during a given hauling cycle. The sum of these segment times becomes the simulated time to make the trip from mill to the woods.

Subroutine HAUL, which generates the travel time from woods to mill, functions in the same manner as that described for EMPTY. The equations used in HAUL are presented in table 2.

When a loaded truck reaches the mill yard, subroutine UNLOAD is called so that a simu-

Table 2.—Expected time equations for subroutine HAUL

Road type	Equation	Standard error of estimate	R^2
2-lane paved	$TH_1 = 5.91 + 1.76\ DIST_1$	8.1	0.90
1-lane paved	$TH_2 = 2.01 + 2.73\ DIST_2$	4.3	.69
Gravel	$TH_3 = 4.00 + 4.34\ DIST_3$	6.0	.82
Woods road	$TH_4 = 3.78 + 11.23\ DIST_4$	7.4	.80

TH_i = Travel time in minutes.
$DIST_i$ = Hauling distance in miles.

lated unloading time can be obtained. UNLOAD utilizes an expected time in conjunction with its standard deviation to generate this time element.

When a truck arrives at a landing in the woods, it is placed in a waiting line if the loader is occupied with another truck. Waiting delay time then accrues until the status of the truck is changed. If the landing is vacant when the truck arrives, the loading phase begins immediately if logs are available; if not, waiting delay time accrues until logs become available for loading.

After the truck is loaded, the driver usually spends a few minutes adjusting the load chains. This time is generated by subroutine DELAY. Provision is also made for generating other self-caused delays during other phases of the hauling cycle.

It is also possible to have the trucks (for those operations with more than one) start at different times in the morning, a staggered departure from the mill (or home). This is accomplished by using lag times that are read in for each truck (input data, card 15).

Both end-of-day and end-of-sale routines for hauling are similar to those used in the other components. However, with hauling, it is possible to have the truck end the day at the landing. In other words, the truck may remain in the woods overnight, either empty or partially loaded. This would occur if the log supply should run out before a truck is loaded, and the felling, skidding, or bucking crews have quit for the day.

ROADBUILDING COMPONENT

This is a relatively simple part of the simulation model because the costs of roadbuilding are treated as an add-on item, computed at the end of the sale. Total roadbuilding cost is computed by subroutine ROADS as follows:

$$RBC = (RB_1)\ (BC_1) + (RB_2)\ (BC_2) + (RB_3)\ (BC_3) + (RB_4)\ (BC_4)$$

Where:

- RBC = Total roadbuilding cost.
- RB_1 = Miles of main skidroads built.
- RB_2 = Miles of spur skidroads built.
- RB_3 = Miles of main haul roads built.
- RB_4 = Miles of secondary haulroads built.
- BC_1 = Construction costs per mile for main skidroads.
- BC_2 = Construction costs per mile for spur skidroads.
- BC_3 = Construction costs per mile for main haulroads.
- BC_4 = Construction costs per mile for secondary haulroads.

The values for RB_1 and RB_2 may be read in with other input data (card 44), or certain assumptions contained in the subroutine may be used instead. If RB_1 and RB_2 are left blank, the simulator automatically assumes that 65 feet of main skidroad and 75 feet of spur road will be constructed for each acre that is logged (*Smithson and Phillips 1970*). Also, if BC_2 is not entered as input, the program will assign a cost-per-mile figure equal to one-fourth of BC_1.

If the separate values of RB_3 and RB_4 are unknown, the simulator uses $DIST_4$ (miles of woods road) as the basis for computing building costs of haulroads. If unknown, RB_3 and RB_4 should be left blank on Card 44. Using $DIST_4$ results in the same cost factor (BC_3) being used for all haulroad construction.

Roadbuilding costs are computed only when the sale has been completed, hence they are only shown in the final table of costs and times. Roadbuilding cost per MBF is based on the total volume hauled from the sale area.

COST COMPONENT

The computation of system cost requires that first the machine rate of each piece of logging equipment in the operation be calculated. T-H-A-T-S does this by means of two subroutines (COSTS and COST), and cost data supplied as part of the input (card 56).

A separate input card is needed for each machine or crew in the logging system. For example, if two fellers are working on the same sale, then two cost-data cards are needed for the felling component. If only one skidder is utilized, then only one card is needed for the skidding component; however, if more than one man is working with the skidder, then the hourly wage rate on this card should reflect the cost of the other men. These rules apply to all other system components as well.

The subroutines calculate two machine rates ($/hour) for each machine or crew in the system; one is applied to the time spent in production (operating time) during the day and the other to time spent idle (delay time). Operating-time machine rates include all cost items, whereas delay-time rates only include the fixed costs (*Martin 1971*).

The operating-time machine rate varies from day to day because the amount of consumables (gas, oil, etc.) changes if the distance traveled changes. A truck may make three trips today and perhaps only one tomorrow. Repair costs and tire wear also vary with the amount of use the vehicle is subjected to.

At the end of each day's operation, and at the end of the sale, the appropriate machine rates are applied to the operating and delay times for all equipment in the system. Summing over all phases of the system provides the daily cost (or sale cost), and dividing by production gives the harvesting cost per MBF (See next section on output from T-H-A-T-S).

OUTPUT FROM T-H-A-T-S

Four types of printed output are generated by T-H-A-T-S:

- System status information.
- Time summaries.
- Production summaries.
- Cost summaries.

The status of the system can be printed for each minute throughout the day, only once a day, or at any interval between these extremes (figs. 8, 9, 10, and 11). The input used to generate the output in these examples is presented in the input instructions.

The interpretation of the columns headed TO, TL, TI, and TU are presented in figure 12. The values found in these columns are simulated minutes for the current cycle. The remaining column headings are explained below:

TP — Time required to position logs at the landing, when it occurs, for the current cycle.

GDT — The generated (self-caused) delay time for the current cycle.

WDT — The waiting delay time that arises during the current cycle.

TOTAL — The total cycle time; TO + TL + TI + TU + TP + GDT + WDT; when the cycle is complete, = 0 otherwise.

VOL/C — The volume (board feet) produced during the current cycle.

	Bucker	*Loader*
VOLLEFT	Volume at the landing that remains to be bucked.	Volume at the landing that is available for loading.
VOLDONE	The cumulative volume bucked during each day; set to zero every morning.	The cumulative volume loaded during each day; set to zero every morning.

CUMOT — The cumulative operating time for each day; set to zero every morning.

CUMDT — The cumulative delay time for each day; set to zero every morning.

By examining the output in figure 8, it is easy to see the methodology of T-H-A-T-S in action. For example, at 8:00, a time of 1 minute was generated for walking to the first tree by the feller. At 8:01 (1 minute later) the felling time of 3 minutes was generated along with the tree's volume (51 board feet). When the 3 minutes have elapsed (at 8:04), the limbing and topping time of 1 minute is generated. And at 8:05, the first felling cycle has been completed and the total time of 5 minutes is printed. Note that no delay time occurred during this cycle; however, the buncher had to wait for this first tree and at 8:05 the accumulated waiting delay time for the buncher is 5 minutes.

Time, production, and cost summaries are printed at the end of each day (and at the end of the simulation run). An example of this can be seen in figure 11.

The production and cost summary is self-explanatory. However, the time summary requires some discussion. The interpretation of these data is best explained by an example or two. Let's consider the information contained in figure 11—the summary of the first day's activities for our illustrative problem (system 272):

The first two rows in the time summary contain felling information; and in the column labeled "Fell", we see the values 147 and 271. This means that 147 minutes were spent during the first day in actually felling trees (operating time). Under the heading "Trees", we find 94—meaning that 94 trees were felled during the day. Therefore, the average felling time was 1.6 minutes (147/94). The standard deviation about the average can also be computed, because 271 is the sum of squares for the individual felling times:

$$s = \sqrt{\frac{271 - (147^2/94)}{94 - 1}} = 0.67 \text{ minutes}$$

Delay-time statistics can also be computed easily by using the figures under the headings DF and F(DF). F(DF) = 8 indicates that delay time occurred during the felling phase of 8 of the 94 trees that were cut during the first day. Therefore, the frequency of occurrence = 0.085 (8/94) or 8.5 percent. The average delay time during the felling phase (when it occurred) was 3.1 minutes (25/8) with a standard deviation of:

$$s = \sqrt{\frac{155 - (25^2/8)}{8 - 1}} = 3.31 \text{ minutes}$$

The same calculations can be performed on the remaining data in the time summary. These values are cumulative so that at the termination of a run, the last time summary printed contains values for the entire run.

The cost summary (including production figures) is also depicted in figure 11. This is printed daily and at the end of the simulation run. It should be noted that the bottom figure in the cost/MBF column is not the sum of this column, but rather is the overall system cost per MBF, calculated as follows:

$$\text{Total system cost/MBF} = \left(\text{Total system cost} \Big/ \text{Volume delivered to mill} \right) 1000$$

and using the data shown in figure 11:

$$\text{Total system cost/MBF} = (\$413.48/5143) \quad 1000 = \$80.40$$

MODEL VALIDITY

The question arises: "How good is T-H-A-T-S?; how much faith should we have in the simulation results?" This is a rather difficult question to answer because, as with most research, the findings and recommendations are based on sample data and generalizations drawn from these data. However, we have made a large number of runs with T-H-A-T-S, using data from seven logging operations and the results show that for most attributes, the simulator resembles the real-world quite well.

As would be expected when an average time and its standard deviation for a particular function are used (read as input), the simulator generates values having an average (and

Figure 8.—A portion of the output from the first day of simulated operation for the example problem, showing the system status printed every minute.

DAY NO. 1 EXAMPLE PROBLEM — SYSTEM 272

TIME		TO	TL	TI	TU	TP	GDT	WDT	TOTAL	VOL/C	VOLLEFT	VOLDONE	CUMOT	CUMDT		
8:00	FELL—1	1	0	0	0		0		0	0					FELLER	1
8:00	BUNCH-1	0					0	0	0	0					BUNCHER	1
8:00	SKID—1	5	0	0	0	0	1	0	0	0					SKIDDER	1
8:00	HAUL—1	77	0	0	0		4	0	0	0					TRUCK	1
8:00	HAUL—2	84	0	0	0		3	0	0	0					TRUCK	2
8:00	LOADER		0					0		0	0	0	0	0	LOADER	
8:00	BUCKER		0					0		0	0	0	0	0	BUCKER	
8:01	FELL—1	1	3	0	0		0		0	51					FELLER	1
8:01	BUNCH-1	0					0	1	0	0					BUNCHER	1
8:01	SKID—1	5	0	0	0	0	1	0	0	0					SKIDDER	1
8:01	HAUL—1	77	0	0	0		4	0	0	0					TRUCK	1
8:01	HAUL—2	84	0	0	0		3	0	0	0					TRUCK	2
8:01	LOADER		0					1		0	0	0	0	0	LOADER	
8:01	BUCKER		0					1		0	0	0	0	0	BUCKER	
8:02	FELL—1	1	3	0	0		0		0	51					FELLER	1
8:02	BUNCH-1	0					0	2	0	0					BUNCHER	1
8:02	SKID—1	5	0	0	0	0	1	0	0	0					SKIDDER	1
8:02	HAUL—1	77	0	0	0		4	0	0	0					TRUCK	1
8:02	HAUL—2	84	0	0	0		3	0	0	0					TRUCK	2
8:02	LOADER		0					2		0	0	0	0	0	LOADER	
8:02	BUCKER		0					2		0	0	0	0	0	BUCKER	
8:03	FELL—1	1	3	0	0		0		0	51					FELLER	1
8:03	BUNCH-1	0					0	3	0	0					BUNCHER	1
8:03	SKID—1	5	0	0	0	0	1	0	0	0					SKIDDER	1
8:03	HAUL—1	77	0	0	0		4	0	0	0					TRUCK	1
8:03	HAUL—2	84	0	0	0		3	0	0	0					TRUCK	2
8:03	LOADER		0					3		0	0	0	0	0	LOADER	
8:03	BUCKER		0					3		0	0	0	0	0	BUCKER	
8:04	FELL—1	1	3	0	1		0		0	51					FELLER	1
8:04	BUNCH-1	0					0	4	0	0					BUNCHER	1
8:04	SKID—1	5	0	0	0	0	1	0	0	0					SKIDDER	1
8:04	HAUL—1	77	0	0	0		4	0	0	0					TRUCK	1
8:04	HAUL—2	84	0	0	0		3	0	0	0					TRUCK	2
8:04	LOADER		0					4		0	0	0	0	0	LOADER	
8:04	BUCKER		0					4		0	0	0	0	0	BUCKER	
8:05	FELL—1	1	3	0	1		0		5	51					FELLER	1
8:05	BUNCH-1	22					0	5	0	51					BUNCHER	1
8:05	SKID—1	5	0	0	0	0	1	0	0	0					SKIDDER	1
8:05	HAUL—1	77	0	0	0		4	0	0	0					TRUCK	1
8:05	HAUL—2	84	0	0	0		3	0	0	0					TRUCK	2
8:05	LOADER		0					5		0	0	0	0	0	LOADER	
8:05	BUCKER		0					5		0	0	0	0	0	BUCKER	

Continued

TIME		TO	TL	TI	TU	TP	GDT	WDT	TOTAL	VOL/C	VOLLEFT	VOLDONE	CUMOT	CUMDT		
8:06	FELL--1	1	1	0	0		0		0	120					FELLER	1
8:06	BUNCH-1	22					0	5	0	51					BUNCHER	1
8:06	SKID--1	5	0	0	0	0	1	0	0	0					SKIDDER	1
8:06	HAUL--1	77	0	0	0		4	0	0	0					TRUCK	1
8:06	HAUL--2	84	0	0	0		3	0	0	0					TRUCK	2
8:06	LOADER		0					6		0	0	0	0	0	LOADER	
8:06	BUCKER		0					6		0	0	0	0	0	BUCKER	
8:07	FELL--1	1	1	0	1		0		0	120					FELLER	1
8:07	BUNCH-1	22					0	5	0	51					BUNCHER	1
8:07	SKID--1	5	0	0	0	0	1	1	0	0					SKIDDER	1
8:07	HAUL--1	77	0	0	0		4	0	0	0					TRUCK	1
8:07	HAUL--2	84	0	0	0		3	0	0	0					TRUCK	2
8:07	LOADER		0					7		0	0	0	0	0	LOADER	
8:07	BUCKER		0					7		0	0	0	0	0	BUCKER	
8:08	FELL--1	1	1	0	1		0		3	120					FELLER	1
8:08	BUNCH-1	22					0	5	0	51					BUNCHER	1
8:08	SKID--1	5	0	0	0	0	1	2	0	0					SKIDDER	1
8:08	HAUL--1	77	0	0	0		4	0	0	0					TRUCK	1
8:08	HAUL--2	84	0	0	0		3	0	0	0					TRUCK	2
8:08	LOADER		0					8		0	0	0	0	0	LOADER	
8:08	BUCKER		0					8		0	0	0	0	0	BUCKER	
8:09	FELL--1	1	3	0	0		0		0	210					FELLER	1
8:09	BUNCH-1	22					0	5	0	51					BUNCHER	1
8:09	SKID--1	5	0	0	0	0	1	3	0	0					SKIDDER	1
8:09	HAUL--1	77	0	0	0		4	0	0	0					TRUCK	1
8:09	HAUL--2	84	0	0	0		3	0	0	0					TRUCK	2
8:09	LOADER		0					9		0	0	0	0	0	LOADER	
8:09	BUCKER		0					9		0	0	0	0	0	BUCKER	
8:10	FELL--1	1	3	0	0		0		0	210					FELLER	1
8:10	BUNCH-1	22					0	5	0	51					BUNCHER	1
8:10	SKID--1	5	0	0	0	0	1	4	0	0					SKIDDER	1
8:10	HAUL--1	77	0	0	0		4	0	0	0					TRUCK	1
8:10	HAUL--2	84	0	0	0		3	0	0	0					TRUCK	2
8:10	LOADER		0					10		0	0	0	0	0	LOADER	
8:10	BUCKER		0					10		0	0	0	0	0	BUCKER	

Figure 9.—A portion of the output from the first day of simulated operation for the example problem, showing the system status printed every 10 minutes.

DAY NO. 1 EXAMPLE PROBLEM — SYSTEM 272

TIME		TO	TL	TI	TU	TP	GDT	WDT	TOTAL	VOL/C	VOLLEFT	VOLDONE	CUMOT	CUMDT		
8:00	FELL—1	1	0	0	0		0		0	0					FELLER	1
8:00	BUNCH-1	0					0	0	0	0					BUNCHER	1
8:00	SKID—1	5	0	0	0	0	1	0	0	0					SKIDDER	1
8:00	HAUL—1	77	0	0	0		4	0	0	0					TRUCK	1
8:00	HAUL—2	84	0	0	0		3	0	0	0					TRUCK	2
8:00	LOADER		0					0		0	0	0	0	0	LOADER	
8:00	BUCKER		0					0		0	0	0	0	0	BUCKER	
8:10	FELL—1	1	3	0	0		0		0	210					FELLER	1
8:10	BUNCH-1	22					0	5	0	51					BUNCHER	1
8:10	SKID—1	5	0	0	0	0	1	4	0	0					SKIDDER	1
8:10	HAUL—1	77	0	0	0		4	0	0	0					TRUCK	1
8:10	HAUL—2	84	0	0	0		3	0	0	0					TRUCK	2
8:10	LOADER		0					10		0	0	0	0	0	LOADER	
8:10	BUCKER		0					10		0	0	0	0	0	BUCKER	
8:20	FELL—1	1	2	0	0		0		0	156					FELLER	1
8:20	BUNCH-1	22					0	5	0	51					BUNCHER	1
8:20	SKID—1	5	0	0	0	0	1	14	0	0					SKIDDER	1
8:20	HAUL—1	77	0	0	0		4	0	0	0					TRUCK	1
8:20	HAUL—2	84	0	0	0		3	0	0	0					TRUCK	2
8:20	LOADER		0					20		0	0	0	0	0	LOADER	
8:20	BUCKER		0					20		0	0	0	0	0	BUCKER	
8:30	FELL—1	1	1	0	1		0		3	132					FELLER	1
8:30	BUNCH-1	22					0	0	0	469					BUNCHER	1
8:30	SKID—1	5	0	0	0	0	1	24	0	0					SKIDDER	1
8:30	HAUL—1	77	0	0	0		4	0	0	0					TRUCK	1
8:30	HAUL—2	84	0	0	0		3	0	0	0					TRUCK	2
8:30	LOADER		0					30		0	0	0	0	0	LOADER	
8:30	BUCKER		0					30		0	0	0	0	0	BUCKER	
8:40	FELL—1	1	1	0	1		0		3	61					FELLER	1
8:40	BUNCH-1	22					0	0	0	469					BUNCHER	1
8:40	SKID—1	5	0	0	0	0	1	34	0	0					SKIDDER	1
8:40	HAUL—1	77	0	0	0		4	0	0	0					TRUCK	1
8:40	HAUL—2	84	0	0	0		3	0	0	0					TRUCK	2
8:40	LOADER		0					40		0	0	0	0	0	LOADER	
8:40	BUCKER		0					40		0	0	0	0	0	BUCKER	
8:50	FELL—1	1	1	0	2		0		4	153					FELLER	1
8:50	BUNCH-1	16					0	0	0	413					BUNCHER	1
8:50	SKID—1	5	17	0	0	0	16	43	0	520					SKIDDER	1
8:50	HAUL—1	77	0	0	0		4	0	0	0					TRUCK	1
8:50	HAUL—2	84	0	0	0		3	0	0	0					TRUCK	2
8:50	LOADER		0					50		0	0	0	0	0	LOADER	
8:50	BUCKER		0					50		0	0	0	0	0	BUCKER	

Figure 10.—A portion of the output from the first day of simulated operation for the example problem, showing the system status printed every hour.

DAY NO. 1 EXAMPLE PROBLEM — SYSTEM 272

TIME		TO	TL	TI	TU	TP	GDT	WDT	TOTAL	VOL/C	VOLLEFT	VOLDONE	CUMOT	CUMDT		
8:00	FELL—1	1	0	0	0		0		0	0					FELLER	1
8:00	BUNCH-1	0					0	0	0	0					BUNCHER	1
8:00	SKID—1	5	0	0	0	0	1	0	0	0					SKIDDER	1
8:00	HAUL—1	77	0	0	0		4	0	0	0					TRUCK	1
8:00	HAUL—2	84	0	0	0		3	0	0	0					TRUCK	2
8:00	LOADER		0					0		0	0	0	0	0	LOADER	
8:00	BUCKER		0					0		0	0	0	0	0	BUCKER	
9:00	FELL—1	1	1	0	1		0		0	92					FELLER	1
9:00	BUNCH-1	16					0	0	0	413					BUNCHER	1
9:00	SKID—1	5	17	0	0	0	16	43	0	520					SKIDDER	1
9:00	HAUL—1	77	0	0	0		4	0	0	0					TRUCK	1
9:00	HAUL—2	84	0	0	0		3	0	0	0					TRUCK	2
9:00	LOADER		0					60		0	0	0	0	0	LOADER	
9:00	BUCKER		0					60		0	0	0	0	0	BUCKER	
10:00	FELL—1	1	1	0	1		0		3	67					FELLER	1
10:00	BUNCH-1	11					0	0	0	611					BUNCHER	1
10:00	SKID—1	1	30	0	0	0	0	0	0	364					SKIDDER	1
10:00	HAUL—1	77	11	0	0		4	28	0	520					TRUCK	1
10:00	HAUL—2	84	0	0	0		3	33	0	0					TRUCK	2
10:00	LOADER		0					109		0	0	520	11	0	LOADER	
10:00	BUCKER		0					9		0	0	520	8	103	BUCKER	
11:00	FELL—1	1	1	0	0		0		0	106					FELLER	1
11:00	BUNCH-1	12					0	0	0	296					BUNCHER	1
11:00	SKID—1	1	25	6	0	0	27	0	0	962					SKIDDER	1
11:00	HAUL—1	77	16	0	0		4	83	0	884					TRUCK	1
11:00	HAUL—2	84	0	0	0		3	93	0	0					TRUCK	2
11:00	LOADER		0					164		0	0	884	16	0	LOADER	
11:00	BUCKER		0					51		0	0	884	14	115	BUCKER	
12:00	FELL—1	1	2	0	1		6		10	180					FELLER	1
12:00	BUNCH-1	9					0	0	0	206					BUNCHER	1
12:00	SKID—1	3	18	0	0	0	24	0	0	797					SKIDDER	1
12:00	HAUL—1	77	36	115	0		20	111	0	2004					TRUCK	1
12:00	HAUL—2	84	2	0	0		3	151	0	228					TRUCK	2
12:00	LOADER		0					10		0	0	2232	38	192	LOADER	
12:00	BUCKER		0					8		0	0	2232	27	205	BUCKER	
1:00	FELL—1	1	1	0	0		0		0	106					FELLER	1
1:00	BUNCH-1	14					0	0	0	526					BUNCHER	1
1:00	SKID—1	3	18	4	0	0	55	0	0	797					SKIDDER	1
1:00	HAUL—1	77	36	115	0		50	111	0	2004					TRUCK	1
1:00	HAUL—2	84	2	0	0		33	181	0	228					TRUCK	2
1:00	LOADER		0					40		0	0	2232	38	222	LOADER	
1:00	BUCKER		0					38		0	0	2232	27	235	BUCKER	

Figure 11.—The output from the first day of simulated operation for the example problem showing the system status printed only once during the day.

DAY NO. 1 EXAMPLE PROBLEM — SYSTEM 272

TIME		TO	TL	TI	TU	TP	GDT	WDT	TOTAL	VOL/C	VOLLEFT	VOLDONE	CUMOT	CUMDT		
8:00	FELL--1	1	0	0	0		0		0	0					FELLER	1
8:00	BUNCH-1	0					0	0	0	0					BUNCHER	1
8:00	SKID--1	5	0	0	0	0	1	0	0	0					SKIDDER	1
8:00	HAUL--1	77	0	0	0		4	0	0	0					TRUCK	1
8:00	HAUL--2	84	0	0	0		3	0	0	0					TRUCK	2
8:00	LOADER		0					0		0	0	0	0	0	LOADER	
8:00	BUCKER		0					0		0	0	0	0	0	BUCKER	

T I M E S U M M A R Y

WALK	FELL	BUCK	LIMB	DW	DF	DB	DL	DUM	TREES	F(DW)	F(DF)	F(DB)	F(DL)
98	147	0	161	0	25	0	72	0	94	0	8	0	12
106	271	0	411	0	155	0	584	0					

OUT	HOOK	SKID	UNHOOK	DO	DH	DS	DU	TREES	TURNS	F(DO)	F(DH)	F(DS)	F(DU)
20	242	42	19	33	115	11	0	63	13	5	11	6	0
48	4754	246	31	299	1867	25	0	331					

OUT	LOGS	HAUL	UNLOAD	DO	DL	DH	DU	CHAIN	TRIPS	F(DO)	F(DL)	F(DH)	F(DU)
161	90	226	10	7	0	15	0	6	2	2	0	2	0
12985	0	25546	52	25	0	173	0	18					

C O S T S U M M A R Y

EQUIPMENT		OPERATING TIME	OPERATING COST	DELAY TIME	DELAY COST	TOTAL TIME	TOTAL COST	VOLUME	COST/MBF
FELLER	1	406	22.79	127	6.25	533	29.04	14420	2.01
BUNCHER	1	490	77.90	56	6.79	546	84.69	14210	5.96
SKIDDER	1	323	73.00	232	38.30	555	111.31	8954	12.43
TRUCK	1	232	31.01	161	13.57	393	44.58	2004	22.25
TRUCK	2	254	35.00	298	28.68	552	63.67	3139	20.28
LOADER		89	14.08	450	63.40	539	77.48	5143	15.07
BUCKER		106	0.91	423	1.79	529	2.70	8194	0.33
TOTALS		---	254.69	---	158.79	---	413.48	5143	80.40

COMPONENT	COLUMN HEADING			
	TO	TL	TI	TU
Felling	Walking between trees	Felling	Bucking in the woods	Limbing and topping
Bunching	Operating time/cycle	—	—	—
Skidding	Return to felling area	Hookup at felling area	Skidding to landing	Unhooking at landing
Hauling	Mill to landing	Loading	Landing to mill	Unloading
Loading	—	Operating time per cycle	—	—
Bucking	—	Operating time per cycle	—	—

Figure 12.—The meaning of certain column headings found in the output from T-H-A-T-S. All values are put in these columns in whole minutes for the current cycle.

standard deviation) very close to the input. Usually the difference, between actual and simulated, is less than ½ minute for both the average and standard deviation.

Empirical distributions are used for dbh, log diameter, and log length; and the use of these by the simulator results in tree and log volumes very close to the actual for the stand being harvested. Also, the volumes per cycle (for bunching, skidding, and hauling) are usually close to observations made on the actual logging operations.

The equations used in some of the felling, skidding, and bucking subroutines were developed from the same data we have used to test the simulator. Needless to say, they work quite well for these seven operations in Appalachia. Whether they would be adequate for other areas is difficult to say, because tests have not been conducted. However, new equations can be substituted easily without affecting the mechanics or performance of the simulator.

Equations used in the hauling subroutines (EMPTY and HAUL) were developed from data collected on entirely different logging operations, yet they have been satisfactory predictors for the seven operations used in testing T-H-A-T-S. Of course, they should be checked before extending their use to areas outside the Appalachian Region. Also, since they are representative of performance with tandem axle trucks, they may not be adequate for tractor-trailer rigs, etc.

Waiting delay time is a good indicator of system interaction; and since it is not a part of the input, it becomes a fairly good criteria for model validity. In collecting data on the seven operations, we coded all delay times as to whether they were self-caused or whether they accrued while waiting for other system components to finish a task. Waiting delays generated by a simulation run usually corresponded rather closely to the observed values. For example, the actual waiting delay time per skidding turn was 1½ minutes for system 272 (the system used as an example problem in this discussion). Using the time summary data generated at the completion of the run, we can compute the simulated average time spent waiting by the skidder:

Total delay time over 9 days = 1,747 minutes

Less self-caused delay:

Return trips	—	168 minutes
Hookup	—	933 minutes
Skidding	—	61 minutes
Unhooking	—	272 minutes
		1,434 minutes

Total waiting delay time = 313 minutes

Averaging this total waiting delay time over 123 turns, we get 2.5 minutes per turn, which is only 1 minute greater than the observed value.

The methodology of the model and its structure need little validation because the simulator simply duplicates the process of logging as it is normally performed in Appalachia. The sequence of operations is the same in both.

CONCLUSION

This paper was designed to provide the information needed for using T-H-A-T-S to analyze a logging system. Although the present subroutines make it peculiar to the Appalachian Region, the simulator is sufficiently flexible to be adapted for use with most logging operations.

Simulation is an invaluable aid to the decision-maker because for the first time he is able to see what effects a proposed change will have on his entire system. T-H-A-T-S is no cure-all, but it can provide new insight into the workings of a logging system, and at a price the manager can well afford to pay.

LITERATURE CITED

International Business Machines Corporation.
1968. SYSTEM/360 SCIENTIFIC SUBROUTINE PACKAGE (360A-CM-03X), VERSION III, PROGRAMMER'S MANUAL H20-0205-3, 454 p.

Martin, A. Jeff.
1971. THE RELATIVE IMPORTANCE OF FACTORS THAT DETERMINE LOG HAULING COSTS. USDA For. Serv. Res. Pap. NE-197. 15 p.

Martin, A. Jeff.
1973. COMPUTER SIMULATION AS A LOW-COST APPROACH TO LOGGING SYSTEM BALANCE. For. Indus. 110(9): 52-54.

Naylor, T. H., J. L. Balintfy, O. S. Burdick, and K. Chu.
1966. COMPUTER SIMULATION TECHNIQUES. 352 p. John Wiley & Sons, Inc., New York.

Smithson, P., and R. A. Phillips.
1970. SKIDROADS FOR LOGGING OPERATIONS IN THE APPALACHIANS. W. Va. Univ. For. Eng. Workshop on Forest Roads Proc.: 17-23.

APPENDIX

A. COMPUTER COMPATIBILITY

T-H-A-T-S is programmed in FORTRAN IV and has been successfully run on an IBM System/370 computer. It should perform equally well with any large computing hardware. Although the program is quite large in terms of source statements, the amount of array storage needed is relatively small (only 2.5K). Consequently, total core storage required is less than 175K.

When the program is to be run, it may be necessary to change the logical unit numbers for the READ and WRITE statements. The IBM system that was used required a 5 for input from cards and a 6 for printed output. To change these values, the user must repunch the first two statements that follow the DIMENSION statement, LUI = 5 and LUO = 6. LUI is the logical unit number for the READ statements, and LUO corresponds to WRITE statements.

B. NORMAL AND LOGNORMAL DISTRIBUTIONS

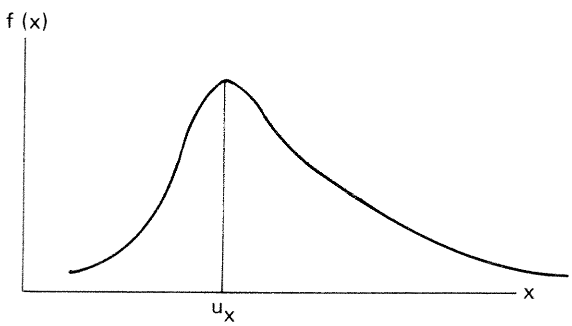
Although subroutine RANDU provides random numbers whenever needed, in most cases the value generated by the simulator must be more than random; it must correspond to a specified distribution. That is, it must be a random variable, not just a random number.

Simulated times are generated by T-H-A-T-S from one of two subroutines. Subroutine NORMAL is based on the normal distribution, subroutine LOGNOR on the lognormal distribution.

The lognormal distribution, which is employed by most of the subroutines in the present version of T-H-A-T-S, is explained as follows (*Naylor et al. 1966, p. 99*):

> If the logarithm of a random variable has a normal distribution, the random variable has a positively skewed continuous distribution known as the lognormal distribution.

In other words, if the data were distributed this



way—it was concluded that a lognormal distribution (and subroutine LOGNOR) should be more appropriate for generating the simulated values than the normal distribution. And after considerable testing with the simulator, the conclusion was verified; "better" values were generated by LOGNOR for those variables having the skewed distribution.

The lognormal distribution is used for all delay times and for most operating times. It was found that those times having an expected value close to zero were nearly always lognormally distributed. Hauling times are generated from a normal distribution, as is bunching time and merchantable height of each felled tree.

Of course, the user may very easily change this procedure should he find that his data behave differently from our Appalachian findings. The "CALLS" to LOGNOR may be changed to NORMAL and vice versa, because the inputs to each routine are the same. More details about the subroutines can be found in Naylor et al. (*1966*).

C. COMPUTER COSTS

Although T-H-A-T-S is a rather complex program, it is quite economical to use. For example, four separate runs were made with the data for our illustrative problem (parts of the output appear as figures 8, 9, 10, and 11); and the costs, using an IBM system/370 computer located on a university campus, were as follows:

- A 10-day run with status-printing only once during each day .... \$ 8.92
- A 10-day run with status-printing each hour 10.08
- A 1-day run with status-printing every 10 minutes 7.22
- A 1-day run with status-printing every minute 11.60

Obviously, computer processing charges vary from place to place; therefore, these figures should serve as a guide only. To assess the magnitude of computer costs, the user should run one or more of the versions used for the example problem. Then, based on the actual expense, he or she would be able to predict the costs of future runs in the following manner.

1. Run the example problem at your computer center.
2. Compute the variable costs of the run (VCC_o) = total costs − fixed costs. The fixed costs will normally be such items as charges for card input, channel time, printing messages, etc.
3. Compute the variable cost per machine-day (VCPMD) as follows:
   $VCPMD = VCC_o \div MD_o$
   Where: MD_o = The number of machine-days in the example problem.
   = 7 machines times 10 days.
   = 70
4. For any other computer run having the same printing frequency, estimate the costs as follows:
   a. Compute the appropriate number of machine-days for the new run.
   MD_n = Number of machines times expected number of days.
   b. Estimate variable costs for the new run.
   $VCC_n = MD_n$ times VCPMD
   c. Estimate total cost of the new run.
   Total cost = fixed costs + VCC_n