DTRAN

Version 1.0

A Multi-Market Timber Supply Model[1]

User's Guide

H. M. Hoganson and D. C. Kapple[2]

July 1991

STAFF PAPER SERIES NO. 82

[1] Research supported by both the University of Minnesota Center for Transportation Studies and The Minnesota Legislature as recommended by the Legislative Commission on Minnesota Resources from the Minnesota Future Resources Fund for the project "Simulating Minnesota's Future Timber Supply." Publication #19,759

[2] The authors are assistant professor Department of Forest Resources, College of Natural Resources, University of Minnesota stationed at the North Central Experiment Station, Grand Rapids, Minnesota, and research assistant, Department of Forest Resources, College of Natural Resources, University of Minnesota, St. Paul, MN 55108.

TABLE OF CONTENTS

ABOUT THE MANUAL

This manual is designed to facilitate the use of **DTRAN**, a forest management scheduling model designed to consider multiple market locations and associated marketing alternatives. This manual is not intended to teach forest management scheduling, timber supply analysis or forest planning. The authors assume that the users of this program have some expertise in these areas, as well as experience in using personal computers running under the DOS operating system environment.

The manual is broken down into several distinct sections that the user may consult as needed. The first section gives a general overview of the importance of transportation considerations in forestry, a brief review of forest management planning methods and a description of the concept of timber supply. The basic solution approach used by DTRAN is also described and compared to the DUALPLAN model for forest management scheduling.

The second section provides a more detailed description of important aspects of the problem including the basic method used to describe the transportation network and its impact on timber stand values. The third section describes briefly the linkages with two other modelling systems that have been developed in conjunction with the development of DTRAN. The DTRAN model is very data intensive as all information impacting timber supply model should be incorporated into the model. Sections four through seven are devoted entirely to the definition and structure of the input information. The main input file and an output file for a test case problem are shown in section eight.

```
6444444444444444444444444444444444444447
5 SOFTWARE DISCLAIMER FOR DTRAN        5
9444444444444444444444444444444444444448
```

All of the software on the DTRAN diskette has been extensively tested and checked for accuracy. To the best of our knowledge, it contains no errors. However, neither the University of Minnesota, the Department of Forest Resources, nor the authors provide any guarantees and are not responsible for errors that may arise during the use of this software. The authors ask that any errors found by users be brought to the authors attention in order to incorporate appropriate changes into future versions.

For permission to use or copy DTRAN and for revised versions or updates to existing DTRAN installations contact:

Dr. Howard Hoganson
University of Minnesota
Department of Forest Resources
North Central Experiment Station
1861 Highway 169 E.
Grand Rapids, MN 55744
(218)327-4490

I. GENERAL OVERVIEW

DTRAN is a computer model designed to help analyze the forest management situation in terms of the forest's ability to supply multiple product flows over time to different market locations. It was designed with the intent of examining differences in timber supply within a large region where specific product mill requirements for specific locations are important. It can be used to help examine a variety of state-wide or region-wide concerns ranging from specific expansion opportunities in a specific location to broad forest-wide policies influencing forest management practices. It is an extension of the DUALPLAN forest management scheduling model (Hoganson and Rose 1984, Hoganson and Rose 1989) that maintains optimality of the solutions generated yet adds the ability to recognize multiple market locations without the need to specify explicitly each potential shipping option for each harvest option. It is the combinatorial nature of problems with alternative market destinations that makes multiple market locations difficult to address using conventional timber harvest scheduling models based on linear programming, especially when multiple product outputs also need to be considered. Similar to DUALPLAN, DTRAN utilizes a basic understanding of the problem in its mathematical solution process for an optimal solution. Compared to DUALPLAN, the multiple-market considerations in DTRAN add significant complications, but the approach can still consider significant detail as the solution process decomposes the problem into parts. Final results include a set of marginal costs or "shadow prices" that can be used in an economic analysis of individual projects to recognize forest-wide objectives or concerns.

The Importance of Transport Costs

Costs of transporting wood to market are an extremely significant component of timber production costs. However, little recognition is given to transportation considerations in forest planning models used to help make strategic decisions on forest management and forestry-based economic development.

The forest industry is a large and growing industry in Minnesota. It is currently the second largest manufacturing industry employing over 54,000 people directly. It is an industry with annual shipments approaching $4 billion and expected to exceed $5 billion by 1995. For most timber harvested in Minnesota, the cost of transporting wood to the mill is greater than the cost of purchasing the standing timber! With the recent growth of the industry one can expect that wood transport costs will increase. Even without growth of the industry, an increase in transport costs would be expected as sites closest to the mills have generally been harvested first.

As important as they are, transport costs have not received adequate attention in strategic planning for forest resource use and development. Compared to other production processes, timber production is unique; extremely long production periods are needed, and the tree itself, is both the product and the factory. Many factors influence forest management decisions. Forests produce many outputs including a range of nontimber and timber products. Timber stands differ significantly in terms of growth rates, wood volumes, size, ownership and access. Computer models have become increasingly important for forest planning to explore the vast number of potential management options. The USDA Forest Service has been the major user of these computer models to date, but large industrial private landowners are also invested, and state and county level governments involved in forest management are at least interested. In Minnesota, efficient management of state and county forest land is especially important; only Alaska has more forest land managed by state and county government.

Computation time requirements associated with large linear programming formulations limit the realism that can be recognized in forest planning models currently in use. Recognizing transport options adds another dimension to a combinatorial problem that is already large. Recent planning efforts on each of the National Forests in the U.S. are a good example of both the use of computer models in strategic forest planning and the inadequate consideration given to transport costs. FORPLAN, a linear programming model serves as the basic analytical tool for developing forest plans (analyses) for each National Forest. Investments in this model have been large as each National Forest has spent on average over 1 million dollars on costs directly related to FORPLAN. Failure to recognize market location is one of the biggest criticisms of these analyses (Bailey, 1986).

By recognizing transportation considerations in strategic forest planning, cost-savings could be achieved through efficient transport of the raw material to the market. With the large forest industry present in Minnesota, the savings could be enormous. Furthermore, improvements in wood transport efficiencies would likely help support additional development of the forest industry, and taxpayers and other road and forest users would also benefit through reductions in road use and unnecessary road development.

Relation to Previous Research

Most models used for forest resource analysis have been either simulation models or linear programming (LP) models (Johnson and Scheurman, 1977). Many simplifying assumptions are almost always needed with both types of models. Simulation models have one advantage in that they can recognize an enormous amount of detail, but they only mimic the system; they do not "solve" the problem. Simulation models in the forestry literature are often referred to as binary search models because they are used to find the desired value of only one system variable. All other variables are assumed to be known. An LP model can recognize many more variables, but many more simplifying assumptions are needed to describe the forest. Usually the forest inventory data must be aggregated significantly before it is used in an LP model. Results often lack the desired specificity, and questions often develop about the possible errors due to oversimplifications.

Recently Hoganson and Rose (1984) developed a method for solving large forest management scheduling problems that incorporates specific concepts of forest economics into the mathematical methods used to solve LP models. It has many of the desirable characteristics of both simulation models and LP models. The method allows for much larger formulations because it uses duality theory to decompose the problem into a series of smaller problems and uses basic concepts of financial maturity to search for the values of the key dual variables that tie the small problems together. According to the latest edition of a widely-used forest management text (Davis and Johnson, 1987, p.669)

> Hoganson and Rose (1984) have presented a heuristic that allows the consideration of alternative management intensities and finds the optimal stand priority for harvest, overcoming two major drawbacks of binary search, while retaining a low cost per run and the ability to consider the inventory in great detail.... Hoganson and Rose's

> work clears away major criticisms aimed at heuristics like binary search. Now the best management intensity and harvest priority can be determined as part of the solution. The technique still retains the criticism that it has difficulties with constraints in addition to harvest amounts, but this pioneering work should breathe new life into the use of heuristics in harvest scheduling problems.

Further improvements in the modelling process seem plausible because the solution process is sequential in nature and thus the process can "learn" about the real world problem from the intermediate results and actually adjust the model formulation itself. In other words, one would not need to specify all possible shipping combinations in a model format initially.

The ability of this type of solution approach to greatly reduce computation time for forest management scheduling problems can at least be partially explained by two important factors. First, linear programming formulations for forest planning define a feasible region that is not as precise as it is expressed in the mathematical definition; optimal solutions that are "almost feasible" in a strict mathematical sense can be feasible solutions in a practical sense. Second, pure mathematical solution techniques do not use any understanding of the problem other than its mathematical representation in attempting to move towards optimality. Incorporating simple principles of forest economics like the basic concepts of procurement zones and optimal rotation age could allow for significant improvements, especially when one recognizes that maintaining strict feasibility is not necessary.

The Timber Supply Problem

Timber supply is an extremely complex concept that does not have a good simple definition. Most anyone concerned about timber supply will generally agree that a physical inventory of the forest is only a good starting point for analysis. An economist's definition of timber supply would read something like "the quantity of timber available to the market at various price levels during a specified period of time." From this definition it is clear that the physical inventory of the forest could be large yet timber supplied to the market could be low at prices considered acceptable by timber users.

The problem in applying the simple economic definition of timber supply is that many complicating factors enter the picture. Some of the complicating factors include: the multi-product/species nature of timber production, other forest uses that influence harvesting decisions, locational aspects of timber supply including the potential large transport costs and options to ship (supply) products from the same timber stand to different markets, and the time element associated with timber production with the opportunity to hold timber or manage more intensively if higher prices might be expected in the future.

A detailed analysis would be needed to obtain good estimates of a supply schedule showing the quantity of a specific timber product that will likely be available to a specific market during a specific time period. This analysis would need to take into account other markets, other products, a long time horizon and even the prices for this same product in this same market in future periods. If one could estimate prices for all relevant forest products in all relevant periods and markets, one would also still need to know how the quantity of this product consumed in this period of interest would influence all of those prices. Typically, one might expect significant interactions both across time and across space. Significant product interactions would also likely be present if significant volumes of this product are produced jointly as with mixed species timber stands. From this brief discussion, it should be evident that the supply of specific timber products to any single mill or market location is almost certainly very difficult to analyze without analyzing timber supply in a much larger regional context.

One way that timber supply can be analyzed in a regional context is by considering how the existing forest resource might best be utilized to meet demands for all products in all markets, both now and in the future. Specific price-quantity demand relationships could be used in this type of analysis, but it is simpler and perhaps more informative to keep the demand side of the situation somewhat removed and focus on the cost of production for meeting potential production targets over time. Without the need for prices as described in the demand relationships, this type of analysis is better suited for nonmarket outputs. Analyses can focus on the costs of production and the trade-offs in cost between alternative production targets. A key output for this type of analysis is the marginal cost of producing specific forest outputs as these costs can be used to help determine whether targets might be set too high or too low. This type of analysis does not necessarily lead to a good single estimate of the most-likely future condition, but it can help gain significant insight into the general timber supply situation.

This type of approach to timber supply analysis can be formulated as a linear programming problem, but model size can become a limiting factor. Options to recognize multiple market locations cause an enormous increase in the size of a linear programming formulation, especially when mixed-species stands and multiple timber products also need to be recognized. DTRAN is a model that determines optimal solutions to very large linear programming formulations of this timber supply problem with multiple market locations, but fortunately, the linear programming formulations never even need to be formulated explicitly.

DTRAN's Solution Approach

The solution method used by DTRAN is relatively easy to explain in general terms by relating the method to the scheduling process one would use for the unconstrained case with the objective of maximizing the net present value of the forest using an infinite planning horizon. For this situation one might not even think of the problem as a scheduling problem as the optimal schedule for each stand in the forest can be determined independently. Alternatives are selected for individual stands by estimating the net present value (NPV) of each alternative using an infinite planning horizon and selecting the alternative with the greatest estimated NPV (Davis and Johnson 1987). A key factor in the analysis is the value of the land for future rotations and the potential gain that might be made by harvesting the stand earlier so that returns from future rotations can also occur earlier. The analysis begins by estimating the value of the land for future rotations with this value dependent on the time at which the land becomes available for future production -- generally at the time of harvest unless some sort of seed tree harvest is applied. Once these "bare land" values are determined for all time periods, they are linked with estimates of the costs and returns for management alternatives for the existing stand to estimate the best management alternative for the stand.

The process followed by DTRAN is essentially the same with the only real complicating factor being the prices or values to use in the analysis. Because DTRAN is based on the assumption that the objective is to meet target production levels for all products in all periods, estimates of actual product prices in each market are not the relevant values to use for evaluating alternatives. The critical value is the marginal cost of production -- the cost of producing the most costly unit (last) each period. Basically, each management alternative is valued based on how much better it produces its outputs compared to production at the margin. In this framework, a negative NPV implies that the corresponding alternative is not desirable as the equivalent outputs can be produced even at the margin at a lower cost. Of course the margin changes and the values of alternatives change with different target levels.

Like its parent model DUALPLAN, the solution process used by DTRAN can be explained by seven basic steps:

1. Identify which forest products in each market have constrained product flows over time. For products with constrained output levels, estimate the desired output levels (target levels or goals) to be considered in this model run. For the products that are unconstrained, input assumed product values.

2. Estimate the marginal costs of production for the constrained products at the desired output level for each period.

3. Using the estimated marginal costs of production as prices, determine the optimal management strategy for each analysis area separately using the basic economic approach for determining the optimal management alternative.

4. Tally the product (and input) flows for the forest to determine the total production level for the forest based on the estimate of the marginal cost of production.

5. Compare the forest-wide outputs for the schedule developed to the desirable output levels to determine whether the schedule is acceptable. If acceptable, report the solution found and stop. Otherwise go to step 6.

6. Use the results of the last solution found and an understanding of the problem to re-estimate the marginal costs of production for each constrained product in each period.

7. Return to step 3 and repeat the scheduling process.

DTRAN differs significantly from DUALPLAN in how it would recognize multiple shipping options. With DUALPLAN, separate management options would need to be included for each possible shipping option. With many shipping options possible for even a "single management alternative," DUALPLAN is not well suited for this type of problem. DTRAN overcomes this problem by basically using a map to describe best shipping strategies. This map is based on the current set of marginal cost estimates for each market. As estimates of the prices in each market change, so do the map boundaries describing shipping strategies.

In terms of the steps described above, step 3 the time consuming aspect of both DTRAN and DUALPLAN. Although this step solves the same stand-level problems in both models, the spatial resolution offered by DTRAN adds significant complications. With DTRAN, prices are location dependent. Not only is a map needed to show boundaries, but a transportation network is also needed to define the map and describe how prices vary by location. Specific prices must be interpolated from the map for each stand during each iteration. Furthermore, because prices are sensitive to location with no simple way to class areas for prices when multiple products and multiple markets are involved, each stand in DTRAN is analyzed separately in terms of its "bare land" value. Compared to DUALPLAN, this significantly increases the computation time (size of the network problem). DUALPLAN assumes a classification scheme can be used to describe stand location and usually shares the "bare land" analysis between many analysis areas.

Similar to DUALPLAN, the marginal costs of production for the harvest level requirements are the focus (key dual variables) in DUALPLAN. The major objective with the modle itself is to develop good marginal cost considering all important aspects of the problem including the opportunity costs associated with harvesting and regenerating a stand in one period versus another period. Once the marginal costs (shadow prices) associated with the forest-wide concerns are adequately estimated, they can be used to help evaluate specific projects not even recognized in the model. They are a key link between forest-wide planning and individual project analysis.

II. THE SPECIFIC MODEL STRUCTURE

Management Inputs, Outputs and Flow Constraints

The total flows of inputs and outputs from the forest are assumed to be the sum of flows from each optimal prescription for each analysis area. Individual flows are defined by the time period of occurrence (TIC), item TYPE, and quantity of flow. Quantity of flows are expressed in flows per land unit -- generally flows per acre. To help speed processing and reduce the size of data files, all of these values are read as a small integer -- values must be less than 32,767. Units used to measure item TYPE flows need to be selected with this maximum value limitation in mind as well as any potential rounding error problems caused by using only integers. For example, rather than describe yields in cords per acre one might want to measure all quantities in tenths of cords per acre to reduce rounding errors.

Item TYPES are aggregated into groups -- item SETS -- for the purposes of summarizing forest flows or constraining flows over time. Item TYPES are also classified as to whether or not transport costs apply. For those TYPES for which transport costs apply, MARKETS also become relevant for the purpose of aggregating item TYPES into item SETS. The following table is an example of how item TYPES and MARKETS are aggregated into SETS. Table entries refer to the item SET for the corresponding item TYPE and MARKET.

Market	Item types - Transported			Item Types - No Transport		
	Aspen pulpwood	Birch pulpwood	Maple pulpwood	Mgmt Costs	Old Growth Acres	Hdwd Sawlogs
Duluth	2	5	5	-	-	-
I Falls	3	0	0	-	-	-
Sartell	4	6	0	-	-	-
"At the stand"	-	-	-	1	7	0

The example shown in the table has seven item SETS labelled 1 thru 7. Note that because of the need to recognize transport costs and specific markets, there are more SETS than item TYPES. Each SET can be constrained to target flow levels, but flows for SETS need not be constrained. Constraints cannot be applied to item TYPES directly so if it is desirable to constrain a single item TYPE, it must also be defined as a SET. In the example above, aspen pulpwood is not aggregated with any other item TYPE for any of the market locations. Specific constraints could thus be applied for aspen pulpwood for each market.

During the solution process DTRAN allows users to examine the flows for item SETS but not item TYPES. If constraints do not apply to an item TYPE, the only reason to define an item TYPE as an item SET would be the desire to view flows of an item TYPE during the solution process. Note that an item TYPE need not be related to any item SET. In the example above, Hardwood sawlogs are not related to any item SET -- zero is used as the SET identifier.

Several options are available for valuing item TYPES if flows do not contribute to the flows of any item SET. An item SET identifier of zero indicates that the value of this item TYPE is fixed at a constant value independent of the flow level and prices for any item TYPE or item SET. This price can be set to any level; the zero does not imply that the item TYPE has a price of zero.

Rather than fix prices for unconstrained item TYPES, prices for these item TYPES can also be made a function of the shadow price of an item SET. For example, one could have aspen sawlog prices as an unconstrained item type but make aspen sawlog prices a function of aspen pulpwood prices. This type of relationship is identified by assigning a negative value to the SET identifier with the absolute value of the identifier equal to the SET to which the price of the item TYPE (and MARKET if transport costs are recognized) is tied.

Analysis Areas, Management Options and Bare Land Classes

The basic unit of the forest for analysis by DTRAN is referred to as an analysis area. Each analysis area is assumed to have a specific location. Each analysis area need not be considered a homogeneous unit, but management alternatives or management prescriptions must refer to the entire analysis area. In the solution, DTRAN cannot split acres of any analysis area between prescriptions; all acres are assigned to one prescription. If flows from a single item type are a significant component of the flows from the forest as a whole, significant deviations from target levels are also likely in the final solution as changing the schedule for only one analysis area could have a significant impact on forest-wide flows.

DTRAN can use either a Model I or Model II (Johnson and Scheurman 1977) type of formulation with the Model II formulation preferable for most problems as it can reduce the number of alternative management prescriptions significantly. A Model II formulation decomposes the decision process for an individual stand into parts by considering each rotation separately. A Model II formulation will also aggregate analysis areas together after harvest if acres are then similar, but that is not possible with DTRAN as each analysis area is unique even after harvest as DTRAN recognizes the location of each analysis area and the impact location has on item TYPE values. Significant computation savings can still be gained from a Model II formulation even if analysis areas are not aggregated together.

DTRAN assumes that each analysis area can be classified as to its type of "bare land" after harvest. Bare land types refer only to the biological characteristics of the site. Generally, there should be a bare land class to describe how each analysis area would regenerate naturally. More intensive management options should also be represented by a bare land type. For example, "site index 70 red pine plantation" might be one type of bare land. Prescriptions for all bare land classes are stored permanently in memory by DTRAN. Each management prescription for each analysis area can be linked to one or more bare land classes. Costs of site conversion can be included in this linkage between initial management prescriptions and bare land classes.

In evaluating management prescriptions for each analysis area, an infinite planning horizon is used. DTRAN assumes all prices and costs are constant for periods beyond the end of the planning horizon. DTRAN allows users to define prices for periods beyond the end of the planning horizon as a weighted average of the shadow price estimates for earlier periods. This helps overcome problems that often occur when users must specify specific ending inventory values. High ending inventory values can cause models to hold timber through the planning horizon while low values can cause models to liquidate timber inventories during the planning horizon.

The solution process followed by DTRAN for an individual analysis area can be thought of as a simple network problem where the network is solved backwards through time. Initially, bare land prescriptions are found for periods beyond the end of the last planning period. Bare land values are then found for each period within the planning horizon by starting with the last period and linking single rotation alternatives for bare land with the bare land values found for later periods. Once bare land values are found for each period, prescriptions for the existing stand are evaluated and linked with the bare land values. Flows for the optimal prescription are then added to the forest totals including all flows from the best bare land alternatives. With a long planning horizon, several rotations of bare land prescriptions might apply with different optimal prescriptions for each rotation because of changing prices over time.

The Transportation Network

The road network is defined in terms of ROADS and POINTS. Each road is a single arc that has two a beginning point and an ending point with no branches or incoming roads associated with it. Points are simply intersections (nodes) where two or more roads come together. For roads, travel is assumed possible in both directions with no significant difference between the definition of the beginning point and ending point. For descriptive purposes, both points can be called end points. All markets are assumed to be located at a point (node). Procurement zone boundaries can occur anywhere along a road (arc) or at a point.

The value of timber located anywhere along the road network depends on how far the timber must be transported to market and the prices offered in the market. To select the market for shipment the important measure to compare is the net return from sales after transport costs are deducted. Of course net returns and best market destinations vary as location varies on the road network. For each product, each market will have a procurement zone surrounding it for which it is the best shipping destination. The boundary for these regions change as the delivered prices change in the different markets. Boundaries between procurement zones can occur anywhere along the road network.

The location of every analysis area is described by the nearest point on the road network and the distance (or transport cost) from that point to the analysis area. The nearest point on the road network is not a point (node) as the term point implies but a relative location along the road. For purposes of defining locations of analysis areas, all roads are assumed to be broken into a specific number of units (UNITSPEROAD). The number of units must be less that 32,767 so that the location can be input and stored as a small integer. The "distance" from analysis areas to the road network must be described in the same units as those used to describe distances or costs of travelling the length of each road. Generally, it would be favorable to use costs as this measure would take into account factors other than distance that influence transport costs.

Note that the distance to the road network refers to the road network recognized in the model and is not necessarily the distance from the analysis area to the nearest road. Distances to the road network likely involve significant travel on secondary roads not recognized explicitly in DTRAN.

Prices on the Road Network

Net returns per wood unit harvested will follow a definite relationship along each road in the road network. Net returns will always be highest at one of the end points. For any road, three cases are possible. To explain, let's consider any road with end points (nodes) N1 and N2. If it is most profitable to ship product P1 through N2 for all points along the road, then net returns will be highest for volumes located at N2 and lowest for volumes located at N1. The difference in net returns between nodes N2 and N1 would equal the cost of transporting a unit volume the length of the road, and net returns for any location along the road would equal the net return for node N2 less the cost of transporting a unit volume to node N2. Likewise, the reverse is true if it is most profitable to ship through point N1 for all points on the road. For the case where a procurement zone boundary is along the road, the situation is complicated only slightly. At the procurement zone boundary, one is indifferent between shipping through node N1 and N2 as this is a condition that makes it a boundary point. The boundary is also the location along the road with the lowest net return. Returns increase as one moves away from the boundary at the rate in which transport costs vary along the road.

The situation with the lowest net return at some midpoint along the road does not necessarily mean that a procurement zone boundary must be at that low point. Situations can exist where the low-cost route to a single market location can vary depending on where one is located along the road -- "if near node N1 take route through N1, if near node N2, take route through N2." This type of situation is easily identified as the net return for the two end point locations differ by less than the cost of transporting the product the length of the road.

The net returns for products "on board" the road network, anywhere on the network (arcs), can easily be estimated if the low point location and net return is known for each road. In terms of the net return at the low point, the net return for any location is simply the net return at the low point plus the cost of transporting the product from the point in question to the low point. Adding this value to the net return at the low point can be thought of as a credit added for not needing to ship all the way from the low point. The market destination for any point is simply the same market destination for the road ending node on the same side of the road low point.

The steps followed by DTRAN to link specific markets with the road network can be summarized by three main steps:

1. Determine all the low-cost routes and associated costs for shipping products from each node to each market. Save this "shortest path" information as it does not change as prices in each market change. This step needs to be done only once and results can be utilized for other forest

management scheduling applications as long as new market locations or new roads are not added to the road network.

2. For each node in the road network, use the shortest path information found in step 1 and the current set of market prices for each period to determine, for each product type and period, both the maximum net return and the corresponding market destination for each node on the road network.

3. For each road in the road network, use the information describing the net returns for each node and the corresponding market destinations to determine the point along each road with the lowest net return. The location of the low point will likely be different for each product type and period. For each product type and period, save the low point location and the corresponding net return per unit shipped.

Steps 2 and 3 in the above process need to be repeated every time price (marginal cost) estimates change in the solution process. Stand-level or Analysis area information is linked to the road network through an identifier that describes the distance or cost required to transport products to the road network. This additional cost could include significant travel on a road system not recognized directly in the model. The road network in the scheduling model serves two primary purposes: (1) it is the basic identifier used to define procurement zone boundaries, and (2) it is the basis identifier for recognizing regional differences in prices. For these two purposes and with the ability to add additional transportation costs to bring products to the main road network, it is likely that the significant detail is not needed in the road network used.

Options to Reduce Model Size

The size of the road network used can have a significant impact on the memory requirements needed by DTRAN. For each road in the network, the model, determines the "price" or net return per unit shipped for the low point along the road for every product in every period. It also determines the "best" market destination for each node in the network for every period. For a problem with 500 nodes, 800 roads, 20 products (item types) and 5 periods, this would result in 80,000 low "prices" for roads and 50,000 "best" market destinations. Just storing each price and road "low point" location as a single precision real number would require 640,000 bytes of memory.

To help minimize storage requirements, DTRAN utilizes two features. First, it breaks item types into two classes: those requiring transport costs and those for which transport costs do not apply. Additional item types not associated with transportation costs add very little to the size of the model as prices do not vary by location.

Determining whether to recognize transportation costs for each product might not be as simple a decision as one might think initially. Certainly transportation costs could be over-estimated if one aggregates market locations to help minimize the number of markets needed in the model. In some regions where many small sawmills are present, it might be most appropriate to not even try to model transportation considerations. Constraints can still be applied to flows associated with products not transported with shadow prices representing costs before any transport costs are incurred.

To help reduce potential model size problems resulting from the size of the road network, DTRAN has the ability to process analysis areas region-by-region. Rather than store information about all road prices and low points, DTRAN decomposes the road network into parts and processes analysis areas in a region-by-region format. Technically, each BLOAD file containing the basic prescription information could be a separate region. Generally, one might consider using each county or survey region as a different region. Regional boundaries are not a consideration as regions act only as a means for organizing the solution process so that only those roads associated with a region need to be considered at that time. To modify an application to utilize regions for analysis all one needs to do is change the value of NREGION on the primary DTRAN input file and edit the master file list to identify the region corresponding with each file name. Computation time will be minimized if the order of the BLOAD files as listed on the master file list (AAMASTERFN$) are such that all files for each region are grouped together.

DTRAN reads all BLOAD files once initially to determine which roads and points are needed for each region. If the analysis areas associated with BLOAD files have not been sorted such that they all apply to the same general region, then the use of "regions" in DTRAN will not likely improve processing efficiency or memory requirements as each region would still likely need a large road network.

Generally, variable names and input parameters that relate to the entire road network use the terms ROADS and POINTS. The terms NODES and ARCS are used to refer to the portion of the network needed in memory at any one time.

Linking Survey Plots with Permanent plots

The Minnesota statewide forest inventory uses a generally accepted and applied two-phase sampling method combining photo plots to estimate the area of each forest type-size-density class and ground plots with numerous measurements of specific tree level characteristics to help estimate inventory volumes and future growth rates. In the past, the measured ground plots have generally serves as the basis for modelling regional timber supply.

DTRAN has the ability to link the sampled forest ground plots with the photo plots to help improve the description of the spatial distribution of the forest. Photo plots from forest survey likely contain valuable information describing the spatial distribution of the forest. With DTRAN, the ground plots can serve as the basic biological unit for which prescriptions are written. Each ground plot can then be linked with a number of photo plots. Each photo plot can be analyzed separately with the optimal prescription likely to vary between photo plots.

More research is needed to determine the most appropriate method for assigning photo plots with ground plots. Criteria for linkages could range from simple factors like distance from ground plot to factors like distance to the nearest road or distance to the nearest market. DTRAN could be used to compare the impact of alternative mapping strategies and help identify which approach might be most appropriate.

Estimating Shadow Prices

After determining the optimal management prescription and summing flows from the forest as a whole, DTRAN compares the flows with the target flows to determine if prices need to be re-estimated and the process repeated. For making this determination, DTRAN uses tolerance levels (acceptable percentage deviations) as specified by the user on the model formulation input file. This process of re-estimating shadow prices and determining the corresponding optimal management schedule (an iteration) continues until a schedule is found with all flows within acceptable tolerance limits or until the user-specified limit is reached as to the maximum number of iterations to perform. A brief status summary is shown on the screen indicating the average percent deviation for the latest iterations and the number of unacceptable constraints (flows).

To adjust shadow price estimates, DTRAN uses the same basic procedures as is used in the DUALPLAN model (Hoganson and Rose 1989). Each procedure compares the desired or target flows for each constrained item SET with the flows that resulted using the latest shadow price estimates. Each approach calculates an average deviation for each period with two of the techniques considering not only the deviation from the target flow level for the period of interest but also the deviations for other periods as well. The size of this average deviation for each period is used to estimate the size of the adjustment to make in the shadow price estimate. Rather than assume that the price adjustment is directly proportional to this deviation, the user defines the size of the adjustment as a piecewise linear function. Typically, one uses this piecewise relationship to greatly reduce the slope of the relationship for large deviations.

In defining each adjustment procedure, the user inputs specific price adjustment levels for specific average deviations (BREAK POINTS) and the model simply interpolates the adjustments for average deviations between these points. In general, it is important to recognize that the shadow prices for products will most likely be serially correlated as several rotation ages are possible for nearly all stands; any large price change between successive periods would likely result in a significant shift in timber production either into or out of that period depending on whether the shift in price is an increase or a decrease. As flows of nontimber outputs are also likely tied to the age structure of the forest, shadow prices for them are also likely to be quite serially correlated. Prices that are serially correlated are generally much easier to estimate as one is essentially searching for one curve rather than a separate price for each period.

The direction of price changes does not depend on whether an item SET is a benefit or a cost. Prices are always lowered (raised) if flows are above (below) the desirable level. Benefits are distinguished from costs only in that costs generally have negative values. Lowering the value of a cost actually makes it more costly thus management prescriptions with those costs are less likely to be optimal prescriptions.

The float procedure uses the average deviation for the item SET over the entire planning horizon. Prices for each period are all raised or lowered the same amount depending on the size of the average deviation. In terms of a graph of the shadow prices over time, one is simply "floating" the entire curve up or down. This adjustment process is most appropriate in the early stages of the search process in an effort to move prices to the general price level associated with the desirable flow levels specified. Large adjustments can likely be used to get prices in the general range where constraints will be satisfied. If the shape of the "prices over time" curve is quite different from the optimal solution -- for example, prices should generally rise over time but the estimates fall over time -- then float iterations alone will likely make very little progress.

The intent with the float iteration procedure is to help find the marginal production area. As prices are raised, it becomes profitable to manage more timber stands and intensify management. Raising or lowering prices the same amount for all periods can also influence the timing of rotations, but such changes are likely to be minimal as compared to the other adjustment procedures. The float procedure also has an automatic cut-off level suggesting more progress can be made using an another adjustment procedure.

The objective of the shape adjustment procedure is to improve the shape of the price curve when plotted against time. It estimates an average deviation for each period by using the deviation for the corresponding period and several neighboring periods as well. If deviations are generally all positive or all negative for several consecutive periods, the shape of the price curve likely needs to be changed for that region. The weights and number of periods to consider in estimating the average percent deviation is input by the user. An average deviation of, say 10 percent, is much more significant if it represents an average over several time periods. It is unlikely that this type of deviation could be overcome by simply shifting harvests from neighboring periods as neighboring periods also tend to have the same problem as those periods were also used in estimating the average deviation.

The smooth procedure uses only the period of interest in estimating the "average deviation." The intent is to fine tune the price-over-time curve to help eliminate large deviations associated with single periods. The primary intent is to shift flows to neighboring periods by raising or lowering prices in periods with large deviations more than periods with small deviations. To shift flows from period to period by adjusting prices, the key issue is the percentage change in stumpage price over time. Stumpage prices are not equivalent to the shadow prices as shadow prices also include costs for harvesting and transport. In general, price adjustments for a given deviation should probably be significantly smaller with the smooth adjustment process than with the other adjustment processes as large deviations for single periods will likely be common in initial stages of the search process.

For calculating "average deviations", the user has the 3 options for "discounting" item flows. The reason for discounting flows is to recognize that volumes grow over time. For example, when trying to shift flows to later neighboring periods, some additional growth would likely take place. Similarly, when trying to move flows to earlier periods, it is likely that equivalent volumes are not moved simply by changing harvest timings as those stands likely added some growth. At least in theory, this factor could be important for both the float and shape adjustment procedures. The 3 options associated with discounting of flows are: (1)no discounting, (2) discount all flows using the general interest rate, and (3) discount all flows using an estimate of the growth rate for stands likely to shift from one period to the next. The third option requires an estimate of the average processing cost. This process cost (per flow unit basis) represents all costs that are incurred in harvesting and transporting the item SET. It is subtracted from the current shadow prices to estimate stumpage price for the average stand. The discount rate used for option (3) is estimated as the growth rate necessary to make one indifferent between harvesting at the current stumpage price or waiting until the next period to obtain that stumpage price. A different rate is estimated for every item SET and TIC.

Land costs are ignored in the estimates. The marginal growth rate will vary significantly over time if prices follow both periods of increasing and decreasing prices. This process of discounting can thus recognize the simple fact that stands likely to shift in the time of harvest are likely to be growing at significantly different rates depending on whether prices are increasing or decreasing.

In general, the adjustment process cannot be expected to move rapidly to the optimal solution in just a few iterations. Attempting to fine tune the adjustment process using a discounting procedure is probably not that important. This aspect of the model has received no significant testing to determine the efficiency of one approach over the other. The potential gain from using these discounting procedures are likely the greatest for the float procedure.

The user has considerable flexibility in controlling the type of price adjustment process to use each iteration. At any time the user can interrupt the solution process by hitting the "hot key" and manually select the type of price adjustment (iteration) to perform in the next iteration. All parameters for all of the iteration procedures can also be edited after any iteration. The user also has the ability to back-up the solution process and start over using a specified earlier iteration if it is felt that those shadow prices are better starting points than the current solution. Manual price adjustments are also possible if the user would like to examine a specific set of prices without restarting the entire search process.

The first iteration in the solution process uses a set of initial price estimates that are on the main input file describing the model formulation. Price estimates on this file can either be those input by the user or those written as output in an earlier run -- DTRAN has an option to save the current set of prices (create new input file) for later use in a later run. This option is also useful for saving the optimal set of prices once the solution process is completed.

The automated price adjustment process is to first perform a series of float iterations, the number is specified by the user (NFLOAT). In each float iteration the prices are adjusted for an item SET only if the "average deviation" for that group is above a level specified by the user (STOPFLOATPC). If none of the item SETS have average deviations above this level, then the adjustment process moves on to the next adjustment method. After the NFLOAT float iterations, NSMTHFLT smooth iterations are performed. This is then followed by a nested loop of iterations where the entire loop is completed NSHAPE times. Within this loop one shape iteration is first performed followed by NSMTHSHP smooth iterations.

Setting the level of price adjustments for each price adjustment process is a key aspect of the modelling process. The DUALPLAN manual (Hoganson and Rose 1989) offer some guidelines for setting adjustment levels for each procedure. The following paragraphs summarize some of those guidelines and suggests several additional considerations related to problems with multiple market locations.

A basic understanding of the actual costs involved and the likely range of those costs over the forest can help in estimating initial price (marginal cost) estimates and in setting price adjustment parameters. Generally users should try to avoid the tendency to jump right in and attempt to solve a specific formulation very rapidly using large price changes each iteration. Before attempting to satisfy any specific flow goals, it is suggested that some general price levels be explored. This can be accomplished relatively easily by using just the float procedure with adjustments for all deviation levels set to a single price increment that might be relatively large, at least at first. One might want to hold the value of all but one item SET constant and float the price through a range of price levels.

Price changes for average percent deviations for the float and shape procedures should be larger than the price changes for the smooth procedure. Unless computation time per iteration is a major concern, it is probably best to keep the price changes small per iteration as then it is more likely that the solution will move toward the optimal solution. Increasing the number of smooth iterations might also help if one is concerned that prices might "jump" significantly far in the wrong direction and have difficulty self-correcting. This kind of jump will likely occur if the parameters for the price adjustments are set extremely high for any of the adjustment procedures causing the process to overshoot the target. For example, with too large parameters, the percent deviations for a single period might go from -20 to +30 to -40 to +50; in other words, the adjustments are overcompensating to the extent that they cause an even larger deviation with opposite sign.

The change in a specific flow level between iterations is likely to exhibit some "lumpiness" because all acres within each analysis area are assigned to one prescription. In other words, a small change in prices might show essentially no change in flow and then another small price change results in a large shift in flows. It is important to get a feel as to how large a change occurs in the solution by shifting a final harvest of one "average" analysis area from one TIC to another TIC. This information should not be hard to determine outside the model.

The piecewise linear relationship (absolute price change plotted against absolute average percent deviation with price change on the y-axis) should probably have a sigmoid shape for all three adjustment procedures. This results in very small adjustments for periods with flows close to the desired level and adjustments that do not differ significantly for deviations with "large" deviations.

A key factor to consider in determining when to stop the iteration process should be when shadow prices have stabilized with signs of deviations fluctuating from positive to negative. Deviations might still seem relatively large, but this is likely due to the fact that flows from a single analysis areas are a relatively large component of the flow or that some analysis areas are nearly identical with specific price levels and price changes influencing them in the same manner. In general, one can probably learn more about the forest through multiple runs than trying to fine tune the solution for one set of flow goals for the item SETS. In trying to find "optimal" solutions, estimating shadow prices to the nearest $.01 is probably not a major concern. Even if deviations seem large, they are unlikely very significant if small price changes can cause them to essentially reverse in sign. It is important to put the modelling process in context with the entire planning process.

With multiple markets, several practical considerations might help in the adjustment process. First it is important to realize that products are now being allocated both across time and across space. The cost of transporting products is the key to understanding the impact of price changes on the size of procurement zones. For example, if market A has a price that is $2.00 greater than market B and it costs .20 per mile to transport the product, then the procurement zone boundary between market A and Market B would be defined by points that are 10 miles closer to market B than to market A. Now suppose the price in market A was lowered by $.50 and the price in market B was raised by $.50 making the price differential between the two markets equal to $1.00. This would change the procurement zone boundary by 5 miles, probably too large a change for a single iteration unless 5 miles is very small compared to the overall size of the procurement zone.

The shape iteration is designed primarily with the intent of considering whether prices are too high or too low for a specific time period. Certainly this procedure can still be of value with multiple market locations, but it is important to recognize that procurement zone boundaries can be influenced significantly with large price changes. With multiple market locations, one should probably keep price change with the shape adjustment procedure closer to those of the smooth procedure so that any over-adjustments with the shape procedure can likely be overcome by successive smooth iterations.

Similar considerations apply to the adjustment levels for the float procedure. In general, one must be careful not to cause major shifts in the procurement zone boundaries between successive iterations. Although applications of DTRAN to date have been limited, results have been excellent for the cases considered. As a result, improvements in the price adjustment process have not been a major priority, but several enhancements seem plausible. First, an adjustment procedure similar to the shape procedure could be used, but rather than base averages on just deviations for neighboring periods for a single market, neighboring markets could also be used. A similar option might raise all market prices for a given item TYPE by the same amount. Potentially this option could be structured such that procurement zone boundaries would not change at all thus eliminating the need to redefine boundaries after this adjustment. Similar modifications might also be desirable for both the Smooth and Float adjustment procedures.

Nontimber Outputs

Although DTRAN was designed primarily with the intent of improving the ability to recognize timber transport costs, it has a number of features that are useful for considering other forest management concerns as well. First, it is important to realize that item TYPES, used to describe flows of forest inputs and outputs, are not limited to timber volumes and dollar costs. Item TYPES could include grouse, recreation user days, acres of old growth or anything else. Of course the production of many nontimber outputs are complicated and have significant interactions between large blocks of the forest. But even those interactions might be possible to address to some extent through outputs describing habitat requirements.

DTRAN also has the ability to recognize values of nontimber uses directly in the management prescriptions. For example if users could estimate the reservation price or net return necessary for private landowners to sell their timber, that minimum return can be recognized in the model. For forest survey ground plots represented by multiple locations, these reservation prices can also vary by location. Similarly, each management prescription for a forest survey ground plot can also have a nontimber value associated with it that varies by location. Reservation prices can also be included to describe minimum returns necessary to keep analysis areas in timber production.

Model Outputs

DTRAN has some basic output options to summarize specific solutions and/or to identify the optimal prescription for each stand. DTRAN uses simple switches to turn on and off the option to print any or all of three table options after each iteration. One table reports the prices and flows for each item

SET. Its' switch is WRITESET. SImilarly, WRITETYPE is a variable switch for printing prices and flows for each item Type. The third optional table compares flows with targets for each item SET and is is controlled by WRIECNSTRNT. Specific details for these switches are given in the description of the Model Formulation File (block 1, group 7).

Each switch be turned on or off using the editor during the solution process. It is not recommended to leave all of these switches on during the entire iteration process as the amount of output will be large if many iterations are needed. Unless one is interested in studying the iteration process itself, a good approach is to leave these switches off until the iteration process is completed and then turn the switches on and run one extra iteration without adjusting the shadow price estimates.

DTRAN also has the option to print a simple flat file that identifies the optimal prescription for each analysis area. This file can easily be linked with most any data base system that is used to describe the analysis areas. Values printed for each analysis area include: the net present value (infinite rotation) of the stand, the optimal first rotation; the time of each regeneration within the planning horizon; the optimal time and prescription for all regeneration options (rotations); and the position within the large array where information on the optimal first rotation is stored. The option to print this file can be selected from the list of menu items following the completion of the selected number of iterations.

Besides the specific output files created, DTRAN offers all of the many options for viewing the current results as offered by DUALPLAN. Several addional features are offered as well. Most notable is a mapping network that relates the procurement zones to the existing road network. Specific descriptions of these maps can be found in the GISTRAN users guide (Kapple and Hoganson 1991).

III. LINKAGES WITH OTHER MODELS

Several other models developed by the University of Minnesota Department of Forest Resources have been specifically designed to be linked with DTRAN. RxWRITE ia a new program that can serve as a "prescription writer" or "alternative generator" (McDill 1991). This program uses the standard U.S. Forest Service statewide forest survey ground plot information to develop possible management alternatives. The basic tree list information from the forest survey is utilized thus overcoming some potential problems caused by data aggregation. RxWRITE uses the same growth projection routine as used by the STEMS growth projection system (USDA Forest Service 1979) to predict yields in future periods. RxWRITE is extremely flexible in its ability to define possible management alternatives, harvest systems, associated harvesting costs, product utilization standards, and species groups. It is a very detailed and user-friendly system which has taken considerable time and effort to develop.

Considerable effort has been made to develop software to help link DTRAN with basic spatial data describing the existing transportation network and the spatial distribution of the forest survey plots in relation to the transportation network. A basic GIS system, GISTRAN, has been developed with major components actually incorporated directly into the DTRAN model so that the road network and procurement zones for each market can be viewed graphically at intermediate stages (iterations) of the DTRAN solution process (Kapple and Hoganson 1991). GISTRAN can be used to generate the necessary transportation related information and input files for input into DTRAN, including the shortest route information for all nodes in the road network and the location identifiers for each basic forest survey plot (analysis area). Mapping routines allow the user to examine the entire study area, a region of the study area, or a particular county in the study area.

IV. THE MODEL FORMULATION FILE

The Model Formulation File contains the information that defines the forest-wide problem to be solved. The file has been structured such that comment lines are included to identify the variables. An example input file is shown in section VIII of this users guide. Many of the parameters can be edited using the edit options within DTRAN. For cases where many changes are desired, an outside editor would probably work best as most editing options in DTRAN change only one value at a time. DTRAN has an option to rewrite this input file at any stage of the solution process. Rewriting the input file is a good way to store the final solution along with all the assumptions that were made for the formulation.

Listed below are definitions describing the specific format for the regeneration prescription file. The numbering system is "block number - group number - entry number." Each group number must begin on a new line in the file. On the input file each block and each group within the block begins on a new row. Some groups contain variable length arrays that stretch out over several lines of input.

ID #	Name	Description
1-1-1	RGPRESFN$	The name of the file that contains all of the prescription information for regeneration after the "bare land" state is reached.
1-2-1	AAMASTERFN$	The name of the file that lists all the names of the BLOAD files containing the location and prescription information for existing stands. This file is created automatically by the program used to write the BLOAD files.
1-3-1	MARKETDATFN$	The name of the file that contains information describing each market location.
1-4-1	COSTROUTEFN$	Name of the file containing the low-cost estimates for all road node-to-market shipping combinations.
1-5-1	COSTROADFN$	Name of the file containing the estimates of the "cost" to travel the length of each road (arc) in the road network.
1-6-1	PATHNETDB$	Name of the DOS path (subdirectory) that contains the data base files describing the road network.
1-7-1	WRITECNSTRNT	If greater than zero then a summary of the constraint achievement levels will be written on the output file after each iteration. Note that this option can be turned on interactively.
1-7-2	WRITESET	If greater than zero then a summary of the flows for the product sets will be written to the output file after each

		iteration. Note that this option can be turned on interactively.
1-7-3	WRITETYPE	If greater than zero then a summary of the specific item flows will be written after each iteration. Note that this option can be turned on interactively.
1-8-1	NTIC	The number of TICs (periods) in the planning horizon.
1-8-2	DCRATE!	The discount rate, expressed as a decimal, for one full period (TIC).
1-8-3	NTYPE	The number of specific items recognized, whether constrained or unconstrained. Note that items can be costs or inputs as well as outputs. If both product size classes and species groups are to be recognized, then each relevant combination must be defined as a separate item type, i.e., aspen pulpwood, pine sawlogs, etc.
1-8-4	NTYPESHIP	The number of item types for which ship-to-market costs will be considered. Management costs are an example of a possible item type for which transport costs are not relevant. All ship-to-market costs are assumed to be directly proportional to the quantity of each item shipped.
1-8-5	NSET	The number of item SETS. Flow constraints can only be applied to item SETS. A set can be a very large aggregate (total) for many market and item type combinations or it can be simply a single item type and market combination. The array ITYPEMARKSET (see 2a1-2-1) is used to define all sets. The maximum number of item sets possible is: NTYPE - NTYPESHIP + (NTYPESHIP*NMARKET)
1-8-6	NMARKET	The number of markets recognized.
1-8-7	NSAVE	The number of prior iterations to retain in memory. This information can be examined interactively and used again as a basis for price adjustments if it is desirable to "back-up" in the solution process and reset some parameters. This value could be reduced in size to help reduce the amount of memory required. Several arrays are dimensioned NSAVE by NSET by NTIC.

1-9-1	NREGBIO	The number of biological land classes identified for classifying analysis areas as "bare land" for valuing future rotations.
1-9-2	MXREGROT	The maximum rotation length possible for any regeneration alternative. This value is expressed in TICs and can be overestimated; however, storage requirements and computation time will increase as this value is increased.
1-9-3	MXTICAAPRES	The last period (TIC) referenced by any prescription for any analysis area. Although a specific planning horizon is defined by NTIC for summing item flows, the model uses an infinite planning horizon for valuing management alternatives. MXTICAAPRES is used to make sure that the time dimension of arrays are dimensioned large enough and that all relevant prices are initialized.
1-9-4	MXPRES	The maximum number of prescriptions considered for any specific analysis area.
1-9-5	MXSTAND	The maximum number of locations for a single analysis area. If photo plots are being used as location estimates for each ground plot, then MXSTAND is the maximum number of photo plots referenced by any ground plot.
1-9-6	NREGION	The number of regions recognized in the model. Regions can be used in DTRAN to help overcome model size problems that would develop if a large transportation network is used. DTRAN allows users to group analysis areas into regions and then use only the road network that applies to that region when analyzing alternatives. No simplifying assumptions are made by utilizing the "regions" option. All analysis areas on each BLOAD file must apply to only one region as identified on the master file list (1-2-1 AAMASTERFN$). Computation time will be reduced if the master file list is ordered such that all BLOAD files for a given region are grouped together so that the number of re-initializations of the road network in memory are minimized.
1-10-1	NPOINT	For the entire forest or study area as a whole, the maximum number of points (nodes) recognized in the road network. DTRAN distinguishes points from nodes by using points to refer to the entire network and nodes to refer to the portion of the network (region) currently being used.

1-10-2	NROAD	For the entire forest or study area as a whole, the maximum number of roads (arcs) recognized in the road network. Roads are defined by a beginning point (node) and an ending point (node). Each analysis area is assigned to a specific road. DTRAN distinguishes roads from arcs by using roads to refer to the entire network and arcs to refer to the portion of the network (region) currently being used.
1-10-3	MXARC	The maximum number of arcs (roads) in any region. If any portion of a road is in a region, then that road is considered in that region. There is no limit as to the number of regions to which a single road can apply. DTRAN automatically assigns roads to regions by linking all roads on each BLOAD file with the regional identifier for that BLOAD file as identified on the master file list AAMASTERFN$ (1-2-1).
1-10-4	MXNODE	The maximum number of nodes (points) in any region. If any portion of a road is a region, then both the beginning point and ending point for that road is considered in that region. Similar to roads, there is no limit as to the number of regions to which a single point can apply. DTRAN automatically assigns points to regions by linking all roads on each BLOAD file with the regional identifier for that BLOAD file as identified on the master file list AAMASTERFN$ (1-2-1).
1-10-5	UNITSPERROAD	The number of unit lengths in which each road is divided for the purpose of linking each analysis area with a location on the nearest road. This value must be a positive integer less than 32,767. The value of UNITSPERROAD must be THE SAME as that used when developing the management prescriptions. An integer value is used as all information describing management prescriptions is loaded as a 2 byte integer.
1-10-6	ACCESSUNITS!	A conversion factor that applies to the travel time (or distance or cost) as measured by ACCESSVAR!, the distance or cost necessary to transport products to the nearest point on the road network (see more detailed description of ACCESSVAR! in the next section). It is the value that one must divide by to convert ACCESSVAR! to the units used in defining the road network. To help clarify, consider the following example. Assume the measure used to describe the road network is cost per cord

transported and the stand of interest has a \$1.4 cost per cord to reach the road network. (Note that this could involve a significant travel distance on secondary roads). Without the option for using a conversion factor, the prescription file would round this cost to \$1 per cord. But by setting accessunits to 10, the cost of transporting 10 cords, \$14, would be written on the prescription file, and DTRAN then divides by ACCESSUNITS! to convert the units back to the units used to measure transport costs on the road network.

1-11-1	MXCNSTRNT	The maximum number of constraints on flows for the item sets. Potentially there could be as many as NTIC times NSET constraints. This value can be set larger than the maximum number of constraints considered, but doing so will increase memory requirements unnecessarily.
1-11-2	TOTRGPRES	The total number of regeneration prescriptions totalled over all bare land types (NREGBIO). This value is used to dimension arrays. It can be set larger than the dimension needed. DTRAN will print a message identifying the potential for reducing this value if it is set significantly larger than necessary.
1-11-3	TOTRGITEM	The total number of item flows from all recognized regeneration prescriptions totalled over all bare land types (NREGBIO). This value is used to dimension arrays. It can be set larger than the dimension needed and DTRAN will print a message identifying the potential for reducing this value if it is set significantly larger than necessary.
1-12-1	NFLOAT	The number of float iterations to perform in the sequence of automatic price adjustments. (see Price adjustments section)
1-12-2	NSMTHFLT	The number of smooth iterations to perform after the float iterations in the sequence of automatic price adjustments. (see Price adjustments section)
1-12-3	NSHAPE	The number of shape iterations to perform in the sequence of automatic price adjustments before starting the process again with a series of float iterations. (see Price adjustments section)

1-12-4	NSMTHSHP	The number of smooth iterations to perform after each shape iteration in the sequence of automatic price adjustments. (see Price adjustments section)
1-12-5	MXBREAK	The maximum number of break points used in defining the price adjustments with any of the price adjustment methods.

******** Block 2 is repeated once for each item type. The format depends on the value of NEEDTOSHIP, the variable used to identify whether transport costs apply to the item type. Input requirements also depend on the value of ITYPEMARKSET.

2-1-1	ZTYPE$	A one-line label for the item type.
2-2-1	NEEDTOSHIP	An identifier used to signal whether transport costs apply (or will be considered) for this item type. If NEEDTOSHIP(JTYPE) is greater than zero then transport costs are considered. The value of NEEDTOSHIP influences the format of the input for this block as identified below.

*** If NEEDTOSHIP is greater than zero then block 2a is followed. Otherwise Block 2b is followed.

******** Block 2a applies to items for which transport costs are recognized.

2a-1-1	TRANCOEF!	The conversion factor for converting road distances or road travel costs as used in describing the road network to costs per unit of item TYPE shipped. For example, if roads are measured in miles, then TRANCOEF! is the cost per item TYPE unit per mile shipped. If road "distances" are in cost per cord transported and this item TYPE is measured in cubic feet, then TRANCOEF! is the factor that converts cost per cord to cost per cubic foot.

******** BLOCK 2a1 IS REPEATED ONCE FOR EACH MARKET LOCATION

2a1-1-1	Marketlabel$	A one-line market label -- used only as a placeholder to help make the input file readable to the user
2a1-2-1	ITYPEMARKSET	The variable used to link item TYPE and MARKET combinations to a SET as used to define flow constraints for each period. If ITYPEMARKSET is:

> 0 then	ITYPEMARKETSET is the SET to which flows of this TYPE and MARKET combination will be linked.

= 0 then	Prices for this TYPE at this MARKET are assumed to be fixed and independent of the level of flows and prices in other markets. The option to use if this MARKET will not accept this item TYPE.
< 0 then	quantity flows for this ITEM to this MARKET are not constrained but prices for this ITEM in this MARKET are a function of the price of ITEM (-ITYPEMARKSET) in a market defined on the next input line by IFIXMARKET.

The next lines of block 2a depend on whether ITYPEMARKSET is greater than, less than or equal to zero. Only ONE of the three applies.

******** Sub-block 2a1> -- Input these values if ITYPEMARKSET is > 0

2a1>-1-1	UNITCONV!	The conversion factor to change units of this item TYPE to the units used for the corresponding SET for this item TYPE and MARKET combination. This value is multiplied by the quantity of the TYPE to determine quantity in terms of the units used to measure the SET.
2a1>-1-2	TYPEMARKINT!	Prices for this ITEM and Market combination are assumed to be a function of the shadow price (or marginal cost) for the corresponding SET. This relationship is simply: P(jTYPE) = a + b * P(jSET) TYPEMARKINT! is the "a" term in this relationship. One could use this variable to build in assumptions like: birch pulp is worth $5 more per cord than maple pulpwood at mill k; if the SET prices are in terms of maple pulpwood, then "a" would equal 5 for this example.
2a1>-1-3	TYPEMARKSLOPE!	As described in the definition for the previous input value (TYPEMARKINT!), per unit prices for this ITEM-MARKET combination are assumed to be a function of the shadow price for the corresponding SET. This relationship is: P(jTYPE) = a + b * P(jSET) TYPEMARKSLOPE is the "b" term in this relationship.

It needs to recognize any differences in the units used to measure the item TYPE and its corresponding SET (as defined by ITYPEMARKSET). For example, if the conversion factor to convert TYPE units to SET units is 20 (as defined by UNITCONV! on this same line of input), and the SET price represents this item TYPE with no other price assumptions, then ITYPEMARKSLOPE! ("b") must also equal 20. This variable can be used to help recognize value differences between TYPE's that have been aggregated into the same SET. For example, to assume that red pine sawlogs are worth 40 percent more than jack pine sawlogs, one would use a "b" term for red pine that is 40 percent greater than the "b" term for jack pine.

******** Sub-block 2a1= -- Input these values if ITYPEMARKSET = 0

2a1=-1-1	PRTM!	With TYPEMARKSET = 0, the prices for this item TYPE in this MARKET are assumed to be fixed and unchanging throughout the solution process. This line of input should include NTIC+1 price estimates, a price estimate for each TIC (period) and a single price estimate that applies for all periods beyond the end of the planning horizon. Note that a price of zero could result in some flows. To insure no flows of this TYPE to this MARKET one could use a large negative price.

******** Sub-block 2a1< -- Input these values if ITYPEMARKSET is < 0

2a1<-1-1	IFIXMARKET	The market for which the price for this item TYPE and MARKET depends. Note that this value could be irrelevant if the "sometype" is a type for which transportation costs are not considered.
2a1<-1-2	FIXINT!	With ITYPEMARKSET < 0, THE price for this MARKET and TYPE is defined by the relationship P(TYPE,MARKET)= a + b * P(sometype, somemarket) FIXINT! is the "a" term in this relationship. Note that "sometype" is defined as -ITYPEMARKSET and "somemarket" is defined by IFIXMARKET.
2a1<-1-3	FIXSLOPE!	FIXSLOPE! is the "b" term in the relationship P(TYPE,MARKET)= a + b * P(sometype, somemarket)

Note that prices in these relationships refer to specific flow units and any differences in units must be incorporated by the user in setting IFIXINT! and IFIXSLOPE!

******** Block 2b applies to items for which transport costs are not recognized. The block IS NOT repeated for each market.

2b-1-1 ITYPESET The variable used to link the item TYPE to a SET as used to define flow constraints for each period. If ITYPESET is:

> 0 then — ITYPESET is the SET to which flows of this TYPE will be linked.

= 0 then — Prices for this TYPE are assumed to be fixed and independent of the level of flows and prices of other items.

< 0 then — quantity flows for this TYPE are not constrained but prices for this TYPE are a function of the price another of SET. The negative of ITYPESET defines the item SET on which prices are based.

******** The next lines of block 2b depend on whether ITYPESET is greater than, less than or equal to zero. Only ONE of the three applies.

**** Sub-block 2b> -- Input these values if ITYPESET is > 0

2b>-1-1 UNITCONV! — The conversion factor to change units of this item TYPE to the units used for the corresponding SET for this item TYPE. This value is multiplied by the flow quantity for the TYPE to convert to the units used to measure the SET.

2b>-1-2 TYPEINT! — Prices for this TYPE are assumed to be a function of the shadow price (or marginal cost) for the corresponding SET. This relationship is simply:

P(jTYPE) = a + b * P(jSET)

TYPEINT! is the "a" term in this relationship. One could use this variable to build in assumptions like: birch pulp is worth $5 more per cord than maple pulpwood; if the SET prices are in terms of maple pulpwood, then "a" would equal 5 for this example.

2b>-1-3 TYPESLOPE!

As described in the definition for the previous input value (TYPEINT!), per unit prices for this TYPE are assumed to be a function of the shadow price for the corresponding SET. This relationship is:

$$P(jTYPE) = a + b * P(jSET)$$

TYPESLOPE is the "b" term in this relationship. It needs to recognize any differences in the units used to measure the item TYPE and its corresponding SET (as defined by ITYPESET). For example, if the conversion factor to convert TYPE units to SET units is 20 (as defined by UNITCONV! on this same line of input), and the SET price represents this item TYPE with no other price relationship assumptions, then ITYPESLOPE! ("b") must equal 1/20 or .05. This variable can be used to help recognize value differences between TYPE's that have been aggregated into the same SET. For example, to assume that red pine sawlogs are worth 40 percent more than jack pine sawlogs, one would use a "b" term for red pine that is 40 percent greater than the "b" term for jack pine.

**** Sub-block 2b= -- Input these values if ITYPESET = 0

2b=-1-** PRT!

With TYPESET = 0, the prices for this item TYPE is assumed to be fixed and unchanging throughout the solution process. This line of input should include NTIC+1 price estimates, a price estimate for each TIC (period) and a single price estimate that applies for all periods beyond the end of the planning horizon. Note that a price of zero could result in some flows. Negative values should be used for costs.

**** Sub-block 2b< -- Input these values if ITYPESET is < 0

2b<-1-1 IFIXMARKET

The market on which the price for this item TYPE depend. This value is relevant only if the item TYPE on which the relationship applies is one that recognizes transportation costs.

2a<-1-2 FIXINT!

The price for this TYPE is defined by the relationship

$$P(TYPE)= a + b * P(sometype, somemarket)$$

FIXINT! is the "a" term in this relationship. Note that "sometype" is defined as -ITYPESET and "somemarket" is defined by IFIXMARKET.

2b<-1-3 FIXSLOPE! — FIXSLOPE! is the "b" term in the relationship

P(TYPE)= a + b * P(sometype, somemarket)

Note that prices in these relationships refer to specific flow units and any differences in units must be incorporated by the user in setting IFIXINT! and IFIXSLOPE!

******** Block 3 is repeated once for each SET.

3-1-1 ZSET$ — A one-line label for the SET.

3-2-** PRSET!() — The initial price estimates, one for each TIC starting with the first TIC and ending with the last TIC in the planning horizon. This value is entered as a real number. This line should include NTIC entries.

3-3-** EWGHT!() — The relative weights to be used for each TIC in estimating prices for the set for periods beyond the end of the planning horizon. The sum of these weights is not important as the model automatically standardizes them. This line should include NTIC entries.

3-4-1 NCNSTRNTSET — The number of constraints for this item group.

****

the remainder of Block 3 is included only if NCNSTRNTSET > 0

****

3-5-1 CONSTRAINTDC — The index of the discount method to use for estimating the deviation from the desired level over the range of the period for which each constraint applies. The relevant values are (0) no discounting (1) discount using the general discount rate (2) discount by estimating the growth rate for stands likely to shift harvests between periods.

3-5-2 FLOATDC — The index of the discount method to use for estimating the average deviation from the desired flow deviation from the desired level in the float procedure. The relevant values are (0) no discounting (1) discount using the general discount rate (2) discount by estimating the growth rate for stands likely to shift harvests between periods.

3-5-3 SMOOTHDC	The index of the discount method to use for estimating the average deviation from the desired flow deviation from the desired level in the smooth procedure. The relevant values are (0) no discounting (1) discount using the general discount rate (2) discount by estimating the growth rate for stands likely to shift harvests between periods.
3-5-4 SHAPEDC	The index of the discount method to use for estimating the average deviation from the desired flow deviation from the desired level in the shape procedure. The relevant values are (0) no discounting (1) discount using the general discount rate (2) discount by estimating the growth rate for stands likely to shift harvests between periods.
3-6-1 STOPFLOATPC	The float procedure is not used if the total deviation in item group flows over the constrained period is less than this percentage (expressed as a percent). This option is included because the direction of the price change is not necessarily clear for small average percent deviations.
3-7-1 NFLOATBREAK	The number of break points used in the float procedure for defining the adjustment in price that will be made for possible "average deviation" values.
3-7-2 NSMOOTHBREAK	The number of break points used in the smooth procedure for defining the adjustment in price that will be made for possible "average deviation" values.
3-7-3 NSHAPEBREAK	The number of break points used in the shape procedure for defining the adjustment in price that will be made for possible "average deviation" values.

******** Block 3a is repeated once for each NFLOATBREAK with a one-line REM identifier at the top of the block (not repeated for each row of input).

3a-1-1 FLOATBREAK	The average percent deviation (integer only) for the price break.
3a-1-2 FLOATVAL!	The dollar amount by which prices should be changed for the average deviation associated with this price break.

*** Block 3b is repeated once for each NSMOOTHBREAK with a one-line REM identifier at the top of the block

3b-1-1 SMOOTHBREAK	The average percent deviation (integer only) for the price break.
3b-1-2 SMOOTHVAL!	The dollar amount by which prices should be changed for the average deviation associated with this price break.

******** Block 3c is repeated once for each NSHAPEBREAK with a one-line REM identifier at the top of the block

3c-1-1 SHAPEBREAK	The average percent deviation (integer only) for the price break.
3c-1-2 SHAPEVAL!	The dollar amount by which prices should be changed for the average deviation associated with this price break.
3-8-* SHAPEWGHT()	An array that defines the weights to be used for each TIC in estimating the average deviation within the shape adjustment procedure. The first NTIC weights input are the weights to be used for each TIC in estimating the deviation for TIC=1 (row 1 of the array). The last NTIC weights are the same except they apply to the estimation of the average percent deviation for the last TIC in the planning horizon. For the set of weights for any one period, the weights are relative and need not sum to any specific value.
3-9-1 PROCESSCOSTINP!	The estimate of the cost per item unit for harvesting and transporting the average stand that produces this SET. Used in estimating the value growth rate if the option 2 discount rate is used.
3-10-1 GOAL ()	The flow goals for each TIC. One goal for every TIC starting with TIC+1.

******** BLOCK 3d is repeated within block 3 once for each constraint (NCNSTRNTSET)

3d-1-1 TICBEGCN	The first period for which the constraint applies.
3d-1-2 TICENDCN	The last period for which the constraint applies.
3d-1-3 ACCEPTPC!	The maximum acceptable percentage deviation for the constraint expressed as a percent, i.e., 1.5 equals 1.5%.

V. AA PRESCRIPTION FILES

The Analysis Area (AA) Prescription files are designed for fast input. They must all be input once for each iteration of DUALPLAN. A short conversion program, BLOADER, is used to create files that can be loaded directly into memory in blocks as large as 64K. Each block becomes a separate BLOAD file for input into DTRAN. An application for a large problem could require a large number of BLOAD files. DTRAN uses a separate file to list the name of each BLOAD file, the number of analysis areas (ground plots) it contains, and the REGION associated with the prescriptions it contains. Information for this file is created automatically by Program BLOADER.

The BLOAD files need not all be developed in a single run of BLOADER. For large problems this might be difficult as many analysis areas would be involved. If multiple runs are to be used, the user will need to combine the file lists as output by BLOADER. The last file name on the list should be a dummy filename DTRAN with the number of analysis areas set less than zero.

The basic raw data file for stand prescriptions is very similar to an input file that might be used for a simple cash-flow program. Essentially it has a separate section for each stand with several nested loops for prescriptions, stand entries, and product flows. All values on this input file must be integers less than 32,767. This limitation and the potential for round-off errors due to integer requirements should be considered in selecting the units to use to represent item TYPES (flows) on a per land unit basis.

Listed below are definitions describing the specific format for the regeneration prescription file. The numbering system is "block number - group number - entry number." Each group number must begin on a new line in the file.

ID #	Name	Description
********	Block 1 is repeated once for each analysis area. The end of the list is signalled by a final Block 1 with a NPRES less than zero.	
1-1-*	AAID$	A one line identifier for the analysis area.
1-2-1	NSTAND	The number of individual "stands" that share these same prescriptions. If forest survey ground plots are serving as the basic unit for developing management prescriptions, NSTAND is the number of photo plots assigned to this ground plot. These "stands" are all evaluated separately.
1-2-2	NPRESAA	The number of management prescriptions
1-2-3	NBARETYPE	The number of bare land types that apply to this analysis area (ground plot).
********	Block 2 is repeated once for each possible NBARETYPE	

2-*-1	KBIOTYPE	The index of a biological bare land type
2-*-2	FTICREG	The first period in which this analysis area could convert the site to this biological bare land type

******** Block 3 is repeated once for each "stand" that shares this set of prescriptions

3-1-1	ACRES1000	The number of 1000's of acres in the analysis area. This value along with ACRE1 (3-1-2) are used together to define the size of the analysis area. Two terms are used because the input file uses (16-bit) integers and thus values greater than 32,767 cannot be represented.
3-1-2	ACRE1	See 3-1-1 ACRE1000
3-1-3	VRESERVEALL	The value of the "stand" if none of the prescriptions are implemented -- the value of the "do nothing" alternative. It could be used to represent the value of nontimber uses if only timber flows are considered.
3-1-4	VRESERVEREG	The value of the "stand" if it is reserved from timber production after the first rotation. This value is discounted based on the time that the first rotation is completed. This value could be used to represent the potential to convert the site to another land use after harvesting.
3-1-5	ACCESSVAR	This value is measured in the same units that are used to measure the road network. It represents the "cost" of transporting products from the "stand" to the road network. It is the "cost" of transporting a quantity equal to ACCESSUNITS! (see description of ACCESSUNITS -- 1-10-6 in Formulation file).
3-1-6	STANDROAD	The road number identifier for the road on the network closest to this "stand"
3-1-7	STANDLOC	The location on the nearest road that is nearest the "stand." It is measured in UNITSPERROAD units (see description of UNITSPERROAD -- 1-10-5 in Formulation file)

******** Block 3a is repeated within block 3 once for each possible bare land type in the order that bare land types are input in block 2.

3a-1-*	ADBENREG	The added benefit (or cost if negative) that would occur if this "stand" location is converted to this bare land type. This variable can be used to recognize different impacts

		based on stand location. This impact is discounted assuming it occurs at the time the "stand" is regenerated. This value is included in the SEV calculation to determine stand values but it is not included in any flow summaries for the forest.

******** Block 3b is repeated within block 3, once for each possible management prescription in the order that prescriptions are input in Block 2 of this file.

3b-1-*	ADDBENPRES --	The added benefit (or cost) if this prescription is applied to this "stand." This value is included in the SEV calculation to determine stand values but it is not included in any flow summaries for the forest.

******** Block 4 is repeated once for each of the NPRES prescriptions.

4-1-*	PRESLAB$	A one-line alphanumeric label for the prescription
4-2-1	NENT	The number of times the stand is "entered" for this prescription. Stand "entries" include any time in which input or output flows are involved that would influence a "cash-flow" analysis.
4-2-2	KREGTIC	The TIC in which the next stand can be regenerated (linked with bare land) with this alternative
4-2-3	NREGBIO	The number of bare land types to consider as possible types for the next rotation. These options can include a site conversion cost.

******** Block 4a is repeated once within block 4 for each of the NENT "stand" entries.

4a-1-1	KTICENTRY	The TIC in which the corresponding stand entry occurs
4a-1-2	NITEM	The total number of items that flow during this stand entry.

******** Block 4a1 is repeated within block 4a, once for each of the NITEM item flows.

4a1-1-1	KTYPE	The index number of the item TYPE for the corresponding flow
4a1-1-2	QUAN	The quantity of the flow. This value will be multiplied times the price for the item to determine the value of the flow.

******** Block 4b is repeated once for each of the bare land types possible for the next rotation

4b-1-1	KREGBIO	The index of a biological "bare land" class to consider for the next rotation.
4b-1-2	CONVCOST	The conversion cost necessary to convert to the bare land type identified by KREGBIO on this same line. This cost is a one time cost that is assumed to occur only for the first rotation and at the time of regeneration.

VI. THE REGENERATION PRESCRIPTION FILE

The Regeneration Prescription File contains the information on the regeneration options for the different "bare land" classes. All item TYPE flows other than transport costs are assumed to be included in the prescriptions. The Regeneration Prescription File is read by DTRAN only once and all information is stored in memory.

The basic raw data file for regeneration prescriptions is very similar to an input file that might be used for a simple cash-flow program. Essentially it has a separate section for each biological land class with several nested loops for prescriptions, stand entries, and product flows. All values on this input file must be integers less than 32767. This limitation and the potential for round-off errors should be considered in selecting the units to use to represent item TYPE flows.

Listed below are definitions describing the specific format for the regeneration prescription file. The numbering system is "block number - group number - entry number." Each group number must begin on a new line in the file.

ID #	**Name**	**Description**
********	Block 1 is repeated once for each biological "bare land" class. The end of the list is signalled by a final Block 1 with a NPRES less than zero.	
1-1-*	BIOTYPE$	An alphanumeric label for the biological "bare land" type.
1-2-1	KBIO	The index number that describes this biological land type. This index number is the KREGBIO number used in the AA prescriptions to link the AA with the different "bare land" types.
1-2-2	NPRES	The number of full-rotation prescriptions for this biological land class.
********	Block 2 is repeated once for each of the NPRES prescriptions.	
2-1-*	PRESLAB$	A one-line alphanumeric label for the prescription
2-2-1	NENT	The number of stand entries. Each stand entry represents a point in time (TIC) when one or more item flows would occur.
2-2-2	ROTLEN	The length of the rotation in TICs for this prescription.
********	Block 3 is repeated once for each of the NENT stand entries. Block 3 is thus nested within both Block 1 and Block 2.	
3-1-1	ENTRYAGE	The age (in TICs) at which this stand entry would occur.

3-1-2	NITEM	The number of item flows that are associated with this stand entry.

******** Block 4 is repeated once for each of the NITEM item flows associated with the stand entry. Block 4 is thus nested within Blocks 1, 2, and 3.

4-1-1	KTYPE	The index number of the item associated with the flow
4-2-2	QUAN	The quantity of the item that flows. This value is assumed to be expressed in a per land unit basis.

VII. TRANSPORTATION FILES

DTRAN reads basic information describing the transportation system from several files. For the Minnesota situation, these files can be generated using the GIS system developed for use with DTRAN (Kapple and Hoganson, 1991). The transportation files are all referenced on the Model Formulation file. A description of the format for each file follows. The numbering convention is the same as used for other files (block number - group number - entry number)

MARKETDATFN$ The file that contains the information describing each market location in terms of the road network.

ID #	Name	Description
1-1-1	NMARKET	The number of markets
1-1-2	NDESTINATION	The number of destinations. Destinations are possible market locations. An array is stored which describes all shortest distances from each node to all possible destinations. By recognizing additional destinations, one can change the number of markets or market locations between DTRAN runs without needing to redevelop the basic transportation data for input.
********	Block 2 is repeated once for each market	
2-1-1	KMARKET	The market number
2-1-2	DBNODEID	The Data Base Node ID for use by any mapping routine. This value is not important if mapping routines are not used
2-1-3	POINTID	The POINT identifier or node number as defined by the road network used by DTRAN
2-1-4	DESTID	The corresponding destination number (or column) in the shortest path array that defines shortest distances from each point to possible market locations (destinations)

COSTROUTEFN$ The file containing the low-cost estimates for all road point-to-market shipping combinations.

ID #	Name	Description
********	Block 1 is repeated once for each possible destination (market location)	
1-1-*	COSTTODEST()	For the given destination, the cost or "distance" of transport to that destination. This value is input for all POINTS (nodes) in the road network starting with the first POINT.

COSTROADFN$ The file containing the estimates of the "cost" to transport the length of each road in the road network. The "costs" can be in any units such as time, dollars per cord, or distance. For each item TYPE the variable TRANCOEF! (see Model formulation file 2a1-1-1)

ID #	Name	Description
********	Block 1 is repeated once for each possible destination (market location)	
1-1-1	DBROADID	Data base road identifier. This value is not used in the solution process.
1-1-2	ROADBEGPT	The beginning POINT for the road
1-1-3	ROADENDPT	The ending POINT for the road
1-1-4	ROADCOST!	The "cost" of travelling the length of the road

PATHNETDB$ The name of the DOS directory containing the data base files describing the road network. These files are used only for the mapping options. For details, see the GISTRAN users Manual (Kapple and Hoganson).

VIII. AN EXAMPLE

A hypothetical example using approximately 2000 ground plots and 20,000 ground plots has been used to test DTRAN in its initial development. The example uses dollar costs and 3 timber types as the item TYPES, a 5 period planning horizon for constraint purposes (infinite horizon for valuation purposes) and constrains flows for 10 TYPE-MARKET combinations. The output shown here for the solution after 221 iterations has flows very close to the targets (goals) for nearly all constraints. Similar results were found for a number of different applications varying in target flows and initial shadow price estimates. For the example shown, the parameters describing the shadow price re-estimation process were never changed in the solution process.

The Model Formulation file

The following listing is a model formulation file that describes the problem. All lines that begin with "REM" are labels to make files easier to read using an editor. Each REM line must be included in the input file. Each is read by DTRAN but not interpreted.

```
REM  FILENAME FOR REGEN DATA
EXFILE.REG
REM  FILENAME FOR LIST OF AA PRESCRIPTION FILES
FNAMES.50
REM  FILENAME FOR MARKET DATA
MARKET.5E
REM  FILENAME CONTAINING SHIPPING COSTS FOR ALL MARKET-POINT COMBINATIONS
NET30M5.PTH
REM  FILENAME CONTAINING INFORMATION FOR EACH ROAD
NET30.ARC
REM  NAME OF PATH FOR NETWORK DATA BASE FILES
C:\DTRAN\NETWORK\
REM  WRITECNSTRNT WRITESET WRITETYPE
      0        0       0
REM  NTIC DCRATE! NTYPE NTYPESHIP NSET NMARKET  NSAVE
    5    .5     4     3      11    5       30
REM  NREGBIO MXREGROT MXTICAAPRES MXPRES MXSTAND  NREGION
     24      10       11       15     12      1
REM  NPOINT  NROAD  MXARC  MXNODE UNITSPERROAD ACCESSUNITS
     30    58    58    30     1000          10
REM  MXCNSTRNT TOTRGPRES TOTRGITEM
     55       200      600
REM  NFLOAT  NSMTHFLT  NSHAPE  NSMTHSHP MXBREAK
     2       2       3       2        3
REM  REPEAT ONE BLOCK FOR EACH ITEM TYPE.  FORMAT
REM  DEPENDS ON VALUE OF NEEDTOSHIP.  IF NEEDTOSHIP(JTYPE) > 0 THEN
REM  THE ITEM TYPE IS SHIPPED AND INFO IS NEEDED FOR EACH MARKET.
REM  THE VALUE OF ITYPEMARKSET IS ALSO IMPORTANT FOR BOTH CASES.
REM  IF ITYPEMARKSET IS:
REM     > 0 THEN  UNITCONV!(JTYPE,JMARKET) TYPEMARKETINT() TYPEMARKSLOPE()
REM     < 0 THEN  IFIXMARKET() IFIXINT!()  IFIXSLOPE()
REM     = 0 THEN  PRICE(JTYPE,JMARKET,JTIC) FOR JTIC = 1 TO NTIC1
```

```
REM  ZTYPE$(JTYPE)
    PRESCRIPTION DOLLAR COSTS
REM  NEEDTOSHIP(JTYPE)
     0
REM  ITYPEMARKSET(JTYPE,1)
        11
REM  DATA CONDITIONAL ON ITYPEMARKETSET(SEE DESCRIPTION ABOVE)
        1 0 1
REM  ZTYPE$(JTYPE)
    ASPEN
REM  NEEDTOSHIP(JTYPE)
     1
REM  TRANCOEF!(JTYPE)
    .02
REM  ASPEN -- MARKET # 1
REM  ITYPEMARKSET(JTYPE,JMARKET)
        1
REM  DATA CONDITIONAL ON VALUE OF ITYPEMARKETSET (SEE DESCRIPTION ABOVE)
        .1 0 .1
REM  ASPEN -- MARKET # 2
REM  ITYPEMARKSET(JTYPE,JMARKET)
        2
REM  DATA CONDITIONAL ON VALUE OF ITYPEMARKETSET (SEE DESCRIPTION ABOVE)
        .1 0 .1
REM  ASPEN -- MARKET # 3
REM  ITYPEMARKSET(JTYPE,JMARKET)
        3
REM  DATA CONDITIONAL ON VALUE OF ITYPEMARKETSET (SEE DESCRIPTION ABOVE)
        .1 0 .1
REM  ASPEN -- MARKET # 4
REM  ITYPEMARKSET(JTYPE,JMARKET)
        4
REM  DATA CONDITIONAL ON VALUE OF ITYPEMARKETSET (SEE DESCRIPTION ABOVE)
        .1 0 .1
REM  ASPEN -- MARKET # 5
REM  ITYPEMARKSET(JTYPE,JMARKET)
        0
REM  DATA CONDITIONAL ON VALUE OF ITYPEMARKETSET (SEE DESCRIPTION ABOVE)
        0 0 0 0 0 0
REM  ZTYPE$(JTYPE)
    BIRCH
REM  NEEDTOSHIP(JTYPE)
     1
REM  TRANCOEF!(JTYPE)
    0.02
REM  BIRCH -- MARKET # 1
REM  ITYPEMARKSET(JTYPE,JMARKET)
        5
REM  DATA CONDITIONAL ON VALUE OF ITYPEMARKETSET (SEE DESCRIPTION ABOVE)
        .1 0 .1
REM  BIRCH -- MARKET # 2
```

```
REM  ITYPEMARKSET(JTYPE,JMARKET)
        6
REM  DATA CONDITIONAL ON VALUE OF ITYPEMARKETSET (SEE DESCRIPTION ABOVE)
        .1 0 .1
REM  BIRCH -- MARKET # 3
REM  ITYPEMARKSET(JTYPE,JMARKET)
        7
REM  DATA CONDITIONAL ON VALUE OF ITYPEMARKETSET (SEE DESCRIPTION ABOVE)
        .1 0 .1
REM  BIRCH -- MARKET # 4
REM  ITYPEMARKSET(JTYPE,JMARKET)
       0
REM  DATA CONDITIONAL ON VALUE OF ITYPEMARKETSET (SEE DESCRIPTION ABOVE)
       0 0 0 0 0 0
REM  BIRCH -- MARKET # 5
REM  ITYPEMARKSET(JTYPE,JMARKET)
       0
REM  DATA CONDITIONAL ON VALUE OF ITYPEMARKETSET (SEE DESCRIPTION ABOVE)
       0 0 0 0 0 0
REM  ZTYPE$(JTYPE)
    SOFTWOOD
REM  NEEDTOSHIP(JTYPE)
     1
REM  TRANCOEF!(JTYPE)
    0.02
REM  SOFTWOOD -- MARKET # 1
REM  ITYPEMARKSET(JTYPE,JMARKET)
        8
REM  DATA CONDITIONAL ON VALUE OF ITYPEMARKETSET (SEE DESCRIPTION ABOVE)
        .1 0 .1
REM  SOFTWOOD -- MARKET # 2
REM  ITYPEMARKSET(JTYPE,JMARKET)
        0
REM  DATA CONDITIONAL ON VALUE OF ITYPEMARKETSET (SEE DESCRIPTION ABOVE)
        0 0 0 0 0 0
REM  SOFTWOOD -- MARKET # 3
REM  ITYPEMARKSET(JTYPE,JMARKET)
        0
REM  DATA CONDITIONAL ON VALUE OF ITYPEMARKETSET (SEE DESCRIPTION ABOVE)
        0 0 0 0 0 0
REM  SOFTWOOD -- MARKET # 4
REM  ITYPEMARKSET(JTYPE,JMARKET)
       9
REM  DATA CONDITIONAL ON VALUE OF ITYPEMARKETSET (SEE DESCRIPTION ABOVE)
       .1 0 .1
REM  SOFTWOOD -- MARKET # 5
REM  ITYPEMARKSET(JTYPE,JMARKET)
       10
REM  DATA CONDITIONAL ON VALUE OF ITYPEMARKETSET (SEE DESCRIPTION ABOVE)
       .1 0 .1
```

```
REM ZSET$(JSET)
    MILL 1, ASPEN
REM  PRSET!(JSET,JTIC),JTIC=1 TO NTIC
    30 40 50 40 30
REM  EWGHT(JSET, JTIC), JTIC=1 TO NTIC
      0 0 0 1 1
REM  NCNSTRNT(JSET)  IF = 0, THEN NEXT SET.
       5
REM  CONSTRAINTDC(JSET), FLOATDC(JSET), SMOOTHDC(JSET), SHAPEDC(JSET)
       0              0             0            0
REM  STOPFLOATPC!(JSET)
       5.0
REM  NFLOATBREAK(JSET), NSMOOTHBREAK(JSET), NSHAPEBREAK(JSET)
        3              3              3
REM  FLOATBREAK(JSET, JBREAK), FLOATVAL!(JSET, JBREAK) -- 1 LINE PER BREAK
        10                .1
        50               1.00
        100               1.50
REM  SMOOTHBREAK(JSET, JBREAK), SMOOTHVAL!(JSET, JBREAK) -- 1 LINE PER BREAK
        10                .02
        50                .30
        100                .40
REM  SHAPEBREAK(JSET, JBREAK), SHAPEVAL!(JSET, JBREAK) -- 1 LINE PER BREAK
        5                 .05
        20                .50
        100                1.50
REM   RELATIVE WEIGHTS BY PERIOD FOR SHAPE ADJUSTMENT- ONE SET FOR EACH TIC
REM  (SHAPEWGHT(JSET, JTIC, JJTIC),JJTIC=1 TO NTIC) -- JTIC * JTIC VALUES
        3 2 1 0 0
        2 3 2 1 0
        1 2 3 2 1
        0 1 2 3 2
        0 0 1 2 3
REM  PROCESSCOSTINP!(JSET)
        25
REM  GOAL!(JSET, JTIC), JTIC=1 TO NTIC
      1500000
      1500000
      1500000
      1500000
      1500000
REM   REPEAT ONCE FOR EACH NCNSTRNT(JSET)
REM   TICBEGCN(JCN), TICENDCN(JCN), ACCEPTPC!(JCN), CNTYPE(JCN)
        1   1  2  0
        2   2  2  0
        3   3  2  0
        4   4  2  0
        5   5  2  0
REM ZSET$(JSET)
    MILL 2 ASPEN
```

```
REM  PRSET!(JSET,JTIC),JTIC=1 TO NTIC
    30 40 50 40 30
REM  EWGHT(JSET, JTIC), JTIC=1 TO NTIC
      0 0 0 1 1
REM  NCNSTRNT(JSET)  IF = 0, THEN NEXT SET.
    5
REM  CONSTRAINTDC(JSET), FLOATDC(JSET), SMOOTHDC(JSET), SHAPEDC(JSET)
       0                    0               0              0
REM  STOPFLOATPC!(JSET)
       1.0
REM  NFLOATBREAK(JSET), NSMOOTHBREAK(JSET), NSHAPEBREAK(JSET)
        3                  3                  3
REM  FLOATBREAK(JSET, JBREAK), FLOATVAL!(JSET, JBREAK) -- 1 LINE PER BREAK
        10                      .1
        50                     1.00
        100                     1.50
REM  SMOOTHBREAK(JSET, JBREAK), SMOOTHVAL!(JSET, JBREAK) -- 1 LINE PER BREAK
        10                      .02
        50                      .30
        100                      .40
REM  SHAPEBREAK(JSET, JBREAK), SHAPEVAL!(JSET, JBREAK) -- 1 LINE PER BREAK
        5                      .05
        20                      .50
        100                      1.50
REM   RELATIVE WEIGHTS BY PERIOD FOR SHAPE ADJUSTMENT- ONE SET FOR EACH TIC
REM  (SHAPEWGHT(JSET, JTIC, JJTIC),JJTIC=1 TO NTIC) -- JTIC * JTIC VALUES
        3 2 1 0 0
        2 3 2 1 0
        1 2 3 2 1
        0 1 2 3 2
        0 0 1 2 3
REM  PROCESSCOSTINP!(JSET)
        25
REM  GOAL!(JSET, JTIC), JTIC=1 TO NTIC
      1000000
      1000000
      1000000
      1000000
      1000000
REM   REPEAT ONCE FOR EACH NCNSTRNT(JSET)
REM   TICBEGCN(JCN), TICENDCN(JCN), ACCEPTPC!(JCN), CNTYPE(JCN)
         1   1  .5  0
         2   2  .5  0
         3   3  .5  0
         4   4  .5  0
         5   5  .5  0
REM ZSET$(JSET)
    MILL 3 ASPEN
REM  PRSET!(JSET,JTIC),JTIC=1 TO NTIC
    30 40 50 40 30
```

```
REM  EWGHT(JSET, JTIC), JTIC=1 TO NTIC
     0 0 0 1 1
REM  NCNSTRNT(JSET)  IF = 0, THEN NEXT SET.
    5
REM  CONSTRAINTDC(JSET), FLOATDC(JSET), SMOOTHDC(JSET), SHAPEDC(JSET)
      0                    0               0             0
REM  STOPFLOATPC!(JSET)
      1.0
REM  NFLOATBREAK(JSET), NSMOOTHBREAK(JSET), NSHAPEBREAK(JSET)
       3                  3                   3
REM  FLOATBREAK(JSET, JBREAK), FLOATVAL!(JSET, JBREAK) -- 1 LINE PER BREAK
       10                      .1
       50                     1.00
       100                     1.50
REM  SMOOTHBREAK(JSET, JBREAK), SMOOTHVAL!(JSET, JBREAK) -- 1 LINE PER BREAK
       10                      .02
       50                      .30
       100                      .40
REM  SHAPEBREAK(JSET, JBREAK), SHAPEVAL!(JSET, JBREAK) -- 1 LINE PER BREAK
       5                      .05
       20                      .50
       100                      1.50
REM   RELATIVE WEIGHTS BY PERIOD FOR SHAPE ADJUSTMENT- ONE SET FOR EACH TIC
REM  (SHAPEWGHT(JSET, JTIC, JJTIC),JJTIC=1 TO NTIC) -- JTIC * JTIC VALUES
       3 2 1 0 0
       2 3 2 1 0
       1 2 3 2 1
       0 1 2 3 2
       0 0 1 2 3
REM  PROCESSCOSTINP!(JSET)
       25
REM  GOAL!(JSET, JTIC), JTIC=1 TO NTIC
     1500000
     1500000
     1500000
     1500000
     1500000
REM   REPEAT ONCE FOR EACH NCNSTRNT(JSET)
REM   TICBEGCN(JCN), TICENDCN(JCN), ACCEPTPC!(JCN), CNTYPE(JCN)
        1   1  .5  0
        2   2  .5  0
        3   3  .5  0
        4   4  .5  0
        5   5  .5  0
REM ZSET$(JSET)
    MILL 4 ASPEN
REM  PRSET!(JSET,JTIC),JTIC=1 TO NTIC
    30 40 50 40 30
REM  EWGHT(JSET, JTIC), JTIC=1 TO NTIC
     0 0 0 1 1
```

```
REM  NCNSTRNT(JSET)  IF = 0, THEN NEXT SET.
    5
REM  CONSTRAINTDC(JSET), FLOATDC(JSET), SMOOTHDC(JSET), SHAPEDC(JSET)
    0            0          0          0
REM  STOPFLOATPC!(JSET)
    5.0
REM  NFLOATBREAK(JSET), NSMOOTHBREAK(JSET), NSHAPEBREAK(JSET)
     3           3           3
REM  FLOATBREAK(JSET, JBREAK), FLOATVAL!(JSET, JBREAK) -- 1 LINE PER BREAK
     10              .1
     50             1.00
     100             1.50
REM  SMOOTHBREAK(JSET, JBREAK), SMOOTHVAL!(JSET, JBREAK) -- 1 LINE PER BREAK
     10              .02
     50              .30
     100              .40
REM  SHAPEBREAK(JSET, JBREAK), SHAPEVAL!(JSET, JBREAK) -- 1 LINE PER BREAK
     5              .05
     20              .50
     100              1.50
REM   RELATIVE WEIGHTS BY PERIOD FOR SHAPE ADJUSTMENT- ONE SET FOR EACH TIC
REM  (SHAPEWGHT(JSET, JTIC, JJTIC),JJTIC=1 TO NTIC) -- JTIC * JTIC VALUES
     3 2 1 0 0
     2 3 2 1 0
     1 2 3 2 1
     0 1 2 3 2
     0 0 1 2 3
REM  PROCESSCOSTINP!(JSET)
     25
REM  GOAL!(JSET, JTIC), JTIC=1 TO NTIC
    1000000
    1000000
    1000000
    1000000
    1000000
REM   REPEAT ONCE FOR EACH NCNSTRNT(JSET)
REM   TICBEGCN(JCN), TICENDCN(JCN), ACCEPTPC!(JCN), CNTYPE(JCN)
      1   1  2  0
      2   2  2  0
      3   3  2  0
      4   4  2  0
      5   5  2  0
REM ZSET$(5)
   MILL 1 BIRCH
REM  PRSET!(JSET,JTIC),JTIC=1 TO NTIC
   30 40 50 40 30
REM  EWGHT(JSET, JTIC), JTIC=1 TO NTIC
    0 0 0 1 1
REM  NCNSTRNT(JSET)  IF = 0, THEN NEXT SET.
   5
```

```
REM  CONSTRAINTDC(JSET), FLOATDC(JSET), SMOOTHDC(JSET), SHAPEDC(JSET)
      0                    0               0             0
REM  STOPFLOATPC!(JSET)
      1.0
REM  NFLOATBREAK(JSET), NSMOOTHBREAK(JSET), NSHAPEBREAK(JSET)
       3                   3                  3
REM  FLOATBREAK(JSET, JBREAK), FLOATVAL!(JSET, JBREAK) -- 1 LINE PER BREAK
       10                       .1
       50                      1.00
       100                      1.50
REM  SMOOTHBREAK(JSET, JBREAK), SMOOTHVAL!(JSET, JBREAK) -- 1 LINE PER BREAK
       10                       .02
       50                       .30
       100                       .40
REM  SHAPEBREAK(JSET, JBREAK), SHAPEVAL!(JSET, JBREAK) -- 1 LINE PER BREAK
       5                       .05
       20                       .50
       100                       1.50
REM   RELATIVE WEIGHTS BY PERIOD FOR SHAPE ADJUSTMENT- ONE SET FOR EACH TIC
REM  (SHAPEWGHT(JSET, JTIC, JJTIC),JJTIC=1 TO NTIC) -- JTIC * JTIC VALUES
       3 2 1 0 0
       2 3 2 1 0
       1 2 3 2 1
       0 1 2 3 2
       0 0 1 2 3
REM  PROCESSCOSTINP!(JSET)
       25
REM  GOAL!(JSET, JTIC), JTIC=1 TO NTIC
     1000000
     1000000
     1000000
     1000000
     1000000
REM   REPEAT ONCE FOR EACH NCNSTRNT(JSET)
REM   TICBEGCN(JCN), TICENDCN(JCN), ACCEPTPC!(JCN), CNTYPE(JCN)
        1   1  .5  0
        2   2  .5  0
        3   3  .5  0
        4   4  .5  0
        5   5  .5  0
REM ZSET$(JSET)
    MILL 2 BIRCH
REM  PRSET!(JSET,JTIC),JTIC=1 TO NTIC
    30 40 50 40 30
REM  EWGHT(JSET, JTIC), JTIC=1 TO NTIC
     0 0 0 1 1
REM  NCNSTRNT(JSET)  IF = 0, THEN NEXT SET.
    5
REM  CONSTRAINTDC(JSET), FLOATDC(JSET), SMOOTHDC(JSET), SHAPEDC(JSET)
      0                    0               0             0
```

```
REM  STOPFLOATPC!(JSET)
      1.0
REM  NFLOATBREAK(JSET), NSMOOTHBREAK(JSET), NSHAPEBREAK(JSET)
       3                  3                  3
REM  FLOATBREAK(JSET, JBREAK), FLOATVAL!(JSET, JBREAK) -- 1 LINE PER BREAK
       10                      .1
       50                     1.00
       100                    1.50
REM  SMOOTHBREAK(JSET, JBREAK), SMOOTHVAL!(JSET, JBREAK) -- 1 LINE PER BREAK
       10                      .02
       50                      .30
       100                      .40
REM  SHAPEBREAK(JSET, JBREAK), SHAPEVAL!(JSET, JBREAK) -- 1 LINE PER BREAK
       5                      .05
       20                      .50
       100                      1.50
REM   RELATIVE WEIGHTS BY PERIOD FOR SHAPE ADJUSTMENT- ONE SET FOR EACH TIC
REM  (SHAPEWGHT(JSET, JTIC, JJTIC),JJTIC=1 TO NTIC) -- JTIC * JTIC VALUES
       3 2 1 0 0
       2 3 2 1 0
       1 2 3 2 1
       0 1 2 3 2
       0 0 1 2 3
REM  PROCESSCOSTINP!(JSET)
       25
REM  GOAL!(JSET, JTIC), JTIC=1 TO NTIC
     500000
     500000
     500000
     500000
     500000
REM   REPEAT ONCE FOR EACH NCNSTRNT(JSET)
REM   TICBEGCN(JCN), TICENDCN(JCN), ACCEPTPC!(JCN), CNTYPE(JCN)
        1   1  .5  0
        2   2  .5  0
        3   3  .5  0
        4   4  .5  0
        5   5  .5  0
REM ZSET$(7)
    MILL 3 BIRCH
REM  PRSET!(JSET,JTIC),JTIC=1 TO NTIC
    30 40 50 40 30
REM  EWGHT(JSET, JTIC), JTIC=1 TO NTIC
     0 0 0 1 1
REM  NCNSTRNT(JSET)  IF = 0, THEN NEXT SET.
    5
REM  CONSTRAINTDC(JSET), FLOATDC(JSET), SMOOTHDC(JSET), SHAPEDC(JSET)
      0                  0              0            0
REM  STOPFLOATPC!(JSET)
      1.0
```

```
REM  NFLOATBREAK(JSET), NSMOOTHBREAK(JSET), NSHAPEBREAK(JSET)
        3                    3                   3
REM  FLOATBREAK(JSET, JBREAK), FLOATVAL!(JSET, JBREAK) -- 1 LINE PER BREAK
        10                       .1
        50                      1.00
        100                      1.50
REM  SMOOTHBREAK(JSET, JBREAK), SMOOTHVAL!(JSET, JBREAK) -- 1 LINE PER BREAK
        10                       .02
        50                       .30
        100                       .40
REM  SHAPEBREAK(JSET, JBREAK), SHAPEVAL!(JSET, JBREAK) -- 1 LINE PER BREAK
        5                       .05
        20                       .50
        100                       1.50
REM   RELATIVE WEIGHTS BY PERIOD FOR SHAPE ADJUSTMENT- ONE SET FOR EACH TIC
REM  (SHAPEWGHT(JSET, JTIC, JJTIC),JJTIC=1 TO NTIC) -- JTIC * JTIC VALUES
        3 2 1 0 0
        2 3 2 1 0
        1 2 3 2 1
        0 1 2 3 2
        0 0 1 2 3
REM  PROCESSCOSTINP!(JSET)
        25
REM  GOAL!(JSET, JTIC), JTIC=1 TO NTIC
      500000
      500000
      500000
      500000
      500000
REM   REPEAT ONCE FOR EACH NCNSTRNT(JSET)
REM   TICBEGCN(JCN), TICENDCN(JCN), ACCEPTPC!(JCN), CNTYPE(JCN)
         1   1  .5  0
         2   2  .5  0
         3   3  .5  0
         4   4  .5  0
         5   5  .5  0
REM ZSET$(8)
    MILL 1 SOFTWOOD
REM  PRSET!(JSET,JTIC),JTIC=1 TO NTIC
    30 40 50 40 30
REM  EWGHT(JSET, JTIC), JTIC=1 TO NTIC
      0 0 0 1 1
REM  NCNSTRNT(JSET)  IF = 0, THEN NEXT SET.
    5
REM  CONSTRAINTDC(JSET), FLOATDC(JSET), SMOOTHDC(JSET), SHAPEDC(JSET)
       0                    0                0              0
REM  STOPFLOATPC!(JSET)
       1.0
REM  NFLOATBREAK(JSET), NSMOOTHBREAK(JSET), NSHAPEBREAK(JSET)
        3                    3                   3
```

```
REM  FLOATBREAK(JSET, JBREAK), FLOATVAL!(JSET, JBREAK) -- 1 LINE PER BREAK
        10              .1
        50             1.00
        100             1.50
REM  SMOOTHBREAK(JSET, JBREAK), SMOOTHVAL!(JSET, JBREAK) -- 1 LINE PER BREAK
        10              .02
        50              .30
        100              .40
REM  SHAPEBREAK(JSET, JBREAK), SHAPEVAL!(JSET, JBREAK) -- 1 LINE PER BREAK
        5              .05
        20              .50
        100              1.50
REM   RELATIVE WEIGHTS BY PERIOD FOR SHAPE ADJUSTMENT- ONE SET FOR EACH TIC
REM  (SHAPEWGHT(JSET, JTIC, JJTIC),JJTIC=1 TO NTIC) -- JTIC * JTIC VALUES
        3 2 1 0 0
        2 3 2 1 0
        1 2 3 2 1
        0 1 2 3 2
        0 0 1 2 3
REM  PROCESSCOSTINP!(JSET)
        25
REM  GOAL!(JSET, JTIC), JTIC=1 TO NTIC
      1500000
      1500000
      1500000
      1500000
      1500000
REM   REPEAT ONCE FOR EACH NCNSTRNT(JSET)
REM   TICBEGCN(JCN), TICENDCN(JCN), ACCEPTPC!(JCN), CNTYPE(JCN)
         1   1  .5  0
         2   2  .5  0
         3   3  .5  0
         4   4  .5  0
         5   5  .5  0
REM ZSET$(9)
    MILL 4 SOFTWOOD
REM  PRSET!(JSET,JTIC),JTIC=1 TO NTIC
    30 40 50 40 30
REM  EWGHT(JSET, JTIC), JTIC=1 TO NTIC
      0 0 0 1 1
REM  NCNSTRNT(JSET)  IF = 0, THEN NEXT SET.
    5
REM  CONSTRAINTDC(JSET), FLOATDC(JSET), SMOOTHDC(JSET), SHAPEDC(JSET)
       0             0           0          0
REM  STOPFLOATPC!(JSET)
       1.0
REM  NFLOATBREAK(JSET), NSMOOTHBREAK(JSET), NSHAPEBREAK(JSET)
        3            3            3
```

```
REM  FLOATBREAK(JSET, JBREAK), FLOATVAL!(JSET, JBREAK) -- 1 LINE PER BREAK
       10                .1
       50               1.00
       100               1.50
REM  SMOOTHBREAK(JSET, JBREAK), SMOOTHVAL!(JSET, JBREAK) -- 1 LINE PER BREAK
       10                .02
       50                .30
       100                .40
REM  SHAPEBREAK(JSET, JBREAK), SHAPEVAL!(JSET, JBREAK) -- 1 LINE PER BREAK
       5                .05
       20                .50
       100                1.50
REM   RELATIVE WEIGHTS BY PERIOD FOR SHAPE ADJUSTMENT- ONE SET FOR EACH TIC
REM  (SHAPEWGHT(JSET, JTIC, JJTIC),JJTIC=1 TO NTIC) -- JTIC * JTIC VALUES
       3 2 1 0 0
       2 3 2 1 0
       1 2 3 2 1
       0 1 2 3 2
       0 0 1 2 3
REM  PROCESSCOSTINP!(JSET)
       25
REM  GOAL!(JSET, JTIC), JTIC=1 TO NTIC
      1000000
      1000000
      1000000
      1000000
      1000000
REM   REPEAT ONCE FOR EACH NCNSTRNT(JSET)
REM   TICBEGCN(JCN), TICENDCN(JCN), ACCEPTPC!(JCN), CNTYPE(JCN)
        1   1  .5  0
        2   2  .5  0
        3   3  .5  0
        4   4  .5  0
        5   5  .5  0
REM ZSET$(JSET)
    MILL 3 SOFTWOOD
REM  PRSET!(JSET,JTIC),JTIC=1 TO NTIC
    30 40 50 40 30
REM  EWGHT(JSET, JTIC), JTIC=1 TO NTIC
      0 0 0 1 1
REM  NCNSTRNT(JSET)  IF = 0, THEN NEXT SET.
    5
REM  CONSTRAINTDC(JSET), FLOATDC(JSET), SMOOTHDC(JSET), SHAPEDC(JSET)
      0             0            0           0
REM  STOPFLOATPC!(JSET)
      1.0
REM  NFLOATBREAK(JSET), NSMOOTHBREAK(JSET), NSHAPEBREAK(JSET)
       3             3             3
```

```
REM  FLOATBREAK(JSET, JBREAK), FLOATVAL!(JSET, JBREAK) -- 1 LINE PER BREAK
      10                .1
      50               1.00
      100              1.50
REM  SMOOTHBREAK(JSET, JBREAK), SMOOTHVAL!(JSET, JBREAK) -- 1 LINE PER BREAK
      10                .02
      50                .30
      100               .40
REM  SHAPEBREAK(JSET, JBREAK), SHAPEVAL!(JSET, JBREAK) -- 1 LINE PER BREAK
      5                .05
      20                .50
      100               1.50
REM   RELATIVE WEIGHTS BY PERIOD FOR SHAPE ADJUSTMENT- ONE SET FOR EACH TIC
REM  (SHAPEWGHT(JSET, JTIC, JJTIC),JJTIC=1 TO NTIC) -- JTIC * JTIC VALUES
      3 2 1 0 0
      2 3 2 1 0
      1 2 3 2 1
      0 1 2 3 2
      0 0 1 2 3
REM  PROCESSCOSTINP!(JSET)
      25
REM  GOAL!(JSET, JTIC), JTIC=1 TO NTIC
     1500000
     1500000
     1500000
     1500000
     1500000
REM   REPEAT ONCE FOR EACH NCNSTRNT(JSET)
REM   TICBEGCN(JCN), TICENDCN(JCN), ACCEPTPC!(JCN), CNTYPE(JCN)
       1   1  .5  0
       2   2  .5  0
       3   3  .5  0
       4   4  .5  0
       5   5  .5  0
REM ZSET$(JSET)
   COSTS
REM  PRSET!(JSET,JTIC),JTIC=1 TO NTIC
   -1 -1 -1 -1 -1
REM  EWGHT(JSET, JTIC), JTIC=1 TO NTIC
     0 0 0 1 1
REM  NCNSTRNT(JSET)  IF = 0, THEN NEXT SET.
      0
```

The Output File

The following is the output file produced for the example input file after 221 iterations. Creation of the file was triggered by using the edit option in DTRAN after iteration number 220 to turn on all three table output options.

SUMMARY OF --PRODUCT SET-- FLOWS FOR ITERATION 221

PRODUCT SET = MILL 1, ASPEN

TIME PERIOD	PRICE	FLO
1	33.64465	1498450
2	39.77335	1490971
3	46.66181	1487557
4	55.07441	1463069
5	55.4	1448561

PRODUCT SET = MILL 2 ASPEN

TIME PERIOD	PRICE	FLO
1	30.52932	1004462
2	36.56838	1006513
3	43.21217	997466.3
4	50.87407	968937.9
5	51.43187	973177.3

PRODUCT SET = MILL 3 ASPEN

TIME PERIOD	PRICE	FLO
1	28.28127	1507324
2	33.30925	1492392
3	38.64045	1505637
4	44.71804	1462240
5	44.40218	1469055

PRODUCT SET = MILL 4 ASPEN

TIME PERIOD	PRICE	FLO
1	26.73092	1000473
2	32.3133	996160.9
3	38.30956	1004991
4	44.83915	978709.7
5	44.39649	974471.1

PRODUCT SET = MILL 1 BIRCH

TIME PERIOD	PRICE	FLO
1	20.26017	1016460
2	22.60888	992529.7
3	24.82421	1014859
4	27.49257	1013328
5	27.41661	1043867

PRODUCT SET = MILL 2 BIRCH

TIME PERIOD	PRICE	FLO
1	17.29981	497377.4
2	19.39238	496016.6
3	21.26827	504783.5
4	23.34385	518123.7
5	23.04444	521297.6

PRODUCT SET = MILL 3 BIRCH

TIME PERIOD	PRICE	FLO
1	8.285419	497009.9
2	10.36718	495033.7
3	11.93394	503274.9
4	13.40414	541174.8
5	13.29251	516351.7

PRODUCT SET = MILL 1 SOFTWOOD

TIME PERIOD	PRICE	FLO
1	31.18632	1513906
2	35.74613	1546225
3	40.70757	1328864
4	46.02636	1349265
5	50.20138	1515408

PRODUCT SET = MILL 4 SOFTWOOD

TIME PERIOD	PRICE	FLO
1	26.07385	980415.8
2	30.38619	946706.7
3	35.45724	1203393
4	40.71073	1122482
5	44.82239	963276.8

PRODUCT SET = MILL 3 SOFTWOOD

TIME PERIOD	PRICE	FLO
1	35.47701	1503305
2	39.76795	1496399
3	44.62167	1499514
4	49.44666	1498887
5	52.70695	1487275

PRODUCT SET = COSTS

TIME PERIOD	PRICE	FLO
1	-1	1.363091E+08
2	-1	1.372617E+08
3	-1	1.38445E+08
4	-1	1.432709E+08
5	-1	1.511572E+08

SUMMARY OF --PRODUCT TYPE-- FLOWS FOR ITERATION 221

*** PRODUCT TYPE = PRESCRIPTION DOLLAR COSTS

TIME PERIOD	PRICE	FLO
1	-1	136309103
2	-1	137261679
3	-1	138444960
4	-1	143270864
5	-1	151157192

*** PRODUCT TYPE = ASPEN
*** MARKET 1

TIME PERIOD	PRICE	FLO
1	3.364465	14984499
2	3.977335	14909713
3	4.666181	14875570
4	5.507441	14630688
5	5.54	14485611

*** PRODUCT TYPE = ASPEN
*** MARKET 2

TIME PERIOD	PRICE	FLO
1	3.052932	10044618
2	3.656838	10065128
3	4.321218	9974663
4	5.087408	9689379
5	5.143187	9731773

*** PRODUCT TYPE = ASPEN
*** MARKET 3

TIME PERIOD	PRICE	FLO
1	2.828127	15073241
2	3.330925	14923918
3	3.864045	15056370
4	4.471805	14622402
5	4.440218	14690546

*** PRODUCT TYPE = ASPEN
*** MARKET 4

TIME PERIOD	PRICE	FLO
1	2.673092	10004728
2	3.23133	9961609
3	3.830956	10049908
4	4.483915	9787097
5	4.439649	9744711

*** PRODUCT TYPE = ASPEN
*** MARKET 5

TIME PERIOD	PRICE	FLO
1	0	0
2	0	0
3	0	0
4	0	0
5	0	0

*** PRODUCT TYPE = BIRCH
*** MARKET 1

TIME PERIOD	PRICE	FLO
1	2.026017	10164602
2	2.260888	9925297
3	2.482421	10148594
4	2.749257	10133279
5	2.741661	10438671

*** PRODUCT TYPE = BIRCH
*** MARKET 2

TIME PERIOD	PRICE	FLO
1	1.729981	4973774
2	1.939238	4960166
3	2.126827	5047835
4	2.334385	5181237
5	2.304444	5212976

*** PRODUCT TYPE = BIRCH
*** MARKET 3

TIME PERIOD	PRICE	FLO
1	.8285419	4970099
2	1.036718	4950337
3	1.193394	5032749
4	1.340414	5411748
5	1.329251	5163517

*** PRODUCT TYPE = BIRCH
*** MARKET 4

TIME PERIOD	PRICE	FLO
1	0	0
2	0	0
3	0	0
4	0	0
5	0	0

*** PRODUCT TYPE = BIRCH
*** MARKET 5

TIME PERIOD	PRICE	FLO
1	0	62755
2	0	56851
3	0	173113
4	0	190786
5	0	105049

*** PRODUCT TYPE = SOFTWOOD
*** MARKET 1

TIME PERIOD	PRICE	FLO
1	3.118632	15139055
2	3.574613	15462247
3	4.070757	13288640
4	4.602635	13492654
5	5.020138	15154081

*** PRODUCT TYPE = SOFTWOOD
*** MARKET 2

TIME PERIOD	PRICE	FLO
1	0	0
2	0	0
3	0	0
4	0	0
5	0	0

*** PRODUCT TYPE = SOFTWOOD
*** MARKET 3

TIME PERIOD	PRICE	FLO
1	0	0
2	0	0
3	0	0
4	0	0
5	0	0

*** PRODUCT TYPE = SOFTWOOD
*** MARKET 4

TIME PERIOD	PRICE	FLO
1	2.607385	9804158
2	3.038619	9467067
3	3.545724	12033928
4	4.071073	11224815
5	4.482239	9632768

*** PRODUCT TYPE = SOFTWOOD
*** MARKET 5

TIME PERIOD	PRICE	FLO
1	3.547701	15033048
2	3.976795	14963990
3	4.462167	14995138
4	4.944665	14988869
5	5.270695	14872753

MILL 1, ASPEN
SUMMARY OF CONSTRAINTS FOR ITERATION 221

FIRST TIC	LAST TIC	DISCOUNTED FLOW	DISCOUNTED GOAL	PERCENT DEVIATION
1	1	1498450	1500000	-.1033417
2	2	1490971	1500000	-.6019083
3	3	1487557	1500000	-.8295333
4	4	1463069	1500000	-2.462075
5	5	1448561	1500000	-3.429258

MILL 2, ASPEN
SUMMARY OF CONSTRAINTS FOR ITERATION 221

FIRST TIC	LAST TIC	DISCOUNTED FLOW	DISCOUNTED GOAL	PERCENT DEVIATION
1	1	1004462	1000000	.4461812
2	2	1006513	1000000	.6512812
3	3	997466.3	1000000	-.2533687
4	4	968937.9	1000000	-3.106206
5	5	973177.3	1000000	-2.682269

MILL 3 ASPEN

SUMMARY OF CONSTRAINTS FOR ITERATION 221

FIRST TIC	LAST TIC	DISCOUNTED FLOW	DISCOUNTED GOAL	PERCENT DEVIATION
1	1	1507324	1500000	.488275
2	2	1492392	1500000	-.5072083
3	3	1505637	1500000	.3758
4	4	1462240	1500000	-2.517317
5	5	1469055	1500000	-2.063025

MILL 4 ASPEN

SUMMARY OF CONSTRAINTS FOR ITERATION 221

FIRST TIC	LAST TIC	DISCOUNTED FLOW	DISCOUNTED GOAL	PERCENT DEVIATION
1	1	1000473	1000000	4.728125E-02
2	2	996160.9	1000000	-.3839062
3	3	1004991	1000000	.4990813
4	4	978709.7	1000000	-2.129031
5	5	974471.1	1000000	-2.552887

MILL 1 BIRCH

SUMMARY OF CONSTRAINTS FOR ITERATION 221

FIRST TIC	LAST TIC	DISCOUNTED FLOW	DISCOUNTED GOAL	PERCENT DEVIATION
1	1	1016460	1000000	1.646019
2	2	992529.7	1000000	-.7470313
3	3	1014859	1000000	1.485944
4	4	1013328	1000000	1.332794
5	5	1043867	1000000	4.386713

MILL 2 BIRCH

SUMMARY OF CONSTRAINTS FOR ITERATION 221

FIRST TIC	LAST TIC	DISCOUNTED FLOW	DISCOUNTED GOAL	PERCENT DEVIATION
1	1	497377.4	500000	-.5245187
2	2	496016.6	500000	-.7966812
3	3	504783.5	500000	.9567
4	4	518123.7	500000	3.624744
5	5	521297.6	500000	4.259519

MILL 3 BIRCH
SUMMARY OF CONSTRAINTS FOR ITERATION 221

FIRST TIC	LAST TIC	DISCOUNTED FLOW	DISCOUNTED GOAL	PERCENT DEVIATION
1	1	497009.9	500000	-.5980188
2	2	495033.7	500000	-.9932563
3	3	503274.9	500000	.6549813
4	4	541174.8	500000	8.234962
5	5	516351.7	500000	3.270344

MILL 1 SOFTWOOD
SUMMARY OF CONSTRAINTS FOR ITERATION 221

FIRST TIC	LAST TIC	DISCOUNTED FLOW	DISCOUNTED GOAL	PERCENT DEVIATION
1	1	1513906	1500000	.9270333
2	2	1546225	1500000	3.08165
3	3	1328864	1500000	-11.40907
4	4	1349265	1500000	-10.04897
5	5	1515408	1500000	1.027208

MILL 4 SOFTWOOD
SUMMARY OF CONSTRAINTS FOR ITERATION 221

FIRST TIC	LAST TIC	DISCOUNTED FLOW	DISCOUNTED GOAL	PERCENT DEVIATION
1	1	980415.8	1000000	-1.958419
2	2	946706.7	1000000	-5.329331
3	3	1203393	1000000	20.33929
4	4	1122482	1000000	12.24815
5	5	963276.8	1000000	-3.672319

MILL 5 SOFTWOOD
SUMMARY OF CONSTRAINTS FOR ITERATION 221

FIRST TIC	LAST TIC	DISCOUNTED FLOW	DISCOUNTED GOAL	PERCENT DEVIATION
1	1	1503305	1500000	.220325
2	2	1496399	1500000	-.2400667
3	3	1499514	1500000	-3.240833E-02
4	4	1498887	1500000	-7.420833E-02
5	5	1487275	1500000	-.8483083

IX. REFERENCES

Bailey, R.G. (editor). 1986. Proceedings of the workshop on lessons from using FORPLAN. April 29-May 1, 1986 in Denver. U.S.D.A. For. Serv. Land Mgmt. Planning, Washington, DC. 268 p.

Davis, L.S. and K.N. Johnson. 1987. Forest Management. 3rd Ed. New York. McGraw-Hill. 790pp.

Hoganson, H.M. and D.W. Rose. 1984. A simulation approach for optimal timber management scheduling. For. Sci. 30(1):220-238.

Hoganson, H. M. 1987. Impacts of wood energy use on timber supply in northeastern Minnesota. In: Achieving wood energy potentials in Northeastern Minnesota. USDA Forest Service, General Technical Report NC-114, pp. 13-31.

Hoganson, H. M. and D. W. Rose. 1987. A model for recognizing forest-wide risk in timber management scheduling. For. Sci. 33(2):268-281.

Hoganson, H. M. 1988. Utilizing basic concepts of forest economics to solve large timber management scheduling problems. In proceedings: Forest Products Research Society's National Procurement Conference. Atlanta, Georgia (in press).

Hoganson, H. M. and D. W. Rose. 1989. DUALPLAN Version 1.0 User's manual. University of Minnesota Department of Forest Resources Staff Series Paper No. 73. 48 pp.

Johnson, K.N. and E.L. Scheurman. 1977. Techniques for prescribing optimal timber harvest and investment under different objectives--discussion and synthesis. For. Sci. Mono. 18. 31pp.

Kapple, D. C. and H. M. Hoganson. 1991. GISTRAN user's guide. University of Minnesota Department of Forest Resources Staff Series Paper No. **. (in progress)

McDill, M. RxWRITE users guide. 1991. University of Minnesota Department of Forest Resources Staff Series Paper No. ** (in progress)

USDA Forest Service. 1979. A generalized forest growth projection system applied to the Lake States Region. USDA Gen. Tech. Rep. NC-49. 96 pp.