Decision Tree Modeling Using Integrated Multilevel Stochastic Networks

Mohamed Moussa[1]; Janaka Ruwanpura, M.ASCE[2]; and George Jergeas[3]

Abstract: Decision trees (DTs) have proven to be valuable tools for decision making. The common approach for using DTs is calculating the expected value (EV) based on single-number estimates, but the single-number EV method has limited the DTs' real-life applications to a narrow scope of decision problems. This paper introduces the stochastic multilevel decision tree (MLDT) modeling approach, which is useful for analyzing decision problems characterized by uncertainty and complexity. The MLDT's advantages are shown through a computer simulation program: the Decision Support Simulation System (DSSS). The DSSS allows users to model probabilistic linear graph networks and provides a hierarchical modeling method for modeling decision trees to present uncertainties more accurately. It consists of three modules: tree analysis networks (TANs), the shortest and longest path dynamic programming analysis network, and cost time analysis networks. The paper only discusses the TAN module by presenting the MLDT concept under the TAN of the DSSS computer application. The content of the paper includes the modeling approach, its advantages, and examples that can be used in modeling stochastic trees. The DT-DSSS was verified by conducting several tests and validated by using it extensively for undergraduate courses in civil engineering at the University of Calgary for the last two academic years.

DOI: 10.1061/(ASCE)0733-9364(2006)132:12(1254)

CE Database subject headings: Decision making; Models; Stochastic models; Networks; Construction industry.

Introduction

This paper introduces the multilevel network as a modeling method for analyzing decision problems by demonstrating the application of the multilevel networks in modeling decision tree (DT) problems. First, background information on the decision tree concept is provided, the expected value (EV) approach of decision trees is explained, and the disadvantages of the EV approach are presented. Background on stochastic decision trees is also provided, with a brief description of the limitations and challenges associated with stochastic decision trees.

Next, the multilevel decision tree (MLDT) is introduced by explaining the modeling specification requirements of the MLDT as the basis for development of a computer simulation application: the Decision Support Simulation System (DSSS). The process of modeling a decision problem and the calculation procedure using the MLDT are also explained, with an example that illustrates the DSSS's outputs. Finally, the paper provides a summary of the tests and verifications made to ensure that the DSSS incorporates the modeling specifications and produces correct results.

Background

DTs have been regarded as suitable tools for making decisions under risk because of their ability to represent the probability of the consequences of decisions. They have been used in problems related to a wide range of knowledge areas, such as legal problems [e.g., Blodgett (1986)] and research methods [e.g., Mock (1972)], in addition to financial [e.g., Hespos and Strassmann (1965)], engineering [e.g., Benjamin and Cornell (1970)], and medical [e.g., Hazen et al. (1998)] applications.

DTs are solved by calculating the EV; the decision that yields the optimum EV is the decision that should be taken (see the appendix and Fig. 1). The EV approach has been covered in a wide range of publications [e.g., Raiffa and Schlaifer (1961); Meredith et al. (1973); Taha (1997)]. The EV has been criticized for being unsuitable for one-time decisions, though it is suitable for repetitive decisions. To make the DT suitable for one-time decisions, attempts have been made to use the utility-converted units of the decision maker. However, using the utility-converted factors is practically difficult and more of a theoretical approach than a practical one (Wideman 1992), and it does not provide a true picture of the ranges of the decision's possible outcomes.

Most scholars tend to agree that the EV approach is suitable for repetitive decisions because it represents the average outcomes for the long run. However, the EV approach does not work for all repetitive problems; it is suitable only for situations where the decisions are not only repetitive but also deterministic, and the realization of their chance events is estimated using a mass prob-

[1]Ph.D. Candidate, Project Management Specialization, Dept. of Civil Engineering, Schulich School of Engineering, Univ. of Calgary, Calgary AB Canada T2N 1N4. E-mail: mohmoussa@shaw.ca

[2]Director and Associate Professor, Project Management Specialization, Dept. of Civil Engineering, Schulich School of Engineering, Univ. of Calgary, Calgary AB Canada T2N 1N4 (corresponding author). E-mail: janaka@ucalgary.ca

[3]Professor, Project Management Specialization, Dept. of Civil Engineering, Schulich School of Engineering, Univ. of Calgary, Calgary AB Canada T2N 1N4. E-mail: gjergeas@hotmail. com

Note. Discussion open until May 1, 2007. Separate discussions must be submitted for individual papers. To extend the closing date by one month, a written request must be filed with the ASCE Managing Editor. The manuscript for this paper was submitted for review and possible publication on June 15, 2005; approved on March 20, 2006. This paper is part of the ***Journal of Construction Engineering and Management***, Vol. 132, No. 12, December 1, 2006. ©ASCE, ISSN 0733-9364/2006/12-1254–1266/$25.00.

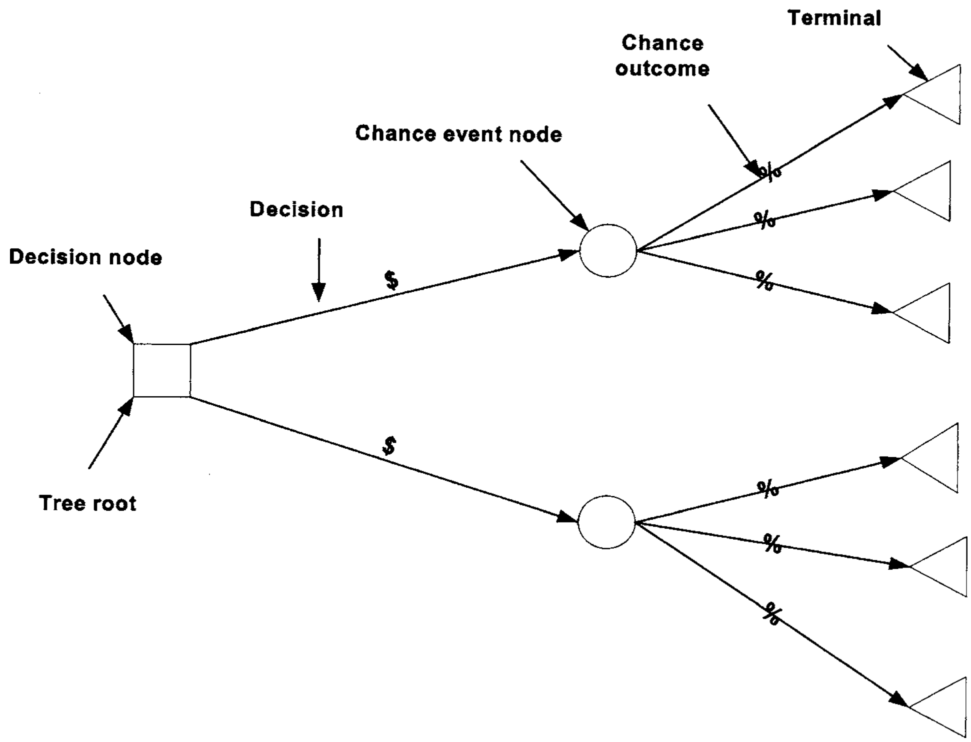


Fig. 1. Conventional decision tree components

ability function distribution. This fits the repetitive gambling example where the decision maker

- Assumes full control over all the decisions' variables (cost, effort, etc.) along all stages of the decision problem;
- Knows with certainty the magnitude of the chance events; and
- Is able to estimate with certainty the probability of each chance event (based on the number of possible outcomes relevant to all possible outcomes).

The EV based on a single number estimate is then suitable only for a very limited scope of applications, which limits the usefulness of DTs to a very narrow scope of applications and makes them unsuitable for most real-life engineering decisions—even the repetitive ones. For more discussion on the shortfalls of the single-number EV approach, readers may refer to Schuyler (2001), Hespos and Strassmann (1965), Smith et al. (1983), and Ferrara and Hayya (1970).

The one-time and repetitive decisions characterized by the probabilistic nature of their parameters require treatment that differs from the single-number EV method. The stochastic approach of DTs overcomes several of the limitations of the single-number EV and can be suitable for such problems. It can examine the DTs under a wide range of variables using ranges rather than single estimates, which makes it more appropriate for uncertainty modeling.

However, very few attempts have been made in modeling using stochastic DTs. Hespos and Strassmann (1965) used stochastic trees in a financial application. Hazen and Pellissier (1996) used stochastic trees in medical decision applications, but the main challenge of stochastic trees—similar to other stochastic modeling—is their ability to represent the uncertainties of the decision problem without sacrificing the necessary details in a framework that can be used for generic decision problems. This paper introduces an MLDT modeling approach that can be used in a wide range of DT applications and presents a simulation application that supports the modeling concept.

In the MLDT, each of the variables could be represented by a range estimate, which includes the probabilistic realization of chance events, the values of the chance events, and the values of decisions. Decisions can be decomposed into smaller components that represent their detailed breakdown. Each decision or combination of decisions can then be evaluated separately or collectively, and the overall decision situation or any part thereof can be evaluated. In the case of repetitive decisions, the EV can be calculated and the optimality index obtained for each decision. The decision optimality index identifies the probability that a decision falls on the optimum decision path (e.g., in a repetitive decision). By examining the decisions' variability (range of outcomes) and the optimality index (optimization probability), the decisions' volatility compared with their optimality can be assessed. Therefore, the decision maker can analyze the decisions based not only on their long-term outcomes' optimality (probability), but also on the risks inherent in the decisions (volatility).

The following section highlights the challenges related to stochastic DT modeling and introduces the need of MLDT for better uncertainty modeling.

Stochastic Decision Trees

The use of the probability distribution functions in DTs was introduced in Hespos and Strassmann (1965), who call DTs that use the EV approach a "conventional decision tree" and the DTs that use the probabilistic estimate a "stochastic decision tree." The same differentiation is used in this paper. Hazen and Pellissier (1996) define a stochastic tree as an "extension of a DT that facilitates the modeling of temporal uncertainties" and permit the use of probability estimates or frequency distributions for some or all factors affecting decisions (Hespos and Strassmann 1965).

Difficulties associated with modeling stochastic DTs arise in four main aspects:

1. The first relates to the ability of the modeling environment to model generic DT problems. While conventional DTs can be solved mathematically using simple arithmetic, simple probabilistic problems require more advanced concepts. With complex problems, mathematical solutions become more difficult. Simulation provides an appealing method for solving complex problems, but most of the simulation platforms are problem specific: that is, reprogramming or program modification is required for each problem. The DT simulation platform should be able to model general types of DTs without the need for reprogramming.

   Special purpose simulation (SPS) allows an experienced simulationist to build a template (a collection of modeling elements targeted for a single domain) that can be used by inexperienced users (Hajjar and AbouRizk 2002). The application of this paper uses Simphony, an SPS computer platform (AbouRizk and Hajjar 1998; Hajjar and AbouRizk 2002); hence, the modeling environment can be used as a generic tool for a wide range of DT problems.
2. The second difficulty relates to the decision model's ability to incorporate the complexity of the decision problems and be dynamic enough to incorporate more details (uncertainties) as information becomes available. Complexity can be addressed in two directions: horizontal and vertical. The horizontal direction relates to the sequential characteristic of decisions along the time line (i.e., each decision leads to another decision or to a state of nature on which other decisions can be based), and the vertical direction relates to the decomposability of decisions.

   The horizontal direction relates to controlling the tree size. One advantage of stochastic DTs is their ability to examine the tree under a wide range of values in a size relatively smaller than conventional trees (Hespos and Strassmann 1965). However, the size of the tree can still go "wild" (large), where several sequential decisions and several chance events are required. With MLDTs, decisions can be broken down in a hierarchical scheme, allowing the tree to expand vertically to accommodate decomposition of decisions, reducing the tree size to manageable decision stages.

   Decomposability relates to the need to break down a decision to smaller components to permit better modeling of the uncertainties. To avoid the need to decompose decisions, modelers use a high-level estimate where decisions are presented using single-density distribution functions, regardless of the number and variability of the states of nature that affect the decision. For example, the cost of a decision such as "Drill a well" would be represented based on only one continuous standard distribution function, although this decision might be affected by several activities and resources that are in turn affected by several uncertainties, such as productivity, soil conditions, and weather conditions.

   The cumulative impact of such uncertainties on the decision could be hard to simplify with a single distribution function. If these uncertainties are simplified by one range (e.g., using a triangular distribution function) of numbers, the results would not truly represent the statistical uncertainty of the decision and could lead to a wrong decision. If the modeler selects to incorporate these uncertainties in a one-level tree, the tree would get unnecessarily large. Therefore, a stochastic DT should be able to carry a hierarchical scheme that allows decisions' decomposability.
3. The third problem relates to estimating the magnitude and probability of realization of chance events when the magnitude, number, and probability of realization of chance events are based on judgment. The one-level stochastic DTs assume that one chance event may originate from a chance node. A chance event would then carry a range of variables represented by a density distribution function. Such an approach assumes that stochastic DTs should be perfectly stochastic: that is, all branches of the tree are to be realized in each simulation run, with states driven by the randomness. This assumption is not always true and does not respond to the yes/no realization of events. Moreover, one or both events may be best modeled in ranges. The situation becomes more challenging when the estimate of the realization of chance events depends on experts' judgments, where a precise estimate of the probability of events realization is not practical. Experts normally provide these probabilities in ranges; for example, the probability of realization of an event could be 90 to 95%.
4. The fourth difficulty arises when one or more decisions in the tree are dependent on the optimization of other sequential decisions, for example, selection of a design method based on life-cost optimization. A single-level tree structure forces the modeler to neglect the variability of the independent decision alternatives and to estimate an approximate, most probable figure to the dependent decisions. The decision maker considers the range of possible outcomes of a single alternative and uses the estimated values of this alternative in modeling the problem at hand, which produces a risk/benefit profile that does not represent the problem. For example, a dependent decision A could be affected by selection of a replacement decision of a piece of equipment (optimization shortest path/dynamic programming problem). Under the one-level tree, the modeler would estimate the cost of one alternative based on the assumption that it is the most likely optimum solution. This creates a false reflection of the decision problem. Considering the other alternatives would then create a true presentation of the system. This is most important when the optimization is to be taken in the future, and the decision maker wants to keep all options open or wants to understand the overall problem behavior.

The MLDT overcomes these difficulties and provides a new method of looking at decision problems. It offers a method of analyzing the problem at hand in a hierarchical way that permits a better representation of the uncertainties. In the next section, the multilevel approach is explained.

Multilevel Decision Tree Conceptual Framework

A stochastic multilevel tree is defined here as a stochastic tree whose variables can be estimated by decomposing the tree elements into smaller components so that the variables can be estimated in ranges. The multilevel tree consists of several layers. The DT is modeled in a top layer in which each of the decisions could be further broken down to another layer (in a lower level). These lower-level layers could have another tree, another network, or a group of variables (e.g., resources such as materials, worker-hours, and so forth). Similarly, any component in this second layer can be broken down to include other components in lower layers. The values of these components could be estimated as a single estimate or as ranges.

The tree breakdown structure enables the modeler to estimate the tree using other types of networks and/or any number of pa-

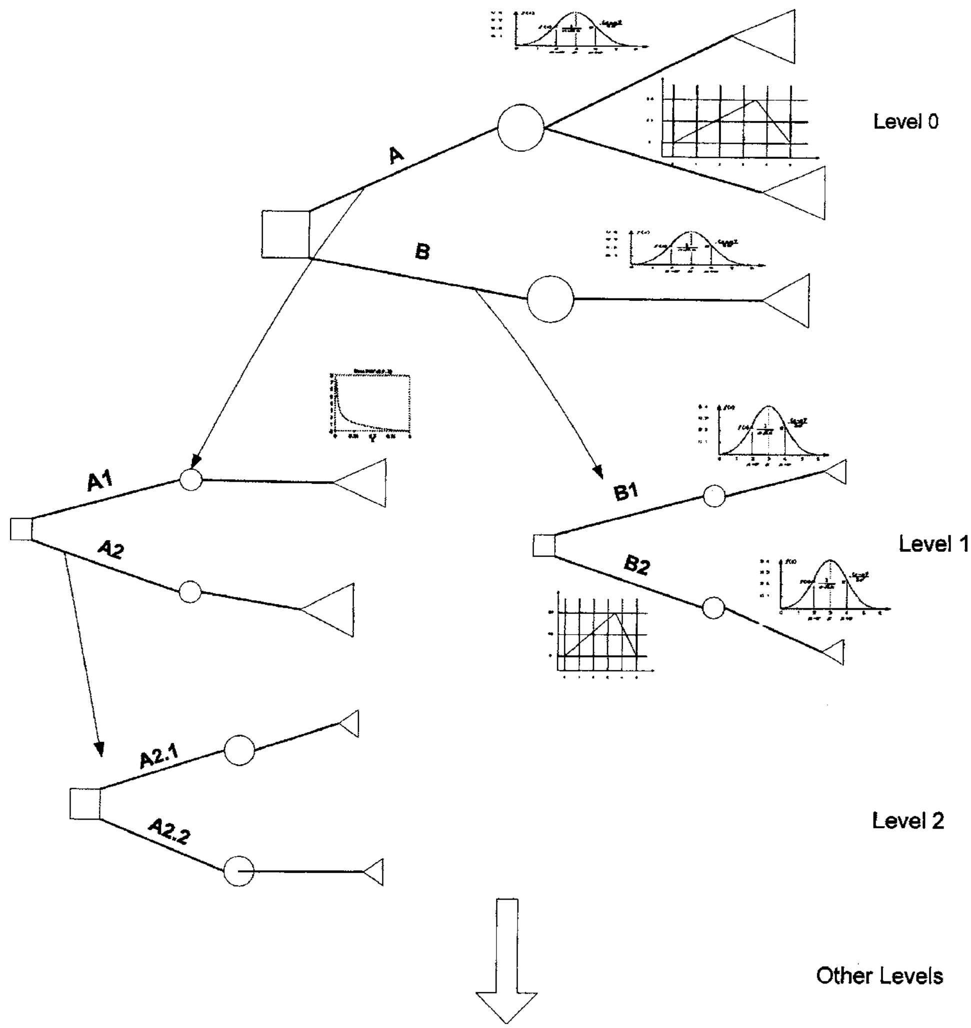


Fig. 2. Conceptual diagram illustrating MLDT

rameters. Fig. 2 shows a conceptual design of an MLDT that consists of two decisions: A and B. Each of these decisions is broken down to another decision tree, so decision A in level 0 is the outcome of solving the decision tree in level 1 (by inheriting the EV of the level 1 DT). The solution of the trees in level 1 (e.g., the selection between decision Al and A2) provides the decision in level 0 with the values (e.g., the cost) of the decisions. Such values are used in solving the tree in level 0. Similarly, any decision in level 1 may be further broken down to DTs. Decisions in an MLDT may get stochastic or single-number estimates.

While Fig. 2 shows the subsystems in the lower layers as DTs, subsystems could be a stochastic optimization shortest/longest route network (using the dynamic programming algorithm), a cost/time network (range estimate), or several resources/parameters. An explanation of how other networks can be modeled inside the MLDT is outside of the scope of this paper [for more explanation of other types of networks, see Moussa et al. (2004)].

A branch in an MLDT can be perfectly stochastic, that is, all chance nodes in the branch precede only one chance event. During simulation, all the elements in the branch are realized; a branch can be semiperfect, that is, some but not all of its chance nodes precede more than one chance event. The realization of the branch depends on the probability assigned to its chance events. For example, the branches connected to decision A in level 0—Fig. 2—are imperfectly stochastic, while that connected to decision B is perfectly stochastic. Values along a perfectly or imperfectly stochastic branch may be deterministic or stochastic (estimated in ranges). During simulation, perfectly stochastic branches realize 100% of the time, while the imperfectly stochastic ones realize a number of times driven by the probability associated to them. On the other hand, decision A is a perfectly stochastic decision because it is realized 100% of the time; however, the chance events succeeding it are imperfectly stochastic because they are realized only part of the time.

Decision Support Simulation System Template

The DSSS template developed within an SPS platform includes the elements required to model three modules: tree analysis network (TAN), shortest and longest path dynamic programming analysis network (DPAN), and cost time analysis network (CTAN). A DSSS TAN module supports both single-level DTs and MLDTs. The template consists of 17 modeling elements, which are color coded so that elements with similar colors are used by one module. The TAN module uses nine modeling elements, which are also required to model a DT.

Fig. 3 shows a conceptual illustration of a DT using DSSS notations. The DSSS has three elements additional to the conventional components shown in Fig. 1. The "DT Branches Sum"

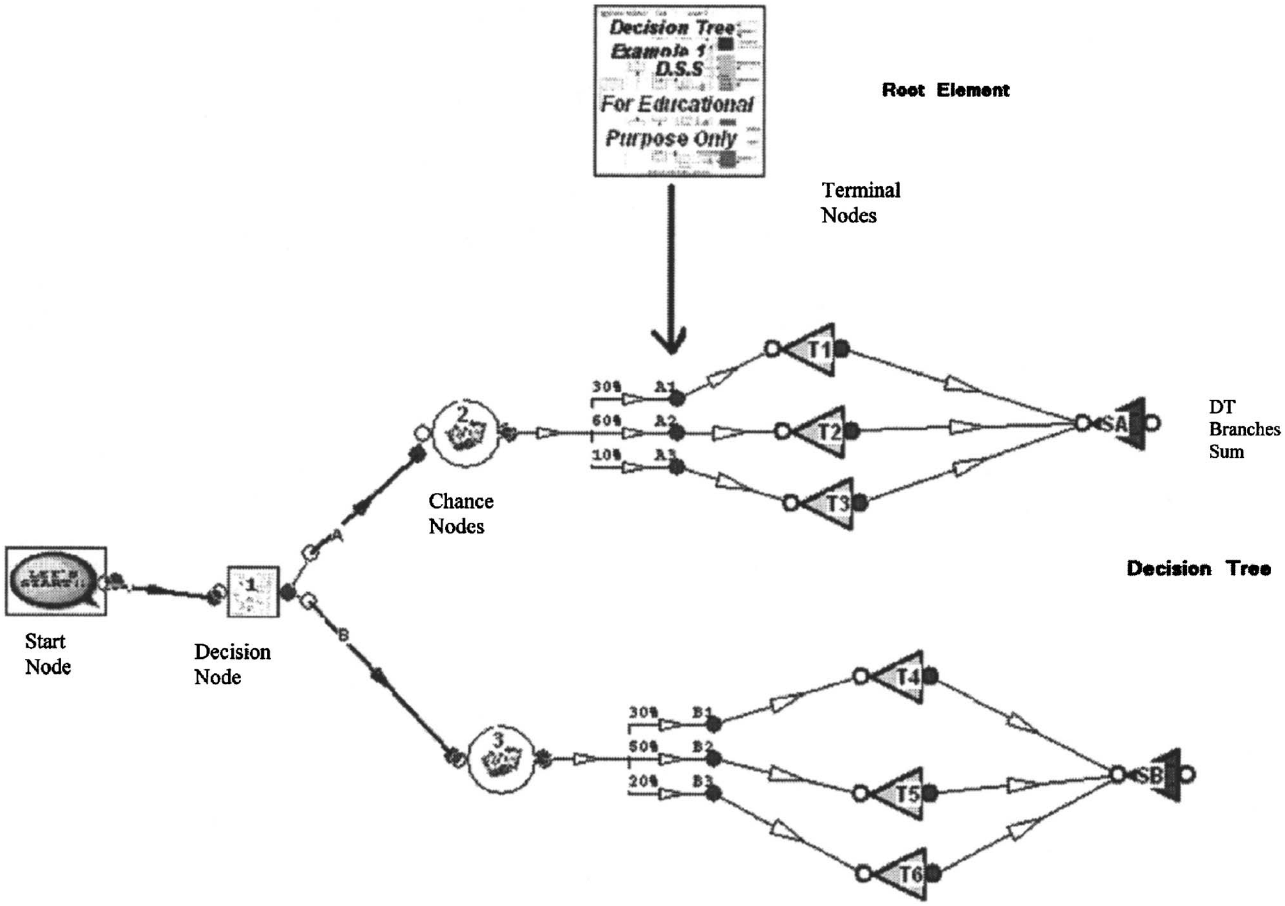


Fig. 3. Decision tree-DSSS main hierarchical structure

component collects the payoffs of connected realized tree branches during simulation to produce the utility/risk profile of any combination of decisions. The "Start Node" creates the required entities for the simulation (i.e., initiates the simulation). The "Root Element" is the root of the vertical hierarchy of a model; it works as an overall envelope to a model and collects the results of calculating a model. For example, in a DT, the root element collects the EV of the tree; in a CTAN, the root element collects the overall cost and duration of the network; and in a shortest/longest route network (DPAN), the root element collects the value of the optimum route. The other six elements are a decision node, a chance node, a terminal node, a decisions element, and two elements for chance events.

A network in the DSSS should be constructed in the lower level of the root element, which collects information from the model created in its lower level. The root element may be inserted inside an element of a model so that the model may be constructed as a child to an element. If the root element is constructed in a lower level of an element, the information collected from its lower-level model is transferred to the higher-level element, and the model in the higher level is then calculated using this information. The MLDT is therefore calculated from the bottom up by calculating the models and parameters in the lower levels first and then transferring the value of these models up to the upper-level elements through their root elements and so on, up until the first root element of the overall model.

For example, the conceptual illustration in Fig. 4 shows the root element at the top of the figure. Underneath the root element is a decision tree that represents a hypothetical example of selection between two decisions regarding the construction of a new terminal: (1) modify an existing harbor, or (2) construct a new harbor. Consider that these decisions do not have to be broken down to other decisions. Each of these two projects consists of activities that need resources (labor, equipment, materials, and so forth). The cost of the resources depends on the quantity of work to be done, the time frame of the activities, the market conditions, and so forth.

Collectively, the project cost depends on its overall duration. Such uncertainties in the project cost can be modeled using a stochastic cost and time analysis network (CTAN). Therefore, for example, the decision "construct a new harbor" has a root element in its lower level. The root element collects the cost of the network in its lower level and transfers this cost up to the decision branch in the decision tree; that is, it collects the sum of the costs of realized activities and transfers this cost to be the cost of the decision "construct a new harbor." The same would be done if the other decision has a CTAN in its lower level. The DT is then calculated with the costs inherited from lower levels. Similarly, any of the activities in the CTAN could be broken down to another CTAN, to a decision network, or to resources. When values are estimated in ranges using a density function, the simulation uses these density functions to generate the random numbers and hence calculates the networks using these generated numbers. At the end of simulation, the statistical results of the calculation are collected, and the modeler can retrieve them in a tabular or graphic format for analysis.

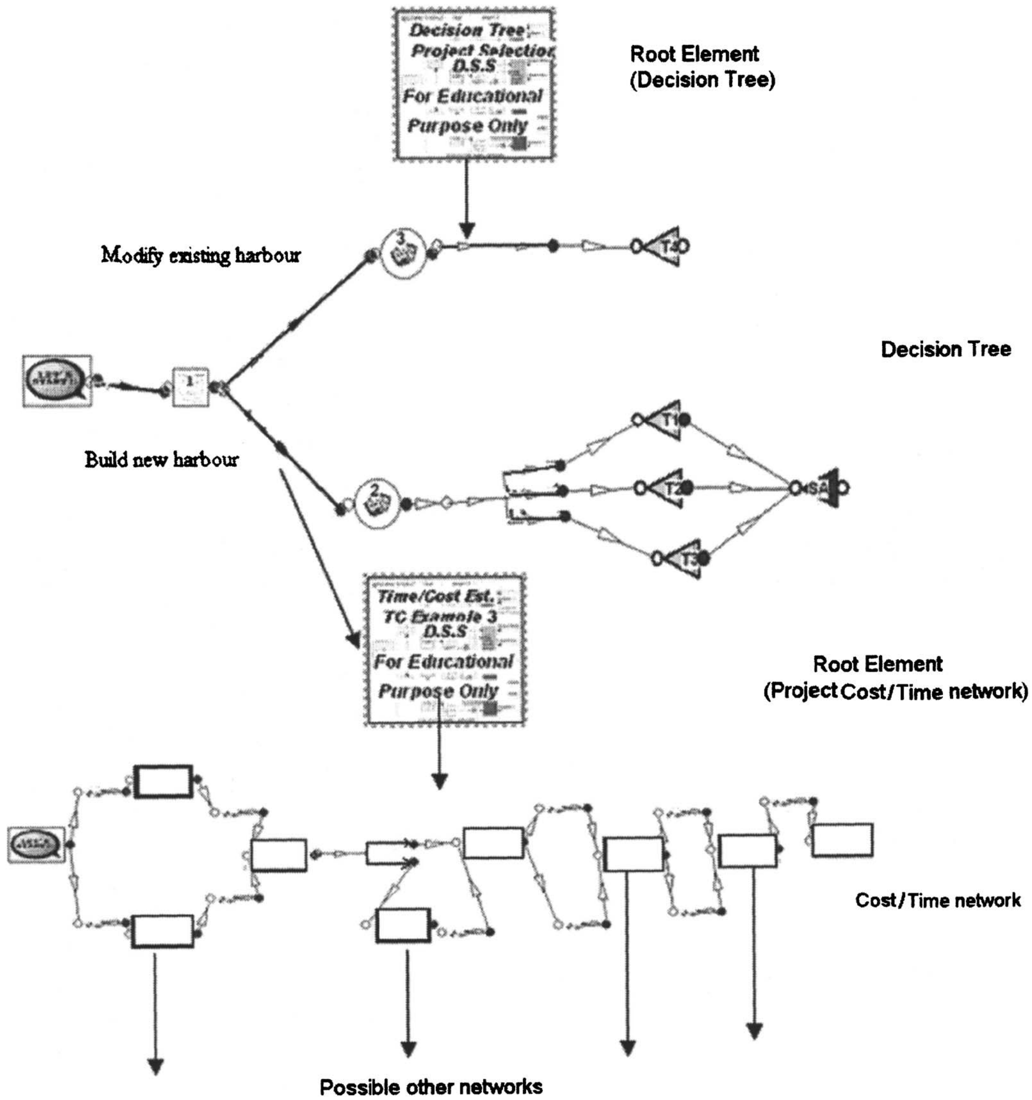


Fig. 4. Decision tree example using DSSS

Multilevel Stochastic Decision Trees Modeling Specifications

The following specifications have been identified as implementation requirements for a TAN in a multilevel network environment. They were successfully incorporated in the existing DSSS. While these specifications represent features incorporated in the DSSS, they are summarized here to present the modeling concept of the MLDTs. The emphasis is in the modeling characteristic, not in the computer program.

- A node (a decision or chance node) may be preceded by any number of branches. Each branch could have probabilistic (density distribution) or/and constant values.
- It should be possible to estimate the cost of a decision by decomposing the decision to its smaller components where the cost of the decision is the sum of these components. Also, the cost of decisions could be estimated as a lump-sum estimate (whether it is a constant or probabilistic estimate). Decisions can be assigned any number of resources (e.g., labor, materials, money, efforts, and so forth). The value of each resource can be a function of the quantity of the resource and the value of its unit of measurement. Both resource quantity and unit rate can be either a constant or a density distribution function.
- The decision's cost may be estimated as a function of the cost of another network. By supporting a hierarchal structure, the output of a network calculation/solution could be the input of an element in another network. For example, the cost of a decision in a DT problem may be estimated as the cost of a project resulting from a CTAN model.
- The modeler should be able to estimate the realization probability of the chance events using single numbers (deterministic) or/and a range of the likely probabilities. If all chance events at a chance node are deterministic, the sum of the probability should be equal to 100%. If the chance events are estimated by ranges, the total upper or lower limit could be not equal to 100%. For example, in a three chance-events node, the user may specify the probability of outcomes as 30 to 40% for the first event, 60 to 75% for the second event, and 5 to 10% for the third event. Such flexibility gives users a more realistic approach in estimating the likelihood of events' occurrence based on judgment. Chance events labeled by such probabilities can be estimated as ranges or/and as single numbers.
- The MLDT should permit feedback to previous decision nodes or event nodes and accept loops (cycles) based on probabilistic

estimates. It also should be able to model coalesced decision trees (decision trees that contain branches that can be collapsed where outgoing links meet in a common node).
- The MLDT should do a forward path calculation to provide the ranges of possible outcomes (payoffs) at each terminal node. The forward path should also collect values along realized branches to produce the utility/risk profile at the end of the simulation. The utility/risk profile is the cumulative density function (CDF) from which the modeler can estimate the value of the branch at any probability level.
- The MLDT should be able to do a backward calculation (folding back the tree) by calculating the EV at each element. Using the EV, the tree should provide the optimality index for the decisions (probability that the decision falls on the optimum solution of the tree, that is, the optimum path that yields the EV). The EV calculation should be based on minimization or maximization optimization criteria.
- The EV in an MLDT is calculated and provided in a range of possible outcomes. In the case of repetitive decisions, the range of the EV of a decision represents the decision's long-term range of outcomes, and the optimality index provides the probability that the decision is the optimum decision. In one-time decisions where the tree branches are perfectly stochastic, the range of EVs represents the range of possible payoffs; hence, the optimality index represents the probability that a decision will fall on the optimum decision path.
- The modeler should be able to retrieve the statistical distribution of the input values as well as the output values (results of calculations) at any element. The user should be able to retrieve the calculation results at any node/element in the tree either as statistical distribution results (CDF graph or tabular format) or as a database table that shows the results for each run.
- The platform should be able to model generic decision trees without the need for reprogramming.

The specifications above have been implemented in the DSSS; accordingly, the presentation of the DSSS in this paper will reflect the criteria specified for an MLDT. Although the MLDT is an extension of conventional trees, constructing an MLDT requires a different decision analysis and building structure. The following section explains the method for analyzing a problem using the MLDT approach.

DT-DSSS Modeling Process

The first step in building an MLDT is to establish the DT hierarchy. The first level of the hierarchy includes the main milestones of the decision processes. Each milestone is characterized by one or more major decisions that characterize the decision process. Then, each of the decisions in the first level can be broken down further, as explained earlier, to other types of networks or resources. Each element in any network can also be decomposed to other networks or resources. The number of levels below the different decisions of the first level may vary based on the breakdown required for each decision.

Fig. 5 shows a hypothetical example of an MLDT. The example shows a DT for the selection between two design methods for an offshore structure. The designers are considering the design of a jetty using piles or quay walls. Each of these decisions could have consequences that relate to the life expectancy of the structure, the maintenance cost during the life cycle, the construction cost, and so forth. The first alternative can be broken down to either concrete piles or steel piles. The tree decisions (the quay wall decision, the concrete piles decision, and the steel piles decision) can then be decomposed to a CTAN. The CTAN of each decision can consist of a cost/time network represented in a precedence logic diagram that outlines the activities' sequence of work. The CTAN carries the uncertainties related to activities' costs, durations, alternative relations, and so forth [see Moussa et al. (2005) for more about the CTAN]. Activities may also be assigned resources such as barges, cranes, concrete, and steel. Each resource may be assigned uncertainties that affect its quantity or unit cost (e.g., a soft soil condition may realize a probability of 15%; if realized, the pile length will increase by 40 to 60%. Similarly, activities could represent decision points and be connected to DTs in a lower level.

As seen in the example above, the decision maker establishes the problem decision hierarchy from top down. The next section describes the calculation of the MLDT.

MLDT-Calculation Procedure

During each simulation run, the MLDT generates random numbers that fit the user input density functions. User inputs include
- Cost and/or benefits of decisions;
- Cost and/or benefits of chance events;
- Probability of chance events (single/constant probability or range probability); and
- Values of any parameter in lower-level layers.

The tree is then calculated using the following order:
1. Networks at the lowest level are calculated first. Based on the discussion above, a layer may be any or a combination of the following:
   - DT: the tree is calculated as explained below.
   - CTAN (project cost/time analysis network): the network is calculated by summing the cost of its realized activities (CTAN models activity networks in a precedence diagram with or without a probabilistic estimate for the realization of the activities).
   - DPAN (a shortest/longest route optimization network using dynamic programming algorithm: the network is calculated according to the optimization criteria specified by the user).
   - Resources: the resources cost values are summed (resources cost = unit price $\times$ unit cost $\times$ impact of any risks that might be assigned to the resource).

   The values calculated by the networks and resources are then transferred to the upper-level element. Once all lower elements of a network are calculated, the network in this layer is calculated (each depending on the type of network). The process continues till the first-level DT.
2. When all the parameters of a decision tree (at any level) are calculated (or assigned), the DT performs the forward path calculation. Through the forward path, the DT calculates the probability of realization of each element (in the case of imperfectly stochastic branches) and calculates the payoffs at the terminals. The probability of realization of an element is the product of probabilities of chance nodes preceding it. The payoff is the algebraic sum of all costs and benefits of a branch and represents a possible outcome from a decision.

   For imperfectly stochastic branches, the DSSS generates independent random numbers during each run at each chance node. Based on this generated number, one branch is realized. If a branch is realized, the utility/risk factor of this branch is assigned a value that equals the payoff calculated earlier. If the branch is not realized, the utility/risk factor gets

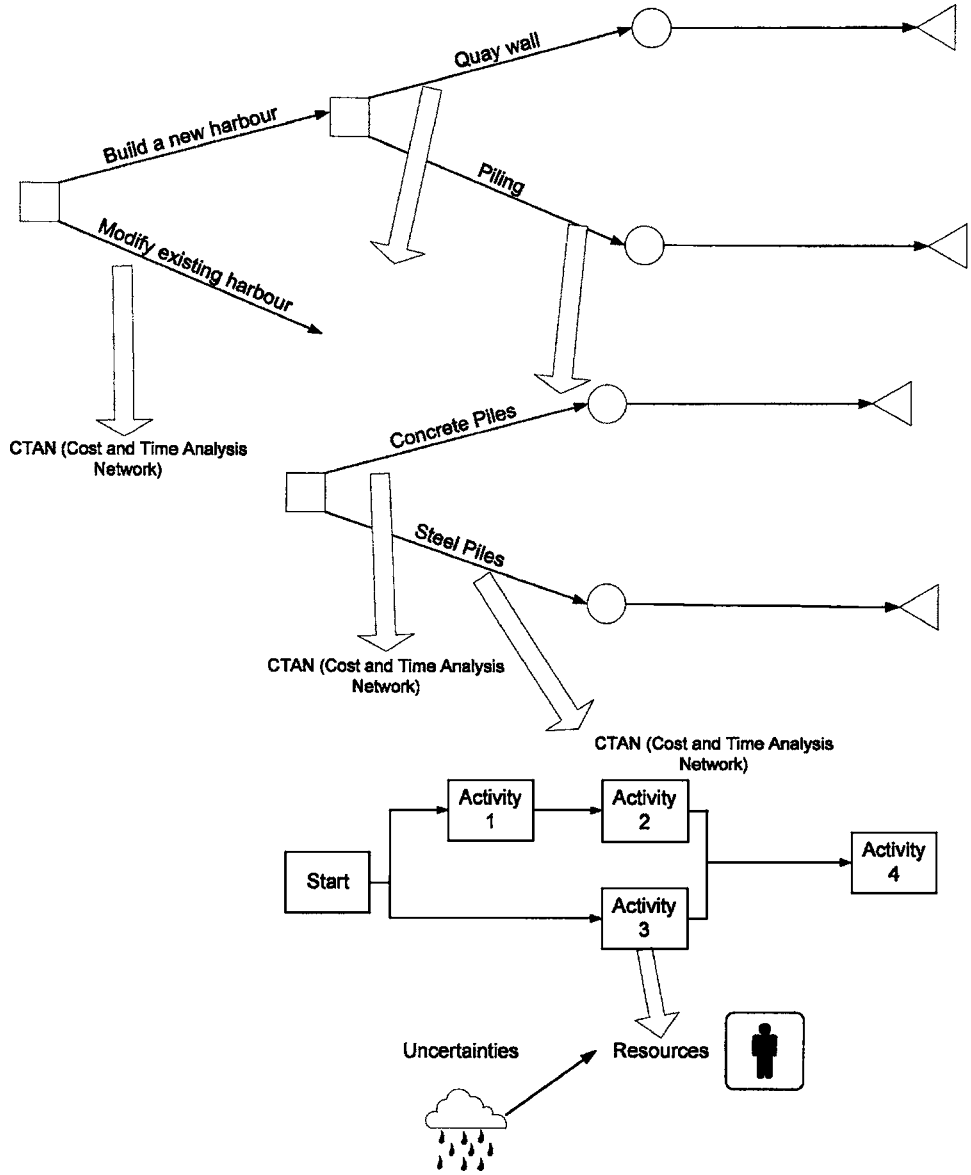


Fig. 5. MLD tree example

a zero value. The chance nodes work as filters that pass values only for realized branches. The program collects the statistics of utility/risk at each terminal, and the statistical results provide the range of outputs associated with a branch versus the probability of realization of this branch; hence they indicate the utility/risk associated with the branch terminal.

The payoff represents the range of possible outcomes based on a scale of probability that ranges from 0 to 100%. It then provides the range of outcomes of a branch independent of other decisions/branches. In contrast, the utility/risk at the terminal provides the possible outcome in relation to all other consequences that originate from the root decision. Therefore, if a branch is realized 10% of the time during simulation, the utility/risk profile at the terminal is the payoff in relation to these 10% chances or realizations. The CDF graph gives the range of payoffs for the probability between 90 to 100%; that is, at a probability less than 90%, the branch payoff is zero, while for the 10% range between 90 and 100%, each probability would have a relevant value indicating the payoff at this particular confidence level. More illustrations of the difference between the payoff and the utility concept are provided later in the paper.

The tree also performs a backward calculation. The program calculates the EV at each element starting from the payoff values calculated in the previous step similar to the algorithm used in conventional trees (see appendix). The EV of an element is transferred to the previous element based on the optimization criteria specified by the modeler (maximum or minimum). When the backward path is complete and the EV of the first node is determined, the path leading to this EV is marked.

3. The processes described above are repeated during every simulation run. (Each simulation iteration generates independent random numbers and provides each element in the multilevel network with a different value than that given in other simulation runs.) At the end of simulation, the DSSS calculates the optimality index for each element (the probability for a decision to fall on the path that yields the optimum EV):

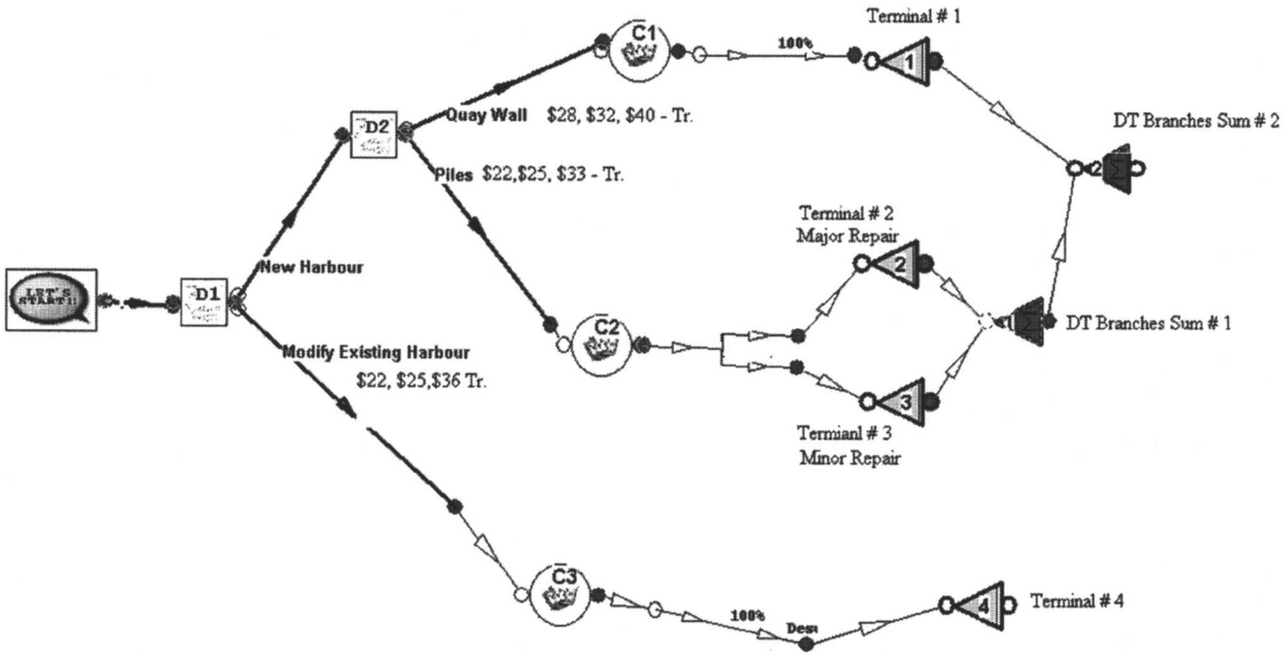


Fig. 6. Stochastic decision tree example

$$\text{Optimality index} = \frac{\text{number of times element falls on tree solution path}}{\text{number of iterations}} \quad (1)$$

4. Finally, the DSSS collects the statistical results for the inputs so that the user may review his/her estimates, including values delivered from other networks or from the costs of allocated resource parameters and for the calculated results (i.e., the EVs, the payoffs at terminals, and the utility/risk for any combination of branches).

DT-DSSS Results and Interpretations

The fundamental difference between conventional and stochastic trees is that the first provides a single solution, that is, it tells the modeler which decision should be taken while the stochastic trees provide a range of outcomes. The range of outcomes shows the behavior of the decision problem for the purpose of understanding it. The decision maker is then able to take the decision that best reflects his/her preference/utility. In the case of repetitive decisions or perfectly stochastic decisions, the MLDT provides the optimality index that reflects the probability associated with each decision being an optimum decision.

As a result of running a simulation a sufficient number of times, the MLDT of the DSSS provides the following results:

- Payoffs at each tree terminal in a CDF graph and a tabular form at each branch end: These provide the decision maker with the range of possible outcomes for each possible scenario.
- Utility/risk at each terminal or any combination of terminals: The "DT Branches Sum" of the DSSS can be connected to the branches' terminals to collect the outcomes of each realized branch and produce the statistical results accordingly. When several branches are connected, the information at the "DT Branches Sum" represents a decision at an early stage.
- The EV of any decision: The EV represents the average range of possible outcomes in the long run. In perfectly stochastic branches, the EV equals the payoff of the branch and hence also shows the utility/risk of the decision.
- Optimality index: When the decision problem is repetitive in nature (e.g., investment/reinvestment decisions) or when the DT is perfectly stochastic (all terminals are realized during all simulation runs), the optimality index represents the probability that a decision is the best decision.
- Results at each DT element (nodes and links): At each node, the DSSS provides the CDF graph and the ranges of results using a tabular form that shows the model inputs/outputs.

To explain the use of the DSSS results, an example is given. The example extends from those shown in Figs. 4 and 5. Consider a decision of either modifying an existing harbor or building a new one. If a decision is made to build a new one, the new construction can be built on quay walls or on piles. The cost of the quay wall construction is estimated as a triangular distribution of $28,000–$32,000–$40,000 (all numbers are in thousands). The average estimated maintenance cost over the life of the project is estimated to be $6,000 with a standard deviation of $1,200. The piling option is estimated as a triangular distribution of $22,000–$25,000–$33,000. At a 5 to 10% probability, the piling structure may require major repairs along its life; the cost of repair and maintenance is estimated as a triangular distribution of $15,000, $18,000, and $22,000. If no major repair is required, however, the average cost is estimated to be $6,000 with a standard deviation of $1,200. The cost of modifying an existing harbor is estimated as a triangular distribution of $22,000, $25,000, and $36,000, and the estimated average maintenance cost is $10,000 with a standard deviation of $2,500.

Fig. 6 shows the decision tree of this problem as modeled by the DSSS. (For the purposes of simplicity and because of limited space, this example does not include lower level networks.) Values in the graph are in millions. As shown in Fig. 6, the two terminals of the piling decision are connected with a "DT

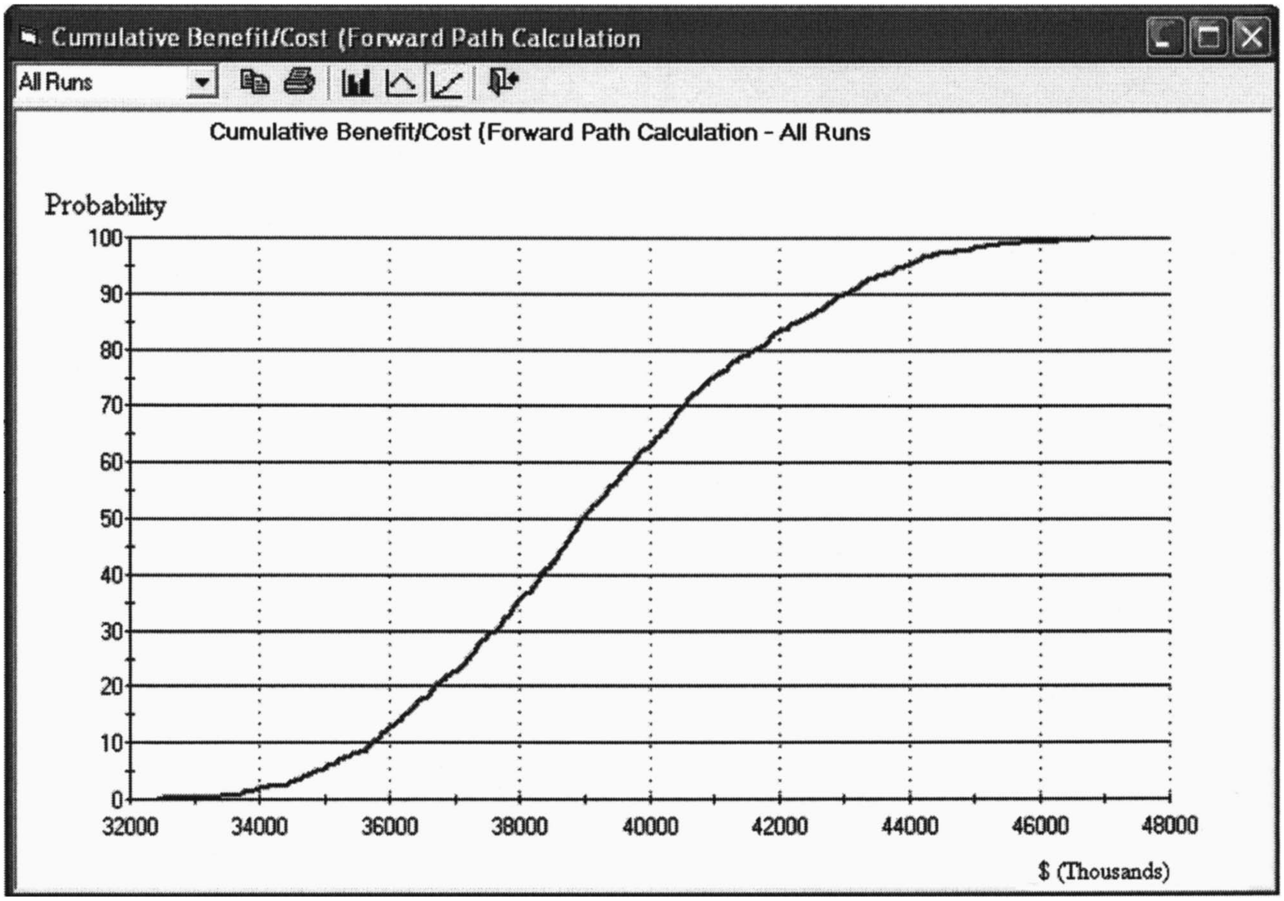


Fig. 7. Stochastic decision tree example—CDF graph at terminal 1

Branches Sum" node (Sum 1). This node collects information from the terminals preceding it to show the utility/risk for the piling decision. Similarly, the node Sum 2 is connected to the terminal of the Quay Wall decision and the node Sum 1 to show the utility/risk of the new harbor decision. The end terminal of modifying the existing harbor does not require a "DT Branches Sum" node because it relates to only one decision.

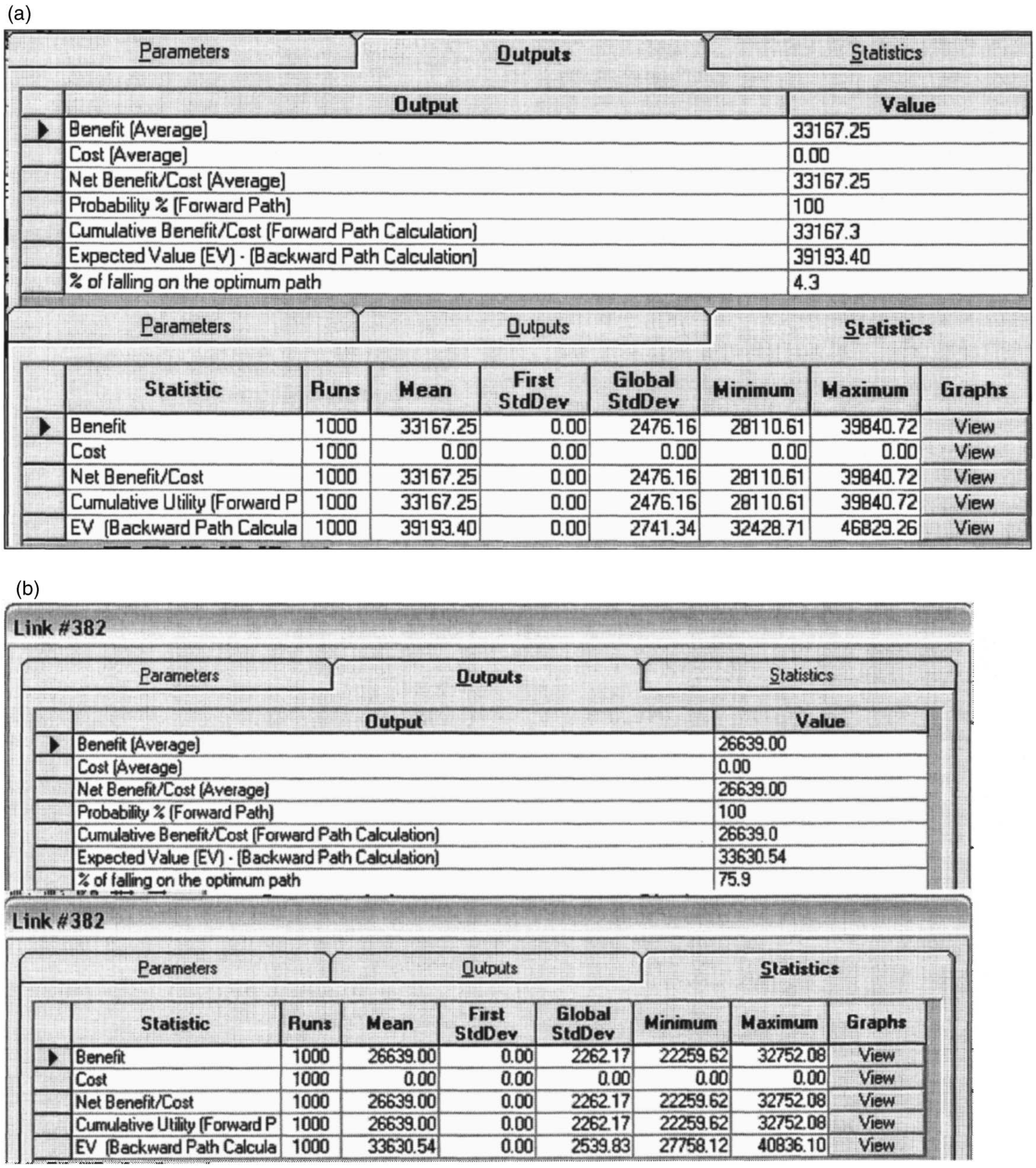
The following explains some of the results that may be obtained from the MLDT of the DSSS:

- Fig. 7 shows the CDF graph at terminal 1 (the node terminates the branch "new harbour-quay wall." The figure shows the CDF graph for this path. The branch is a perfectly stochastic branch, and hence the payoff, the EV, and the utility/risk are the same. From the CDF graph, it is possible to estimate the cost at any confidence level. For example, at a 70% confidence level, the cost is approximately $41,000 or less. Similar CDF can be obtained by investigating the results at the decision "quay wall." The tabular results at this decision show the average EV (also the payoff and risk/utility) and its standard deviation (Fig. 8). Fig. 8(a) shows that the quay wall decision is realized 100% of the time-and that the optimality index is only 4.3%. Results relating to the input at the decision are also given. Fig. 8(b) shows that the optimality index for the pile decision is 75.9%.
- The results of the calculations at terminal 2 and terminal # 3 may be retrieved in CDF graphs and in tabular forms. Fig. 9 shows the utility/risk of terminal 3. The figure also shows that the results of the curve fall between approximately 8% to 100%: i.e., a range of 92%. (The results of terminal 2 are then from 92 to 100%: i.e., in a range of 8%). The graph shows the consequences of the decision if only minor repairs are required.
- The analyst may obtain the results at any node and compare results to understand how each decision behaves. Based on his/her utility, the decision maker is then able to select the decision that best reflects his/her risk tolerance.
- Once a decision is made (or once the alternatives are narrowed down to a smaller number of decisions), the analyst can eliminate other decisions from the model and run the simulation again to study further the selected decision(s). The rerun of the simulation after eliminating discarded decisions will produce results that can be studied. The decision maker can estimate his/her budget based on the results of the simulation based on the risk tolerance level of the organization.

With the MLDT, the decision maker has the tool to support his/her position rationally and estimate more accurately the risk exposure of his/her problem by better modeling of the uncertainty. The MLDT provides better modeling of uncertainty than does the single level because of its ability to reflect the complexity of decisions without sacrificing the necessary details.

Testing and Verification

To ensure that the simulation platform provides the expected results and incorporates the specifications described earlier, several tests and verifications were conducted on the DSSS. Below is a


(a)

Parameters | Outputs | Statistics

	Output	Value
▶	Benefit (Average)	33167.25
	Cost (Average)	0.00
	Net Benefit/Cost (Average)	33167.25
	Probability % (Forward Path)	100
	Cumulative Benefit/Cost (Forward Path Calculation)	33167.3
	Expected Value (EV) - (Backward Path Calculation)	39193.40
	% of falling on the optimum path	4.3

Parameters | Outputs | Statistics

	Statistic	Runs	Mean	First StdDev	Global StdDev	Minimum	Maximum	Graphs
▶	Benefit	1000	33167.25	0.00	2476.16	28110.61	39840.72	View
	Cost	1000	0.00	0.00	0.00	0.00	0.00	View
	Net Benefit/Cost	1000	33167.25	0.00	2476.16	28110.61	39840.72	View
	Cumulative Utility (Forward P	1000	33167.25	0.00	2476.16	28110.61	39840.72	View
	EV (Backward Path Calcula	1000	39193.40	0.00	2741.34	32428.71	46829.26	View

(b)

Link #382

Parameters | Outputs | Statistics

	Output	Value
▶	Benefit (Average)	26639.00
	Cost (Average)	0.00
	Net Benefit/Cost (Average)	26639.00
	Probability % (Forward Path)	100
	Cumulative Benefit/Cost (Forward Path Calculation)	26639.0
	Expected Value (EV) - (Backward Path Calculation)	33630.54
	% of falling on the optimum path	75.9

Link #382

Parameters | Outputs | Statistics

	Statistic	Runs	Mean	First StdDev	Global StdDev	Minimum	Maximum	Graphs
▶	Benefit	1000	26639.00	0.00	2262.17	22259.62	32752.08	View
	Cost	1000	0.00	0.00	0.00	0.00	0.00	View
	Net Benefit/Cost	1000	26639.00	0.00	2262.17	22259.62	32752.08	View
	Cumulative Utility (Forward P	1000	26639.00	0.00	2262.17	22259.62	32752.08	View
	EV (Backward Path Calcula	1000	33630.54	0.00	2539.83	27758.12	40836.10	View

Fig. 8. Simulation outputs and statistics results: (a) quay wall decision; (b) piling decision

summary of the tests and verification conducted:

- Verification test: performed by executing the program using examples to find errors. The verification test was done in two stages:
  1. During development of the DSSS, several deterministic conventional DTs and deterministic networks were constructed in a multilevel framework. The results of the DSSS were compared with the results of solving the networks manually. The template was also tested using indeterministic examples. Template results using stochastic examples were retrieved using database files, and the results were manually checked for several iterations. Statistical analysis to the results stored in the database files were also conducted and compared with the template results. Wherever applicable, corrections and modification to the template were made.
  2. After developing the tool, the DSSS was used by students in the Project Management Specialization at the University of Calgary at both the graduate and undergraduate levels for the academic years 2003 to 2005. Students were asked to create examples and apply the DSSS to these examples. (Moussa et al. 2005; Ruwanpura et al. 2005).
- Code test: during the coding and development of the DSSS, the logic was examined through test cases to validate the execution of each instruction in the program.
- Unit test (bottom-up testing): each module was divided into several submodules, and each submodule was tested separately to locate errors. For example, the DT is a submodule to the TAN. The DT is then divided into two main processes: forward path and backward path. Each subsection is tested to ensure that the calculations for any and all iterations are correct.
- Up-down test: each module was tested from the upper-level down by checking that the upper-level results collected infor-

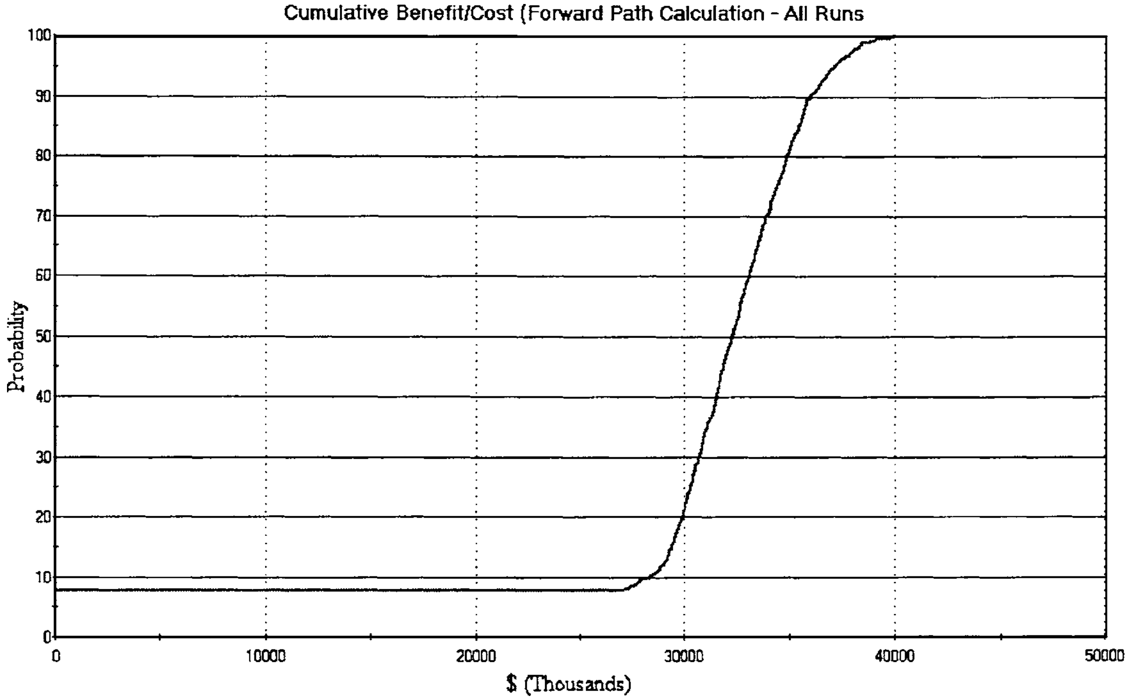


Fig. 9. Terminal 3: CDF graph of utility/risk profile

mation from the submodels correctly and that submodules' interactions did not provide wrong results.

- System test: tested the integration of the components of each module and the interrelation among several modules to ensure that the hierarchal relationship was error free.
- Objectives test: during coding and upon completion of the program development, the program results were checked against the identified specifications to ensure that original objectives, current specifications, and system results were compatible.
- Peak load and performance time test: the fully loaded *Time and Cost* network, which consists of approximately 75 activities, three of which were linked to DTs and two of which were linked to dynamic program networks, was developed to determine whether the program could handle the volume of activities that could occur in a real-life midsize project. The simulation time for 1,000 runs was approximately 60 min.
- Human factors test: the DSSS was given to users at different computer and modeling levels to determine how users would interact with the program. The test aimed to make sure that the program was user friendly. There were no complaints about the difficulty of the program. Modeling a small network that consisted of 20 nodes took novice users 15 to 25 min. Tests were done under the Windows XP operating system using a Pentium 4 and 3 Intel processors.

Conclusion

Decision trees (DTs) provide a powerful method for visualizing and analyzing decisions. Conventional methods of solving DTs do not respond to stochastic needs and are not suitable for most real-life decisions. This paper introduced the MLDT as a better method for uncertainty modeling. The MLDT introduced in the paper is supported by a DSSS computer application that is developed under a special purpose simulation platform. The MLDT overcomes the limitation of decision trees in that it accepts decomposability to decisions and enables modeling complex decisions without sacrificing the detailed presentation of uncertainty in a manageable framework. The modeling computer application presented in the paper accepts constant and density distribution values and integrates with other networks supported by the DSSS template. The DSSS-DT module accepts loops (feedback to previous decisions), allows estimating probability of events based on a range estimate, provides the optimality index for each decision, and provides the utility/risk profile for any decision and any combination of decisions.

The DT module was used by students in the civil engineering program at the University of Calgary at the undergraduate levels for the academic years 2004 to 2005. Students were asked to create examples and apply the DSSS to these examples. Ruwanpura et al. (2005) have emphasized the value of DSSS for teaching purposes: "The civil engineering program at the University of Calgary has successfully taught the risk and simulation related concepts and applications over the last two years. The development of DSSS has enabled students to learn these concepts with ease. Further, the wide range of applications that the DSSS can model makes it a suitable tool for both practical and educational purposes."

Acknowledgments

The writers acknowledge the funding of the Natural Sciences and Engineering Research Council of Canada (NSERC)'s Research Grant Programs–Individual for "Comprehensive Modeling Framework for Managing Construction Projects." The writers wish to acknowledge Dr. Simaan AbouRizk, NSERC/Alberta Construction Industry Research Chair, for his permission to use Simphony to develop DSSS and to use it for teaching purposes.

Appendix. Conventional Decision Trees

Conventional DTs are covered by a wide range of publications (Meredith et al. 1973; Revelle et al. 1997; Hillier and Lieberman 2001; Taha 1997). A brief explanation of the DT components and computational procedures are provided below for easy reference.

A DT consists of five main components (Fig. 1):

- Decision nodes: represented by squares; they precede variables or actions that the decision maker can control.
- Chance event nodes: represented by circles; they precede events that the decision maker cannot control.
- Terminal: end points where outcome values are attached (tree leaves); they present possible outcomes of the tree decisions.
- Decision branches: represent alternative decisions available to decision makers.
- Chance events: represent possible outcomes of decisions; they are assigned probability.

If $X_1, X_2, X_3, \ldots, X_n$ are mutually exclusive chance events that could possibly occur at a chance node, then

$$\sum_{i=1}^{i=n} P(X_i) = 1 \tag{2}$$

where $P(X)$=probability of the realization of chance i that n's chances originated from a chance node.

A tree starts with a decision node. The start node represents the tree's root. Fig. 1 shows a simple DT with two decisions and three chance events resulting from each decision. To solve a DT, the analyst calculates the EV at each node and takes the decision that yields the maximum EV at the root. The following summarizes the EV calculation procedure:

1. Identify alternative decisions and their cost.
2. Identify the possible outcomes of each decision.
3. Estimate the probability of the outcomes; if there are subsequent decisions or outcomes, steps 1 to 3 are to be repeated.
4. Draw the tree chronologically from left to right and calculate the payoffs at the end of each branch.
5. Fold back the tree to calculate the EV and take the decision that has the optimum EV.

A DT is constructed from the bottom up (from time zero up), analyzed from the top down, and finally implemented from the bottom up (Revelle et al. 1997). Values in the tree should be discounted to the net present value to reflect the to-date estimate. The EV is calculated so that

- At chance events, the EV equals the sum of the values of the chances multiplied by their probability

$$EV = \sum_{i=1}^{i=n} X_i P(X) \tag{3}$$

where X=value of chance event i, $P(X)$=probability of realization of chance event value for n chance events.

- At the decision nodes, the EV equals the optimum EV at the node (minimum or maximum as per the utility optimization criterion required).

$$EV = \mathrm{MAX}(U) \tag{4}$$

where U=utility of decision maker.

References

AbouRizk, S. M., and Hajjar, D. (1998). "A framework for applying simulation in construction: NRC Canada 1998." *Can. J. Civ. Eng.*, 25(3), 604–617.

Benjamin, J. R., and Cornell, C. A. (1970). *Probability, statistics, and decision for civil engineers*, McGraw-Hill, New York.

Blodgett, N. (1986). "Decision trees." *ABA Journal*, 72(1), 33.

Ferrara, W. L., and Hayya, J. C. (1970). "Toward probabilistic profit budgets." *Management Accounting*, October, 23–28.

Hajjar, D., and AbouRizk, S. M. (2002). "Unified modeling methodology for construction simulation." *J. Constr. Eng. Manage.*, 128(2), 174–185.

Hazen, G. B., and Pellissier, J. M. (1996). "Recursive utility for stochastic trees." *Oper. Res.*, 44(5), 788–809.

Hazen, G. B., Pellissier, J. M., and Sounderpandian, J. (1998). "Stochastic tree models in medical decision making." *Interfaces*, 28(4), 64–80.

Hespos, R. F., and Strassmann, P. A. (1965). "Stochastic decision trees for the analysis of investment decisions." *Manage. Sci.*, 11(10), B244–B259.

Hillier, F. S., and Lieberman, G. J. (2001). *Introduction to operation research*, 7th Ed., McGraw-Hill, New York.

Meredith, D. D., Wong, K. W., Woodhead, R. W., and Wortman, R. H. (1973). *Design and planning of engineering systems*, Prentice-Hall, Englewood Cliffs, N. J.

Mock, T. J. (1972). "Decision tree approach to the methodological decision process." *Account. Rev.*, 47(4), 826.

Moussa, M., Ruwanpura, J. Y., and Jergeas, G. (2005). "Multi level integrated stochastic networks for cost and time risk assessment." *Proc., CSCE 6th Construction Specialty Conf.*, Canadian Society for Civil Engineering, Toronto.

Raiffa, H., and Schlaifer, R. (1961). *Studies in managerial economics: Applied statistical decision theory*, Graduate School of Business Administration, Harvard Univ., Boston.

Revelle, C. S., Whitlatch, E. E., and Wright, J. R. (1997). *Civil and environmental systems engineering*, Reprinted with correction, June 1999. Prentice-Hall, Upper Saddle River, N.J.

Ruwanpura, J., Moussa, M., and Hewage, K. (2005). "Risk and simulation-based educational module in construction project management for civil engineering undergraduates." *Canadian Civil Engineer*, 22(0) 10–13.

Schuyler, J. (2001). *Risk and decision analysis in projects*, 2nd Ed., Project Management Institute, Newtown Square, Pa., 41–74.

Smith, A. A., Hinton, E., and Lewis, R. W. (1983). *Civil engineering systems analysis and design*, Wiley, New York.

Taha, H. A. (1997). *Operations research: An introduction*, 6th Ed., Prentice-Hall, Upper Saddle River, N.J.

Wideman, M. R., ed. (1992). *Project and program risk management: A guide to managing project risks and opportunities*, Project Management Institute, Newtown Square, Pa.