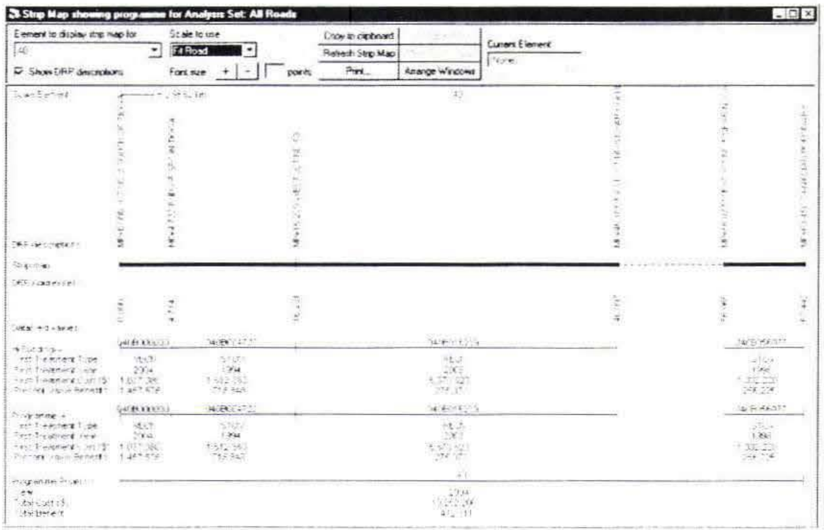
SD98-05-F

South Dakota DOT

Connecting South Dakota and the Nation

SD Department of Transportation
Office of Research



Pavement Management Segment Consolidation

Study SD98-05

Final Report

Prepared by:

deighton

Information tools for managing resources

June 1998

DISCLAIMER

The contents of this report reflect the views of the authors who are responsible for the facts and accuracy of the data presented herein. The contents do not necessarily reflect the official views or policies of the South Dakota Department of Transportation, the State Transportation Commission, or the Federal Highway Administration. This report does not constitute a standard, specification, or regulation.

ACKNOWLEDGEMENTS

This work was performed under the supervision of the SD98-05 Technical Panel:

David Voeltz..................Planning and Programming
Jon Becker...Research
Steve Ulvestad............................Operations Support
Steve Gramm.................Planning and Programming
Gill Hedman.......................Materials and Surfacing
Blair Lunde...Research
Mark Leiferman............Planning and Programming
Ben Orsbon...................Planning and Programming
Brett Hestdalen..FHWA

TECHNICAL REPORT STANDARD TITLE PAGE

1. Report No SD98-05-F	2. Government Accession No	3. Recipient's Catalog No
4. Title and Subtitle SDDOT Pavement Management Segment Consolidation		5. Report Date June 30, 1998
		6. Performing Organization Code
7. Author(s) Richard Deighton		8. Performing Organization Report No
9. Performing Organization Name and Address Deighton Associates Limited 112 King Street East Bowmanville, Ontario, Canada L1C 1N5		10. Work Unit No
		11. Contract or Grant No 310563
12. Sponsoring Agency Name and Address South Dakota Department of Transportation Office of Research 700 East Broadway Avenue Pierre, SD 57501-2586		13. Type of Report and Period Covered Final; October 1997 - June 1998
		14. Sponsoring Agency Code
15. Supplementary Notes		

16. Abstract

During this research project, the consultant investigated how optimal dTIMS strategies on adjacent road segments can be combined into construction projects and modifications to dTIMS were made so that this process can be done both automatically and manually.

The investigation of the process by which optimal strategies can be combined took the form of a questionnaire that was distributed to a group of experienced dTIMS users. The results of the questionnaire were used as a guide to the development of a process within dTIMS that would allow users to define a set of parameters, in decision tree format, that would result in the consolidation of pavement management segments.

The consultant then enhanced dTIMS to include a function called Automatic Programme Development. This function automatically goes through the list of selected strategies and determines which adjacent sections can be combined together into projects. To allow the user to manipulate the projects that are created, dTIMS was also enhanced to include a strip map feature, which visually displays the projects created alongside the optimal strategies. The strip map is a line diagram of a selected road that shows were the elements from the inventory file occur along the road.

17. Keywords		18. Distribution Statement No restrictions. This document is available to the public from the sponsoring agency.	
19. Security Classification (of this report) Unclassified	Security Classification (of this page) Unclassified	21. No. of Pages	22. Price

Table of Contents

1. Executive Summary

This document has been prepared by Deighton Associates Limited (Deighton) and is the final report for project SD98-05: Pavement Management Segment Consolidation. This report will present the results of the study that lead to the enhanced feature in dTIMS that is known as Automatic Programme Development (APD).

1.1 The Study's Purpose

dTIMS is a multi year prioritization computer system used by the South Dakota Department of Transportation (SDDOT) as part of the State's pavement management system. dTIMS selects an "optimal" strategy for each segment in the highway network in such a way as to ensure that the sum of the costs for all segments does not exceed the budget and that the total network wide benefits are maximized.

Unfortunately, dTIMS makes the "optimal" strategy selection for each segment independent of the segment's location. This strategy selection independence is a problem whose symptoms take many forms. For instance, dTIMS can recommend a treatment on one road segment in one year, and the same treatment on an adjacent segment in the next year.

It takes a tremendous (impractical) amount of time for SDDOT staff to sit down with dTIMS and rationalize reasonable construction projects by manually combining adjacent road segments. dTIMS, therefore, required a procedure that would reduce this incredible time burden down to a reasonable level. Such a feature will combine adjacent segments and make minor adjustments to the optimal strategy to come up with a recommended, practical set of construction projects. Then, the feature should allow a user to sit down and manually confirm and/or adjust the resulting project sections, treatments and timing using a strip map interface.

From a construction programming point of view it is most economical to create construction projects that are as long as possible. Some of the benefits of longer projects are, for instance: they save money in plant mobilization costs, and they reduce the impact of traffic obstruction.

1.2 Approach to Study

The approach to the study was centered around Deighton identifying a set of parameters that can be used to automatically combine optimized dTIMS strategies into a first cut set of construction projects and the implementation of the procedures in dTIMS.

The project began with the consultant developing a questionnaire to be distributed to a cross section of experienced dTIMS users including SDDOT staff. The questionnaire was designed with the intent of gaining an insight into the manual processes that are currently used by dTIMS users to develop final construction projects.

The questionnaire was sent to SDDOT for review and distributed following modification to incorporate the comments received by Deighton. Following several weeks the questionnaires were returned and the answers/comments were summarized in a report to SDDOT. The report and its recommendations made by Deighton was the basis for the programming work that followed.

From the results of the questionnaire Deighton concluded that the parameters needed to process the logic required to automatically generate construction projects from optimized section strategies would be user specified by way of decisions trees. The branches of the decision trees would address the following questions regarding each candidate section for inclusion in a construction project.

- Is the maximum project length exceeded?

- Is the treatment compatible with the project treatment?
- Can the treatment be switched?
- Can the treatment year be switched?
- Can the section be skipped?

As the dTIMS users were completing the questionnaires, Deighton began modifying dTIMS to include a strip map function. This work proceeded independent of the automatic project generation phase of this study. It was intended that this feature would be used to display the results of the APD and would allow the user to modify the recommended set of projects manually.

Subsequent to the development of the strip map capabilities in dTIMS the programming of the automatic project generation logic was completed and a rigorous in house testing program was established to identify and correct any faults.

To conclude the project the integrated Help System and Reference Manual were modified to reflect the outcome of the programming. Following this was the publication of a software patch that was distributed to all dTIMS users via the internet so that the current version of dTIMS could be upgraded to version 6.1 and include the features developed under this study.

1.3 Conclusions and Recommendations

At the completion of the project the consultant made the following conclusions and recommendations.

1.3.1 Decision Tree Development

Following the review of the software documentation it is recommended that SDDOT commence with the development of decision tree theory that will be used to model the current programme development process in South Dakota. Any assumptions or background used in the development of the theory should be recorded in the documentation that accompanies the decision tree summary.

1.3.2 Further Testing

The SDDOT decision tree structure should be applied to the SDDOT network for testing the reliability of the APD. Results of the test program should be used to, not only, debug the decision tree setup but, also to give additional feedback to Deighton regarding the functionality of the software.

1.3.3 Automatic Programme Development Implementation

SDDOT must develop an implementation plan that will enable the APD to be used in the development of practical construction projects from the dTIMS optimized construction programme. It may be wise, as part of the implementation plan, to check results from APD with the programme that was produced using current procedures in a previous year.

1.3.4 Review and Revise

Following the generation of the first automatic programme, SDDOT should review the programme developed with the intent of evaluating the suitability of using decision trees as a foundation in the construction programme process.

2. Problem Description

This research project was initiated in 1997 when the South Dakota Department of Transportation (SDDOT) issued a Research Project Statement (RPS). The RPS stated the following in the form of a Problem Description.

SDDOT uses Deighton Associates Limited software, dTIMS, to analyze life-cycle costs for various rehabilitation strategies on each segment of road in its network. To use dTIMS, SDDOT divides its road network into "homogeneous" segments. Homogeneous means that the source data describing the attributes of a segment are the same throughout the segment's entire length. Unfortunately, dTIMS makes the "optimal" strategy selection for each segment independent of the segment's location, length, or adjacent segment status.

> *For instance, dTIMS might recommend a particular repair strategy for one segment, a different strategy for the next and a third strategy for the next. But each of these three segments may not be of sufficient length to be a stand alone project or it might be more cost effective to do one project at one time instead of spreading projects over more years. To combine these segments requires Planning and Programming staff to manually combine segments so that viable construction projects can be proposed.*
>
> *Research is needed to define rules for consolidating segments and to automate those methods in the dTIMS software.*

This is a fundamental problem with dTIMS; not only for SDDOT, but, for dTIMS users all over the world. As a result, this enhancement has been on Deighton's list for enhancements to dTIMS for a number of years. Due to other commitments and priorities, it has remained on the enhancement list. Prior to this project the enhancement was scheduled for a 1999 completion date.

Deighton responded to the RPS and based on their proposal was awarded a contract on November 25th, 1997. The work was jointly funded by SDDOT and Deighton. The software, complete with Reference Manual and an integrated Help System, was delivered to SDDOT on April 30, 1998.

2.1 Background Summary

Dividing roads into "homogeneous" segments has been a major problem for all areas of highway engineering. Yet, there is little or no work done in this area guiding us to a proper theoretical solution to a tough practical problem. Some specific work has been done, but no purely theoretical work. For example, Appendix J in the AASHTO Design of Pavement Structures manual is devoted to describing "a powerful analytical method for delineating statistically homogeneous units from pavement response measurements." This powerful analytical method, however, presupposes that project lengths are already known; how unfortunate.

dROAD has a function called "automatic sectioning" whose purpose is to produce homogeneous sections. This function, however, is not based on rigorous theory. Rather, it is based on brute force. The reason for this is the lack of direction from the literature. This too, by the way, is why the construction project development problem has stayed on the shelf at Deighton for so long.

One of the biggest problems this research project encountered was how to aggregate data from small sections to bigger sections. Issues such as the following required careful study, for example:

1. First, assume that dTIMS has been set up to produce a list of feasible strategies for each segment. Second, assume that the recommended strategy for segment one is not even in the list for segment two. Will this new procedure generate the strategy for segment two so that it can analyze the effect of joining segment one and segment two into construction projects? If not, why? If so, why wasn't dTIMS set up to have the strategy generated for the section segment in the first place?

2. Presumably, when two smaller segments are joined into a larger construction project the cost for the same strategy is not simply the sum of the costs for two individual strategies. (Consider the savings in mobilization costs as an example of why this may happen.) If this is true, then the optimization analysis to initially select the two individual strategies was not using the same information as we are ending up with. Does this mean we are moving away from the optimal solution? Or, is the amount of movement so small it's not significant? Or, do we change the original cost to remove the mobilization factor out of the original cost equations and just add it in later?

3. Using a similar line of thought, when two identical strategies are added for two adjacent segments, are the benefits additive?

4. If a one mile-long 'oddball' segment exists between two fifteen mile segments which have the same recommended strategy, should the oddball be 'thrown in' and the two segments joined together? What if the two segments were each one mile and the oddball was fifteen miles? If you answered "Yes" to the first and "No" to the second, where is the dividing line?

The specific answers to the above questions are not important. What is important, though, is the number of possible questions that must be answered when performing a function such as the one proposed in this project. Deighton had to be sure that as many of these questions were asked before the software code was modified so that the design can be complete. Deighton's law of code bloat says "the number of lines of code is inversely proportional to the amount of forethought that goes into a design."

For this reason, at the last Deighton user meeting an informal group of people was assembled to form the "hop scotch user group." The mandate of the group was to contribute to this forethought.

The new version of dTIMS will have three software functions not currently available in dTIMS. These are:

1. A window which accepts the user's parameters regarding how the construction project generation analysis will act. This screen will get such things as the minimum and maximum project lengths.

2. A function (presumably initiated from a "Execute" button on the above window) which uses the Road, From, and To fields in the DT2699 file to automatically generate construction projects.

3. A window that allows for manual override of the construction projects. This window will use a strip map interface which shows the analysis segments along each road with their optimal strategies. It also shows the resulting construction projects with their strategies. This function will allow the user to adjust the length and strategy for each construction project interactively.

3. Project Objectives

This project has three research objectives. They are listed in the following with a short discussion that is intended to communicate the perception of Deighton's understanding of the objectives.

Objective One: To develop a set of user definable parameters, or a mechanism to handle 'open-ended' parameters which will determine how dTIMS will generate construction projects from the analysis segments and its corresponding list of recommended strategies.

One of the most predominant considerations when making construction projects is *minimum length.* For instance, it is impractical to initiate a 500 foot long construction project in the middle of nowhere. Therefore, anyone can say with certainty that one of the parameters will be minimum length. However, can anyone say that the minimum length parameter is the same for construction projects in urban areas? How about treatment type or funding category? Issues such as this must be identified and clearly understood.

Other parameters are not quite as certain as minimum length. The project must either develop an exhaustive list of prioritized parameters; or it must develop a mechanism to accommodate 'open-ended' parameters. An 'open-ended' parameter is user specified. Without them the next objective will be difficult to satisfy.

Objective Two: To develop procedures that use the above parameters to automatically generate a set of viable construction projects.

This objective prompts the development of a function in dTIMS that can use the parameters to go through the list of analysis segments and test various combinations of them to see which best meets the criteria for a construction project. Building this function will not be trivial. It will possibly have to have a recursive component that keeps calling itself until all criteria are satisfied. Although it is premature to design the function here, we do not want to trivialize the effort required here.

One of the difficulties in demonstrating that this objective has been satisfied is getting a consistent, reproducible definition of viable construction project. The word viable was used here to distinguish the resulting construction project from "optimum." To get the "optimum" construction projects would require a totally different project going in a totally different direction. The basic idea is to produce a set of construction projects that are reasonably close to those that are currently produced by hand using dTIMS' output.

Objective Three: To provide functions in dTIMS which will allow a user to (a) enter the parameters of objective one, (b) perform the analysis of objective two, (c) interactively review and adjust the construction projects which result from the previous function on a strip map, and (d) print and export the results.

Finally, the project must produce results that are actually implemented in dTIMS. After all the symptom we are trying to reduce is the tremendous (impractical) amount of time it currently takes SDDOT staff to sit down and rationalize reasonable construction projects by manually combining adjacent road segments. This objective is aimed at ensuring this symptom is corrected.

The interface will 'borrow' the strip map technology already in use in dROAD version 6.0. The word borrow is used because the objects in dROAD 6.0 are not directly transferable to dTIMS for various reasons. The technology will have to be custom written for dTIMS 6.1.

4. Project Tasks

The list of research tasks accomplished during this project is shown below. The first two tasks, A and B, are general to all Deighton projects. All other tasks, 1 to11, are specific to this project. Each task is presented as it appeared in the proposal and discussed.

Task A: Initiate and maintain the project in Deighton's Quality System.

Over the past several years Deighton Associates has been designing and implementing a Quality System for its off-the-shelf software. This quality system is scheduled for ISO 9000 certification in October 1997.

Before any project involving the software is initiated Deighton Staff must initiate the project in the Quality system. To do this the project manager must sit down and create a file with a number of documents in it. These documents include such things as the Project Plan, the Requirements Document, the Test Plan, just to name a few.

Each of these documents are meticulously maintained throughout the life of the software. The specific steps are described in detail in the Deighton's Quality Procedures Manual.

Task B: Change project scope and work plan to incorporate the comments of SDDOT's project's technical panel.

Deighton adjusted and modified the project scope and workplan to incorporate all reasonable requests and ideas from the SDDOT's technical panel.

The word reasonable was used to make it perfectly clear that Deighton had the final say on all matters dealing with the deliverables of this project. SDDOT had the right to recommend and make suggestions. However, Deighton reserved the right to refuse suggestions that might jeopardize the successful completion of the project. For instance, Deighton would have refused a request by SDDOT to include a button on the manual adjust window which would allow the user to view a video clip of the affected segments.

Preparing and submitting the proposal for this project was how this task was accomplished. In the proposal Deighton described the project, its objectives, and its scope. SDDOT had the opportunity to review the document and suggest changes.

Task One: Write strip map function for dTIMS.

No matter what the results of the remaining tasks were, the end product of this project had to be manifested in the dTIMS software. The primary user interface for this is a strip map function similar to the one used in dROAD 6. (See the "View|Element Locations" function in dROAD 6)

The idea was to use the dROAD 6 strip map function as a prototype for a similar function in dTIMS. The reason a new function had to be built is primarily related to the difference between data structures used in both systems; dROAD has a much more sophisticated data structure than dTIMS.

Nevertheless, building the foundation for this strip map function in dTIMS could commence immediately. Objects such as information windows and buttons were added to the basic strip map later to accommodate specific functions and procedures that were developed in subsequent tasks.

While Deighton was waiting for the results of Task Two work began on the writing of the strip map function within dTIMS. This was the single most time consuming task in the project.

The whole idea of the strip map is to (a) draw a line diagram of a particular road showing where the elements from the inventory file occur along the road. and (b) let the user compare the selected strategies for each element from a particular budget scenario with the selected strategies in another budget scenario that can be manipulated to form programme projects.

Task Two: Investigate and recommend a set of parameters to be used by dTIMS to control the automatic generation of construction projects.

Although this was proposed as Task Two, the project began with the investigation of the parameters to be used by dTIMS to control the APD. This task was tackled first because of the need to solicit input from the SDDOT Technical Panel as well as a number of experienced dTIMS users that were not directly involved in this project. Deighton correctly perceived this as potentially being a time consuming process.

Task Two specifically involved the investigation and recommendation of a set of parameters to be used by dTIMS to control the automatic generation of viable construction projects.

To accomplish the objectives of Task Two, Deighton proposed the following:

- develop a questionnaire which would allow various experienced dTIMS user's to describe the criteria used to decide on construction projects;

- get comments from SDDOT on the questionnaire by fax, modify it, then distribute it to SDDOT as well as other experienced dTIMS users;
- collect the questionnaire and analyze the results;
- summarize the results in the form of a draft report which suggests a list of criteria to be used in the automated procedure;
- distribute the above report to the original participants;
- have a conference call between Deighton and SDDOT to discuss changes to the report;
- incorporate the comments into a revised report and redistribute;
- use the report as the basis for the development of the viable projects enhancement.

The results of the survey are summarized in the dTIMS Viable Projects Questionnaire Summary. Deighton anticipated that the task at hand was going to be complicated. The questionnaire responses confirmed just that. Although Deighton never asked directly for any "secrets" that might make the job easier, none came. Therefore, the problem was investigated from several different angles and an approach developed. Which, in the absence of anything better, is a little complicated, but, will work.

A copy of dTIMS Viable Projects Questionnaire Summary is included in Appendix '1' of this report. It includes the survey questions, answers from the participants and summation by Deighton.

Task Three: Create a window in dTIMS which allows the user to enter the required parameters and to initiate the automatic generation of construction projects.

The general logic required to have dTIMS perform APD was developed as part of the summary of the dTIMS user questionnaire. (See Appendix '1') It was only left to determine how dTIMS would process the logic. The approach taken laid the logic out in a flowchart and identifies the decisions that need to be made. To help dTIMS answer the questions, Deighton developed a decision tree concept that allows the user to define the parameters needed to initiate the automatic generation of construction projects.

Task Four: Write a function which uses the parameters and dTIMS' list of strategies to automatically generate a 'first-cut' set of construction projects.

Deighton developed a function, which uses the user defined decision tree parameters and the list of dTIMS' strategies to automatically generate a set of construction projects. Some initial ideas about this function are listed below (NOTE the highlighted words are specific objects in dTIMS described in the help system):

- A "construction project" will be a set of one or more analysis segments. The function will not subdivide the analysis segments into smaller pieces when forming construction projects.

- The strategy for a construction project will be the aggregation of the ***selected strategies*** for each of its analysis segment's. This means that dTIMS will simply modify the selected strategies of an existing ***Budget Scenario.***

- This function will have the ability to automatically add another strategy to an analysis segment's ***list of strategies***. This strategy will be ***generated*** in a manner similar to dTIMS' ***committed treatment***. The difference will be that this new strategy will be added to the existing list of strategies.

Task Five: Conduct a conference call with SDDOT to review progress of the overall project.

A conference call between Deighton and SDDOT was arranged to discuss other ideas about how this function would. From the discussions the specifications for the APD summary report were defined. This summary report is available for viewing from dTIMS.

Task Six: Use the Strip Map function as the basis of a function that allows the user to manually create and edit construction projects, and their strategies.

This task added the information windows and buttons to the basic strip map function created in task three. The basic idea for this function was to give dTIMS a function that would allow a user to create construction projects manually. The user can create them from scratch, or, modify the projects recommended by the function developed in Task Four.

Task Seven: Develop a print and output function so that optimum and viable construction projects' budget information can be output to a printer or a file.

This function includes a **recalculate** button and an **undo** button, which affect the budgeting factors. It also has a **print** and **export** button which, among other things, prints the original optimal strategy for each section along side of the construction project segment.

The programming of dTIMS, the second most time consuming task, included Tasks 4, 6 and 7. In total, 8656 lines of code were added to dTIMS. This does not include the modifications to the integrated Help System. The results of these tasks can be seen in the software that was delivered to SDDOT. The new code was delivered as a patch via the Internet. Through the application of the patch dTIMS 6.0 was upgraded to version 6.1.

Task Eight: Make necessary changes to the on-line Help system and the hard copy User Manual.

The new functions and definitions developed in this project were incorporated into dTIMS' existing on-line help system. This task ensured that not only was the text added, but, the context mapping is performed and linked into dTIMS.

Deighton Quality Procedures describe the processes involved in transferring the help system text into the hard copy manual for dTIMS. Deighton followed these steps to ensure that an up-to-date hard copy manual is completed by the end of this project and a copy of this manual shall be provided to SDDOT for each license they possess.

The relevant sections added in the Reference Manual are as follows: Chapter 3, pages 87 – 95, Chapter 4, pages 182 – 189 and Chapter 4, pages 201 – 205. These sections of the Reference Manual give specific information on the functionality and capabilities of the APD feature of dTIMS 6.1 and have been reproduced in Appendix '2' of this report.

Task Nine: Develop a test plan and test the software according to the plan to verify that the software performs its intended function.

Prior to the delivery of the dTIMS 6.1, Deighton committed a significant effort towards the in-house testing to verify that the software was working as intended. This involved the development of a test plan, as outlined in Deighton's Quality Procedures manual, necessary to carry out a series of tests to verify that the software is functioning as planned.

Task Ten: Classify faults and correct all "bugs.".

Deighton's Quality Procedures manual outlines the steps required by the vice president of R&D, the project manager and the programmer/analysts to classify all faults discovered during the testing process and correct the bugs. Deighton has procedures which track each and every fault discovered during all phases of testing.

Basically, a fault is a perception that the software did not perform according to the requirements. Faults can be (a) bugs (the software did not provide a function that was part of the requirements), (b) enhancements (mistaken belief that the software should have provided a function and the function is NOT in the requirements, but, is desirable), or (c) misconceptions (mistaken belief that the software should have provided a function and the function is NOT in the requirements and is NOT desirable). This task requires that Deighton correct all identified bugs before the software is released.

The test plan developed in Task Nine, yielded a number of faults. In total, 19 bugs were discovered, documented and corrected prior to the delivery of the product.

Task Eleven: Write a report presenting the results of this study and provide to SDDOT for review and comments.

Deighton concluded this project with the delivery of this final report that summarizes the results of this study. Following a review and comment by SDDOT, Deighton issued a final version of the report.

5. Conclusions and Recommendations

At the completion of the project the consultant made the following conclusions and recommendations.

5.1.1 Decision Tree Development

Following the review of the software documentation it is recommended that SDDOT commence with the development of decision tree theory that will be used to model the current programme development process in South Dakota. Any assumptions or background used in the development of the theory should be recorded in the documentation that accompanies the decision tree summary.

5.1.2 Further Testing

The SDDOT decision tree structure should be applied to the SDDOT network for testing the reliability of the APD. Results of the test program should be used to, not only, debug the decision tree setup but, also to give additional feedback to Deighton regarding the functionality of the software.

5.1.3 Automatic Programme Development Implementation

SDDOT must develop an implementation plan that will enable the APD to be used in the development of practical construction projects from the dTIMS optimized construction programme. It may be wise, as part of the implementation plan, to check results from APD with the programme that was produced using current procedures in a previous year.

5.1.4 Review and Revise

Following the generation of the first automatic programme, SDDOT should review the programme developed with the intent of evaluating the suitability of using decision trees as a foundation in the construction programme process.

Appendix '1'

Questionnaire Responses and Summary

Combining Recommended Strategies into Construction Projects using dTIMS
Questionnaire Responses and Summary

Submitted By
Deighton Associates Limited
112 King Street East
Bowmanville, Ontario, Canada
L1C 1N5
(905) 697-2644
(905) 697-2645

Submitted to
South Dakota Department of Transportation
Office of Research Room B-116
700 East Broadway Avenue
Pierre, SD 57501-2586

February 24, 1998

Table of Contents

List of Figures

1. Introduction

This report has been prepared as part of task two of the research and development project that is intended to allow users of dTIMS to combine recommended strategies into viable construction projects. Task two specifically involves the investigation and recommendation of a set of parameters to be used by dTIMS to control the automatic generation of viable construction projects.

To accomplish the objectives of Task Two, Deighton proposed the following:

- develop a questionnaire which will allow various experienced dTIMS user's to describe the criteria used to decide on construction projects;
- get comments from SDDOT on the questionnaire by fax, modify it, then distribute it to SDDOT as well as other experienced dTIMS's users;
- collect the questionnaire and analyze the results;
- summarize the results in the form of a draft report which suggests a list of criteria to be used in the automated procedure;
- distribute the above report to the original participants;
- have a conference call between Deighton and SDDOT to discuss changes to the report;
- incorporate the comments into a revised report and redistribute;
- use the report as the basis for the development of the viable projects enhancement.

This report presents the results of the above proposal and offers recommendations for the design modifications to dTIMS.

Throughout this report the term "section" is used to describe a piece of road originally defined to dTIMS on one record of the DT2699 file. The term "project" is used to describe a piece of road that is created by combining one or more consecutive sections. The term "treatment" is used rather loosely to refer to the initial treatment in a strategy. In other words, we use the term treatment to refer to a dTIMS strategy.

2. The Questionnaire

To solicit input from a number of existing dTIMS users regarding this project, a questionnaire was prepared and distributed. The questionnaire explained the project objectives and gave the recipients directions for completing the questionnaire.

The South Dakota Department of Transportation (SDDOT) participated in the review of the draft questionnaire and provided comments prior to its distribution. These comments were incorporated into the questionnaire. A copy of the final questionnaire can be found in Appendix A of this report.

The list of dTIMS agencies that were contacted to participate in this questionnaire included the following:

- The State of South Dakota DOT, USA
- The State of Indiana DOT, USA
- The State of Utah DOT, USA
- CERTS International Pty. Ltd., Australia
- Riley Partnership, UK
- PLB Geotechnical Division, Malaysia
- ViaGroup Ltd., Switzerland
- St. Michel Consulting Inc., Canada
- PAWC, South Africa

A total of nine surveys were distributed. Five questionnaires were returned, of which one was from SDDOT and one was returned without answering the questions. In the later case, the agency said they did not combine smaller sections into larger projects, therefore, had no comment on the subject.

3. The Responses

Each of the questions contained in the questionnaire is reproduced below. Under each question the responses are listed anonymously. All responses appear as they were received with the exception of changes made to obvious typographical errors. The interpretation of the responses will be done in the following section of the report.

3.1 Question 1a

When developing a viable construction project, do you combine different treatment types into a single construction project?

1. Yes

2. Yes, in situations of a long rural segment adjacent to a short urban segment, the treatment may vary between the urban and rural segments, especially for reconstruction treatments. For rural resurfacing treatments, it is very rare to have projects cross over different pavement categories (asphalt vs. concrete), though an overlay project may have a few short segments of mill & overlay within it is not uncommon. Urban resurfacing treatments do not get combined into one project.

3. Yes, we consider viable construction limits by related planned and associated activities to include similar pavement construction history, geometrics, safety, structures, functional classification, and political impact. In rural Utah project mobilization costs and efforts requires lengthy projects to justify the hot plant set up resource requirements.

4. I try to avoid combining different treatments into one, as this would hide the true nature of the work. Here I would define "project" as the rehab/reconstruction work on a piece of road. In theory, many treatments can be combined, but if we stay with project level planning, I suggest keeping the different treatments apart. However, some treatments may be and are combined, such as "re-shape and seal" or "mill and overlay".

3.2 Question 1b

If yes, are all treatment types eligible for combination into a viable project and explain your criteria for determining which treatments are complementary?

1. I would separate hot mix overlays and surface treatments (ST) and maybe replace some short ST sections with hotmix to give a uniform surfacing material. I have no problem mixing mill & fill with overlay - the resulting surface looks the same to the road user.

2. Asphalt treatments are not combined with concrete treatments for resurfacing projects. For pavement management purposes, asphalt and concrete treatments are not combined into one project, though the timing of the projects might be adjusted to coincide with each other. Most asphalt resurfacing treatments are eligible for combination into one project, but more often we change all the treatments to correspond to the section of predominate length.

3. Yes, adjacency is more important than treatment type. Spot intermediate preparatory treatments are often performed to bring small sections up to the same level as the majority of the project length. Major treatment differences are not allowed on the same project.

4. No comment.

3.3 Question 1c

If not, do you consider treatment dominance for adjoining sections that do not have the same treatment types and explain your particular criteria for determining which treatments are dominant?

1. N/A

2. N/A

3. Yes dominant treatment by extent. Treat spots as individual treatment by percentage of project cost and requirements of specialized contractor equipment and skills. Compatibility and constructability of treatment types for inclusion or exclusion.

4. When combining sections the dominant issue is whether the base/sub-base of the existing pavement has to be treated or not. E.g., stabilization, milling, drainage or reconstruction are dominant. Overlay, spray seal are not.

3.4 Question 1d

Is there a maximum project length where you would not continue to combine sections and what parameters would you suggest be used to determine the maximum length?

1. I would not normally set a maximum length as my links ("roads" in dTIMS) are usually shorter than the desirable contract size. I am usually adding links together to make a viable contract.

2. Yes, there are physical limitations to the length of a project that can be done by any one contractor, varying by treatment type and geographic location in the state. South Dakota recommends that the maximum length parameter be user defined.

3. Project cost tends to determine length. Minimum length by viable fixed project costs such as mobilization. Geographic statewide distribution requires staging projects throughout the state.

4. The maximum project length would depend on the circumstances, such as:

   Availability of treatment (easy - moderate - difficult); e.g., asphalt may need a mobile plant; in this case the minimum job size would be larger, or spray sealing is usually easily available anywhere. One way to address this issue is to introduce "bulk discount", i.e. allow a full price and a discount depending on the quantity.

   E.g., cost of treatment: $<1000\ m^2 = \$10/m^2$, $1000\text{-}10000m^2 = \$9.5/m^2$, $>10000m^2 = \$9.0/m^2$

   The underlying issue is the conflict between the condition driven treatments and practical realities. Another option would be to create two different section lists; one would reflect the uniform conditions, and the other the practical project sections. After analyzing the uniform sections, the results could be "transformed" into the project level sections, and these could be optimized again, using the treatments for the uniform sections as committed treatments.

3.5 Question 2

If dTIMS does not generate a desired treatment in a desired year because it does not satisfy some other criteria, should dTIMS generate the treatment for consideration in a viable project?

1. Not sure what this question means but I do need some treatment for bridging gaps where dTIMS has not recommended a strategy (maybe the road is fine) but I want continuity.

2. As long as the generated treatment would be eligible for the section's pavement type. It would not be feasible for the concrete pavement grinding project to be generated on an asphalt pavement.

3. No, desired treatment and time is a judgement decision that needs to be value added after the dTIMS model evaluation. dTIMS should allow for project commitment based on individual preferences and then re-run the model and analysis. The dTIMS process is not the entire complete actions within the decision making process but it certainly provides the major source of data and analysis to the process. It should accurately reflect the way the agency makes its decisions, reflects how the agency believes its pavements perform, and the impact of various treatment applications.

4. This would be nice, but it would require a fairly complex logic, including some political considerations.

3.6 Question 3

Is there a time window that is considered practical for the combining of sections that have similar treatment types in different years? That is, do you consider moving a treatment one, two, three, etc. years from its optimal date of application?

1. Normally I consider 3 years as a reasonable period for programming and switching treatment years.

2. Both a time frame window and a condition range would need to be considered. The timing could vary dependent upon treatment type and length of the shorter segment. South Dakota recommends that ranges be user defined.

3. Yes, but the window is dependent on the treatment being considered. One year for seal coats; up to five years for reconstruction and three years for major rehabilitation projects.

4. This depends again on the nature of the work. An overlay or re-seal can be separated by one year easily. Reconstruction could be combined if they are max 2 years apart. Again, a tick box may give some options on the treatment menu, e.g., "what is the minimum frequency this treatment can be tolerated?" This may also be different in urban and rural areas, or depending on the AADT.

3.7 Question 4

When considering a recommended section for combination into a viable project, do you examine only adjacent sections for their potential to be included in a viable project?

1. Normally look for adjacent sections for continuity.

2. No, some sections may be skipped over due to being a different pavement type, rural vs. urban, etc. Also, geographically adjacent sections on different routes (intersection arms) may be combined into a viable project, though South Dakota recognizes the difficulty in automating this scenario, and does not expect it to be included.

3. Yes. Adjacency and constructability are key contract project parameters. Simple, singular project requirements provide better contractor performance and improve overall pavement performance.

4. Combination of treatment over 2-3 km length would probably defy the purpose of the PMS optimization - but this would depend on the local conditions. Some aspects of combining projects:

   - economics: distance from plant / supplier; bulk discount, lane occupancy costs
   - technical: would the previously completed work be damaged (drainage, heavy vehicle traffic, detour); presence of intersections (in many cases, intersections are considered as a separate section)
   - politics: what is tolerable for the public (this may mean a compulsory internal ranking of the network)

3.8 Question 4a

If not, what location criteria do you use?

1. N/A

2. Multi-sectioned non-adjacent projects are usually limited by Regional Office jurisdiction and by geographic proximity. The range of a multi-sectioned project varies by treatment type.

3. No response required.

4. No comment.

3.9 Question 4b

Is there a maximum project length where you would not combine adjacent sections?

1. No maximum length.

2. Yes, there is both a physical and a political limit to the length any one project can be. This would vary by treatment type. South Dakota recommends that this maximum length be user definable.

3. Yes, project cost limited by treatment type. Seal coats have longer project lengths while reconstruction projects are of limited project length.

4. No comment.

3.10 Question 5

Do you consider economics when developing viable projects? That is, are budgets required to be maintained when treated sections are shifted from year to year?

1. Yes - use weighted NPV or NPV/cost to decide on programme year for aggregated sections.

2. If costs are going to be added up, and the timing of a project is not to be re-optimized, the maximum allowable budget for each year should not be allowed to exceed a user definable percentage of the maximum allowable. If the timing of a project is to be re-optimized after project sections are combined, the maximum allowable budget should be maintained.

3. Yes, available funds are geographically distributed. They are also distributed by treatment type and effort required. There are ranges of resource allocations by geographic region.

4. Yes.

3.11 Question 6

Are committed strategies eligible to be moved to create a viable construction project?

1. Not usually as these are already contracted.

2. No, committed strategies are fixed. It is not pavement manageent's responsibility to alter a project after a project is programmed into the 5 yr. STIP.

3. Yes, based on the project scoping results and review. The present pavement condition and predicted performance prior to treatment application is considered on a project by project basis by the project design team. This allows for project details and site specific conditions to be considered that could not be modeled nor noted within the system level modeling process. Identified projects can be either lengthened or shortened based on the information and financial constraints imposed on the project. Project available funding also impacts the length of projects and this can cause the splitting of similar treatments.

4. Yes.

3.12 Question 7

When combining sections to create a viable construction project are costs simply added or do you consider cost savings? What method of cost savings calculation would you suggest?

1. Cost savings due to scale of project normally allowed for at section level as I know that the sections are going to be combined later to form larger contracts.

2. Because all of our costs are a per mile average, costs are simply added together. South Dakota does realize that there are certain per mile costs which would be reduced by having a longer project (mobilization & traffic control).

3. Cost Savings, minimum effective length for fixed project costs, mobilization in remote rural areas, costs of acquiring and transporting equipment and materials. Anticipated contractor project savings is the method for combining or splitting projects.

4. Cost saving comes from bulk discount, reduced management costs and lower ancillary cost, such as lane occupancy costs, improved safety. I would suggest a bulk discount cost as described before.

3.13 Question 8

Is it all right for dTIMS to place the results of the viable project generation into one of the five existing scenarios? If not, how would you suggest that this happen?

1. Sorry, don't understand the question.

2. It would be OK for dTIMS to do this. Currently South Dakota does not use all five available scenarios.

3. Yes, allow for integration in the same way as other model selected treatments.

4. I think that viable projects should be either flagged only, or this should be a separate or third run after generating strategies and optimization. This way the committed projects can be considered. The flag option would allow a presentation on dMAP.

3.14 Question 9

Do you believe dTIMS should allow the user to set different criteria for minimum and maximum length of projects for the following? Please check the ones that apply and add others that you feel should be listed.
- Treatment Type
- Analysis Set
- Budget Scenario

1. User should specify minimum length of project and maximum gap to be filled i.e. put in a treatment for continuity even though no economic treatment is given during the programme period. These parameters could be for different treatment groups e.g. seal, overlay, reconstruction, widening.

2. Treatment Type, Urban vs. Rural.

3. Treatment Type - yes; Analysis Set - yes; Budget Scenario – yes. No additional others.

4. Suggested "window" or "box" items:
   - minimum practical length of the treatment
   - acceptable min time span between treatments on adjacent road sections
   - bulk discount
   - mutually excluding treatments (surface treatments as opposed to "deep" or intrusive treatments)

4.2 Question 1b

If yes, are all treatment types eligible for combination into a viable project and explain your criteria for determining which treatments are complementary?

This question tries to get at whether or not there are any restrictions in combining sections with different treatments together into one project. The responses indicate that there are restrictions. In fact, the response: "Major treatment differences are not allowed in the same project." captures the general feeling of the respondents.

The question becomes: "How do we identify major treatments that cannot be mixed?" Or, put another way: "How do we identify treatments that can be mixed in a project (i.e., are complimentary)?"

4.3 Question 1c

If not, do you consider treatment dominance for adjoining sections that do not have the same treatment types and explain your particular criteria for determining which treatments are dominant?

This question tries to clarify whether there is a treatment that would dominate the project. In other words, when combining adjacent sections with different treatments will dTIMS substitute a treatment for one of the sections with a dominant treatment. Although this and the previous question were presented as if they were mutually exclusive, the responses seem to indicate that indeed there is some kind of dominance being considered.

It seems as though there are cases where a certain treatment would be "switched" in certain circumstances. For example a response from the previous question indicated that the programmer would "replace some short ST [surface treatment] sections with hotmix to give a uniform surfacing material." Once again, the question becomes: "How do we identify a 'short' section with a less dominant treatment?"

4.4 Question 1d

Is there a maximum project length where you would not continue to combine sections and what parameters would you suggest be used to determine the maximum length?

This question ponders whether or not we keep combining adjacent sections together if all other constraints are met. If we answer yes to this question, in theory at least, we could end up with a single project for each road. (dTIMS identifies a road as all sections with the same road name in the DT2699 file.)

The overwhelming response is that there is a maximum length. However, the maximum length is not just a single number for the entire network. It seems to be related to things like: (a) treatment type, (b) urban vs. rural, and (c) some geographic considerations.

One obvious question that arises is: "If we are allowed to mix treatment types in a project, then how can we specify maximum length by treatment type?" Or, from a logic point of view, "If treatments are allowed to be mixed in a single project, and if maximum length is given by treatment type, then, what happens if the maximum length is different for compatible treatments?"

4.5 Question 2

If dTIMS does not generate a desired treatment in a desired year because it does not satisfy some other criteria, should dTIMS generate the treatment for consideration in a viable project?

This question adds an additional level of logic to the previous questions. This question's main idea is to ascertain whether or not dTIMS should automatically generate a treatment in the case where adjacent sections would otherwise be joined, but the two treatments are not compatible. The question presupposes that the only treatments dTIMS can use are those in the strategy list, and that the strategy list is already exhaustive. Therefore, if a treatment is desired for project development should dTIMS generate a strategy that involves this as the initial treatment?

The responses indicate a definite desire for this feature, although they admit that capturing the logic to automate this would be very difficult. Perhaps we should answer the following question first: "Should dTIMS automatically generate a strategy and add it to the list, or, should it have a mechanism for the user to manually 'commit' an additional strategy and have dTIMS add this to the list?" In the manual case, the user would consider all of the additional criteria, including political considerations, involved in the decision to override the triggers. In either case, it would seem consistent for dTIMS to generate the strategy using its normal analysis process so the costs and benefit calculations would be consistent for all strategies.

4.6 Question 3

Is there a time window that is considered practical for the combining of sections that have similar treatment types in different years? That is, do you consider moving a treatment one, two, three, etc. years from its optimal date of application?

This question is similar to the previous one in that it gets to the issue of dTIMS automatically generating additional strategies. In this case, however, the question tries to simplify the logic by having dTIMS create the additional strategy simply by moving the initial treatment of an existing strategy by a certain number of years.

As in most of the other responses, however, it is desirable but it's not that simple. There seems to be a need to consider other things than just the year of application. The decision depends on: (a) the nature of the treatment and (b) the condition. This raises the following question: "Should dTIMS have another set of triggers for each treatment that are used in deciding how and when to generate an additional treatment?"

4.7 Question 4

When considering a recommended section for combination into a viable project, do you examine only adjacent sections for their potential to be included in a viable project?

This question tries to gather information about skipping sections in an otherwise viable project. From a logic point of view, it is extremely desirable to make decisions about including a section in a project as one travels along the road in the direction of travel. That is, whenever a 'different' section is encountered along the direction of travel, a new project is initiated. Skipping sections implies an ability to make the decision while travelling ahead and back. In the extreme, this could create an endless loop of analysis.

The responses seem to acknowledge the difficulty of skipping over certain sections. It does appear, however, that if logic can be worked out it would be a nice feature to have. Cases cited in the responses involve a small section of different pavement type or different cross section included in the middle of an otherwise homogeneous project.

Once again we must answer the question of: "How do we consistently and unambiguously identify 'small' and 'different' sections within a project so we can skip over them?"

4.8 Question 4a

If not, what location criteria do you use?

This question tried to capture an idea of the 'skipping' logic. Since the respondents didn't specifically require skipping, few responses were given. In this case therefore, we could simply ask the question: "Should dTIMS allow the user to manually create viable projects which 'skip' over certain sections?"

It seems as though this question is similar to an earlier question about mixing treatments. In this case, however, the 'skipped' section would have the 'do-nothing' strategy selected.

4.9 Question 4b

Is there a maximum project length where you would not combine adjacent sections?

This question was not as clear as it should have been. Since we already asked the question about maximum lengths, the intent of this question was to determine whether there indeed was a maximum length of a skipped section. Poor wording or not, since there does not appear to be any reducible logic for automatically skipping sections, the question is irrelevant anyway.

4.10 Question 5

Do you consider economics when developing viable projects? That is, are budgets required to be maintained when treated sections are shifted from year to year?

This is an age-old question. It asks: "Do we shift around treatments and years without regard to over-running the budget?" The responses all indicated that there has to be budget considerations.

This opens up a very difficult area. The question was asked in the hope that everyone would agree that we could do it without regard to budgets; which is by far the easiest way. If we have to include budgets we probably need some sort of re-optimization. But, to re-optimize we need a list of choices for each project just as the initial optimization needed a list of strategies for each section from which to choose.

4.11 Question 6

Are committed strategies eligible to be moved to create a viable construction project?

This question was asked because we needed verification that committed strategies are exactly that, committed. It was hoped that the response: "No, committed strategies are fixed." would be universal.

Since the responses were split, it would seem that either: (a) some users regard committed strategies as something different than their original intent in dTIMS, or (b) the respondents understood the term 'committed strategy' to be something other than the literal definition in dTIMS. dTIMS defines 'committed strategy' as: "the only strategy in the list for a committed section."

We prefer the later explanation for the yes responses. However, it may be possible to include a flag specifying whether committed strategies can be moved from year to year.

4.12 Question 7

When combining sections to create a viable construction project are costs simply added or do you consider cost savings? What method of cost savings calculation would you suggest?

This question wanted to get at the heart of considering 'bulk discounting' when combining sections. The responses all agree that some form of bulk discounts are realized. However, there is a split in how bulk discounts should be accommodated. In some cases, the user accounted for them before hand by presupposing the bulk discount when supplying the initial treatment costs. In other cases, the users wanted dTIMS to calculate a bulk discount after the project length was determined.

4.13 Question 8

Is it all right for dTIMS to place the results of the viable project generation into one of the five existing scenarios? If not, how would you suggest that this happen?

This question wanted verification of an initial thought as to how dTIMS could accommodate this project creation function. We thought that dTIMS could analyze the selected strategies in one budget scenario and put the changed (or not) strategy in a second scenario, we would not have to do a major rebuild of dTIMS. It seems as though the responses indicate that this approach would not have an adverse effect.

4.14 Question 9

Do you believe dTIMS should allow the user to set different criteria for minimum and maximum length of projects for the following? Please check the ones that apply and add others that you feel should be listed.

This question wanted to get an idea of what criteria are important when specifying minimum and maximum lengths. The responses all agree on treatment type (or some grouping of treatments) as desirable criteria. They also indicate, however, that there are other criteria also as important. Since not all of the criteria are 'base data items' in dTIMS, we will need a mechanism to accommodate listing criteria using other discernable data values.

4.15 Question 10

If you have any additional comments or descriptions of the manual procedures used currently to develop viable construction projects, please specify the criteria below.

The responses to this question were useful. One particular response was thought provoking: "The treatment for the consolidated section should correspond to the individual section having the most length. The year of the treatment for the consolidated section should be no sooner than the earliest individual section, and within 3 years of the section with the predominate length."

This statement seemed to wrap the entire thought of how this could work into one coherent idea. The question remaining is: "How can we get a set of parameters together that could be translated into this statement?" The answer to this question is the topic of the next chapter.

5. Conclusions

The results of the survey were not all that surprising. We knew the task at hand was going to be complicated. The questionnaire responses confirmed just that. Although we never asked directly for any secretes that might make the job easier, none came. Therefore, we looked at the problem from several different angles and have developed an approach. Which, in the absence of anything better, is a little complicated, but, it will work. Therefore, we recommend adopting this approach we call "Automatic Programme Development". (NOTE: dTIMS uses the term 'programme' to refer to a construction programme as opposed to a computer program.)

5.1 Discussion of Requirements

The survey results indicated that dTIMS must have the following capabilities in its Programme Development module:

Section data

dTIMS will assume that the user will provide both the road name and the from and to offset information in the DT2699 file for this function to operate. As well, the user must provide an "Other" data field in the DT2699 file where dTIMS will put a unique project identifier.

Treatment compatibility

dTIMS must have the capability of allowing different treatments to exist in the same project; as long as the treatments are 'compatible' as defined by the user.

Treatment substitution

dTIMS must have the capability to substitute a section's treatment with another treatment from the section's list of strategies to make the section compatible in the project. The user will define the criteria for treatment substitution.

Treatment adjustment

dTIMS must have the capability to move a section's treatment ahead or back in time. The criteria for moving a treatment will be defined by the user

DRP descriptions:

(not shown) This line shows the description of each data reference point (DRP) on the road. You define these descriptions in the FDESCFROM and FDESCTO fields in the inventory file. You can suppress this part by unchecking the Show DRP descriptions control (see below.)

Strip map:

This part is a thick black line with cross hatches at each DRP along the road. If the road does not have elements for its entire span, dTIMS displays these as a dotted line in this part.

DRP Addresses:

This part shows the offset for each DRP. You define these offsets in the FOFFFROM and FOFFTO fields in the inventory file.

DataField values:

This part shows the data of interest on the strip map. There are three line diagrams in this part. Each line shows a specific set of data as follows:

First Line:

Shows the elements from the inventory file and displays data describing the first treatment of the selected strategy as assigned by optimization for the budget scenario you selected when you entered this function. You cannot change these selected strategies in this function: they are displayed for your information only so you can compare what the optimization function selected for these elements. The label on top of these elements is the element ID from the FSECTION field of the inventory file. The labels on the left hand side of this line are:

XXXXXXXXXX->:

The name of the budget scenario you selected when you first entered this function. This is shown in the above figure as "High Funding->".

First Treatment Type:

The name of the first treatment of the selected strategy in this budget scenario.

First Treatment Year:

The year of the first treatment of the selected strategy in this budget scenario.

First Treatment Cost (yy):

The cost of the first treatment of the selected strategy in this budget scenario. The cost units are shown as (yy).

Present Value Benefits:

The present value benefits of the selected strategy in this budget scenario.

Second Line:

Shows the elements from the inventory file and displays the first treatment of the selected strategy for the fifth budget scenario in the current analysis set. The selected strategies in the fifth budget scenario are the ones used to form the programme project. You can select one of these elements by clicking on it and use the control buttons on the strip map to change these selected strategy so you can develop your programme projects. The label on top of these elements is the element ID from the FSECTION field of the inventory file. The labels on the left hand side of this line are:

XXXXXXXXXX->:

The name of the fifth budget scenario in the analysis set you selected when you first entered this function. This is shown in the above figure as "No Funding->".

First Treatment Type:

The name of the first treatment of the selected strategy in this budget scenario.

First Treatment Year:

The year of the first treatment of the selected strategy in this budget scenario.

First Treatment Cost (yy):

The cost of the first treatment of the selected strategy in this budget scenario. The cost units are shown as (yy).

Present Value Benefits:

The present value benefits of the selected strategy in this budget scenario.

Third Line:

Shows the programme projects and displays some information summarizing the treatments from the elements that make up the project. The label on top of these elements is the project ID from the

PROGPROJID field of the inventory file. The labels on the left hand side of this line are:

PROGRAM PROJECT->:

A label.

Year:

The year of the programme project. dTIMS defaults this to the year of first treatment of the first element in the project. Since the idea of programme projects is to group together elements for programming purposes, dTIMS assumes that the first treatment for all elements in the project will be the same, therefore, it defaults the year to the first element. It is up to you to ensure this is so when you manually adjust the strategies for the elements in the project.

Total Cost (yy):

The sum of the costs of all the first treatments from all the elements in the project. The cost units are shown as (yy).

Total Benefit:

The sum of the Present Value Benefits of all selected strategies from all elements in the project.

Total Length (zz):

The sum of the lengths of all elements in the project. This does not include the length of missing pieces (dotted lines) along the road. The length units are shown as (zz).

The top part of the strip map has a number of different controls which allow you to manipulate the strip map and the program projects. The top part looks like this:

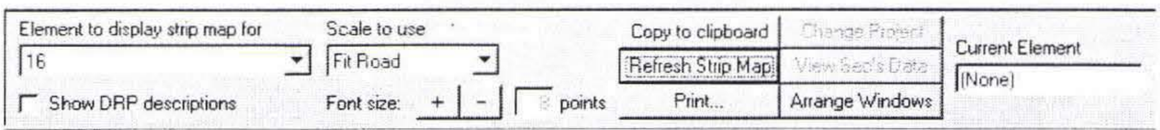


Figure 2: The Strip Map Controls

These controls are as follows:

Element to display strip map for:

Select the road element you want the strip map displayed for. dTIMS compiles this list of road names by scanning the inventory file and selecting all unique names from the FNAME field. When you select an element in this control, dTIMS redraws the strip map.

Scale to use:

You have a choice of three scales to display the strip map for. When you change the scale, dTIMS redraws the strip map.

Fit Road:

Select this to cause dTIMS to force the entire length of the road element into the window.

By Length:

Select this to cause dTIMS to force a certain length of the road element into the window. When you select this, dTIMS asks you for the length you wish to **Show**.

By DRP:

Select this to cause dTIMS to force a certain number of DRPs on the road element into the window. When you select this, dTIMS asks you for the number of DRPs you wish to **Show**.

Show:

This is where you specify the length, or the number of DRPs you want dTIMS to fit into the current window size.

Show DRP descriptions:

Check this box to have dTIMS show descriptions for each DRP. dTIMS writes these descriptions vertically above the DRP's location. To do this dTIMS must rotate the font. When you uncheck this control, dTIMS suppresses the DRP descriptions. This will save an incredible amount of screen space to help you display strip maps in lower resolution.

Font size:

Use the + and - buttons to increase and decrease the size of the font dTIMS uses to place information on the strip map.

Copy to clipboard button:

Press this button to send a copy of the strip map to the clipboard. dTIMS will redraw the strip map to the clipboard. You can then paste this drawing as a bitmap into any document.

Refresh Strip Map button:

Press this button to refresh the data on the strip map. dTIMS will redraw the strip map with the latest information from all the source files. You use this button after you select another strategy for one of the elements in on the strip map.

Print button:

Press this button to produce a print out of the current strip map. dTIMS redraws the strip map to the printer object. Therefore, you must make sure that your printer can accept graphic characters and is capable of rotating fonts.

View Sec's Data button:

Press this button to see the Inventory Data Window and the Strategy List Window for the current element. This button is enabled when you select a Current element (see Current Element below). When you press this button dTIMS opens the Inventory Data Window (if it is not already opened) and opens the Strategy List Window (if it is not already opened). dTIMS finds the record in the Inventory Data Window for the current element and sets Strategy List Window to this element for the fifth budget scenario. You can change the selected strategy for the current element on the Strategy List Window.

Arrange Windows button:

Press this button to conveniently arrange the windows for better viewing of the data. When you press this button dTIMS maximizes itself and displays the strip map across the top half of the available space. dTIMS then puts the Inventory Data Window on the left half of the available space below and the Strategy List Window on the right half. This arrangement works best on computers with high resolution.

Current Element:

Shows the element ID for the currently selected element. You select an element by placing the mouse cursor on top of the element's line (shown in the fifth budget scenario's part) and clicking. dTIMS highlights the selected element's line in red.

Change Project button:

Press this button to change the programme project for the current element. This button is enabled when you select a Current element. When you press this button dTIMS displays the Change Project Window. This window looks like this:

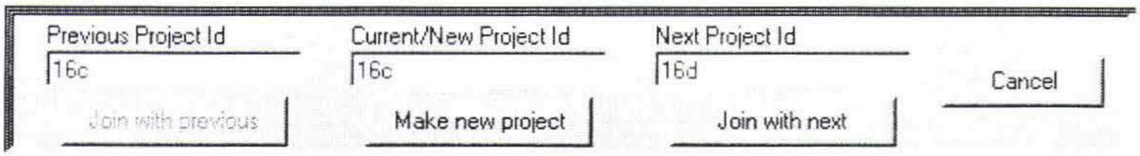


Figure 3: The Change Project Window of the Strip Map

The Change Project Window has the following components:

Previous Project Id:

Shows the project id for the programme project that the element to the immediate left of the current element belongs. The project Id is the value found in the PROGPROJID field of the Inventory File.

Current/New Project Id:

Shows the project id for the programme project that the current element belongs. The project Id is the value found in the PROGPROJID field of the Inventory File. If you want to make the current element belong to a new project you put the new project's Id in this field.

Next Project Id:

Shows the project id for the programme project that the element to the immediate right of the current element belongs. The project Id is the value found in the PROGPROJID field of the Inventory File.

Join with previous button:

dTIMS enables this button only when the Previous Project Id is different than the Current/New Project Id. When you press this button dTIMS removes the current element from the current project and adds it to the previous project. dTIMS does this by changing the value in the PROGPROJID field of the Inventory File for the current element to the value shown in the Previous Project Id control.

Make new project button:

dTIMS enables this button only when the Previous Project Id is different than the Next Project Id. When you press this button dTIMS removes the current element from the current project and adds it to a new project. Before you press this button you must enter a project Id for the new project in the Current/Next Project Id Field.

Join with next button:

dTIMS enables this button only when the Next Project Id is different than the Current/New Project Id. When you press this button dTIMS removes the current element from the current project and adds it to the next project. dTIMS does this by changing the value in the PROGPROJID field of the Inventory File for the current element to the value shown in the Next Project Id control.

Cancel Button:
Press this button to remove the Change Project Window.

5.2.2 Automatic Project Development

dTIMS will also have a function which automatically generates a 'first-cut' set of projects. This function will automatically go through an existing budget scenario and, to the extent possible, will do an analysis to determine the projects.

The basic idea of this function will be to work on one treatment type at a time. dTIMS will process treatments in an order set by the user in a Treatment Precedence Table.

Figure 4 shows a flowchart of how this process will work. Basically, dTIMS will find the first section with the highest precedence treatment on it. Then, dTIMS will start a new project, which includes only that section. Next, dTIMS will check to see if there is a 'next' section. (The term 'next' refers to increasing offsets and 'previous' refers to decreasing offset.) dTIMS will keep processing next sections until there are no more. Then, dTIMS will check the 'previous' section and will keep processing until there are no more previous.

With the general logic of the project development function in place, the only thing remaining is to determine how dTIMS will answer each of the questions in the flowchart. Questions are denoted as diamond boxes in the flowchart. The questions that the user must help dTIMS answer are highlighted in the flowchart.

To help dTIMS answer the questions, we propose the user will define a decision tree. In all, dTIMS will have a total of six decision trees; one for each highlighted question diamond. The questions are:

Will the maximum project length be exceeded?

Is the new section compatible with the project?

Can this section be included in the project by being 'skipped' over?

Is the treatment on the section compatible with the project?

Can the treatment on the section be switched?

Can the year of application of the treatment be switched?

These decision trees are described in the next section of this report.

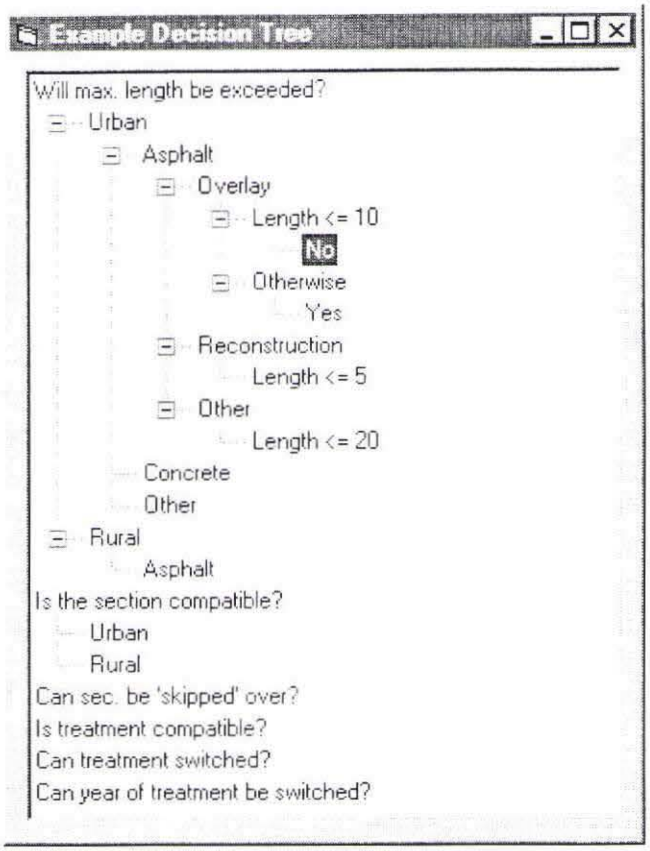


Figure 5: Example of a decision tree

5.2.4 Branch Condition Table

We propose that the user set up a Branch Condition Table for each data field needed to be use in the analysis. The Branch Condition Table consists of a unique name and either one column for character fields or two columns for numeric fields. The column for a character field contains a list of values the user wants dTIMS to treat as similar. The two columns for a numeric field will define the minimum and maximum values for a range.

For example, the Branch Condition Table for the 'Asphalt' example would be dedicated to the pavement type field called "P_TYPE" and would have four rows: (1) THK, (2) FD, (3) TonW and (4) TonS.

5.2.5 Treatment Precedence Table

The idea behind the Treatment Precedence Table is to have the user define the order that treatments are to be processed. This is similar to defining dominant treatments for a project. dTIMS will make the first project with the highest treatment in this table. As dTIMS moves forward and backwards along the road, it will look for sections with either the same treatment or a compatible treatment to add to the project.

Appendix A: The Questionnaire

December 3, 1997

Attention:

Re: Pavement Management Segment Consolidation in dTIMS Research Project

Dear :

Deighton Associates' research and development team is embarking on the development of an enhancement to dTIMS that will enable users to combine optimized strategies, based on certain criteria, to develop a "viable" construction programme.

The problem, as described by Deighton users, which leads to this research and development project is as follows:

> *dTIMS is used to analyze life-cycle costs for various rehabilitation strategies on each segment of road within a network. To use dTIMS, an agency must divide its network into homogeneous segments. Homogeneous means that the source data describing the attributes of a segment are the same throughout the segment's entire length. Unfortunately, dTIMS makes the optimal strategy selection for each segment independent of the segment's location, length or adjacent segment status.*
>
> *For instance, dTIMS might recommend a particular repair strategy for one segment, a different strategy for the next and a third strategy for the next. But each of these segments may not be of significant length to be a stand alone project or it might be more cost effective to do one project at one time instead of spreading projects over more years. To combine these segments requires staff to manually combine segments and propose complementary strategies so that viable construction projects can be proposed.*

Questions

1a. When developing a viable construction project, do you combine different treatment types into a single construction project?

1b. If yes, are all treatment types eligible for combination into a viable project and explain your criteria for determining which treatments are complementary?

1c. If not, do you consider treatment dominance for adjoining sections that do not have the same treatment types and explain your particular criteria for determining which treatments are dominant?

4

1d. Is there a maximum project length where you would not continue to combine sections and what parameters would you suggest be used to determine the maximum length?

2. If dTIMS does not generate a desired treatment in a desired year because it does not satisfy some other criteria, should dTIMS generate the treatment for consideration in a viable project?

3. Is there a time window that is considered practical for the combining of sections that have similar treatment types in different years? That is, do you consider moving a treatment one, two, three, etc. years from its optimal date of application?

Appendix '2'

Reference Manual Changes

The collection of all selected strategies is called the construction programme. You can change the construction programme by changing the selected strategy in the Strategy List window.

Developing a Construction Programme

Once dTIMS has generated a strategy list for each element, the next challenge becomes which strategy to choose. Optimization selects the best strategy for each element taking the objectives and constraints into consideration. The selected strategies form construction programme recommended by optimization.

However, optimization cannot possibly consider all of the other factors involved in developing a construction programme. This is why dTIMS allows you to review and adjust the construction programme. You do this in the Strategy List window. This window allows you to make any strategy in the element's strategy list, the selected strategy; thereby changing the construction programme.

Some of the reasons you may want to change the selected strategy for any element are, for example:

(1) Political considerations;
(2) To combine the work on this element with an adjacent element to make one construction project;
(3) To spread the work around your geographical area;
(4) To see the network impacts of various "what-if" scenarios.

Alternatively, dTIMS has a function that will allow you to combine elements with similar strategies into programme projects. The section called Programme Development discusses this function.

Automatic Programme Development

dTIMS has a function (see Programme Development window) which automatically generates a 'first-cut' set of programme projects. This function automatically goes through the selected strategies for an existing budget scenario and, to the extent possible, will do an analysis to determine which adjacent elements (e.g., sections) can be joined together into programme projects.

When joining adjacent elements together into programme projects, dTIMS will automatically move selected strategies to different years, or, will switch selected strategies altogether. To help you keep track of the changes it makes, dTIMS stores these new selected strategies in the last (i.e., fifth) budget scenario for the current analysis set. Then, dTIMS gives you a function (see Strip Map window) to compare the new programme with the original optimized one, and to adjust the new programme manually. This points out the need for you to name the fifth budget scenario accordingly (e.g., 'Programme') and give it the budget you want the Programme Development function to use.

Programme Development Logic

It would seem the easiest way for dTIMS to join adjacent elements is to start at the beginning of each road and proceed along it. This, however, involves very complicated logic. Therefore, as the flowchart shown later reveals, the basic idea of the Programme Development function is to select a road and to work on one treatment at a time. dTIMS finds the first element with a certain treatment and proceeds in both directions along the road from there. (In this context we use the term 'treatment' to refer to the first treatment in the selected strategy of the current budget scenario for an element.) dTIMS processes treatments in the order you set in a **Treatment precedence** tab in the Treatment Property window. dTIMS selects sections along the road using this treatment precedence and year of application. In other words, dTIMS processes all of the sections with the highest precedence treatment in year one, then, in year two, etc before dTIMS processes the next treatment type. The flowchart, shown in Figure 15, illustrates how this function works.

dTIMS will find the first element whose selected strategy has the highest precedence treatment as its first treatment in the earliest year. Then, dTIMS will start a new programme project, which includes only that element. Next, dTIMS will check to see if there is a 'next' element. (The term 'next' refers to increasing offsets along the road where a section is 'next' if and only if its from offset equals the current section's to offset. Similarly, a section is 'previous' if and only if its to offset is the same as the current's section's from offset.) dTIMS will keep processing next elements until there are no more. Then, dTIMS will check the 'previous' element and will keep processing until there are no more previous elements.

After processing all roads with the above logic, dTIMS will use a second pass to adjust the application years on the newly selected strategies of the programme projects based on the available budget. This second pass (shown on the above flowchart as 'Budget ranking process') is based on ranking the programme projects in each year of the first five years by their present value benefits. Then, programme projects at the bottom of the list (i.e., with the least benefits) will be moved into the next year if the budget for the current year is exceeded. The total yearly cost dTIMS uses for comparing against the budget is estimated by summing the costs of the first treatment and any associated ancillary treatments for each selected strategy.

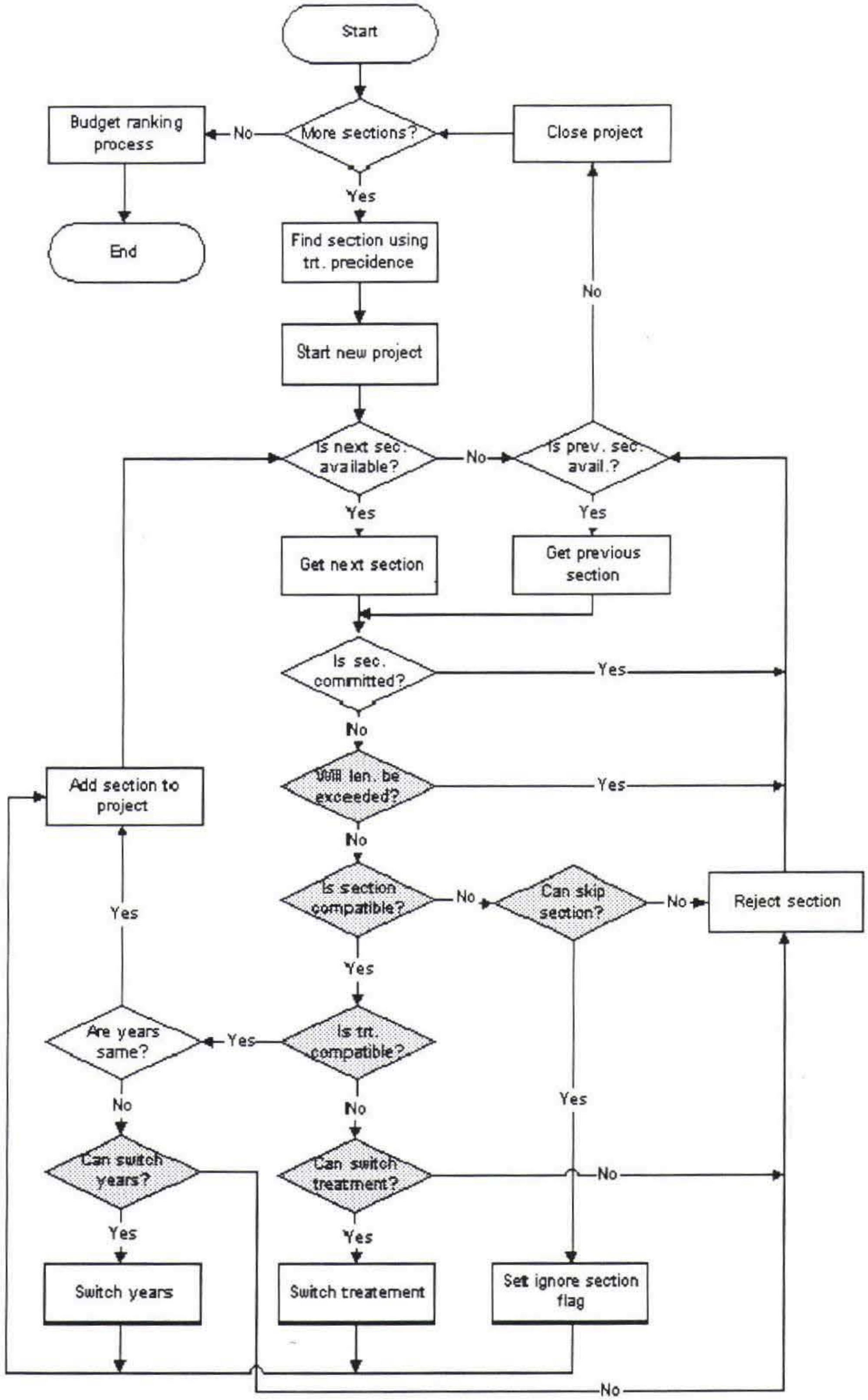

Start
More sections?
No
Budget ranking process
End
Yes
Find section using trt. precidence
Start new project
Close project
No
Is next sec. available?
No
Is prev. sec. avail.?
Yes
Yes
Get next section
Get previous section
Is sec. committed?
Yes
No
Will len. be exceeded?
Yes
No
Add section to project
Is section compatible?
No
Can skip section?
No
Reject section
Yes
Yes
Are years same?
Yes
Is trt. compatible?
Yes
No
No
Can switch years?
Can switch treatment?
No
Yes
Yes
Switch years
Switch treatement
Set ignore section flag
No

Figure 15

✎ ***Note:***
It is possible for you to declare your own programme projects by combining elements manually. (You do this by giving adjacent elements the same value for PROGPROJID.) If the programme project you combine manually have different first treatments and/or different first treatment years, dTIMS will treat the entire programme project as if all elements had the same first treatment year as the first element. (You will notice this particularly on the Strip Map window.) Also, you will notice that during this budget ranking process dTIMS will change treatment years for all elements in the programme project to the year after the first treatment year of the first element in the programme project.

Decision Trees

To generate programme projects dTIMS must answer each of the questions in the flowchart. Questions are denoted as diamond boxes. To help dTIMS answer these questions, you must define **decision trees**. You define these decision trees in the Programme Development window. In all, dTIMS has a total of six decision trees; one for each highlighted question diamond in the flowchart. The questions are:

1. Will maximum project length be exceeded by adding adjacent section?

2. Is adjacent section compatible with project?

3. Can adjacent section be included in project by being 'skipped' over?

4. Is treatment on adjacent section compatible with project?

5. Can treatment on adjacent section be switched?

6. Can application year of adjacent section's treatment be switched?

In order to unambiguously answer any of these questions in the flowchart, dTIMS must analyze a huge number of possible combinations of values. Making logical statements with AND's and OR's to cover this would be very complex. To reduce this complexity, you construct branches of a decision tree, one variable at a time. (You define these branches in the Decision Tree Branch property window.)

By picturing the answer as an exploration along limbs of a "tree" of possibilities, this model will help you visualize the vast number of possible combinations more easily. The root of the tree signifies the question that must be answered, and that's the point where the exploration starts. Each branch of the tree represents a decision, and the leaves of the tree indicate the possible answer of either 'Yes' or 'No'. The exploration ends when you arrive at a single leaf. As you take one path or another, the ambiguity in the problem is reduced and an answer is eventually arrived at. The further out you go along the decision tree's branches, the less ambiguity there is to resolve. When you reach the final answer, all decisions have been made and there is no remaining ambiguity.

Decision Tree Branches

For example, consider the decision tree shown in Figure 16.

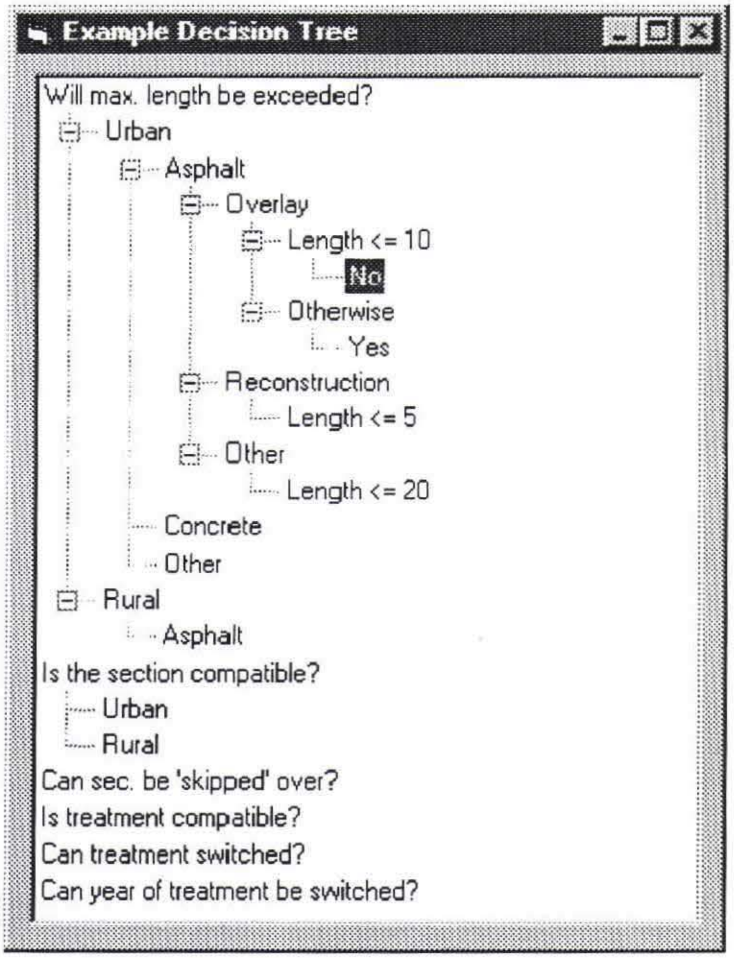


Figure 16

Figure 16 shows one path for the "Will maximum length be exceeded?" question which leads to a 'No' answer. You can interpreted this path as:

```
IF Urban AND Asphalt AND Overlay AND Length <= 10
THEN No
```

In this example path, each branch is labeled with a name that describes a certain condition. Fortunately, we selected the names so that the condition was obvious. For instance, the name 'Asphalt' stands for condition which groups asphalt pavement types together. The Decision Tree Branch property window you define for the Asphalt branch might look like Figure 17.

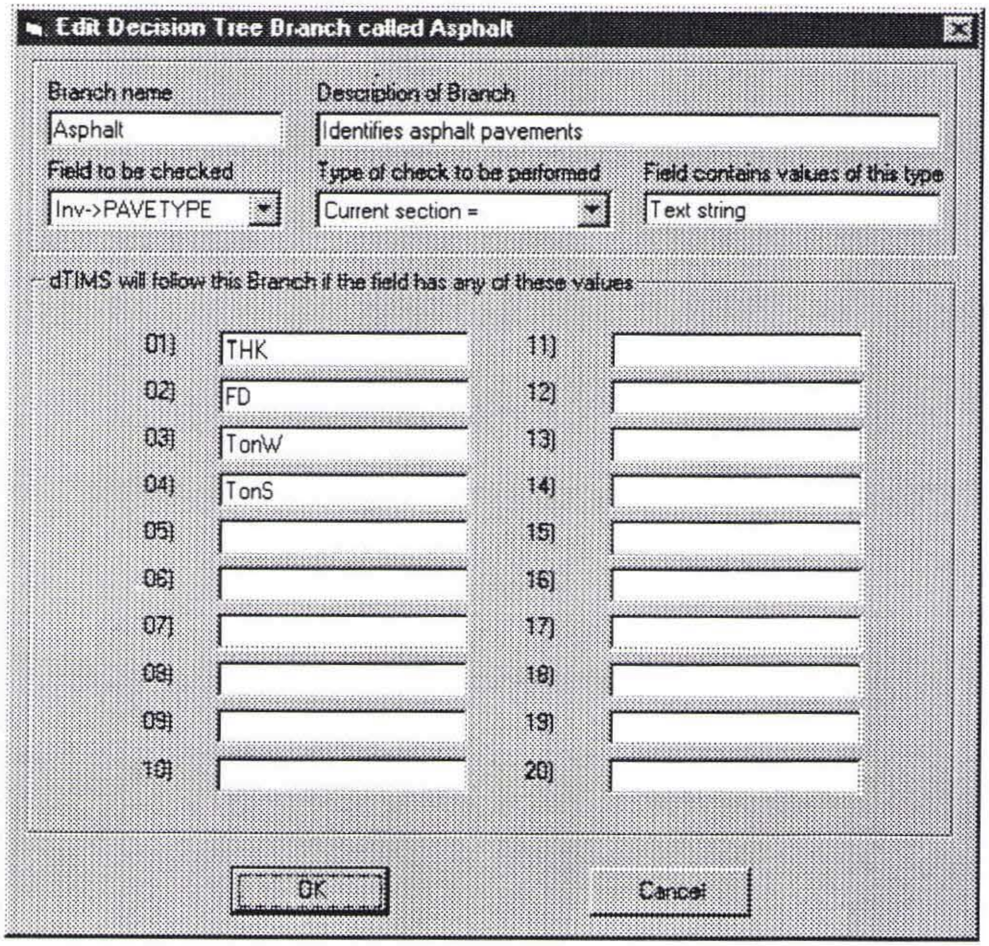


Figure 17

The pavement type field in this dTIMS is 'PAVETYPE'. Since PAVETYPE is defined in the inventory file, dTIMS appends 'Inv-' in front of its name. The branch tells dTIMS to check the value for the current section only. The condition that the 'Asphalt' branch defined above represents is:

```
PAVETYPE = "THK" OR PAVETYPE = "FD" OR PAVETYPE =
"TonW" OR PAVETYPE = "TonS"
```

Substituting this condition in the path for the 'Asphalt' branch, the original path condition statement shown earlier would now look like this:

```
IF Urban AND (PAVETYPE = "THK" OR PAVETYPE = "FD" OR
PAVETYPE = "TonW" OR PAVETYPE = "TonS") AND Overlay
AND Length <=10 THEN No
```

Before dTIMS evaluates a path on the decision tree it substitutes the conditions defined for each branch. For example, the above path might end up looking like this:

```
IF (FUNC CLASS = "Urban") AND (PAVETYPE = "THK" OR
PAVETYPE = "FD" OR PAVETYPE = "TonW" OR PAVETYPE =
"TonS") AND (FTRT1TYPE = "SOVL" OR FTRT1TYPE =
"TOVL") AND (LENGTH <=10) THEN No
```

Order of processing branches

A final note about how dTIMS processes the decision trees involves a discussion regarding the order in which it checks the branches. In the above example, dTIMS first checked the Urban branch. To continue down that path, dTIMS had to get a 'True' from that branch. In other words, the FUNC CLASS value had to be 'Urban'. If the value had been anything other than 'Urban', dTIMS would have abandoned the path and would have checked the Rural branch. You set the order that dTIMS follows through the decision tree by the order you place the branches at each fork along the tree. Although not shown in the above decision tree figure, dTIMS automatically supplies a default 'Otherwise' branch as the last branch in every decision point. The 'Otherwise' branch is always 'True', therefore, dTIMS will always find a path through the decision tree.

Steps To Implement the Programme Development Function

To implement the Programme Development function, you must do the following:

1. Place valid values in the FNAME, FOFFFROM, FOFFTO, FDESCFROM and FDESCTO fields in the inventory file (see Base Data Fields for a description of these fields).

✎ ***Note:***

dTIMS considers all elements in the inventory file with the same FNAME values as being on the same 'Road'. By extension then, the FOFFFROM and FOFFTO values tell dTIMS the relative placement of the elements along the road; they should not overlap one another. If you are using dTIMS to analyze elements which are not roads you must 'trick' this function into believing that the elements that belong together occur consecutively on the same road. For example, if you are analyzing bridge components as separate elements, (e.g., deck, superstructure, substructure, etc.) You must place these components on the same road by giving all of them the same value in the FNAME field (e.g., the bridge name). Then you must set the FOFFFROM and FOFFTO values so that it appears to dTIMS that they occur consecutively (e.g., deck goes from 0 to 1, superstructure goes from 1, to 2, substructure goes from 2 to 3, and so on).

2. Place two fields in the inventory file: FSEC REC and PROGPROJID (see Special Data Fields for a description of these two fields).

✎ ***Note:***

You must create the PROGPROJID text field to be wide enough to hold a unique identifier for each programme project (e.g., 20 bytes to be safe). Before you can use the Strip Map window you must ensure that this field has valid identifiers in it. You can fill these in manually (e.g., by copying the values from FSECTION which essentially places each element in its own programme project). Or, you can have dTIMS automatically fill in this field by executing the Program Development function. Elements in the inventory file that belong to the same programme project will have the same identifier in this field.

3. Name the fifth budget scenario in the respective analysis set to something that will remind you that dTIMS will place the selected strategies for the programme projects in this scenario.

4. Supply the desired budget to the fifth budget scenario.

5. Adjust the treatment precedence for your rehabilitation treatments in the Treatment precedence tab of the Treatment property window.

6. Generate the strategies for the respective analysis set (from the Tools menu).

7. Run Optimization for all budget scenarios in the analysis set including the fifth one (from the Tools menu).

8. Design your programme development decision tree **ON PAPER FIRST!**

> ***Note:***
> dTIMS does not 'know' anything at all about deciding which elements to combine into programme projects. The only thing that dTIMS 'knows' is that it must answer the six questions listed earlier, and, that to answer them it will follow branches along a path of the decision tree. Which path it will follow depends on the branches you define. Therefore, to be sure dTIMS follows the correct path, you must carefully define your branches. Always make sure that the branches at any decision point are in the correct order because dTIMS checks them in the same order that you place them at each decision point. Don't forget that there is a default branch called "Otherwise" which is always true and which is always at the bottom of every decision point.

9. Define your branches (see the Decision Tree Branch property window).

10. Print a copy of your decision tree branches (use the Print function on the File menu) and keep it handy for defining your decision tree.

11. Open the Programme Development window (from the Tools menu) and define your decision trees.

12. Run the Programme Development function from the same window.

13. Open the Strip Map window (from the View menu) to review the results and adjust your programme manually.

DRP descriptions: (not shown) This line shows the description of each data reference point (DRP) on the road. You define these descriptions in the FDESCFROM and FDESCTO fields in the inventory file. You can suppress this part by unchecking the Show DRP descriptions control (see later).

Strip map:
This part is a thick black line with cross hatches at each DRP along the road. If the road does not have elements for its entire span, dTIMS displays these as a dotted line in this part.

DRP Addresses:
This part shows the offset for each DRP. You define these offsets in the FOFFFROM and FOFFTO fields in the inventory file.

DataField values:
This part shows the data of interest on the strip map. There are three line diagrams in this part. Each line shows a specific set of data as follows:

First Line: Shows the elements from the inventory file and displays data describing the first treatment of the selected strategy as assigned by optimization for the budget scenario you selected when you entered this function. You cannot change these selected strategies in this function; they are displayed for your information only so you can compare what the optimization function selected for these elements. The label on top of these elements is the element ID from the FSECTION field of the inventory file. The labels on the left hand side of this line are:

XXXXXXXXXX->: The name of the budget scenario you selected when you first entered this function. This is shown in Figure 69 as "High Funding->".

First Treatment Type: The name of the first treatment of the selected strategy in this budget scenario.

First Treatment Year: The year of the first treatment of the selected strategy in this budget scenario.

First Treatment Cost (yy): The cost of the first treatment of the selected strategy in this budget scenario. The cost units are shown as (yy).

Present Value Benefits: The present value benefits of the selected strategy in this budget scenario.

Second Line: Shows the elements from the inventory file and displays the first treatment of the selected strategy for the fifth budget scenario in the current analysis set. The selected strategies in the fifth budget scenario are the ones used to form the programme project. You can select one of these elements by clicking on it and use the control buttons on the strip map to change these selected strategy so you can develop your programme projects. The label on top of these elements is the element ID from the FSECTION field of the inventory file. The labels on the left hand side of this line are:

XXXXXXXXXX->: The name of the fifth budget scenario in the analysis set you selected when you first entered this function. This is shown in the above figure as "No Funding->".

First Treatment Type: The name of the first treatment of the selected strategy in this budget scenario.

First Treatment Year: The year of the first treatment of the selected strategy in this budget scenario.

First Treatment Cost (yy): The cost of the first treatment of the selected strategy in this budget scenario. The cost units are shown as (yy).

Present Value Benefits: The present value benefits of the selected strategy in this budget scenario.

Third Line: Shows the programme projects and displays some information summarizing the treatments from the elements that make up the project. The label on top of these elements is the project ID from the PROGPROJID field of the inventory file. The labels on the left hand side of this line are:

PROGRAM PROJECT->: A label.

Year: The year of the programme project. dTIMS defaults this to the year of first treatment of the first element in the project. Since the idea of programme projects is to group together elements for programming purposes, dTIMS assumes that the first treatment for all elements in the project will be the same, therefore, it defaults the year to the first element. It is up to you to ensure this is so when you manually adjust the strategies for the elements in the project.

Total Cost (yy): The sum of the costs of all the first treatments from all the elements in the project. The cost units are shown as (yy).

Total Benefit: The weighted average of the present value benefits of all selected strategies from all elements in the project. We use weighted average to combine these because the benefits are given as per unit length.

Total Length (zz): The sum of the lengths of all elements in the project. This does not include the length of missing pieces (dotted lines) along the road. The length units are shown as (zz).

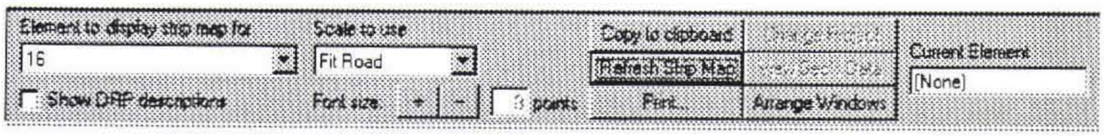


Figure 74

The top part of the strip map has a number of different controls which allow you to manipulate the strip map and the program projects. The top part looks like this:

These controls are as follows:

Element to display strip map for: Select the road element you want the strip map displayed for. dTIMS compiles this list of road names by scanning the inventory file and selecting all unique names from the FNAME field. When you select an element in this control, dTIMS redraws the strip map.

Scale to use: You have a choice of three scales to display the strip map for. When you change the scale, dTIMS redraws the strip map.

Fit Road: Select this to cause dTIMS to force the entire length of the road element into the window.

By Length:
Select this to cause dTIMS to force a certain length of the road element into the window. When you select this, dTIMS asks you for the length you wish to show.

By DRP: Select this to cause dTIMS to force a certain number of DRPs on the road element into the window. When you select this, dTIMS asks you for the number of DRPs you wish to show.

Show: This is where you specify the length, or the number of DRPs you want dTIMS to fit into the current window size.

Show DRP descriptions: Check this box to have dTIMS show descriptions for each DRP. dTIMS writes these descriptions vertically above the DRP's location. To do this dTIMS must rotate the font. When you uncheck this control, dTIMS suppresses the DRP descriptions. This will save an incredible amount of screen space to help you display strip maps in lower resolution.

Font size: Use the + and - buttons to increase and decrease the size of the font dTIMS uses to place information on the strip map.

Copy to clipboard button: Press this button to send a copy of the strip map to the clipboard. dTIMS will redraw the strip map to the clipboard. You can then paste this drawing as a bitmap into any document.

Refresh Strip Map button: Press this button to refresh the data on the strip map. dTIMS will redraw the strip map with the latest information from all the source files. You use this button after you select another strategy for one of the elements in on the strip map.

Print button: Press this button to produce a print out of the current strip map. dTIMS redraws the strip map to the printer object. Therefore, you must make sure that your printer can accept graphic characters and is capable of rotating fonts.

Change Project button: Press this button to change the programme project for the current element. This button is enabled when you select a Current element (see Current Element later). When you press this button dTIMS displays the Change Project window.

View Sec's Data button: Press this button to see the Inventory Data window and the Strategy List window for the current element. This button is enabled when you select a current element (see Current Element later). When you press this button dTIMS opens the Inventory Data window (if it is not already open) and opens the Strategy List Window (if it is not already opened). dTIMS finds the record in the Inventory Data window for the current element and sets Strategy List window to this element for the fifth budget scenario. You can change the selected strategy for the current element on the Strategy List window.

Arrange Windows button: Press this button to conveniently arrange the windows for better viewing of the data. When you press this button dTIMS maximizes itself and displays the strip map across the top half of the available space. dTIMS then puts the Inventory Data window on the left half of the available space below and the Strategy List window on the right half. This arrangement works best on computers with high resolution.

Current Element: Shows the element ID for the currently selected element. You select an element by placing the mouse cursor on top of the element's line (shown in the fifth budget scenario's part) and clicking. dTIMS highlights the selected element's line in red.

Change Project Window

The Change Project window appears on the top part of the strip map when you press the Change Project button. The window looks like this:

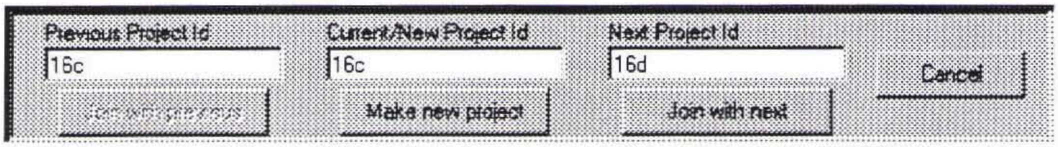


Figure 75

The Change Project window has the following components:

Previous Project Id: Shows the project ID for the programme project that the element to the immediate left of the current element belongs. The project ID is the value found in the PROGPROJID field of the Inventory file.

Current/New Project Id: Shows the project ID for the programme project that the current element belongs. The project ID is the value found in the PROGPROJID field of the inventory file. If you want to make the current element belong to a new project you put the new project's ID in this field.

Next Project Id: Shows the project ID for the programme project that the element to the immediate right of the current element belongs. The project ID is the value found in the PROGPROJID field of the inventory file.

Join with previous button: dTIMS enables this button only when the Previous Project ID is different than the Current/New Project ID. When you press this button dTIMS removes the current element from the current project and adds it to the previous project. dTIMS does this by changing the value in the PROGPROJID field of the inventory file for the current element to the value shown in the Previous Project Id control.

Make new project button: dTIMS enables this button only when the Previous Project ID is different than the Next Project ID. When you press this button dTIMS removes the current element from the current project and adds it to a new project. Before you press this button you must enter a project ID for the new project in the Current/Next Project Id field.

Join with next button: dTIMS enables this button only when the Next Project ID is different than the Current/New Project ID. When you press this button dTIMS removes the current element from the current project and adds it to the next project. dTIMS does this by changing the value in the PROGPROJID field of the inventory file for the current element to the value shown in the Next Project Id control.

Cancel Button: Press this button to remove the Change Project window.

Programme Development Window

The Programme Development window gives you access to the programme development decision tree and the function which actually generates the programme projects. See the section called Automatic Programme Development for a complete background to the concept of programme development and decision tree branches.

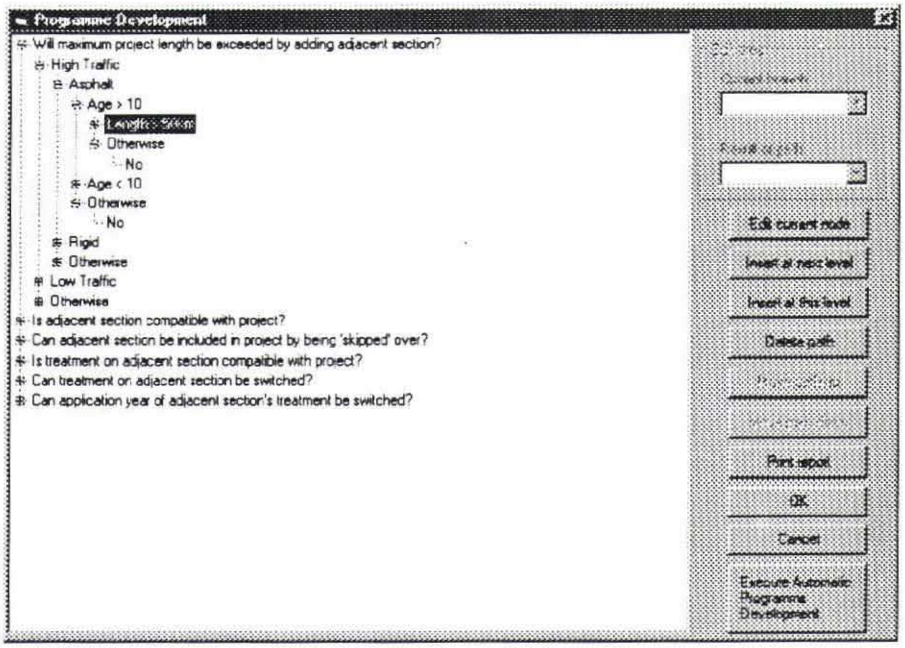


Figure 83

The window contains the following information:

Decision tree: The area to the left hand side of the window contains the programme development decision tree. You modify the paths on this decision tree using the facilities provided on this window. The components on this decision tree are as follows:

Path: A path is a collection of related nodes which starts at a question node and ends at a result node. The relationship between the nodes along the path can be best described using the family analogy. Each node (except the question node) along the path has a parent. And, each node (except the result node) has a child. A path is best described as the collection of parent nodes for any result node. If you press the Delete current path button while on a branch node, dTIMS will delete all paths that that are related to that branch node.

When transversing a path it is convenient to think of the nodes as being at different levels. The level of a node is relative. The node's parent is up one level, its child is down one level and its siblings are at the same level. We use the term 'level' in this window to describe where to insert a node. dTIMS allows a maximum of twenty levels (not including the question level) for any path.

Question node: There are a total of six questions shown here. These are the roots of the decision tree. They correspond to the decision points highlighted in the flowchart presented in the Automatic Programme Development section of this manual. You cannot perform any functions on this node other than expand and collapse it.

Branch node: A branch node represents a decision point on the decision tree. It contains the name of one of the branches you created in the Decision Tree Branch properties window. A branch node is a point where dTIMS will make a decision when developing the programme. If dTIMS finds the branch to be true, it will continue along that branch's path by jumping to the first node of the next level. If dTIMS finds the branch to be false, it will jump to the node directly beneath it at the same level (which could be a branch node or an otherwise node).

You can edit the name of the branch in a branch node using the edit button. You can also change the order of the branch node by moving it up and down within its level using the respective move buttons. Finally, you can add a new branch node either at the same level or at the next level using the respective insert buttons.

Otherwise node: An otherwise node represents a decision point that is always true. In other words, when dTIMS is developing the programme and it has found all of the nodes at a particular level to be false, it will eventually come to the otherwise node of that level. dTIMS always evaluates an otherwise node as true, and, the otherwise node always has a result node as its only child. This way dTIMS is always guaranteed a path through the decision tree.

You cannot add, delete, edit or change the order of an otherwise node. dTIMS maintains the otherwise nodes automatically for you.

Result node: A result node is the last node of a path. A result node tells dTIMS the answer to the question at the root of the path. The result node can have a value of either 'Yes' or 'No'. You can edit the result node by pressing the edit node button. However, you cannot add, delete or change the order of a result node. dTIMS maintains the result nodes automatically for you.

Edit area: This is the area on the window where you can edit the branch for a branch node, or the result for a result node. This area is disabled until you press the Edit current node button. When you press that button dTIMS enables one of the following controls depending on what type of node you were on when you pressed the button:

Current Branch: If you were on a branch node when you pressed the edit button, dTIMS will enable this control. It contains a list of available branch names. Select the new branch name you want from this list. dTIMS will change the name of the branch and will disable the control again.

Note: The list of branch names dTIMS presents will not have the names of any of the parent branches or any of the branches at the same level. dTIMS does this to remind you that having the same branch name at two levels of the same path is not good logic. For this reason, sometimes the list only contains the current branch name meaning you have no more branches to change this one to.

Result of path: If you were on a result node when you pressed the edit button, dTIMS will enable this control. It contains a list of two items: 'Yes' and 'No'. Select the result you want for the current path from this list. dTIMS will change the result and will disable the control again.

Edit current node button:
This button is only enabled when you are on a branch or a result node. Press this to edit the value for this node. dTIMS will enable the Edit area and the respective control within it according to the description provided above.

Insert at next level button:
This button is only enabled when: (a) you are on a branch node that is not at the twentieth level, or (b) when you are on a question node whose only child is the otherwise node. Press this to add a branch node as a child of the current node. dTIMS will find the first available branch name and will insert a branch node with that name at the top of the next level.

NOTES:

1) dTIMS automatically adds a result node as a child to the new node, and sets its value to 'Yes' by default.
2) If the current node only has one child (i.e., a result node) when you press this button, dTIMS will also add an otherwise node below the new node. dTIMS gives the otherwise node a child result node whose value is defaulted to 'No'.
3) If all of your branches you defined in the Decision Tree Branch properties window are already involved in the current path, dTIMS does not insert a new branch. Rather, it issues a message saying 'Cannot insert. All available branches have been used in path already.'

Insert at this level button:
This button is only enabled when you are on a branch node. Press this to add a branch node as a sibling of the current node. dTIMS will find the first available branch name and will insert a branch node with that name at the same level as the current node.

NOTES:

1) dTIMS inserts the new branch node directly after the current node.
2) dTIMS automatically adds a result node as a child to the new node, and sets its value to 'Yes' by default.
3) If all of your branches you defined in the Decision Tree Branch properties window are already involved in either the current path or the current level, dTIMS does not insert a new branch. Rather, it issues a message saying 'Cannot insert. All available branches have been used in path already.'

Delete path button:
This button is only enabled when you are on a branch or a result node. Press this to delete the path involving this node. dTIMS will remove the path to the current node. .

NOTES:

1) If the current node has any children, dTIMS will delete those paths as well.
2) If the current node has an otherwise path as its only sibling, dTIMS automatically: (a) removes that path as well, and (b) adds a result node to the current path's parent node and sets its value to 'Yes' by default.

Move path up button:
This button is only enabled when you are on a branch node and (a) that branch node is not at the top of the current level, and (b) that branch node has more siblings than just the otherwise node. Press

this button to move the current branch node up in the current level. dTIMS moves all of the paths associated with the current branch node above the sibling node appearing directly above it at the same level.

Move path down button:
This button is only enabled when you are on a branch node and (a) that branch node is not directly above its sibling otherwise node, and (b) that branch node has more siblings than just the otherwise node. Press this button to move the current branch node down in the current level. dTIMS moves all of the paths associated with the current branch node below the sibling node appearing directly below it at the same level.

Print report button:
dTIMS prints a report showing the current programme projects.

OK button:
dTIMS saves all changes to the decision tree and closes the window.

Cancel button:
Press this button to cancel all the changes you made to the decision tree since entering this function. dTIMS cancels all changes and reverts the decision tree to the state it was in when you entered this function. dTIMS then closes the window.

Execute Automatic Programme Development button:
Press this button to have dTIMS generate programme projects using the current configuration of the decision tree. After you press this button, dTIMS shows the Analysis Set selection window and asks you to select the analysis set and the budget scenario you want dTIMS to evaluate when creating the programme projects. At the bottom of this same window, dTIMS asks you whether you want it to 'blank out' the PROGPROJID values in the Inventory file for the current analysis set. The notes below explain the significance of blank PROGPROJID values.

Notes:
1) dTIMS automatically puts the results of this function in the fifth budget scenario of the analysis set. This means that the selected strategy in the fifth budget scenario for each element is the strategy dTIMS wants for the programme project associated with that strategy. In fact, before dTIMS starts generating programme projects, it copies the selected strategy from the budget scenario you selected to the fifth budget scenario for all elements with blank PROGPROJID's. Then, during the function, dTIMS modifies the selected strategy in the fifth budget scenario according to the programme project development rules.

2) If you check the 'blank out PROGPROJID' check box, dTIMS will go through the Inventory file and will replace the PROGPROJID values with blanks for each element in the current analysis set. At the same time, dTIMS will delete any strategies it automatically generated for each of these elements during previous executions of this function.

> * The only way dTIMS has to identify which element belongs to the analysis set at this stage, is to find elements that have selected strategies in the budget scenario you selected earlier. Furthermore, one of the only ways an element gets a selected strategy in a budget scenario is for you to run optimization for that budget scenario. Therefore, YOU MUST HAVE ALREADY RUN OPTIMIZATION ON THE SELECTED BUDGET SCENARIO before this function will work correctly.

3) During this function, dTIMS will only generate programme projects for elements with blank PROGPROJID fields. You can make the PROGPROJID values blank using the above check box, or, you can make them blank by hand in the inventory file. The advantage of this second approach is that it allows you to 'commit' programme projects. You can commit consecutive elements in the inventory file to a programme project by providing similar values in their PROGPROJID field.

4) When developing programme projects, dTIMS assumes that an element with the do-nothing strategy as the selected strategy cannot be joined to an existing programme project. Therefore, dTIMS will place each such element in its own programme project.