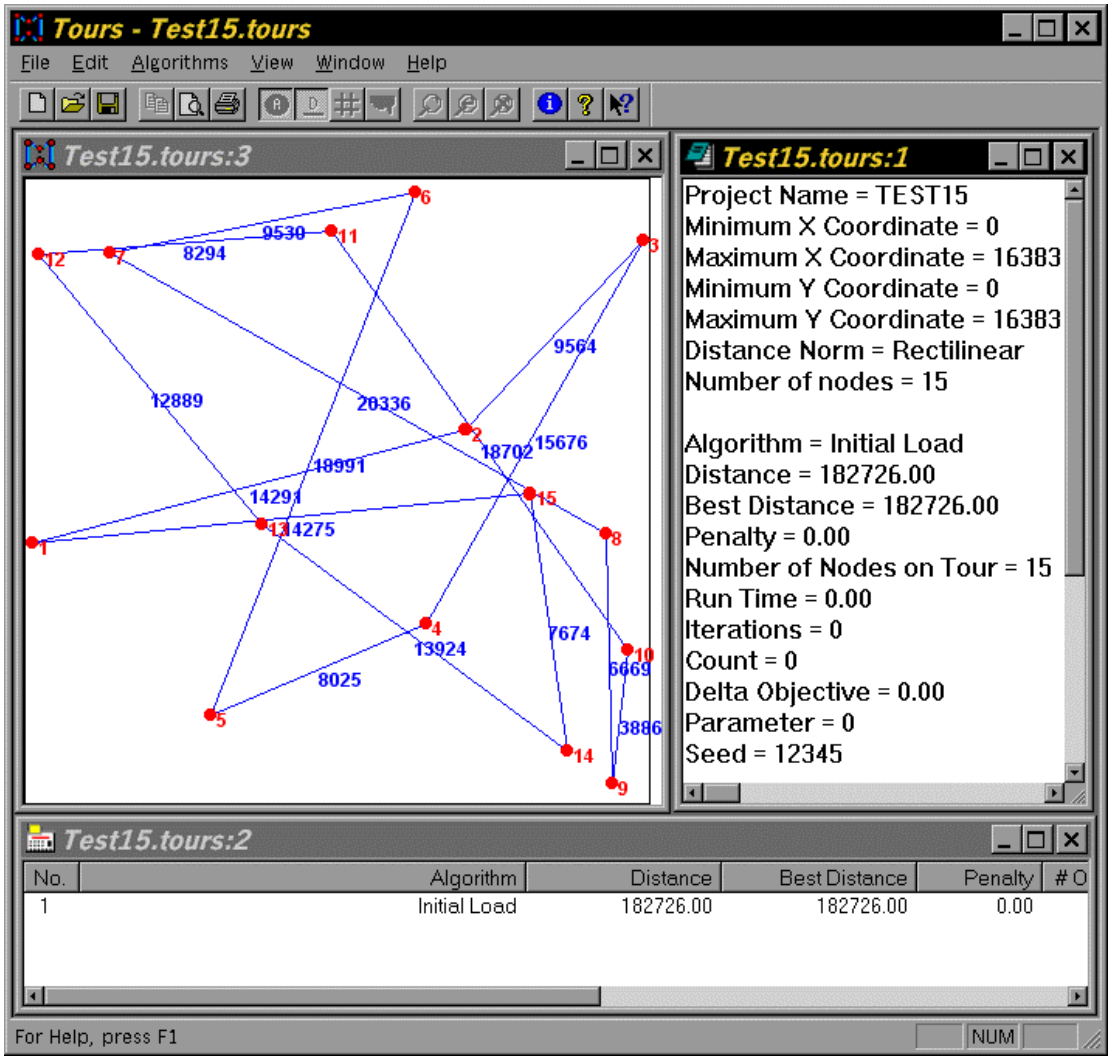
# Tours

## User's Manual




By Marc Goetschalckx

***Tours User's Manual, Version 4.00, April 19, 2000***


**Marc Goetschalckx**
**4031 Bradbury Drive**
**Marietta, GA 30062-6165**
**+1-770-578-6148**
**+1-770-565-3370**
**Fax: +1-770-578-6148**

# Contents

# Disclaimer

## Warranty

Marc Goetschalckx's entire liability and your exclusive remedy under this warranty (which is subject to you returning the program to Marc Goetschalckx) will be, at Marc Goetschalckx's option, to attempt to correct or help you around errors with efforts which Marc Goetschalckx believe suitable to the problem, to replace the program or diskettes with functionally equivalent software or diskettes, as applicable, or to refund the purchase price and terminate this agreement.

Marc Goetschalckx warrants that, for a period of ninety (90) days from the date of delivery to you as evidenced by a copy of your receipt, the diskettes or CD-ROM on which the program is furnished under normal use will be free from defects in materials and workmanship and the program under normal use will perform without significant errors that make it unusable.

Except for the above express limited warranties, Marc Goetschalckx makes and you receive no warranties, express, implied, and statutory or in any communication with you and Marc Goetschalckx specifically disclaims any implied warranty of merchantability or fitness for a particular purpose. Marc Goetschalckx does not warrant that the operation of the program will be uninterrupted or error free. It is your responsibility to independently verify the results obtained by this program.

In no event will Marc Goetschalckx be liable for any damages, including loss of data, lost profits, cost of cover or other special, incidental, consequential or indirect damages arising from the use of the program or accompanying documentation, however caused and on any theory of liability. This limitation will apply even if Marc Goetschalckx or any authorized dealer has been advised of the possibility of such damage. You acknowledge that the license fee reflects this allocation of risk.

Some states do not allow the exclusion of implied warranties so the above exclusions may not apply to you. This warranty gives you specific legal rights. You may also have other rights, which vary from state to state.

# Proprietary Notice



# Version

Version 4.00, April 8, 2000.

# Chapter 1. Installation

## Installing Tours

To install **Tours** you must run the **Setup** program on the distribution disk. The exact method of executing the **Setup** program depends on the version and type of Windows operating system that is installed on your computer.

Copying the files from the distribution disk to your computer or executing the program from a file or application server is not sufficient to run **Tours**. Several dynamic link libraries, such as the **Scientif** application library, and active-x controls, such as the grid control, are required for the proper execution of the **Tours** program and must be registered on your computer. The **Setup** program copies and registers these libraries and controls during its installation process.

To remove the **Tours** application completely and safely from your computer see the instructions in the section on Removing Tours.

### Installation Instructions for Windows NT 4.00 and Windows 95 and 98

1. Insert the distribution disk 1 into the floppy or CD-ROM disk drive ***d:***.
2. In the **Control Panel**, select **Add/Remove Programs** and then press the **Install/Uninstall** tab.
3. The Windows operating system will search for the installation program on the floppy or CD-ROM drives ***d:*** and will identify ***d:\setup.exe*** as the installation program.
4. Press **Finish** to start the installation procedure.
5. Follow the instructions of the SETUP program.

If your floppy or CD-ROM disk drive is not drive ***d:***, substitute the appropriate disk drive letter with colon for ***d:*** in the above instructions.

Alternatively, you can also install **Tours** using the installation instructions for Windows NT version 3.51.

### Installation Instructions for Windows NT 3.51

1. Insert the distribution disk 1 into the floppy or CD-ROM disk drive ***d:***.

2. In the Windows Program Manager, select the **Run** command from the **File** menu.
3. In the Command Line box type:

   ***d:\setup***

4. Choose **OK** to start the installation procedure.
5. Follow the instructions of the SETUP program.

If your floppy or CD-ROM disk drive is not drive ***d:***, substitute the appropriate disk drive letter with colon for ***d:*** in the above command.

## Installation Notes

### *Scientif Dynamic Link Library*

**Tours** uses the common **Scientif** application dynamic link library (DLL). Other programs may use the same library with the same file name ***scienmfc.dll*** but may require a different version of the library. We recommend that you store the ***scienmfc.dll*** library file included on the distribution disk in the **Tours** application directory. This will ensure that the **Tours** program always will use the proper version of the library, even if you install other applications that use the same **Scientif** application library.

### *Write Access Privileges*

When installing to a Windows NT system, make sure you have write access privileges to the directory where **Tours** will be installed and to all the files in this directory and its subdirectories. You must also have write access privileges to the \Winnt\System32 directory and the ***msflxgrd.ocx*** file in that directory. We recommend that you install **Tours** while being logged on as administrator.

### *Grid Controls msflxgrd.ocx*

Several commands, such as the **All Distances** command of the **Edit** menu, require that the active-x grid control ***msflxgrd.ocx*** is present and registered on the computer that is executing the **Tours** program. The automated **Setup** program copies and registers the grid control during its installation process. If you run **Tours** from a file or application server, the control must be installed on every client computer. Many commercial applications use the same grid control, so it may already be installed on your computer. To verify the presence and registration of the grid control, open the Tours15 project, which is included on the distribution disk, and execute the **All Distances** command. If the two-dimensional distance matrix is not shown, the grid control is not installed or registered. In that case run the **Setup** program to install **Tours** and the grid control on this computer.

# Removing Tours

You can remove the **Tours** program and its application libraries from your computer, as well as remove its keys from the **Registry**. The exact method of removing **Tours** depends on the version and type of Windows operating system that is installed on your computer.

Since the removal procedure actually deletes files from your computer, some of which may be system level libraries or common controls, it is strongly recommended that you make a complete backup of all the files on your computer before proceeding with the removal procedure.

To install **Tours** on your computer see the instructions in the section on Installing **Tours**.

## Removal Instructions for Windows NT 4.00 and Windows 95 and 98

Select the **Add/Remove Programs** command in the **Control Panel** of your computer. **Tours** will be listed as one of the applications that can be automatically removed. Select **Tours** and press **Add/Remove**. All the files specific to **Tours**, such as the executable program file, the application help file and the **Scientif** application library, will be removed from your computer. All keys associated with **Tours** will also be removed from the **Registry**. Common libraries, such as the Microsoft Foundation class library and the Microsoft C runtime library will not be removed. **Tours** project files that you created in different directories will not be removed either.

## Removal Instructions for Windows NT 3.51

During installation, an icon to remove **Tours** from your computer was placed in the same program group that you selected for the **Tours** program. Press this **Uninstall** icon. All the files specific to **Tours**, such as the executable program file, the application help file and the **Scientif** application library, will be removed from your computer. All keys associated with **Tours** will also be removed from the **Registry**. Common libraries, such as the Microsoft Foundation class library and the Microsoft C runtime library will not be removed. **Tours** project files that you created in different directories will not be removed either.

This page left intentionally blank.

# Chapter 2. Tutorial

## Creating a Small Tutorial Project

Following are the step by step instructions to create a small Traveling Salesman Problem (TSP) project and design its tour in an interactive manner. For large projects it might be more convenient to create the necessary data files outside **Tours** with a file editor capable of creating pure ASCII files and then to import the project from these data files.

**Tours** follows the standard Windows graphical user interface (GUI) conventions. This tutorial will focus on the features unique to the **Tours** program and assumes that you are familiar with the Windows environment and the execution of Windows applications. More information on the Windows user interface can be found in the Windows User's Guide.

More detailed explanations and instructions can be found in the sections on the Project Data and Design Algorithms. A summary of all the available actions in **Tours** can be found in the section on the Command Reference. A list of traditional and World Wide Web (WWW) references for further reading is given in the Reference section. Finally, we will be using the Test15 example in this tutorial and the data for this project are summarized in the section on Sample Projects.

### Steps to Be Done Before Starting A New Project

#### *Create a project directory*

Determine in which directory you want to save this new project. It is strongly recommended that you use a separate directory for each project. If necessary, create this directory on your hard drive. Note that some versions of the Windows operating environment allow you to create this directory from the **Save As** dialog box while executing the **Tours** program. **Tours** is compatible with long file names and directory names.

#### *Copy the background map file*

The optional background map for a new project is imported from a map file. The default extension for a map file is *.map. Further information on the structure of the Map Data File is given in the section on **Map Data**. The map file ***usa.map*** contains the background map for the continental United States and has been placed in the Projects directory of the **Tours** directory during the program installation.

Copy the appropriate map file to the newly created directory. You can then later navigate to this map file from the **New Project** dialog window.

## Defining a New Project

Select the **New** command from the **File** menu or press the **New** button on the toolbar. The **New** dialog window will be shown. The initial **New** dialog window for this project is illustrated in Figure 2.1.

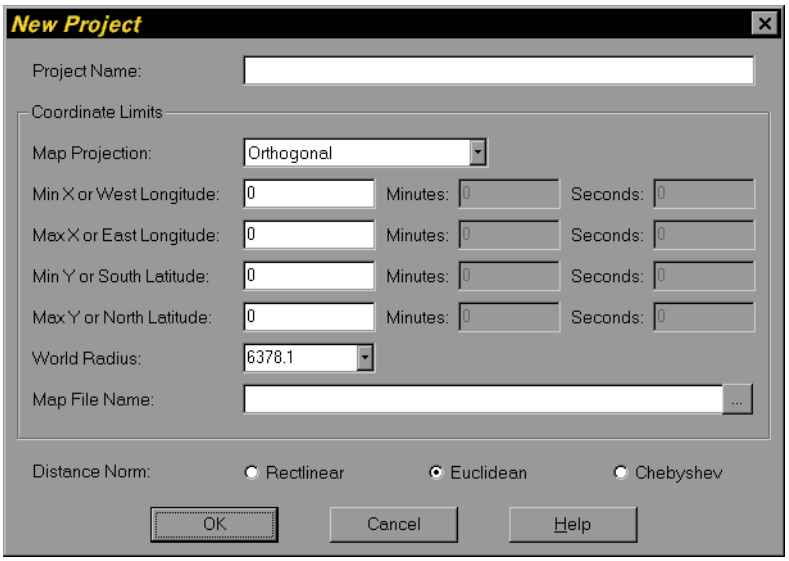



*Figure 2.1. Initial New Project Dialog Window for the Tutorial Example*

Enter a project name with a maximum of 63 characters. Only letters, digits, spaces, and underscores are allowed in the project name. For this project enter Tutorial as the project name. Select a map projection method. We will be using standard x and y coordinates in this tutorial, which corresponds to an orthogonal map projection method. Determine the minimum and maximum x and y coordinates for the project and enter them in the appropriate fields. For this tutorial project the minimum and maximum x and y coordinates are 0 and 16383, respectively. Since we are using the orthogonal map projection, the minutes and seconds fields of the world coordinates are not used. Since we are using the orthogonal map projection, the world radius is not used in the project. We also will not be using a background map in this project, so we leave the map file name field blank. The final **New** dialog window for this project is illustrated in Figure 2.2. Press **OK** and a new project without any points will be created.

Note that any data item of this project except the data items entered on the new dialog window can be changed later from inside the project with commands from the **File** and **Edit** menu. You will also be able to change the project name and the distance norm after the project has been created. You can get context sensitive help for any dialog window by pressing the Help button when the dialog windows is displayed.

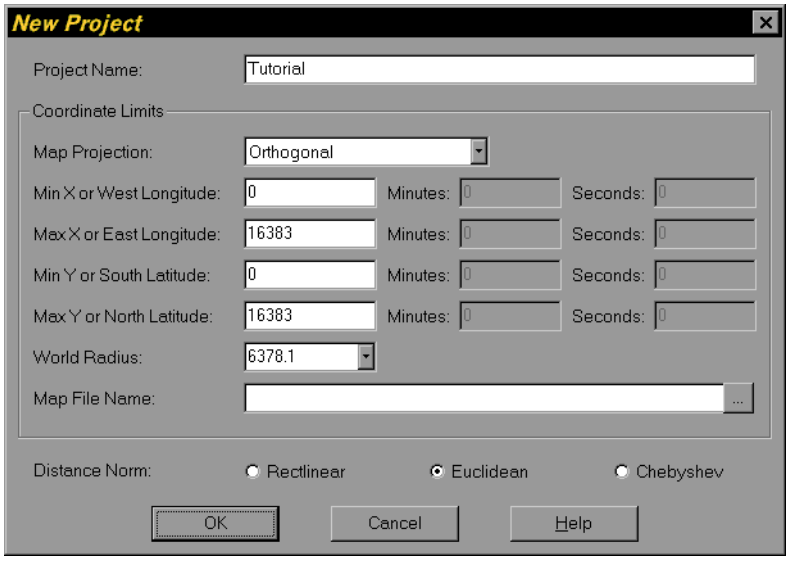



*Figure 2.2. Completed New Dialog Window for the Tutorial Project*

The **Tours** program shows three views of the new project in three cascaded windows. The original views are illustrated in Figure 2.3. Observe that the application title bar indicates that this is a new project that not has been saved yet. You can select the **Tile** command from the **Window** menu to display all the views simultaneously.

Each view has individual characteristics and display attributes. The type of view is indicated by an icon in the top left corner of the title bar of each view. The three view types are 1) **Project Notes** view, 2) **Algorithm Statistics** view, and 3) **Tour** view. Changing the attribute of one view does not affect the same attribute in other views.

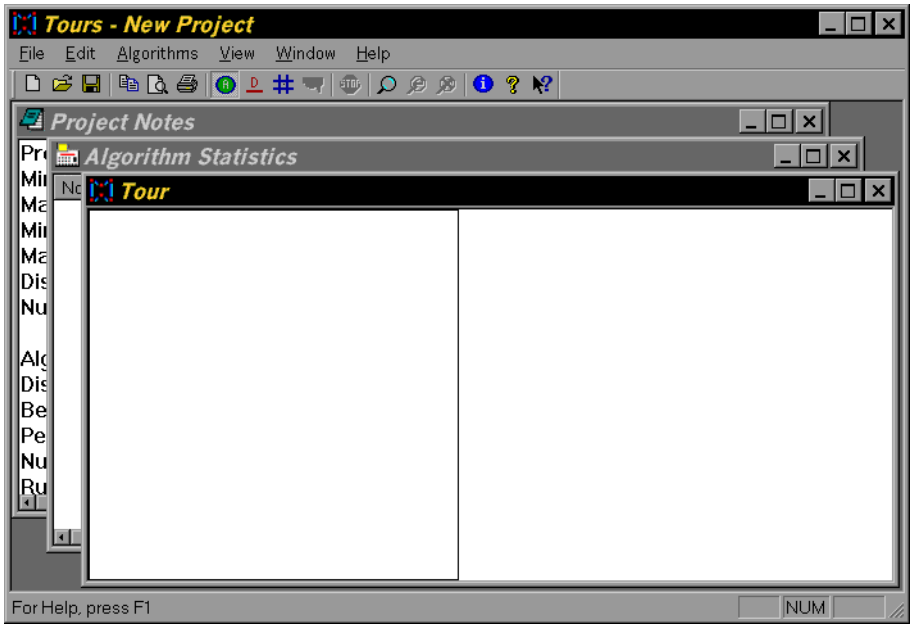



*Figure 2.3. Initial Views for a New Project*

## *Specifying additional project information*

The project name and other project information can be changed with the **Properties** command of the **File** menu. The properties of the tutorial project are illustrated in Figure 2.4. You can also provide here additional project information and comments and they will be saved with the project.

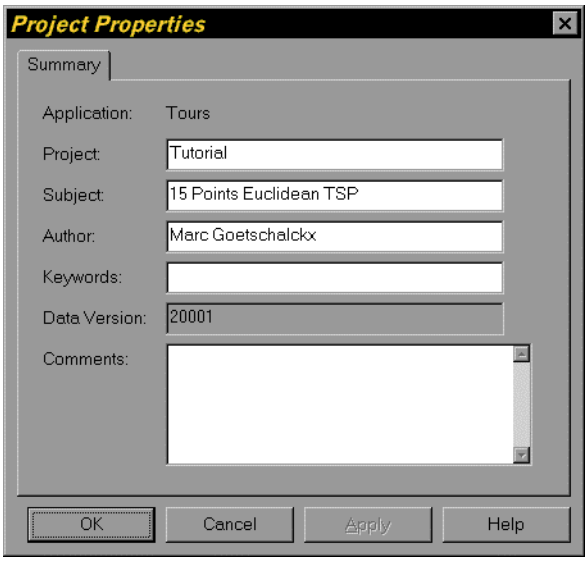



*Figure 2.4. Properties Dialog Window for the Tutorial Project*

## *Controlling the display of the background map*

Select the **Tour** view, which will display the tour but does not contain any points at this moment, by either clicking on its title bar or from the **Window** menu.

You can control the display the background map in this view by selecting the **Map** command from the **View** menu or by pressing the **Map** button on the toolbar. You can also control the display of the background map by pressing the shortcut keys **Ctrl+Shift+M**. The **Draw Map** dialog window for this tutorial project is illustrated in Figure 2.5. The possible options are None: the background map is not displayed at all, Outline: the boundary of the map objects is shown only, and Area Filled: the boundary of the map objects is shown and the objects are filled in. The default value is Area Filled. We recommend using the Outline or Area Filled modes, since the map provides visual feedback for the location of points. Usually, either the background map or the background grid displayed. In this tutorial we will be using the grid, so select None to turn the display of the background map off.

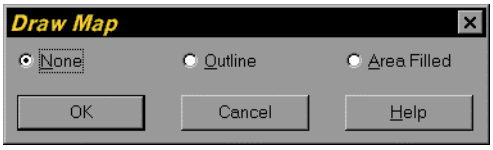



*Figure 2.5. View Map Dialog Window*

## *Controlling the display and size of the location grid*

Display the location grid in this view by selecting the **Grid** command from the **View** menu or by pressing the **Grid** button on the toolbar. You can also display the location grid by pressing the shortcut keys **Ctrl+Shift+G**.

Set the grid size for this view equal to 1000 units with the **Grid Size** command of the **View** menu. The Grid Size dialog window for this tutorial project is illustrated in Figure 2.6. Enter 1000 for the grid size. Press **OK** and only this view of tutorial project will use this new grid size. The perpendicular gridlines in this **Network** view are now 1000 units apart.

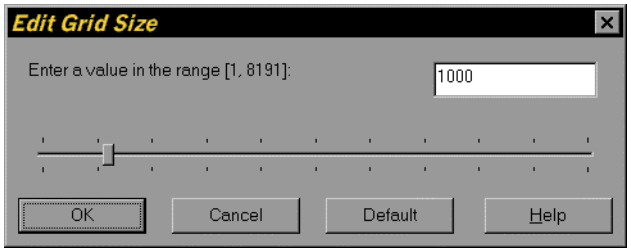



*Figure 2.6. Edit Grid Size Dialog Window*

If you have selected to display the grid in the **Tour** view, then this view is updated to reflect the new grid size.

## *Saving the project for the first time*

Save the current project with the **Save As** command from the **File** menu. In the **Save As** dialog box, select the directory where you want to save the project file. We strongly recommend that you create a separate directory for each project before creating the new project with **Tours**. Specify a name for the project data file, which by default has the *tours* extension. Some versions of the Windows operating environment will truncate the default *tours* extension to the three letters *tou*, so we recommend that you explicitly add the *tours* extension to the file name in the Save As dialog window. The **Save As** dialog box for this tutorial project is illustrated in Figure 2.7. Press **Save** and the file for the current project will be saved to disk.

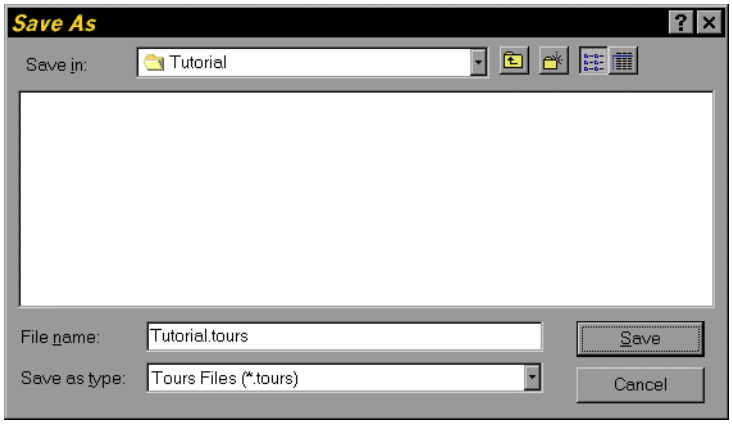



*Figure 2.7. Save As Dialog Box for the Tutorial Project*

The different views of the current project at this time are illustrated in Figure 2.8. Observe that the title bar of the application has changed from New Project to Tutorial to indicate that this project has been saved.

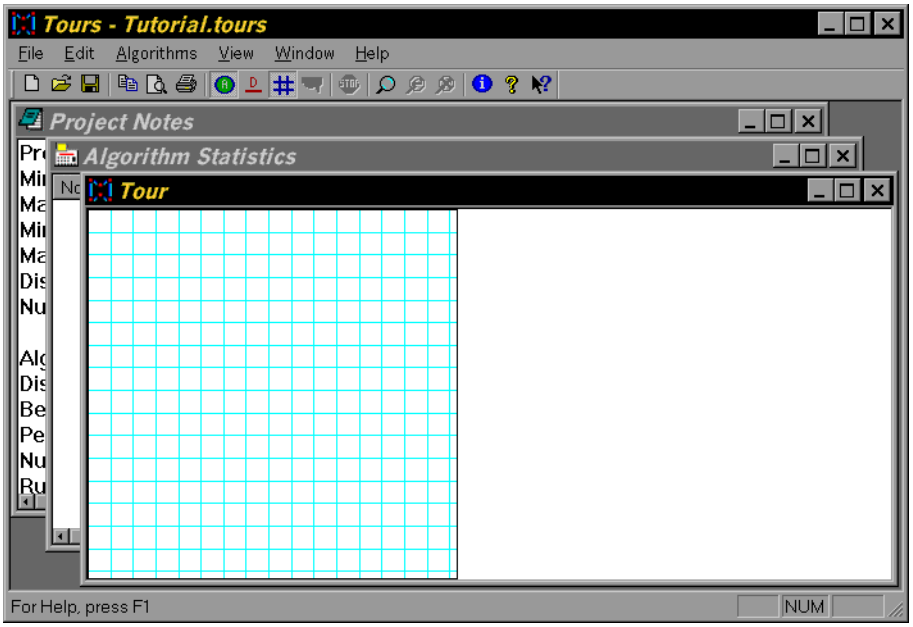



*Figure 2.8. Initial Views of the Tutorial Project*

## Sensitivity Analysis and Evaluating Tours

You can modify the distance data in the project with the **Distance** or **All Distances** commands of the **Edit** menu. To evaluate the impact of the changes you made use the **Evaluate** command of the **Algorithms** menu. The Evaluate command computes the tour length if a tour has been created. The results are displayed in the **Notes** and **Statistics** views. The Evaluate command is most frequently used after interactive editing of the distances or after you have dragged a point and have updated the distances based on its new location. The Evaluate command does not create a new tour, but rather computes the length of the current tour based on the current distances. The command also displays dialog window with the length of the current tour and the number of points included on the current tour. The **Evaluate** dialog window is illustrated in Figure 2.9.

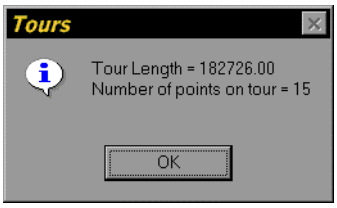



*Figure 2.9. Evaluate Dialog Window*

The **Algorithm Statistics** View displays the history of algorithm statistics. This view can be printed to the default printer with the **Print** command of the **File** menu. Move and size this window to suit your taste. The result of the tour construction and improvement algorithms that you have executed so far is illustrated in Figure 2.10. The corresponding **Tour** view is shown in Figure 2.11.

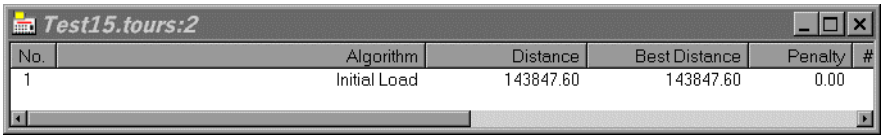


*Figure 2.10. Algorithm Statistics View for the Tutorial Project*

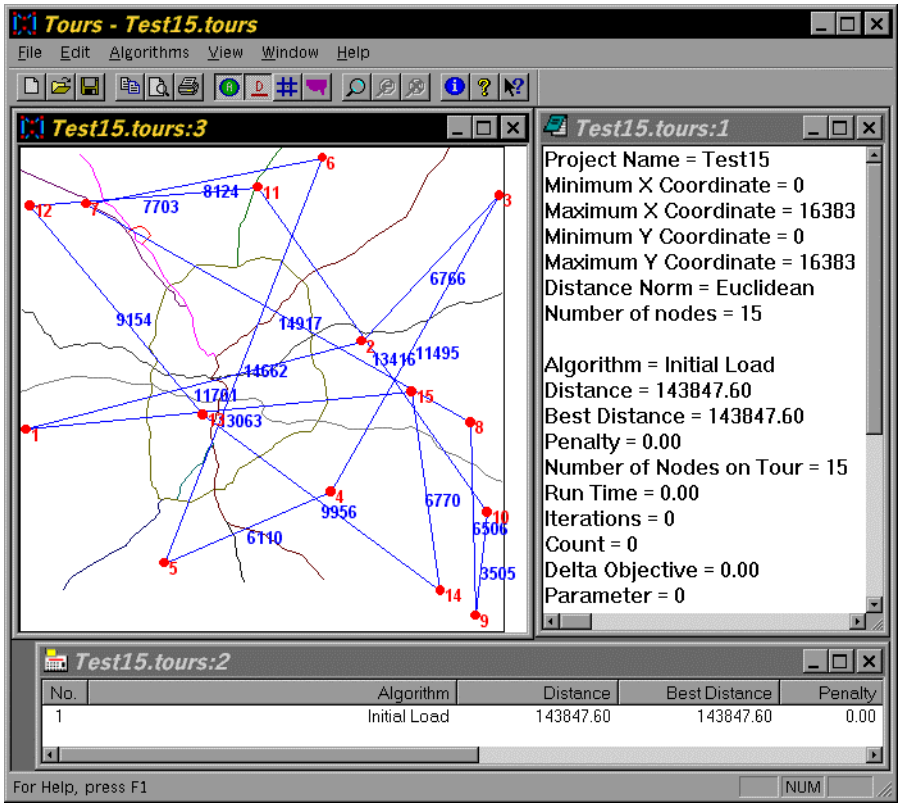



*Figure 2.11. Final Views for the Tutorial Project*

# Using Tour Views in Other Windows Programs

The results of the **Tours** design algorithms can be used in other Windows programs in basically three ways. Unless otherwise indicated, the actions described below can be executed on all three of the **Tours** views. First, we will print the **Tours** views to any installed printer, then we will copy and paste **Tours** views into other Windows applications. Finally, the current project can be send as an attachment to an electronic mail message, if you have an active mail client installed on your computer.

## Printing Tours Views

Select the printer on which you wish to print the **Tours** views with the **Print Setup** command of the **File** menu. This command presents a **Print Setup** dialog box, where you specify the printer and its connection. This printer and these options will then be used by all subsequent **Print** operations. The same changes can also be made from the main Windows **Control Panel**. The printer must have been previously installed from the Windows **Control Panel** or **Print Manager**. The **Print Setup** dialog box is illustrated in Figure 2.12.

The **Print Setup** dialog box is a common dialog box and its exact appearance depends on the version of your Windows operating environment.

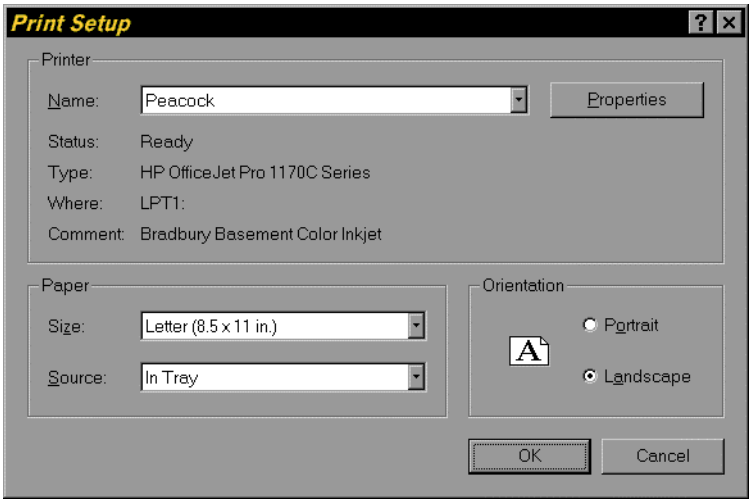



*Figure 2.12. Print Setup Dialog Box*

You can preview the image that will be printed by selecting the **Print Preview** command from the **File** menu. When you choose this command, the main window will be replaced with a **Print Preview** window in which one or two pages will be displayed in their printed format. The toolbar of the Print Preview window offers you options to view either one or two pages at a time; move back and forth through the document; zoom in and out of pages; and initiate a print job. The **Print Preview** dialog box is illustrated in Figure 2.13.

The **Print Preview** dialog box is a common dialog box and its exact appearance depends on the version of your Windows operating environment.

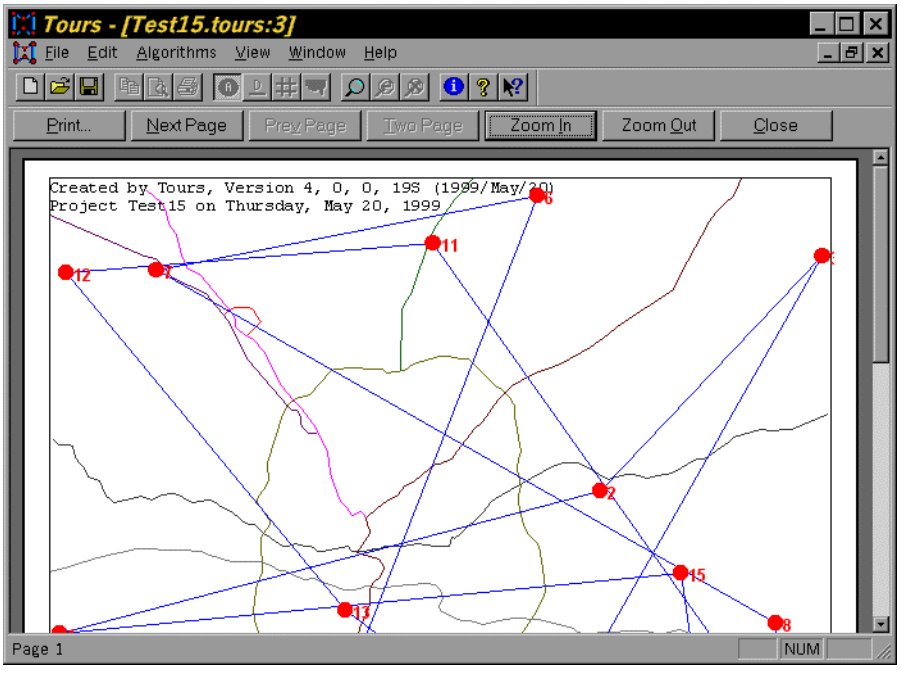



*Figure 2.13. Print Preview Window*

You can also directly select the view that you wish to print and then execute the **Print** command from the **File** menu. This command presents a **Print** dialog box, where you may specify the range of pages to be printed, the number of copies, the

destination printer, and other printer setup options. If you press **OK** the selected view will be printed. For **Tour** views of large projects, printing such a complex view might require substantial processing times. The **Print** dialog box is illustrated in Figure 2.14.

The **Print** dialog box is a common dialog box and its exact appearance depends on the version of your Windows operating environment.

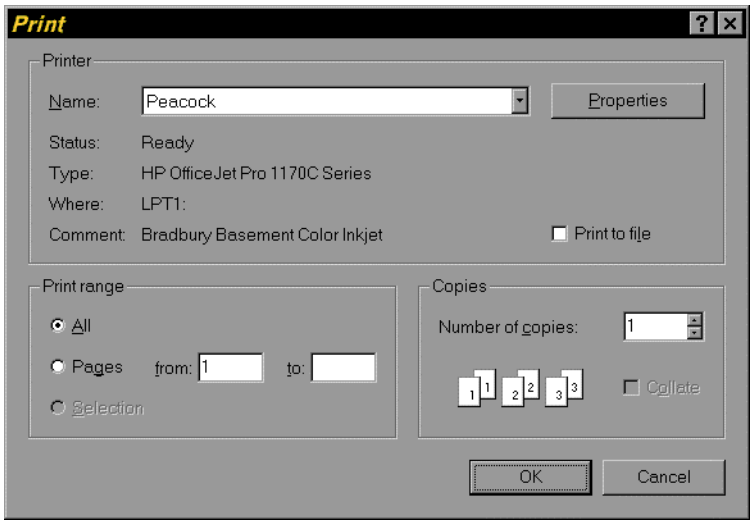



*Figure 2.14. Print Dialog Box*

## Copying and Pasting Tours Views with the Clipboard

To paste the contents of one of the **Tours** views in another application, select the view that you wish to paste then execute the **Copy** command from the **Edit** menu. For the **Tour** views, the section of the tour currently displayed in the view will be copied in graphical format to the clipboard. For the **Notes** and **Statistics** views, all the text, whether it is currently displayed or not, will be copied in text format to the clipboard. For the **Notes** and **Statistics** a header will generated which indicates the version of the **Tours** program, the name of the project and the date the view was copied to the clipboard.

Select the **Tour** view and execute the **Copy** command from the **Edit** menu. After the copy operation either activate the clipboard and verify its contents or paste the clipboard data into an application that accepts graphical data, such as Microsoft Word.

Select in turn the **Notes** and **Statistics** view and execute the **Copy** command from the **Edit** menu. After each copy operation either activate the clipboard and verify its contents or paste the clipboard data into an application that accepts text data, such as Microsoft Excel.

If you want to use a screen shot from one of the **Tours** views, maximize this view and then capture the view with the **Alt-PrintScreen** command. For all views, the section of the tour or text currently displayed in the view will be copied in graphical format to the clipboard. No header indicating the **Tours** version or the project data will be added to the image on the clipboard. You can use the same techniques if you want a screen shot from the main **Tours** window as it is currently displayed. The same technique can also be used to capture images of the various dialog boxed used

by **Tours** to the clipboard. In this case, obviously the dialog box cannot and does not have to be maximized.

Select in turn the **Notes**, **Statistics**, and **Tour** view and maximize this view. Execute the **Alt-PrintScreen** command. After each copy operation activate the clipboard and verify its contents or paste the clipboard data into an application that accepts graphical data, such as Microsoft Word.

## Sending Project as an E-mail Attachment

You can send the current **Tours** project as an attachment to an electronic mail message, if you have an active mail client installed on your computer. The currently saved project will be sent, so you should save the project before sending it.

Select the **Send** command from the **File** menu to activate your mail client and to send the saved version of the current project as an attachment.

This concludes the tutorial. Further information on the project data can be found in the Project Data chapter, further information on the design algorithms can be found in the Design Algorithms chapter. A complete list of all commands is given in the Command Reference chapter. You can also find more information in the references given in the References chapter. Finally, the data used in the tutorial are listed in the appendix Sample Projects.

# Chapter 3. Project Data

## Specifying and Editing Project Data

### Project Data

Every project has a number of data items associated with for which there is only one value per project. These data items are called scalar data items.

#### *Project Name*

Every project has a name, assigned to it when the project was created with the **New** command or when the project was read from an external ASCII file with the **Import** command. Both these commands are located on the **File** menu. The project name should be a maximum of 63 characters and should contain only letters, digits, spaces, and underscore characters. The term project title is used synonymously with project name.

The project title can be changed with the **Properties** command of the **File** menu after the project has been created.

If the project name contains spaces, it will be exported correctly with the **Export** command, but only the segment before the first space will be imported by the **Import** command. If you plan to export and import the project, you should only use underscore characters and not spaces to separate the different segments of the project name.

#### *Minimum X or West Longitude*

All world coordinates are bounded by a leftmost, rightmost, topmost, and bottommost value. The minimum x is the leftmost boundary value of valid coordinates if the orthogonal map projection is used. The minimum west longitude is the leftmost boundary value of valid coordinates when the Mercator or Albers projections are used. Longitudes can range from –180 to +180 degrees or from -1800000 to +1800000 in the integer latitude/longitude format. See the section on the **Common Data Items** for further explanation on the integer latitude/longitude format.

#### *Maximum X or East Longitude*

All world coordinates are bounded by a leftmost, rightmost, topmost, and bottommost value. The maximum x is the rightmost boundary value of valid

coordinates if the orthogonal map projection is used. The maximum east longitude is the rightmost boundary value of valid coordinates when the Mercator or Albers projections are used. Longitudes can range from –180 to +180 degrees or from -1800000 to +1800000 in the packed integer latitude/longitude format. See the section on the **Common Data Items** for further explanation on the packed integer latitude/longitude format.

### *Minimum Y or South Latitude*

All world coordinates are bounded by a leftmost, rightmost, topmost, and bottommost value. The minimum y is the bottommost boundary value of valid coordinates if the orthogonal map projection is used. The minimum south latitude is the bottommost boundary value of valid coordinates when the Mercator or Albers projections are used. Latitudes can range from –90 to +90 degrees or from –900000 to +900000 in the packed integer latitude/longitude format. See the section on the **Common Data Items** for further explanation on the packed integer latitude/longitude format.

### *Maximum Y or North Latitude*

All world coordinates are bounded by a leftmost, rightmost, topmost, and bottommost value. The maximum y is the topmost boundary value of valid coordinates if the orthogonal map projection is used. The maximum north latitude is the topmost boundary value of valid coordinates when the Mercator or Albers projections are used. Latitudes can range from –90 to +90 degrees or from –900000 to +900000 in the packed integer latitude/longitude format. See the section on the **Common Data Items** for further explanation on the packed integer latitude/longitude format.

### *Map Projection*

A map projection projects the three dimensional surface of the earth on the flat two-dimensional surface of the map and the screen. All map projections make some approximation errors during projection. Different map projections make different errors with respect to distance between two points and areas of continents.

At the current time, three map projections are supported: orthogonal, Mercator, and Albers. The orthogonal projection assumes perpendicular meridians and latitude lines and assumes that the distance between two meridians is constant everywhere. The orthogonal projection is equivalent to the standard two-dimensional coordinate system. The Mercator projection is best suited to map situated around the equator, since it distorts distance and area significantly at regions close to the poles. The Albers projection is particularly suited for the projection of the continental United States and areas at intermediate north latitudes.

### *Map Data File Name*

The optional background map for a new project is imported from a map file. The default extension for a map file is **.map*. Further information on the structure of the Map Data File is given in the section on **Map Data**.

Typically, all the files associated with a project are stored in a separate directory. It is most convenient to create this directory in advance and to copy the appropriate map file to this directory. You can then navigate to this map file from the **New** dialog window.

A new map for the current design project can be loaded with the **Load Map** command of the **Utilities** menu. The display of the background map is controlled by the **Map** command of the **View** menu.

If the map data file name contains spaces, it will be exported correctly with the **Export** command, but only the segment before the first space will be imported by the **Import** command. If you plan to import and export the project, you should only use underscore characters and not spaces to separate the different segments of the map data file name.

## *World Radius*

You specify the basic distance unit for the current project by giving the radius of the earth in these units. For example, the radius of the earth is approximately 6366.2 kilometers and 3955.8 miles. If you use a distance unit different from miles or kilometers, the earth radius must least be at least one thousand of these distance units. The world radius is not used by the orthogonal map projection.

## *Report Level*

The Report Level controls the level of detail written to the **Output Log** file and the number of pauses during algorithm execution. There are six levels ranging from zero to five, which generate increasingly more detailed output and frequent algorithm pauses.

### Levels of Detail

There are six levels of detail and pauses for reporting:

0. NONE generates no output per algorithm and does not halt the algorithm execution. This level is used when maximum execution speed and minimal reporting is desired.
1. DATABASE generates one line of strictly numerical output per algorithm. No titles or headers are included. This level is primarily used to create a data base file, which can then be manipulated in a spreadsheet or statistical analysis program.
2. SUMMARY displays the total cost plus the algorithm run time. It is useful if you are only interested in the final results. This level of output should be used if you are interested in performing timing studies. Higher level of details corrupt timing results due to user interaction delays and graphics creation delays. The screen is not updated until the algorithm has run to its completion.
3. STANDARD generates the total cost for each of the algorithm components. The program runs without interruption until the complete algorithm is finished. If you have selected ALL, then the program runs uninterrupted for the 18 different combinations. The screen is updated periodically to show the progress of the algorithm.
4. EXTENDED displays the total cost during each of the algorithm modules and the run time so far. The program halts frequently to allow you to observe the algorithm progress and the screen is updated before each pause.
5. DEVELOP generates extremely detailed output plus a very large number of intermediate results. This mode is only useful for debugging purposes or to observe the most detailed workings of the algorithms. The output is extremely long for large problems.

The Report Level can be modified at any time with the **Report Level** command of the **Edit** menu. It can also be changed by pressing the **Report Level** button of the

**Pause** dialog window, when an algorithm is paused. The algorithm will then use this new report level for the rest of its execution.

### *Seed*

The tour construction and improvement algorithms often need to make a random choice among several alternative sequences for the points. This random choice is made based on pseudo random numbers, generated from an initial seed. An algorithm will always make the same random choices if it is given the same random seed, and hence will create the same tour sequence. The seed has to be a positive number in the range of [1,32767]. If a seed of zero is given, then the computer will pick a random seed based on the computer clock.

### *Tolerance*

At the current time the tolerance parameter is not used in the program.

### *Time Limit*

The maximum time limit is the maximum amount of time a single algorithm is allowed to execute. The time limit is expressed in seconds. Currently, the time limit is used to terminate the two and three exchange algorithms if they have exceeded the time limit after one complete iteration, i.e. after all possible two or three exchanges have been tested. So it is possible that the execution time of the improvement algorithm is actually larger than the time limit specified.

### *Number of Replications*

The construction and improvement algorithms often need to make a random choice among several equivalent tour sequences. Different replications of the same algorithm can thus provide different tours. The higher the maximum number of replications, the more likely a good adjacency tour will be constructed. Of course, more replications require more computation time. The default number of the maximum number of replications is equal to 20.

## Map Data

In **Tours**, background maps are considered as a collection of map objects. Each map object can be thought of as a line on a paper map that can be drawn without lifting the pencil from the paper. **Tours** describes each map object as a series of map points denoted by latitude and longitude or by x and y coordinates, depending on the map projection of the project. Latitudes and longitudes are represented in one of two formats, a packed integer numeric format: (+/-)dddmmss or a fractional format: (+/-)ddd.ffff. For example, in the packed integer format the longitude 23°20'15" W is represented as -0232015, and the latitude 35°20'43" N as 0352043. In the fractional format, the longitude 23°30' W is represented as –23.5, and the latitude 35°06' is represented as 35.10. The packed integer format of the latitude and longitude and described in more detail in the section on **Common Data Items**.

The background map for a new project is imported from a map file. The default extension for a map file is **.map*. The name of this file is shown in the **Notes** view as the **Map Data File Name**. A new map for the current design project can be loaded with the **Load Map** command of the **Utilities** menu. The display of the background map is controlled by the **Map** command of the **View** menu.

### *Map File Format*

The file format is as follows:

```
Header -          number of objects          total number of points for all objects
          map format       either "integer" or "fractional"
Object 1 -        number of points           boundary color    area fill color
          Point 1.1 -       latitude                  longitude
          Point 1.2 -       latitude                  longitude
          Point 1.3 -       latitude                  longitude

Object n -        number of points           boundary color    area fill color
          Point n.1 -       latitude                  longitude
          Point n.2 -       latitude                  longitude
```

If no map format is specified, the default format is integer.

Valid color names are given in the section on **Common Data Items**. For example the file ***usa.map*** contains the 57 map objects with a total of 15702 points. The first object contains 203 points, has a boundary color of GREEN, and an area fill color of FOREST. An extract of the corresponding map data records is given next.

*Table 3.1. A Partial Example of Map Data Records in Packed Integer Format*

```
      57      15702
     203 GREEN    FOREST
  302500    -882400
  304400    -882500
  306000    -882500
...
     391 GREEN    NAVY
  423000    -903900
  423300    -904000
...
```

For example the file ***South America North.map*** contains one map object with a total of 493 points. The map data is given in the fractional format. The first, and only object, has a boundary color of BROWN, and an area fill color of GRAY.

*Table 3.2. A Partial Example of Map Data Records in Fractional Format*

```
       1         493 FRACTIONAL
     493 BROWN  GRAY
  -22.933     -43.167
   -22.75     -43.267
  -22.667     -43.083
  -22.983     -43.033
...
```

## Common Data Items

### *Longitude*

The location of any facility is given by its latitude and longitude. The longitude is given in the "(+/-)dddmmss" packed integer format, that is, the last two digits indicate the seconds, the third and fourth last digit indicate the minutes, and the remaining digits indicate the degrees. Longitudes east and west of the Greenwich

meridian are positive and negative, respectively. Valid longitudes are in the range from -180 to +180 degrees, which correspond to field values from -1800000 to +1800000. For example, the longitude 23°20'15" W is represented as -0232015, and the latitude 35°20'43" N as 0352043.

### *Latitude*

The location of any facility is given by its latitude and longitude. The latitude is given in the "(+/-)dddmmss" packed integer format, that is, the last two digits indicate the seconds, the third and fourth last digit indicate the minutes, and the remaining digits indicate the degrees. Latitudes above and below the equator are positive and negative, respectively. Valid latitudes are in the range from -90 to +90 degrees, which correspond to field values from -900000 to +900000. For example, the longitude 23°20'15" W is represented as -0232015, and the latitude 35°20'43" N as 0352043.

### *Color*

All objects in **Tours** can be displayed in one of the following standard colors.

*Table 3.3. Valid Color Names*

| | |
|---|---|
| BLACK | DARKGRAY |
| BLUE | NAVY |
| GREEN | FOREST |
| CYAN | OCEAN |
| RED | BROWN |
| MAGENTA | PURPLE |
| YELLOW | OLIVE |
| WHITE | GRAY |

### *Display Symbol*

All facilities in **Tours** can be displayed with one of the following standard geometrical shapes.

*Table 3.4. Valid Display Symbol Names*

| |
|---|
| CIRCLE |
| SQUARE |
| TRIANGLE |
| DIAMOND |
| HEXAGON |

# Importing Data Files from Previous Versions

The current version of the **Tours** program can import data files created by the previous versions of the **Tours** program with the **Import** command of the **File** menu. The description of the data files for this previous version of **Tours** is given next.

A project is completely described by two files. The first file is the **Project Data** file that holds all the scalar information about this project. The second file is the **Points Data** file that contains all point data.

## Project Data File

The Project Data file can be created with any editor or word processor capable of generating pure ASCII files. Word processors usually insert special formatting codes into their regular document files, such as page breaks, which cannot be read by the **Tours** program, so special care should be taken when using a word processor to generate a pure ASCII file. The project file also should not contain any blank lines. Once an input file has been created, it can be used repeatedly by the **Tours** program.

Each line in the input file is associated with a single data item and contains two fields. The first field is the description or name of the data item. The name of the item is enclosed in square brackets and may not contain any spaces. The second field is then the value of the item to be used. The two fields are separated by one or more space or tab characters. For example:

```
[problem_name]          TEST15
[norm_name]             RECTILINEAR
[number_of_points]      15
[minimum_xcoord]        0
[maximum_xcoord]        16383
[minimum_ycoord]        0
[maximum_ycoord]        16383
[points_file_name]      TEST15.PTS
[map_file_name]         ATLANTA.MAP
[tolerance]             0.01000
[seed]                  12345
[time_limit]            120
[maximum_iterations]    500
[report_level]          3
```

### *Problem Name*

The first item is the name of the current project. This name should be a maximum of 15 characters and should contain only letters, digits, and underscore characters. In the version 4.0 of **Tours** or higher the size of this data field has been expanded and spaces are allowed.

However the old size of **Problem Name** still should be used in the import files and the name should not contain any spaces.

### *Points File Name*

The next item is the name of the points data file. This file contains for each point the x and y coordinates and the index of the successor point, respectively. For versions of **Tours** before 4.0 the file name must satisfy the historical 8.3 DOS file name conventions. A path must precede the file name if it is not in the current directory or in the data path. This path cannot contain spaces. In the version 4.0 of **Tours** or newer long file and path names may be used but they still may not contain any spaces. An example of a points data file is the file ***test15.pts*** that holds the point information. This file is included in the appendix and on the distribution disk. This file must be created outside the **Tours** program with an editor capable of generating pure ASCII files. The extension ***.pts*** is solely a convention, other extensions can be used.

### *Seed*

The tour construction and improvement algorithms often need to make a random choice among several alternative tour sequenes for the points. This random choice is

made based on pseudo random numbers, generated from an initial seed. An algorithm will always make the same random choices if it is given the same random seed, and hence will create the same tour sequence. The seed has to be a positive number in the range of [1,32767]. If a seed of zero is given, then the computer will pick a random seed based on the computer clock.

### *Tolerance*

At the current time the tolerance parameter is not used in the program.

### *Time Limit*

The maximum time limit is the maximum amount of time a single algorithm is allowed to execute. The time limit is expressed in seconds. Currently, the time limit is used to terminate the two and three exchange algorithms if they have exceeded the time limit after one complete iteration, i.e. after all possible two or three exchanges have been tested. So it is possible that the execution time of the improvement algorithm is actually larger than the time limit specified.

### *Maximum Iterations*

The construction and improvement algorithms often need to make a random choice among several equivalent tour sequences. Different replications of the same algorithm can thus provide different tours. The higher the maximum number of replications, the more likely a good adjacency tour will be constructed. Of course, more replications require more computation time. The default value for the maximum number of replications is equal to 20. In version 4.0 or higher of **Tours**, the name of this field has been changed to **Number of Replications**.

### *Report Level*

Report level is the level of detail the program will use in generating output reports. There are six levels of detail, ranging from 0 through 5. The higher the report level the more information is written to the Output Log File and the more frequent halts during program execution.

## Points Data File

The Points Data file can be created with any editor or word processor capable of generating pure ASCII files. Word processors usually insert special formatting codes into their regular document files, such as page breaks, which cannot be read by the **Tours** program, so special care should be taken when using a word processor to generate a pure ASCII file. The points file also should not contain any blank lines. Once an input file has been created, it can be used repeatedly by the **Tours** program.

Each line in the input file is associated with a single point and contains three fields. The first line corresponds to the point with index or label one, the second line to the point with index two, and so on. The first and second field are the x and y coordinates of the corresponding point. The third field is the index of the point that follows the current point in the tour. If no tour has been created then all the successor indices for the points should be set to zero. No partial tours are allowed. In other words, either all points should have a successor point or no points should have a successor point. The three fields are separated by one or more space or tab characters. For example:

```
220     9526    2
11537   6552    3
…
```

# Chapter 4. Design Algorithms

## Introduction

The Traveling Salesman Problem is one of the most studied problems in several areas of mathematics such as graph theory, mathematical programming, and combinatorial optimization. Even though the basic problem is easy to define and explain, efficient optimal design algorithms do not exist to this date and may never be found. The problem has been shown to belong to a class of computationally hard problems for which it is difficult to find the exact solution for problems of even modest size. This difficulty has given rise to the development of a large number and variety of heuristic algorithms. An overview of the TSP problem, its history, fundamental properties and of a large variety of its design algorithms is given in Lawler et al. (1985).

### Algorithm Taxonomy

#### *Exact versus Heuristic*

Algorithms are called optimal or exact if they find the optimal solution, algorithms are called heuristics if they attempt to find a high quality but not necessarily optimal solution.

#### *Construction versus Improvement*

Algorithms are called constructive (construction algorithms) if they create a solution from the original data without requiring an initial feasible solution. Algoritms are called improving (improvement algorithms) if they require an initial feasible solution and attempt to improve the quality of this solution.

#### *Primal versus Dual*

Algorithms are called primal if they maintain the feasibility of their solution while they attempt to reach optimality. Algorithms are called dual if they maintain optimality of their solution, while they attempt to reach feasibility. Typically dual algorithms ignore one or more constraint sets, solve the resulting problem to optimality, and then attempt to add the ignored constraints while maintaining optimality of the current solution with respect to the current constraints.

# Construction Heuristics

## Nearest Neighbor

The Nearest Neighbor algorithm starts the tour with an initial point and then appends the nearest unvisited or free point to the tour. This algorithm was originally described by Rosenkrantz et al. (1977).

Different initial starting points may give different tour sequences. Since the Nearest Neighbor algorithm executes very fast, a possible alternative would be to start a tour at each point and then to select the shortest tour among them.

## Random

The random construction algorithm creates a tour by selecting randomly the next unvisited point. Since this algorithm does not make any attempt to minimize the length of the tour it creates, the resulting tour length is typically much larger than tour lengths created by the other algorithms.

The implementation of the random algorithm uses a pseudo random number generator to randomly select the next point on the tour. Pseudo random number generators create identical sequences of random numbers if they are started from the same initial condition. This starting condition is called the random number seed. Hence, the random algorithm will create identical traveling salesman tours when started with the same seed. If you want to generate different random tours for the same data, then you must change the seed before every run of the random algorithm. Difference processors, operating systems, or compiler versions may generate different random number sequences for the same random number seed.

## Sweep

The Sweep algorithm creates a tour by appending the points to the tour when they are traversed by a ray rotating around the center of de points. The center coordinates are equal to the average the x and y coordinates. The relative polar coordinates of each point with respect to the center point are then determined and the points are inserted in the tour by increasing polar angle. The algorithm was first described by Gillet and Miller (1974).

The initial starting angle of the rotating ray is an algorithm parameter that you can specify. However the total tour length is independent of the starting angle, since all the points will always be visited in the same sequence irrespective of the starting angle.

## Savings

Clarke and Wright (1964) developed a construction procedure that extends a partial route or route primitive on its two end points. Conceptually the algorithm defines a base point and constructs an Eulerian tour that visits each of the other points and the returns to the base point. The Eulerian tour is then reduced in length by finding and executing the shortcut with the largest savings. The savings are computed as the sum of the distances to the base point of the two points minus the distance between the two points.

$$\max_{i,j}\left\{s_{ij} = c_{i0} + c_{0j} - c_{ij}\right\} \tag{4.1}$$

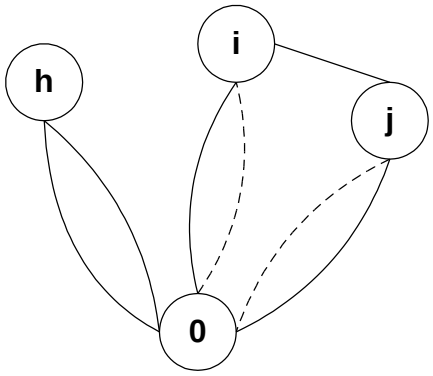



*Figure 4.1. Clarke and Wright Tour Extension Illustration*

Once two points have been joined by a shortcut they are never separated again by the Clarke and Wright algorithm. This equivalent to extending the partial route at its end points, which are connected to the base point. The next point is then selected by finding the point with the largest savings shortcut to the current end points of the partial tour.

$$\max_{i,j}\left\{\max_{h}\{s_{ih} = c_{i0} + c_{h0} - c_{ih}\}\right\} \tag{4.2}$$

***Algorithm 4.1 Clarke and Wright Savings Algorithm (TSP Version)***

1. Select base point {0}
2. Construct a tour primitive by finding the two points with the largest savings shortcut
3. While not all points have added to the partial tour
4. Update computation of savings of combining tours
5. Append point with largest savings shortcut to endpoints of the partial tour

### Band

### Space Filling Curve

# Partial Tour Construction Heuristics

Partial tour construction heuristics are construction algorithms that create a tour through a subset of the points.

### Quad

### Convex Hull

# Insertion Heuristics

Insertion algorithms insert the remaining unvisited or free points into a partial tour. They have to make two decisions: which point to insert next and on which link to insert this point. Depending on the answer to those two questions, different variations of insertion algorithms have been developed.

### Nearest Insertion

### Cheapest Insertion

### Farthest Insertion

### Nearest Addition

### Minimum Ratio Insertion

### Optimal Insertion

# Improvement Heuristics

## Introduction

### *Improvement Heuristic Classification*

Exchange improvement heuristics can be divided into four classes depending on which exchange they test for possible improvement and which exchange they select to execute. For a minimization problem, such as the TSP, where we want to a tour with the lowest possible length, the categories are

1. First Descent
2. Steepest Descent
3. Simulated Annealing
4. Tabu Search

## First Descent

All possible edge exchanges that can result in a new tour are examined in a structured way until an exchange is found that reduces the tour length. This exchange is executed immediately and the process of examining all possible exchanges starts all over. Hence, the first exchange in each iteration that yields a reduction is executed. The process terminates when no further exchanges can be found that yield a cost reduction.

## Steepest Descent

All possible edge exchanges that can result in a new tour are examined in a structured way and the exchange that yielded the largest reduction in the tour length is retained. If this exchange reduces the tour length then it is executed and the process of examining all possible exchanges starts all over. Hence, the exchange that yields the strongest reduction in each iteration is executed. The process terminates when no further exchanges can be found that yield a cost reduction.

## Simulated Annealing

Both previous improvement algorithms are deterministic, i.e. each algorithm will convert an initial tour into specific final tour. Since they are heuristics, this final tour may not be of high quality. To remedy this problem, a probabilistics exchange improvement algorithm was developed. There exists an analogy between the optimization method of simulated annealing and the laws of thermodynamics, specifically with the way in which liquids freeze and crystallize or metals cool and anneal.

The simulated annealing algorithm selects a set of edges for exchange evaluation at random. If the exchange yields a cost reduction, then the exchange is executed immediately. If the exchange yields a cost increase, then the exchange is executed with probability *P*, which is computed in function of the cost increase Δ and the temperature *T*. *T* is a search control parameter that is systematically reduced during the algorithm execution.

$$\begin{aligned} &if \quad \Delta < 0 \quad P[Exch] = 1 \\ &if \quad \Delta \geq 0 \quad P[Exch] = e^{-\Delta/T} \end{aligned} \tag{4.3}$$

This allows early on exchanges with large cost increases. As the temperature is reduced, the number of such exchanges and the size of the allowed cost increases are gradually reduced. The objective of these non-improving exchanges is to avoid a first descent into a local minimum. The process repeats itself until no further improvements can be made. Since the exchanges were selected at random, the improvement algorithm may generate a different final tour if run from the same initial tour if different seeds are used to generate different pseudo-random number streams for sampling the probability function of *P*.

For further information on two and three exchanges see Goetschalckx (1992). For further information on simulated annealing see Kirkpatrick et al. (1983) and Vechi and Kirkpatrick (1983).

Computational processing time increases sharply with the amount of improvement processing.

## 2-Opt Exchange

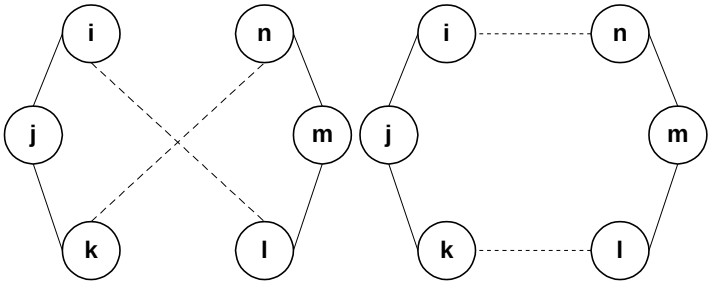
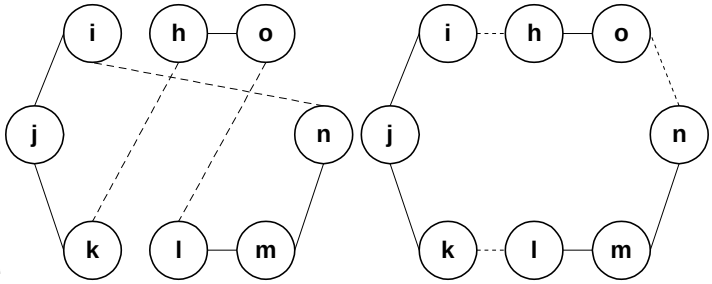



*Figure 4.2. Two Exchange Improvement Illustration*

## 3-Opt




## Exchange

*Figure 4.3. Three Exchange Improvement Illustration*

## Or-Exchange

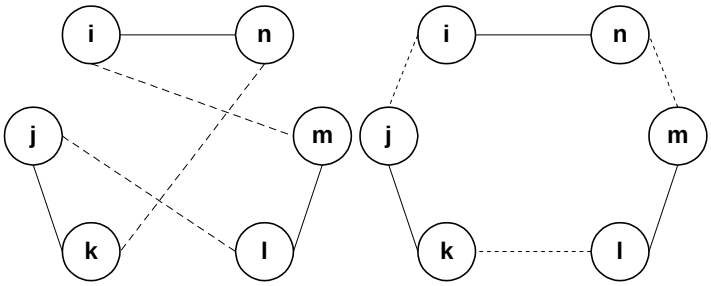



*Figure 4. Or (Chain) Exchange Improvement Illustration*

## Simulated Annealing

# Lower Bound Algorithms

## Quad

## Convex Hull

## 1-Tree Relaxation

# Optimal Construction Algorithms

## Assignment

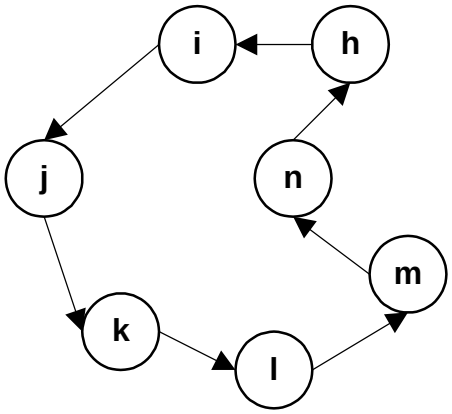



*Figure 4.5. Asymmetric Traveling Salesman Problem Illustration*

***Formulation 4.1. Asymmetric Traveling Salesman Problem***

$$
\begin{aligned}
Min \quad & \sum_{i=1}^{N}\sum_{j=1}^{M} c_{ij}x_{ij} & & \\
s.t. \quad & \sum_{i=1}^{N} x_{ij} = 1 & & \forall j \\
& \sum_{j=1}^{N} x_{ij} = 1 & & \forall i \\
& \sum_{i\in S}\sum_{j\in S} x_{ij} \le |S|-1 & & \forall S \subset N \\
& x_{ij} \in \{0,1\} & &
\end{aligned}
\tag{4.4}
$$

The Asymmetric Traveling Salesman (ATSP) is basically an Assignment Formulation (AP) with additional constraints that eliminate subtours.

## Subtour Elimination Constraints

$$\sum_{i\in S}\sum_{j\in S} x_{ij} \le |S|-1 \quad \forall S \subset N \tag{4.5}$$

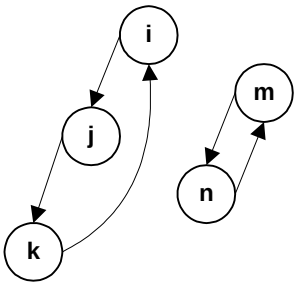



*Figure 4.6. Subtour Elimination Illustration*

## Transformed Assignment

# Chapter 5. Command Reference

## Menu Overview

An overview of the **Tours** program menu structure is shown in Figure 5.1.

| File | Edit | Algorithms | View | Window | Help |
|---|---|---|---|---|---|
| New | Add Facilities | Nearest Neighbor | Facility Labels | New Notes | Help Topics |
| Open | Edit Facilities | Sweep | Facility Quantities | New Statistics | About Lineback |
| Close | Edit Links | Savings | Link Distances | New Routes | |
| Save | Delete Facilities | Band | Colors by Route | Cascade | |
| Save As | Delete All Routes | Two Exchange | Grid | Tile | |
| Import | Parameters | Or Exchange | Map | Arrange Icons | |
| Export | Copy | Three Exchange | Label Size | Toolbar | |
| Send | | Move | Grid Size | Status Bar | |
| Output Log | | Swap | Zoom | | |
| Close Log | | Select | Zoom Previous | | |
| Properties | | Evaluate | Zoom Original | | |
| Print | | | Redraw | | |
| Print Preview | | | | | |
| Print Setup | | | | | |
| Exit | | | | | |

*Figure 5.1.* ***Tours*** *Menu Structure*

Several commands have shortcut keys so that you can easily control the program with the keyboard. Several commands are also shown on the toolbar, to allow easy program control with the mouse.


*Figure 5.2.* ***Tours*** *Toolbar*

## File Menu

The **Tours** program requires the project data before any algorithm can be executed. The projects are managed with the commands on the **File** menu.

### New

The **New** command of the **File** menu allows you to interactively create a new facilities design project.

If there is a project currently open and if it has been modified, then the program will display the **Save Changes** dialog window and ask if you wish to save those

changes, discard the changes, or if you wish to abort the creation of new project. The **Save Changes** dialog window is illustrated in Figure 5.3.

The command displays the **New** dialog window, which is illustrated in Figure 5.4. To save the new project use the **Save As** command. To open a previously saved project use the **Open** or **Import** commands.

To create the new project with the values shown in the dialog window press the **OK** button. When this project is created it has no points. If you press **Cancel**, no new project will be created.

## New Project Shortcuts

Toolbar: 

Keys: CTRL+N

## Save Changes Dialog Window

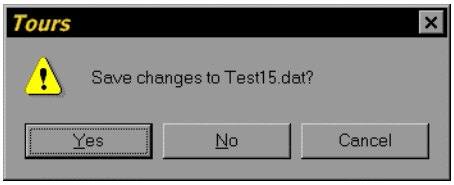



*Figure 5.3. Save Changes Dialog Window*

## New Project Dialog Window

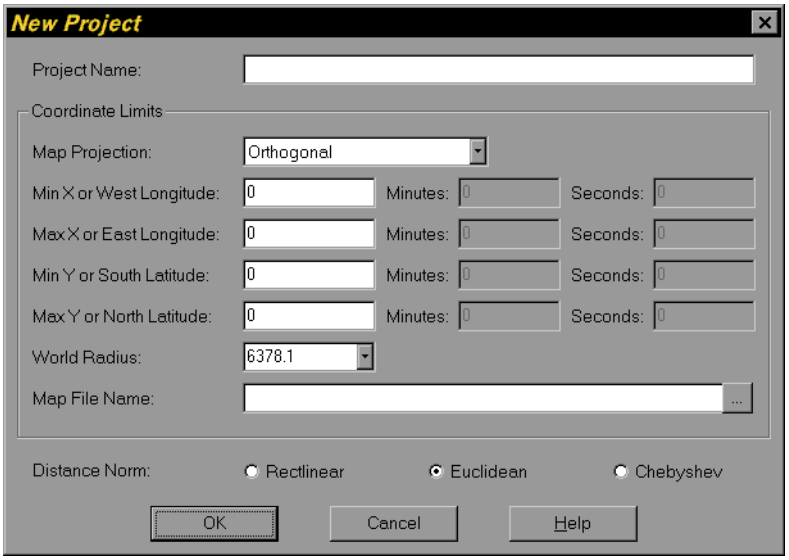



*Figure 5.4. New Project Dialog Window*

## New Project Data Items

### Project Name

The project name refers to the title of the project to be used in reports and printouts. It consists of a maximum of 63 alphanumeric, spaces or underscore characters.

Punctuation marks or tab characters are not allowed. The title is also included in the project **Properties**. The term project title is used synonymously with project name.

Further information can be found in the Project Data section under **Project Name**. The project name can be changed after the project creation with the **Properties** command of the **File** menu.

If the project name contains spaces, it will be exported correctly with the **Export** command, but only the segment before the first space will be imported by the **Import** command. If you plan to export and import the project, you should only use underscore characters and not spaces to separate the different segments of the project name.

### Minimum X or West Longitude

All world coordinates are bounded by a leftmost, rightmost, topmost, and bottommost value. The minimum x is the leftmost boundary value of valid coordinates if the orthogonal map projection is used. The minimum west longitude is the leftmost boundary value of valid coordinates when the Mercator or Albers projections are used. Longitudes can range from –180 to +180 degrees or from -1800000 to +1800000 in the integer latitude/longitude format. See the section on the **Common Data Items** for further explanation on the integer latitude/longitude format.

### Maximum X or East Longitude

All world coordinates are bounded by a leftmost, rightmost, topmost, and bottommost value. The maximum x is the rightmost boundary value of valid coordinates if the orthogonal map projection is used. The maximum east longitude is the rightmost boundary value of valid coordinates when the Mercator or Albers projections are used. Longitudes can range from –180 to +180 degrees or from -1800000 to +1800000 in the packed integer latitude/longitude format. See the section on the **Common Data Items** for further explanation on the packed integer latitude/longitude format.

### Minimum Y or South Latitude

All world coordinates are bounded by a leftmost, rightmost, topmost, and bottommost value. The minimum y is the bottommost boundary value of valid coordinates if the orthogonal map projection is used. The minimum south latitude is the bottommost boundary value of valid coordinates when the Mercator or Albers projections are used. Latitudes can range from –90 to +90 degrees or from –900000 to +900000 in the packed integer latitude/longitude format. See the section on the **Common Data Items** for further explanation on the packed integer latitude/longitude format.

### Maximum Y or North Latitude

All world coordinates are bounded by a leftmost, rightmost, topmost, and bottommost value. The maximum y is the topmost boundary value of valid coordinates if the orthogonal map projection is used. The maximum north latitude is the topmost boundary value of valid coordinates when the Mercator or Albers projections are used. Latitudes can range from –90 to +90 degrees or from –900000 to +900000 in the packed integer latitude/longitude format. See the section on the **Common Data Items** for further explanation on the packed integer latitude/longitude format.

### Map Projection

A map projection projects the three dimensional surface of the earth on the flat two-dimensional surface of the map and the screen. All map projections make some approximation errors during projection. Different map projections make different errors with respect to distance between two points and areas of continents.

At the current time, three map projections are supported: orthogonal, Mercator, and Albers. The orthogonal projection assumes perpendicular meridians and latitude lines and assumes that the distance between two meridians is constant everywhere. The orthogonal projection is equivalent to the standard two-dimensional coordinate system. The Mercator projection is best suited to map situated around the equator, since it distorts distance and area significantly at regions close to the poles. The Albers projection is particularly suited for the projection of the continental United States and areas at intermediate north latitudes.

### Map Data File Name

The optional background map for a new project is imported from a map file. The default extension for a map file is **.map*. Further information on the structure of the Map Data File is given in the section on **Map Data**.

Typically, all the files associated with a project are stored in a separate directory. It is most convenient to create this directory in advance and to copy the appropriate map file to this directory. You can then navigate to this map file from the **New** dialog window.

A new map for the current design project can be loaded with the **Load Map** command of the **Utilities** menu. The display of the background map is controlled by the **Map** command of the **View** menu.

If the map data file name contains spaces, it will be exported correctly with the **Export** command, but only the segment before the first space will be imported by the **Import** command. If you plan to import and export the project, you should only use underscore characters and not spaces to separate the different segments of the map data file name.

### World Radius

You specify the basic distance unit for the current project by giving the radius of the earth in these units. For example, the radius of the earth is approximately 6366.2 kilometers and 3955.8 miles. If you use a distance unit different from miles or kilometers, the earth radius must least be at least one thousand of these distance units. The world radius is not used by the orthogonal map projection.

## Open

The **Open** command of the **File** menu allows you to read a previously saved project.

The program will read the corresponding **Project Data File**. Starting with version 4.0 of the **Tours** program, the **Project Data File** is a binary file that can no longer be viewed or manipulated outside the **Tours** program. Use the **Import** command to import project data created with previous versions of the **Tours** program or with the **Export** command. The default extension for **Tours** project files is *.tours*. The command then displays the **Open** dialog window, which is illustrated below in Figure 5.5.

### *Open Project Shortcuts*

Toolbar: 

Keys: CTRL+O

### *Open Project Dialog Window*

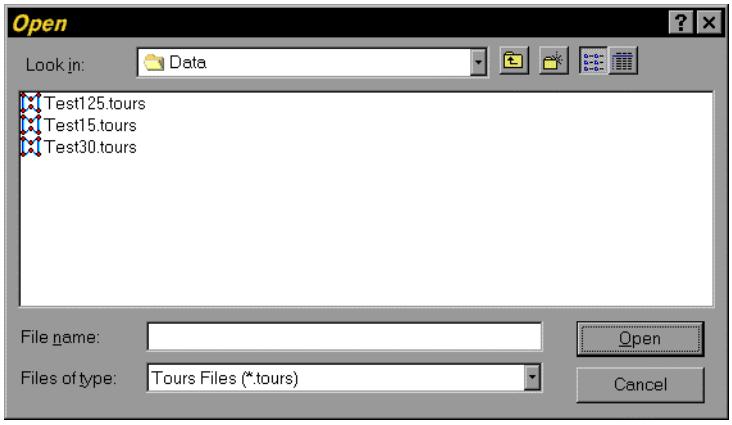



*Figure 5.5. Open Project Dialog Window*

## Close

The **Close** command of the **File** menu allows you to close the current project.

If the current project has been modified, then the program will display the **Save Changes** dialog window and ask if you wish to save those changes, discard the changes, or if you wish to abort the closing of the current project. The **Save Changes** dialog window is illustrated in Figure 5.3.

## Save

The **Save** command of the **File** menu will save the current project data in the **Project Data File**. If no file name for the current **Project Data File** has been defined, then the **Save** command will execute as the **Save As** command.

### *Save Project Shortcuts*

Toolbar: 

Keys: CTRL+S

## Save As...

The **Save As** command of the **File** menu will query you for the file name of the **Project Data File** with the **Save As** dialog window. If you press the **Save** button the current project will then be saved as if the **Save** command was executed.

### *Save As Dialog Window*

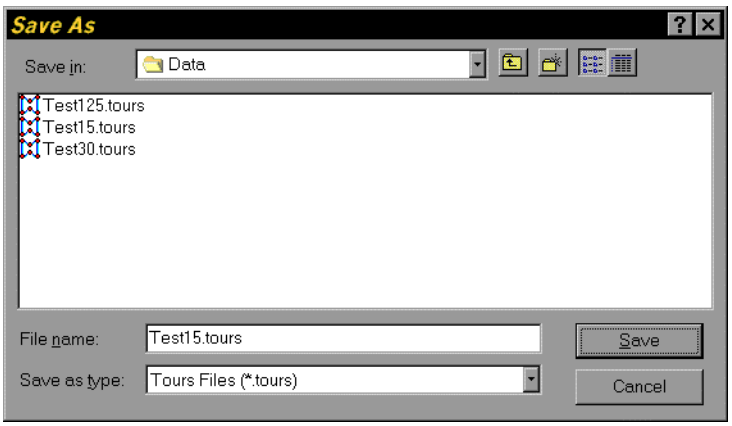



*Figure 5.6. Project Save As Dialog Window*

### *Save As Data Fields*

#### File Name

The file name identifies the **Project Data File**. **Tours** is compatible with long file and directory names that follow the Windows conventions. Some versions of the Windows operating environment will truncate the default *tours* extension to the three letters *tou*, so we recommend that you explicitly add the *tours* extension to the file name in the Save As dialog window.

#### File Type

Starting with version 4.0 of the **Tours** program, the **Project Data File** is a binary file that can no longer be viewed or manipulated outside the **Tours** program. Use the **Export** command to create a set of ASCII files that contain the major data for the current project. These files can then be manipulated outside the **Tours** program with an ASCII text editor and then imported again into the **Tours** program using the **Import** command.

## Import...

The **Import** command of the **File** menu allows you to read project data saved in ASCII files with a version of the **Tours** program earlier than 4.0, created with the **Export** command, or created manually outside the **Tours** program with an ASCII editor.

Starting with version 4.0 of the **Tours** program, the **Project Data File** is a binary file that can no longer be viewed or manipulated outside the **Tours** program. Use the **Open** command to read project data files created with the **Save** command of version 4.0 or higher of **Tours**. The default extension for **Tours** project files is *.tours*, the default extension for the **Tours** project files of previous versions or exported data files is *.dat*.

The command then displays the **Open** dialog window, which is illustrated in Figure 5.7. The program will read the corresponding **Project Data File** (**.dat*) and **Points Data File** (**.pts*). It will also read the corresponding **Map Data File** *(*.map*) if a map file has been specified.

The **Project Data File** and the **Points** and **Map Data Files** can be created with any text editor or word processor capable of generating pure ASCII files or with the **Export** command. Since word processors usually insert special formatting codes into their regular document files, which cannot be read by the **Tours** program, special care should be taken when using a word processor to generate a pure ASCII file. The project data file also should not contain any blank lines. Once these input files have been created, they can be used repeatedly by the **Tours** program.

### *Import Project Dialog Window*

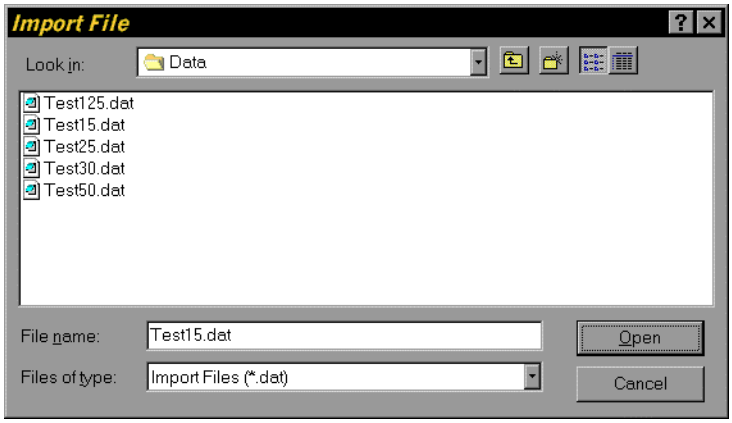



*Figure 5.7. Import Project Dialog Window*

## Export...

The **Export** command of the **File** menu will save the current project data in a set of ASCII files. The program will create the corresponding **Project Data File** (**.dat*) and **Points Data File** (**.pts*). Once these data files have been created, they can be read by the **Import** command and by an ASCII editor. Not all program settings will be saved, but only the major project and point data.

Starting with version 4.0 of the **Tours** program, the **Project Data File** created by the **Save** command is a binary file which can no longer be viewed or manipulated outside the **Tours** program. Use the **Open** command to read project data files created with the **Save** command of version 4.0 or higher of **Tours**. The default extension for **Tours** project files is *.tours*, the default extension for the **Tours** project files of previous versions or exported data files is *.dat*.

### *Export Dialog Window*

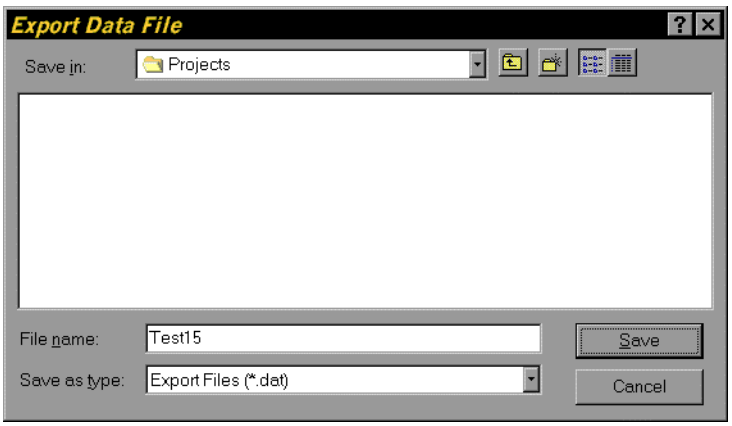



*Figure 5.8. Export Dialog Window*

## Send...

The **Send** command of the **File** menu uses the electronic mail application installed on your computer to send the saved version of the current project as an attachment to an electronic mail message. It is recommended that the current project first be saved before using the **Send** command. The exact execution of this command will depend on which electronic mail application has been installed on your computer. This command will not be enabled if you do not have an electronic mail client installed on your computer.

## Properties...

The **Properties** command of the **File** menu allows you to add project information, such as authors, subjects, and comments to the current project. The **Properties** command displays the **Properties** dialog window for the current document. You can provide additional information in this dialog window about the current project.

### *Properties Dialog Window*

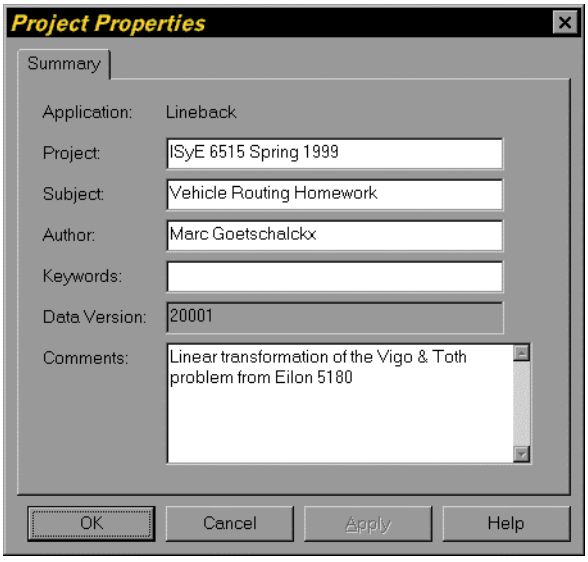


*Figure 5.9. Properties Dialog Window*

### *Properties Data Items*

#### Application

The name of the application that is creating this dialog window. In this case, **Tours**. You cannot change this data field.

#### Project

The project name. This is the only place where the name of the current project can be changed after it has been initially entered in the **New Project** dialog window or was imported from the **Project Data** file.

Further information can be found in the **Project Data** section under Project Name.

#### Subject

The subject of the current project.

#### Author

The author of the current project.

#### Keywords

A list of one or more keywords describing this project.

#### Data Version

The data version of the current project. You cannot change this data field. It is displayed for information purposes only.

#### Comments

You can enter comments about the current project.

## Output Log

The **Output Log** command of the **File** menu allows you to specify the file name for the log file created and used by the **Tours** program. **Tours** writes the results and intermediate information generated by the various design algorithms to the **Output Log file**. The amount of information written to the **Output Log File** depends on the level of detail selected with the **Report Level** command in the **Edit** menu. The file and all of its previous contents will be erased every time the **Output Log** command is executed. To select another log file or to restart the current log, execute the **Output Log** command again. To stop recording algorithm results without deleting the log file itself, use the **Close Log** command of the **File** menu.

### *Output Log Dialog Window*

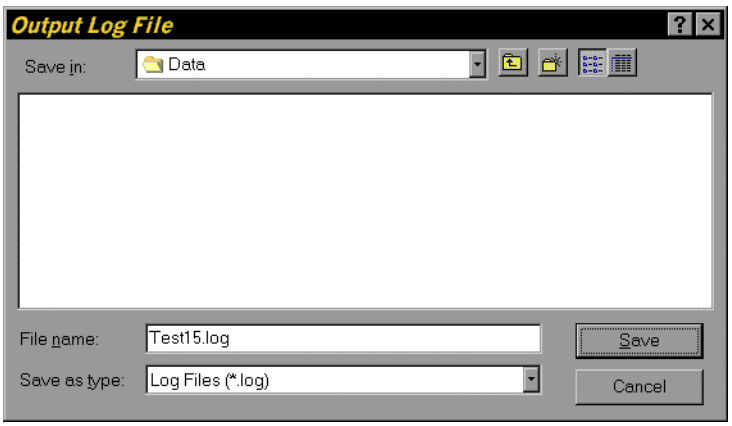



*Figure 5.10. Output Log Selection Dialog Window*

## Close Log

Use this command to close the current output log. No further information or algorithm results will be written to the output log, but the output log file itself will not be deleted. Reopening the same log file with the **Output Log** command will erase all the information in the log file, since the Output Log command always creates a new file.

## Print

Use this command to print a document. This command presents a **Print** dialog window, where you may specify the range of pages to be printed, the number of copies, the destination printer, and other printer setup options.

You can also copy the all views to the clipboard with the **Copy** command of the **Edit** menu and then paste the views in other Windows applications.

### *Print Shortcuts*

Toolbar: 

Keys: CTRL+P

## Print Dialog Window

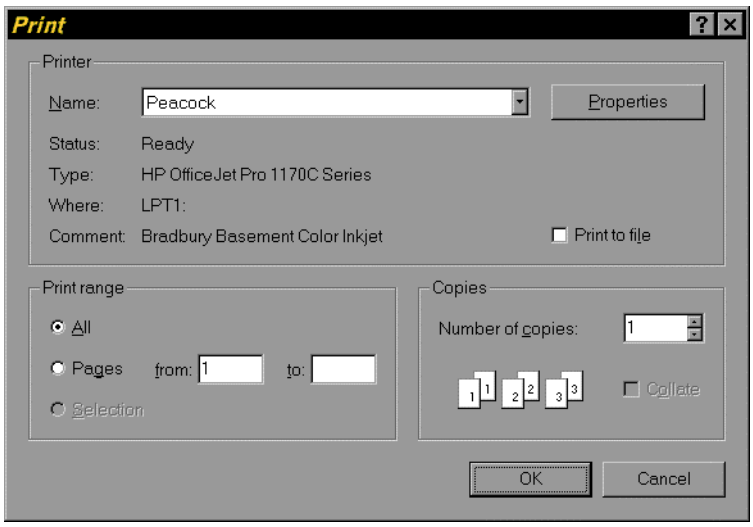



*Figure 5.11. Print Dialog Window*

## Print Data Items

The following options allow you to specify how the current view should be printed:

### Printer Name

This is the active printer and printer connection. Choose the Properties button to specify printing options for this particular printer.

### Print Range

Specify the pages you want to print:

*Table 5.1. Print Range Options*

| Option | Legend |
|---|---|
| All | Prints the entire document. |
| Selection | Prints the currently selected text. |
| Pages | Prints the range of pages you specify in the From and To boxes. |

### Copies

Specify the number of copies you want to print for the above page range.

### Collate Copies

Prints copies in page number order, instead of separated multiple copies of each page.

## Print Progress Dialog

The Printing dialog window is shown during the time that **Tours** is sending output to the printer. The page number indicates the progress of the printing.

To abort printing, choose **Cancel**.

## Print Preview

Use this command to display the active view as it would appear when printed. When you choose this command, the main window will be replaced with a **Print Preview** dialog window in which one or two pages will be displayed in their printed format. The toolbar of the Print Preview window offers you options to view either one or two pages at a time; move back and forth through the document; zoom in and out of pages; and initiate a print job.

### *Print Preview Shortcuts*

Toolbar: 

### *Print Preview Window*

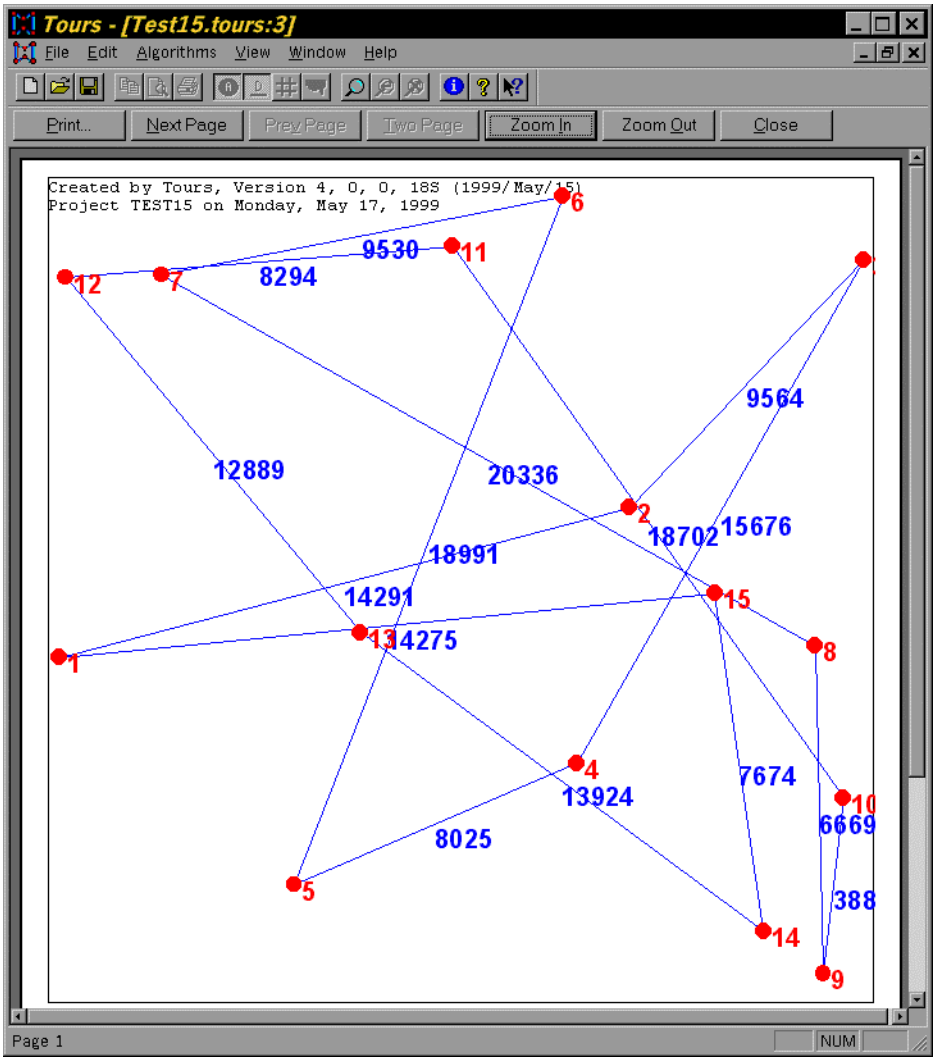



*Figure 5.12. Print Preview Window*

### *Print Preview Commands*

The print preview toolbar offers you the following options:

### Print

Bring up the print dialog window, to start a print job.

### Next Page

Preview the next printed page.

### Prev Page

Preview the previous printed page.

### One Page / Two Page

Preview one or two printed pages at a time.

### Zoom In

Take a closer look at the printed page.

### Zoom Out

Take a larger look at the printed page.

### Close

Return from print preview to the editing window.

## Print Setup

Use this command to select a printer and a printer connection. This command presents a **Print Setup** dialog window, where you specify the printer and its connection. This printer and these options will then be used by all subsequent **Print** commands. The same changes can also be made from the main Windows Control Panel.

### *Print Setup Dialog Window*

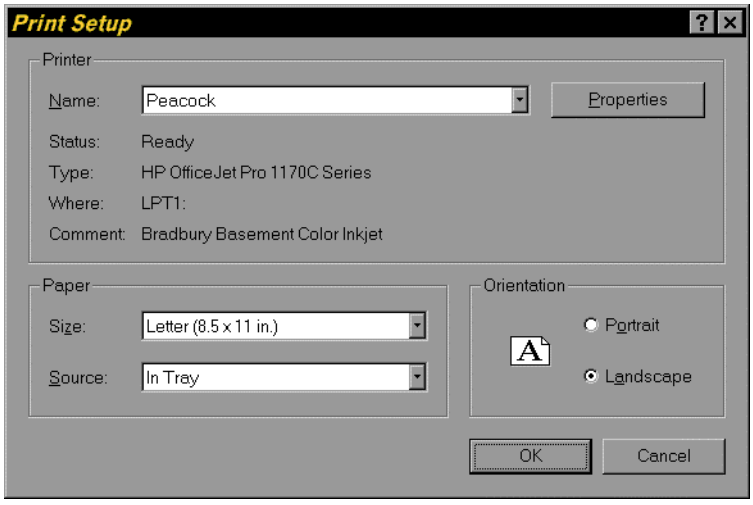



*Figure 5.13. Print Setup Dialog Window*

### *Print Setup Data Items*

The following options allow you to select the destination printer and its connection.

#### Printer

Select the printer you want to use. Choose the Default Printer; or choose the Specific Printer option and select one of the current installed printers shown in the box. You install printers and configure ports using the Windows Control Panel.

#### Orientation

Choose Portrait or Landscape.

#### Paper Size

Select the size of paper that the document is to be printed on.

#### Paper Source

Some printers offer multiple trays for different paper sources. Specify the tray here.

#### Options

Displays a dialog window where you can make additional choices about printing, specific to the type of printer you have selected.

## Most Recently Used Files

Every time a project is saved, **Tours** adds the fully qualified path and file name of the project data file to the list of the most recently used (MRU) files. The eight most recently used files are listed in the **File** Menu and saved in the Registry between sessions. You can bypass the **Open** command and open one of those projects directly by clicking on its file name.

## Exit

The **Exit** command of the **File** menu terminates the **Tours** program.

If there is a project currently open and if it has been modified, then the program will display the **Save Changes** dialog window and ask if you wish to save those changes, discard the changes, or if you wish to abort the termination of the application. The **Save Changes** dialog window is illustrated in Figure 5.3.

The **Tours** program can be terminated in the same way as all Windows programs by double clicking on the system menu box or by selecting the **Exit** command.

# Edit Menu

Most of the data items of the current project can be modified while executing the **Tours** program. The exceptions are the boundary values for the world coordinates and the map projection. Most of the data items can be edited with commands from the **Edit** menu. Some overall project characteristics can be changed with commands on the **File** menu. Finally, some algorithm parameters can be changed with the commands on **Algorithms** menu.

## Distance…

The **Distance** command of the **Edit** menu allows you to interactively change the distances between two points. Select the **Distance** command and the cursor will change to an up arrow in the **Tour** views. Move the cursor on top of the node of the first point and click the left button of the mouse. Then move the cursor on top of the node of the second point and click the left button of the mouse again. The **Edit Distance** dialog window will be shown. The distance dialog window is illustrated in Figure 5.14.

Edit the distance value. Click on **OK** to accept the modifications that you have made to the distance. If you click on **Cancel** then all the modifications that you made to the distance will be discarded and the distance will not be modified.

Alternatively, you can select the distance to be edited by holding down the Ctrl key and clicking the right mouse button while the cursor is over the first and second point. A third way to select the distance to be edited is to click the right mouse button while the cursor is on the link between the points itself. The context sensitive menu for links will be shown, from which the edit command can be selected. This latter method works only for links currently in the tour.

### *Edit Distance Dialog Window*

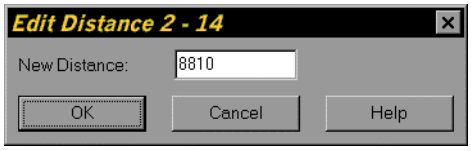



*Figure 5.14. Edit Distance Dialog Window*

## All Distances…

Use this command to display or edit all the distances between the nodes. The command will display the **All Distances** dialog window, which allows the modification of the distances. The All Distances dialog window is illustrated in Figure 5.15.

This dialog window shows the distances between the nodes. Each row represents an origin node and each column represents a destination node. You can use the scrollbars to move around the distance matrix, if there are more nodes than can be displayed simultaneously in the dialog window. You can also move the arrow keys to move around the distance matrix.

Double click on any distance and you will be able to modify that distance with the **Edit Distance** dialog window. You can also click on the **Edit** button or press **Alt-E** to edit the currently selected distance, which is indicated by a thin black border.

You can edit a distance as many times as you want and as many distances as you want.

When you have finished editing the distance, click on **OK** to accept the modifications that you have made to the distances. If you click on **Cancel** then all the modifications that you have made to the distances will be discarded and none of the distances will be modified.

### *Edit All Distances Dialog Window*

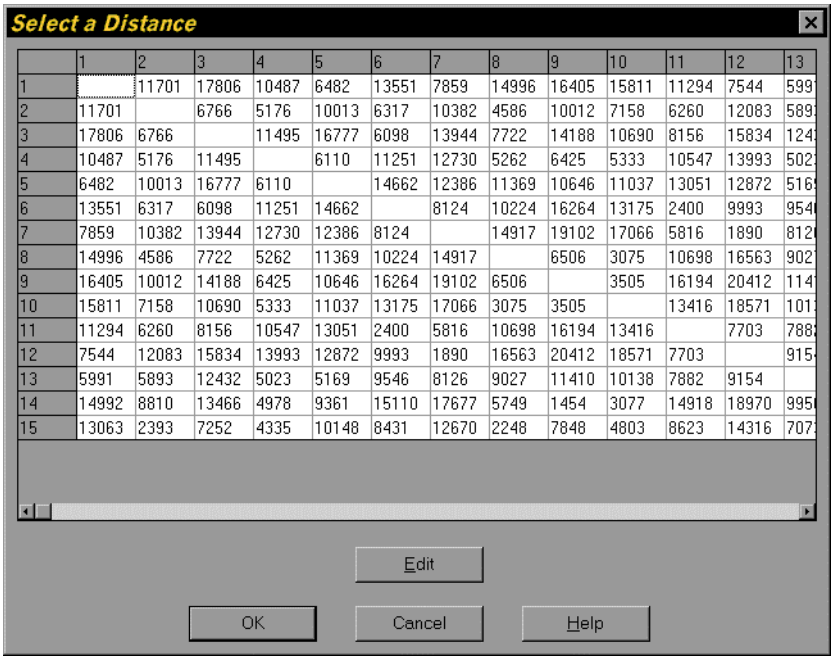



*Figure 5.15. Edit All Distances Dialog Window*

## Report Level...

The **Report Level** command of the **Edit** menu displays the **Report Level** dialog window, which allows you to set the level of detail written to the **Output Log** file and the number of pauses during algorithm execution. There are six levels ranging from 0 to 5, which generate increasingly more detailed output and algorithm pauses.

Click on **OK** to accept the modifications that you have made to the report level. If you click on **Cancel** then all the modifications that you made to the report level will be discarded and the report level will not be modified.

### *Report Level Dialog Window*

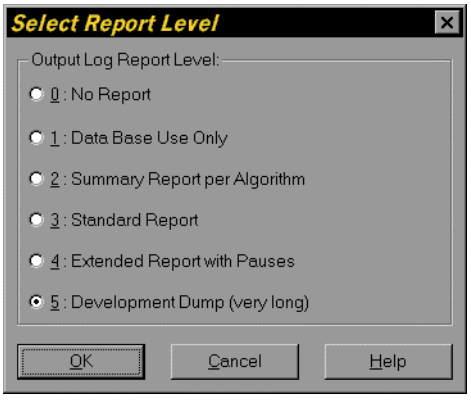



*Figure 5.16. Report Level Dialog Window*

### *Report Level Data Items*

#### Levels of Detail

There are six levels of detail and pauses for reporting:

0. NONE generates no output per algorithm and does not halt the algorithm execution. This level is used when maximum execution speed and minimal reporting is desired.
1. DATABASE generates one line of strictly numerical output per algorithm. No titles or headers are included. This level is primarily used to create a data base file, which can then be manipulated in a spreadsheet or statistical analysis program.
2. SUMMARY displays the total cost plus the algorithm run time. It is useful if you is only interested in the final results. This level of output should be used if you is interested in performing timing studies. Higher level of details corrupt timing results due to your interaction and graphics creation delays.
3. STANDARD generates the total cost for each of the algorithm components. The program runs without interruption until the complete algorithm is finished. If you have selected ALL, then the program runs uninterrupted for the 18 different combinations.
4. EXTENDED displays the total cost during each of the algorithm modules and the run time so far. The program halts frequently to allow you to observe the algorithm process.
5. DEVELOP generates extremely detailed output plus a very large number of intermediate results. This mode is only useful for debugging purposes or to observe the most detailed workings of the algorithms. The output is extremely long for large problems.

## Seed...

The **Seed** command of the **Edit** menu displays the **Seed** dialog window, which allows you to set the new seed for the random number generator. Several algorithms make random choices based on the random numbers. For example, the **Random** algorithm selects the next point on the tour randomly from all unvisited points. The tour improvement algorithms based on simulated annealing use random numbers to select the next two points to be tested for a possible exchange. An algorithm will always produce the same identical results if you select the same algorithm settings and sets the same random number seed.

Click on **OK** to accept the modifications that you have made to the random number seed. If you click on **Cancel** then all the modifications that you made to the random number seed will be discarded and the random number seed will not be modified.

### *Seed Dialog Window*

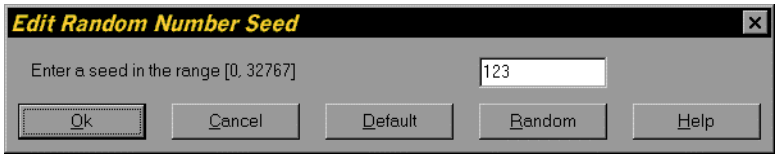



*Figure 5.17. Seed Dialog Window*

### *Seed Data Items*

Any positive seed value between 1 and 32767 is a valid starting seed for the random number generator. If a zero seed value is specified, the computer will create a random seed based on the computer clock.

#### Default

The default value for the seed is equal to one. This is the seed value when the program is originally started.

#### Random

If a value of zero is entered for the seed, then the program will select a random seed based upon the computer clock.

## Max. Replications...

The **Max. Replications** command of the **Edit** menu allows you to change the maximum number of replications of an algorithm. The default number of replications is equal to 20.

Click on **OK** to accept the modifications that you have made to the maximum number of replications. If you click on **Cancel** then all the modifications that you made to the maximum number of replications will be discarded and the maximum number of replications will not be modified.

### *Maximum Replications Dialog Window*

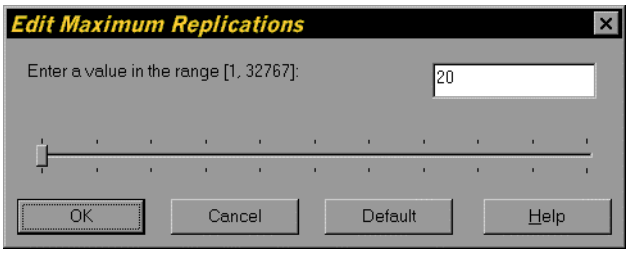



*Figure 5.18. Maximum Replications Dialog Window*

### *Maximum Replications Data Items*

The tour improvement algorithms based on simulated annealing select random pair of points to be tested for possible exchange. Different replications of the same algorithm can thus provide different tours. The higher the number of replications, the more likely it is that a high quality tour will be constructed. Of course, more replications require more computation time.

## Time Limit...

The maximum time limit is the maximum amount of time a single algorithm is allowed to execute. The time limit is expressed in seconds. Click on **OK** to accept the modifications that you have made to the time limit. If you click on **Cancel** then all the modifications that you made to the time limit will be discarded and the time limit will not be modified.

If an algorithm exceeds the time limit, then you will be asked either to abort or continue the algorithm with the **Time Expiration** window. At that time you have also the option to set a new time limit.

### *Time Limit Dialog Window*

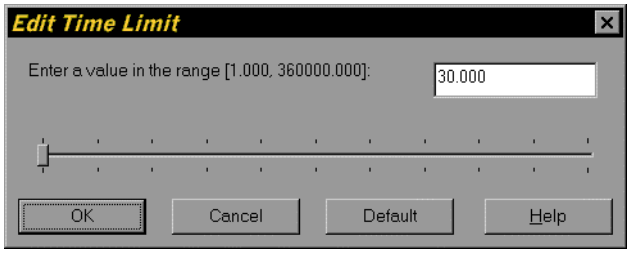



*Figure 5.19. Time Limit Dialog Window*

### *Time Limit Data Items*

Currently, the time limit is only used to terminate the two, three and annealing exchange algorithms if they have exceeded the time limit after one complete iteration, i.e. after all possible two or three exchanges have been tested or when the annealing temperature is decreased. So it is possible that the execution time of the improvement algorithm is actually larger than the time limit specified.

### *Time Limit Expiration Dialog Window*

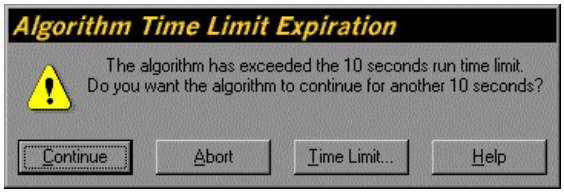



*Figure 5.20. Time Limit Expiration Dialog Window*

## Copy

The **Copy** command of the **Edit** menu copies the contents of the currently active view to the Windows Clipboard. The contents can then be pasted into other Windows applications such as CAD to design the layout in further detail. The **Tours** views copy the view as currently displayed to the clipboard. The **Notes** and **Statistics** views copy all the data in text format to the clipboard.

### *Copy View Shortcuts*

Toolbar: 

Keys: CTRL+C

# Algorithms Menu

## Nearest Neighbor

The Nearest Neighbor algorithm starts the tour with an initial point and then appends the nearest unvisited or free point to the tour. This algorithm was originally described by Rosenkrantz et al. (1977).

Different starting points may give different tour sequences and different tour lengths. The initial starting point is an algorithm parameter that you can specify on **Nearest Neighbor** tab of the **Select** command of the **Algorithms** menu. Since the Nearest Neighbor algorithm executes very fast, a possible alternative would be to start a tour at each point and then to retain the shortest tour among them.

## Random

The random construction algorithm creates a tour by selecting the next point on the tour randomly from all unvisited points. The random choices depend on the random number seed. The random algorithm will create identical traveling salesman tours when started with the same seed. If you want to generate different random tours for the same data, then you must change the seed before every run of the random algorithm. You can specify the seed of the random number generator with the **Seed** command of the **Edit** menu. Further information on the **Random** algorithm can be found in chapter on **Design Algorithms**.

## Sweep

The Sweep algorithm creates a tour by appending the points to the tour when they are traversed by a ray rotating around the center of de points. The center coordinates of all the points are found by averaging the x and y coordinates. The relative polar coordinates of each point with respect to the center point are then determined and the points are inserted in the tour by increasing polar angle. The algorithm was first described by Gillet and Miller (1974).

The initial starting angle of the rotating ray is an algorithm parameter that you can specify on **Sweep** tab of the **Select** command of the **Algorithms** menu. However the total tour length is independent of the starting angle, since all the points will always be visited in the same sequence irrespective of the starting angle.

## Savings

The Savings algorithm creates the tour with the sequential savings algorithm of Clark and Wright (1964). Conceptually the algorithm defines a base point and constructs an Eulerian tour that visits each of the other points and the returns to the base point. The Eulerian tour is then reduced in length by finding and executing the shortcut with the largest savings. The savings are computed as the sum of the distances to the base point of the two points minus the distance between the two points.

The base point is algorithm parameter that you can specify on the **Savings** tab of the **Select** command of the **Algorithms** menu.

# Select

The Select command allows you to specify an algorithm, to set the parameters for this algorithm, and then to start the execution of this algorithm. Only algorithms that take additional parameters will be displayed as one of the tabs in this dialog window. The Nearest Neighbor, Sweep and Savings algorithm require such additional parameters.

## *Select Algorithm Dialog Window (Nearest Neighbor Page)*

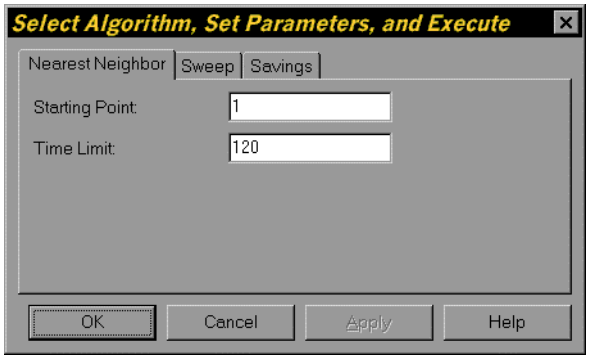



*Figure 5.21. Select Algorithm Window Nearest Neighbor Tab*

## *Select Algorithm Data Items (Nearest Neighbor Page)*

### Starting Point

The Nearest Neighbor construction algorithm starts the tour in one of the points. Depending on the starting point a different tour may be constructed. If no other data are changed, the algorithm will generate always the same tour if starting from the same point. You select any of the points in the current project as the starting point. The default value is to pick point 1 as the starting point. The starting point remains the same until you change it explicitly in the dialog window.

### Time Limit

The maximum time limit is the maximum duration a single algorithm is allowed to execute. The time limit is expressed in seconds. The time limit remains the same until you change it explicitly in any of the dialog windows.

The Time Limit can also be set with the **Time Limit** command of the **Edit** menu.

## *Select Algorithm Dialog Window (Sweep Page)*

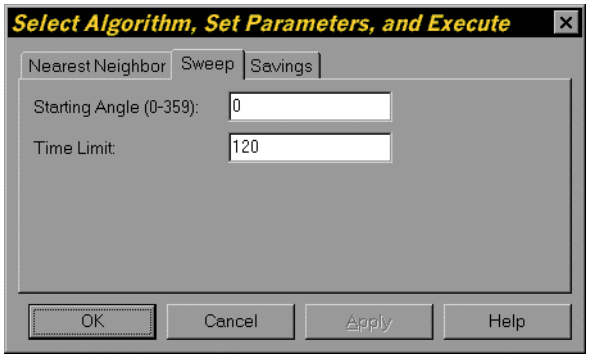


*Figure 5.22. Select Algorithm Window Sweep Tab*

## *Select Algorithm Data Items (Sweep Page)*

### Starting Angle

The initial starting angle of the rotating ray is an algorithm parameter. However the total tour length is independent of the starting angle, since all the points will always be visited in the same sequence irrespective of the starting angle.

### Time Limit

The maximum time limit is the maximum duration a single algorithm is allowed to execute. The time limit is expressed in seconds. The time limit remains the same until you change it explicitly in any of the dialog windows.

The Time Limit can also be set with the **Time Limit** command of the **Edit** menu.

## *Select Algorithm Dialog Window (Savings Page)*

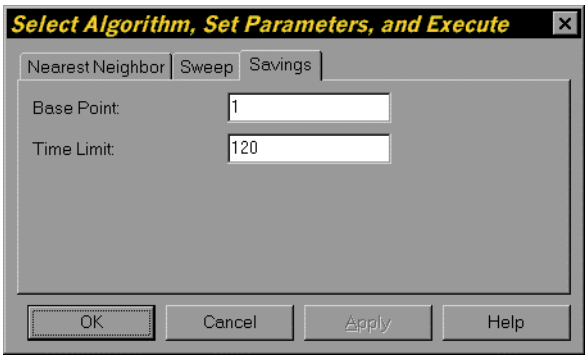



*Figure 5.23. Select Algorithm Window Savings Tab*

## *Select Algorithm Data Items (Savings Page)*

### Base Point

Conceptually the Savings algorithm defines a base point and constructs an Eulerian tour that visits each of the other points and the returns to the base point. The Eulerian tour is then reduced in length by finding and executing the shortcut with the largest savings. The savings are computed as the sum of the distances to the base point of the two points minus the distance between the two points. Different base points may generate different tours sequences and tour lengths.

### Time Limit

The maximum time limit is the maximum duration a single algorithm is allowed to execute. The time limit is expressed in seconds. The time limit remains the same until you change it explicitly in any of the dialog windows.

The Time Limit can also be set with the **Time Limit** command of the **Edit** menu.

## Manual

The **Manual** command allows you to construct a tour in an interactive manner. When you execute the Manual command, the cursor changes to a vertical arrow indicating that you can now select points to be added to the tour and the algorithm Abort button on the tool bar is enabled. You start and extend the tour by clicking the left button of the mouse when the cursor is over an unvisited or free point. If this is the first point that you clicked then the tour is started and this point is called the starting point. If this is not the first free point that you clicked then the tour is extended with this point. If you click for the second time on the starting point, then the tour is closed and the algorithm terminates. Not all points have to be included on the tour when the algorithm closes the tour and terminates, in other word, partial tours are allowed.

If you click again on the last point on the tour, then this point is removed from the tour and becomes free again and the next to last point on the tour becomes the last point. If the starting point is the only point on the tour and is clicked again, then it is removed and the complete tour is removed. Clicking then again on a free point will make that point the starting point.

In general, clicking on a free point adds this point to the tour and clicking on the last point of the tour removes this point from the tour. Clicking on intermediate points of the tour has no effect.

You can abort the Manual construction algorithm by pressing the Abort button on the toolbar, which is shown as a stop sign. The current partial tour is deleted and the algorithm statistics are not updated.

### *Manual Algorithm Shortcuts*

Keys: Ctrl+Alt+M

## Evaluate

The **Evaluate** command computes the length of the current tour if a tour has been created. The results are displayed in the **Notes** and **Statistics** views and in the Evaluate dialog window. This command is most frequently used after you have edited interactively the distances or after you have created a tour manually. The **Evaluate** command does not create a new tour, but rather computes the length of the current tour based on the current distances.

### *Evaluate Algorithm Shortcuts*

Keys: Ctrl+Alt+E

### *Evaluate Algorithm Dialog Window*

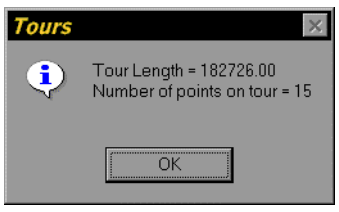



*Figure 5.24. Evaluate Algorithm Result Window*

## Aborting an Executing Algorithm

While the algorithms are executing, they will display the **Abort Algorithm** dialog window. The algorithms check at certain points during their execution if you have pressed the **Abort** button and, if so, will terminate at that time.

Note that for computationally intensive algorithms, there may be a significant delay between the moment you press the **Abort** button and the moment the algorithm checks for the button press. This is especially true for computers with single or slow processors.

Since the algorithm did not run to completion, the reported scores for the tour length and number of points on the tour may be incorrect. It is strongly recommended that you execute immediately the **Evaluate** command from the **Algorithms** menu if you plan to use the tour shown to ensure that the correct tour length and number of points on the tour are computed.

### *Abort Algorithm Dialog Window*


*Figure 5.25. Abort Algorithm Dialog Window*

# View Menu

All settings and switches in the **View** menu apply only to the currently active view. The equivalent setting in other views will not be affected by the commands of this menu.

## Grid

The **Grid** command is a toggle switch that displays or hides the orthogonal grid in the **Tours**. This grid is primarily of use when you want to move one or more of the points by dragging them in the **Tours** view. The distance between two grid lines and the size of the unit squares in the **Tours** view can be changed with the **Grid Size** command of the **View** menu.

### *Grid Project Shortcuts*

Toolbar: 

Keys: CTRL+SHIFT+G

## Map

The Map command allows you to specify if and how the background map will be displayed in the current View window with the **Map** dialog window. This dialog window is illustrated in Figure 5.26. The possible options are:

NONE: the background map is not displayed at all

OUTLINE: the boundary of the map objects is shown only

AREA FILLED: the boundary of the map objects is shown and the objects are filled in.

The default value is AREA FILLED. The **Map Data**, i.e., the map objects and their boundary and area fill color, have been read in from the **Map Data File** during the creation of the current project. They are saved from then on in the **Project Data File**. You can load a new background map with the **Load Map** command from the **Utilities** menu.

### *Map Dialog Window*

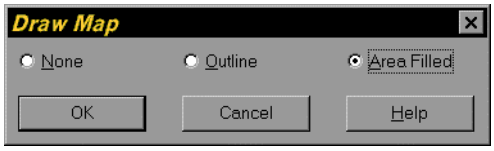



*Figure 5.26. View Map Dialog Box*

For further information on the function of this dialog window see the **Map** command.

### *Map Shortcuts*

Keys: CTRL+SHIFT+M

## Label Size...

The **Label Size** command allows the user to specify the size of the text labels used to identify the facilities and the links connecting them. Windows displays the labels in a font size that most closely matches the desired label size.

The display of the facility labels and link distances is controlled by the **Facility Labels** and **Link Distances** command of the **View** menu, respectively.

Click on **OK** to accept the modifications that you have made to the label size. If you click on **Cancel** then all the modifications that you made to the label size will be discarded and the label size will not be modified.

### *Label Size Dialog Window*

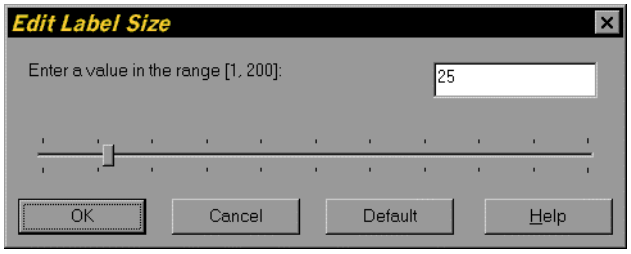



*Figure 5.27. Edit Label Size Dialog Window*

## Grid Size...

The **Grid Size** command allows the user to specify the distance between two adjacent grid lines. A smaller distance will display a grid with more grid lines, smaller squares, and higher resolution. The display of the grid is controlled by the **Grid** command of the **View** menu.

Click on **OK** to accept the modifications that you have made to the grid size. If you click on **Cancel** then all the modifications that you made to the grid size will be discarded and the grid size will not be modified.

### *Grid Size Dialog Window*

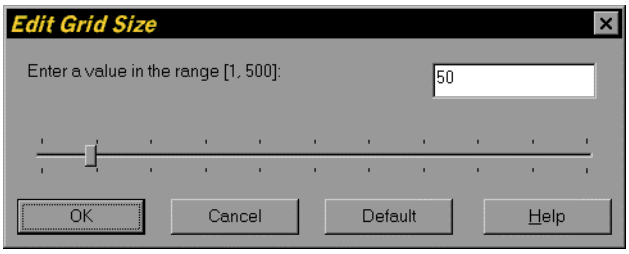



*Figure 5.28. Edit Grid Size Dialog Window*

## Zoom

The **Zoom** command allows you to select a rectangular portion of the current **View** window and to enlarge that rectangular region so that it will fill the complete **View** window. After the command has been selected you start a rubber band bounding rectangle by pressing and holding down the left mouse button in any **Tours** view. The rectangle will shrink or grow following the cursor until you release the left mouse button. The **Zoom** operation preserves the length to width aspect ratio of the **View** window. The zoom option is most useful to display the routes in more detail. The full, original **View** can be displayed by using the **Zoom Original** command. The previous **View** screen can be viewed by using the **Zoom Previous** command.

### *Zoom Shortcuts*

Toolbar: 

Keys: CTRL+SHIFT+Z

## Zoom Previous

The **Zoom Previous** command displays again the previous **View** window before the last **Zoom** command was executed.

### *Zoom Previous Shortcuts*

Toolbar: 

Keys: CTRL+SHIFT+P

## Zoom Original

The **Zoom Original** command displays the original, full **View** window before any **Zoom** command was executed for the current case.

### *Zoom Original Shortcuts*

Toolbar: 

Keys: CTRL+SHIFT+O

## Redraw

The **Redraw** command of the **Display** menu redraws the currently active view immediately, be it either a **Tours** view, **Notes** view, or **Statistics** view. It is used primarily to remove any remaining screen artifacts created by either a zoom operation on the current view or dragging a point in the current view.

### *Redraw Shortcuts*

Keys: CTRL+R

# Windows Menu

The **Windows** menu allows the opening, closing, arrangement, and selection of the **Tours** Views, the **Notes** Views, and the **Statistics** Views. In addition, the windows can be tiled and cascaded in standard Windows fashion as described in the Windows User's Guide.

## New Notes Window

The **New Notes View** command displays a new window showing the overall, aggregate project data. This view can be printed to the default printer with the **Print** command of the **File** menu. This window can be moved and sized to suit your taste.


*Figure 5.29. Notes View*

## New Statistics Window

The **New Statistics View** command displays a new window showing the history of algorithms statistics. This view can be printed to the default printer with the **Print** command of the **File** menu. This window can be moved and sized to suit your taste.

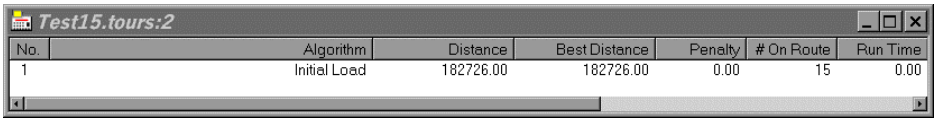



*Figure 5.30. Statistics View*

## New Tours Window

The **New Tours View** command adds a new window that displays the points and the tour of the current project. The display options for the new view are the standard options. You can then modify these options in the normal fashion described under the **Display** menu. Each **View** window can be moved and sized to suit your taste.

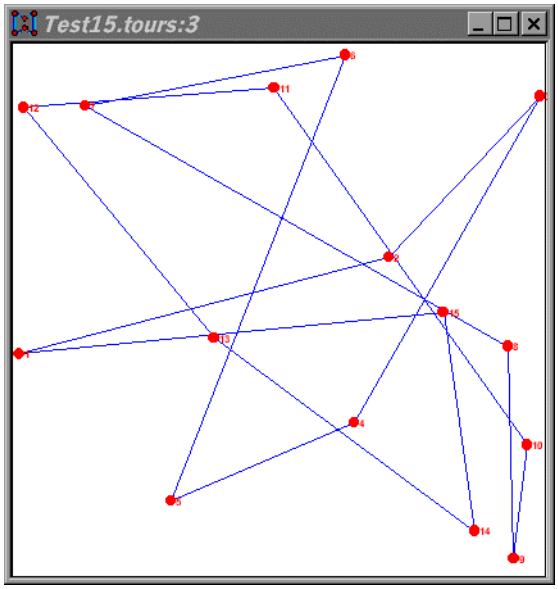



*Figure 5.31.* ***Tours*** *View*

## Cascade

This command cascades or arranges all child views in an overlapping manner, with the currently active child view on top.

## Tile

This command tiles or arranges all child views in a non-overlapping manner, attempting to make each view window the same size.

## Arrange Icons

This command arranges all icons of child views that have been minimized at the bottom of the **Tours** window.

## Opened Windows

You can activate any of the opened view windows by clicking on its name in the Window menu or by clicking anywhere in the window area. When you activate a window it displayed on top of all other child view windows.

## Toolbar

This command toggles the display of the toolbar of the **Tours** program. The toolbar contains short cut buttons to the most commonly used commands. When the mouse point is held immobile for a short time on any button of the toolbar a tool tip which gives the buttons functions will be displayed.


*Figure 5.32.* ***Tours*** *Toolbar*

The toolbar is dockable, i.e. it can be moved to any part of the application window and be reshaped.

*Figure 5.33. Dockable Tours Toolbar*

## Status Bar

This command toggles the display of the status bar at the bottom of the Tours window. The Status Bar displays a description of the currently highlighted command and the status of the keyboard.


*Figure 5.34. Status Bar*

# Utilities Menu

The Utilities menu allows the execution of miscellaneous support task.

## Load Map

The Load Map command allows you to specify of a new background map for the current project with the **Load Map** dialog window. The dialog window is illustrated in Figure 5.35. The coordinates of all the objects of the new map must fall inside the world boundary coordinates of the current project. After they have been loaded, the map data will be saved in the **Project Data File**. Further information on the data format of the Map File is given in the **Map Data** section. The new **Map Data File Name** of the file will be shown in the **Notes** view.

### *Load Map Dialog Window*

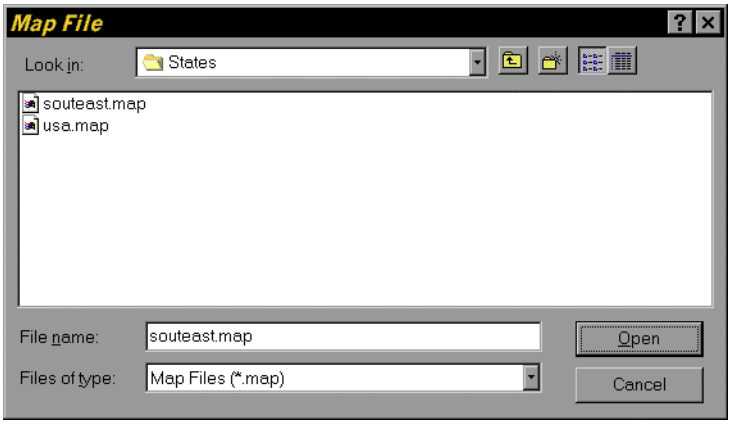



*Figure 5.35. Load Map Dialog Window*

For further information on the function of this dialog window see the **Load Map** command.

# Help Menu

## Help Topics

This command displays the Contents page of the **Tours** interactive help system as shown in Figure 5.36.

The **Tours** program contains an interactive help system. The instructions in the help system always take precedence over those in the printed User's Manual. The Help system can be started from the **Help** menu or by pressing F1.

### *Help Topics Window*

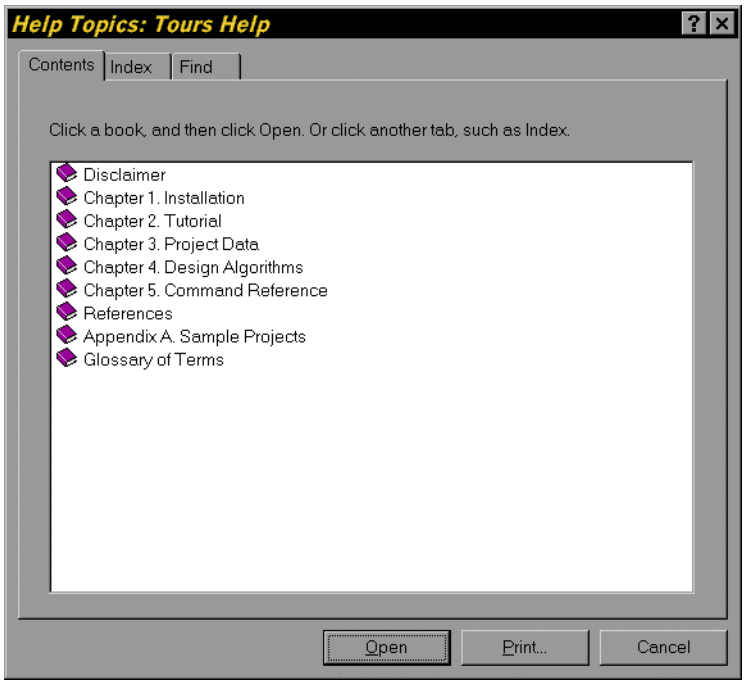



*Figure 5.36. Help Contents Window*

### *Help Shortcuts*

Toolbar: 

Keys: CTRL+H
F1

## Context Sensitive Help

You can request help for a specific topic by pressing SHIFT-F1 or by clicking the button for context sensitive help in the toolbar. The **Tours** application is placed in Help mode. You can then specify the topic by a mouse click on a menu command or

an area of the screen or by the key stroke(s) for a menu command. The help file will be opened on that particular topic.

Pressing the Esc button while the application is in Help mode will cancel the Help mode and return the application to its normal operation.

Pressing the Help button in the various dialog windows will also activate the context sensitive help for that dialog window.

### *Context Sensitive Help Shortcuts*

Toolbar: 

Keys: Shift+F1

## About Tours

The **About** command of the **Help** menu shows the **About Tours** dialog window with the **Tours** program and the **Scientif** library information. This information includes the name, version, date, and copyright. It also shows the program and library icon. The **About Tours** dialog window is illustrated it Figure 5.37.

### *About Tours Dialog Window*

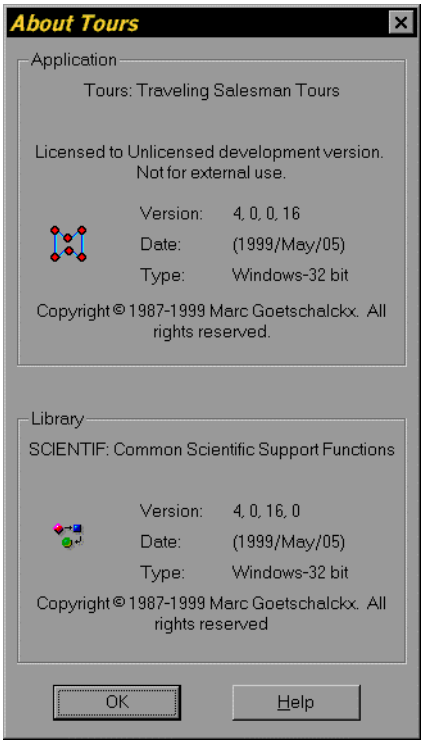



*Figure 5.37. About Dialog Window*

### *About Shortcuts*

Toolbar: 

Keys: Ctrl+I

# References

## Book and Journal References


1. Allison D. C. and Noga, M. T., (1984), "The $L_1$ Traveling Salesman Problem," ***Information Processing Letters***, Vol. 18, No. 4, pp. 195-199.
2. Bellmore, M. and Nemhauser, G. L., (1968), "The Traveling Salesman Problem: A Survey," ***Operations Research***, Vol. 16, No. 3, pp. 538-558.
3. Boyd S. C., W. R. Pulleyblank and G. Cornuejols, (1987), "TRAVEL - An Interactive Traveling Salesman Problem Package for the IBM Personal Computer," ***Operations Research Letters***, Vol. 6, No. 3, pp. 141-143.
4. Bozer, Y. A., Schorn, E. C. and Sharp, G. P. (1990), "Geometric Approaches to Solve the Chebyshev Traveling Salesman Problem," ***IIE Transactions***, Vol. 22, No. 3, September, pp. 238-254.
5. Christofides N. and Eilon S., (1972), "Algorithms For Large-Scale Traveling Salesman Problems," ***Operations Research Quarterly***, Vol. 23, pp. 511-518.
6 Clark, G. and J. Wright, (1964), "Scheduling of Vehicles From a Central Depot to A Number of Delivery Points," ***Operations Research***, Vol. 12, pp 568-581.
7 Gillett, B. and L. Miller, (1974), "A Heuristic Algorithm For the Vehicle Dispatch Problem," ***Operations Research***, Vol. 22, pp 340-349.
8. Golden, B. L., Bodin, L., Doyle, T., and Stewart, W. Jr., (1980), "Approximate Traveling Salesman Algorithms," ***Operations Research***, Vol. 28, No. 3, pp. 694-711.
9. Helbig Hansen, K. and Krarup, J., (1974), "Improvements on the Held-Karp Algorithm for the Symmetric Traveling Salesman Problem," ***Mathematical Programming***, Vol. 7, pp. 87-96.
10. Held, M. and Karp, R. M., (1970), "The Traveling-Salesman Problem and Minimum Spanning Trees," ***Operations Research***, Vol. 18, No. 6, pp. 1138-1162.

11. Held, M. and Karp, R. M., (1971), "The Traveling-Salesman Problem and Minimum Spanning Trees: Part II," ***Mathematical Programming***, Vol. 1, pp. 6-25.

12. Laporte, G., (1992), "The Traveling Salesman Problem: An Overview of Exact and Approximate Algorithms," ***European Journal of Operational Research***, Vol. 59, pp. 231-247.

13. Lawler E. L., Lenstra J. K., Rinnooy Kan A. H. G., and Schmoys, D. B., (1985), **The Traveling Salesman Problem**, John Wiley & Sons, Chichester, Great Britain.

14. Lenstra, J. K. and A. H. G. Rinnooy Kan (1981), "Complexity of Vehicle Routing and Scheduling Problems," ***Networks***, Vol. 11, No. 2, pp. 221-227.

15. Lin S., and B. Kernighan, (1973), "An Effective Heuristic Algorithm for the Traveling Salesman Problem," ***Operations Research***, Vol. 21, pp. 498-516.

16. Lin, S., (1965), "Computer Solutions of the Traveling Salesman Problem," ***Bell System Technical Journal***, Vol. 44, pp. 2245-2269.

17. Little J. D., Murty K. G., Sweeney D. W. and Karel C., (1963), "An Algorithm for the Traveling Salesman Problem," ***Operations Research***, Vol. 11, No. 6, pp. 972-989.

18. Or I., (1976), "Traveling Salesman-Type Combinatorial Problems and their Relation to the Logistics of Regional Blood Banking," Unpublished Ph.D. Thesis, Northwestern University, Evanston, Illinois.

19. Parker R, G. and Rardin R. L., (1983), "The Traveling Salesman Problem: An Update of Research," ***Naval Research Logistics Quarterly***, Vol. 30, pp. 69-99.

20. Platzman, Loren K. and John J. Bartholdi, III (1984), "Spacefilling Curves and the Planar Traveling Salesman Problem," PDRC Report Series 83-02, School of Industrial and Systems Engineering, Georgia Institute of Technology.

21. Rosenkrantz, D. J., R. E. Stearns, and P. M. Lewis, (1977). "An Analysis of Several Heuristics for the Traveling Salesman Problem." ***SIAM Journal of Computing***, Vol. 6, pp. 563-581.

22. Smith T. H. and Thompson G. L., (1977), "A LIFO Implicit Enumeration Search Algorithm for the Symmetric Traveling Salesman Problem using Held and Karp's 1-Tree Relaxation," ***Annals of Discrete Mathematics***, Vol. 1, pp. 479-493.

23. Volgenant, T. and R. Jonker, (1982). "A Branch and Bound Algorithm for the Symmetric Traveling Salesman Problem based on the 1-Tree Relaxation." ***European Journal of Operations Research***, Vol. 9, pp. 83-89.

# World Wide Web Sites

24. Goetschalckx, Marc. www.isye.gatech.edu/~mgoetsch/index.html

# Appendix A. Sample Projects

## Test15

### Test15.dat Project Data File Example

```
[data_version]              20000
[project_name]              Test15
[distance_norm]             Euclidean
[number_of_points]          15
[minimum_x_coordinate]      0
[maximum_x_coordinate]      16383
[minimum_y_coordinate]      0
[maximum_y_coordinate]      16383
[points_file_name]          Test15.pts
[map_projection}            orthogonal
[map_file_name]             atlanta.map
[world_radius]              3960.00
[tolerance]                 0.01000
[time_limit]                120
[maximum_iterations]        500
[report_level]              3
[seed]                      123
[time_limit]                30
```

### Test15.pts Points Data File Example

```
220     9526    2
11537   6552    3
16181   1632    4
10495   11622   5
4880    14032   6
10214   375     7
2240    1931    8
15214   9293    9
15379   15797   10
15783   12315   11
8029    1367    12
351     1983    13
6191    9032    14
```

```
14196   14951   15
13221   8252    1
```

# Glossary of Terms

## ASCII

ASCII is an acronym in computer science for American Standard Code for Information Interchange. It is a standardized coding scheme that assigns numeric values to letters, numbers, punctuation marks, and certain other characters. By standardizing the values used for these characters, ASCII enables computers and computer programs to exchange information. ASCII provides for 256 codes divided into two sets of 128 each. The standard ASCII character set consists of the first 128 codes. The first 32 values of standard ASCII are assigned to communication and printer control codes, i.e., non-printing characters, such as backspace, carriage return, and tab, that are used to control the way information is transferred from one computer to another or from a computer to a printer. The remaining 96 codes are assigned to common punctuation marks, the digits 0 through 9, and the uppercase and lowercase letters of the Roman alphabet. Since ASCII characters just consists of characters, numbers, and punctuation marks and none of the special formatting codes associated with word processors, most programs are able to read ASCII files. Common synonyms are "DOS text" or "text".

## GUI

GUI is the abbreviation for graphical user interface. **Tours** follows the conventions of the standard Windows user interface.

## MRU

MRU is the abbreviation for Most Recently Used. Most applications display a list of most recently used files in their **File** menu to allow quick access to the project and document files that have been recently saved.

## RGB

RGB is an acronym for Red-Green-Blue and it denotes a color described by three numbers for its red, green, and blue components, respectively. Windows allows values from 0 to 255 for each component for a total of more than 16.7 million colors.

## Scientif

**Scientif** is a library containing common scientific functions used in Windows programs. The current implementation is ***scienmfc.dll***, which is the 32-bit dynamic link library required to run the **Tours** application program. **Scientif** in turn requires a 32-bit Windows operating system, which is denoted by Win32. The safest location for ***scienmfc.dll*** is in the directory where **Tours** was installed. For the 32-bit Windows NT operating system ***scienmfc.dll*** can also be placed in the system32 directory of the directory where Windows NT has been installed. Usually this directory is ***c:\winnt\system32***. For the Windows 95 and 98 operating systems ***scienmfc.dll*** can also be placed in the system directory of the directory where Windows 95 or 98 has been installed. Usually this directory is ***c:\windows\system***. The automated **Setup** program with its default selections will place all the application and library files in the appropriate directories.

## Setup

**Setup** is the automated, Windows based installation program which copies the **Tours** program, help, libraries, and example data files to your hard disk. In addition, it installs **Tours** in the selected program group on your desktop. The **Setup** program file is ***setup.exe*** and it is located on the first distribution diskette or CD-ROM.

## TSP

TSP is the acronym for the Traveling Salesman Problem, which is the mathematical optimization problem to construct a cycle of minimum length that visits all points in a set exactly once. A cycle that visits all the points in the set exactly once is called a Hamiltonian cycle. The TSP is thus the problem of finding the Hamiltonian cycle of shortest length.

## Win32

**Tours** is a 32-bit Windows program and as such requires a 32-bit version of the Windows operating environment, denoted by Win32. Current implementations of Win32 are Windows NT (all versions), Windows 95 (all versions), and Windows 98 (all versions). This version of **Tours** is no longer compatible with the 32-bit extensions Win32s to the 16-bit Windows 3.1.

## WWW

WWW is the acronym for the World Wide Web and denotes the collection of sites on the Internet that contain a large variety of information.

# Index

## D



## E



## F



## G



## H



## I



## K



## L



## M



## N



## O

## P



## Q



## R



## S



## T



## U



## V



## W



## Z