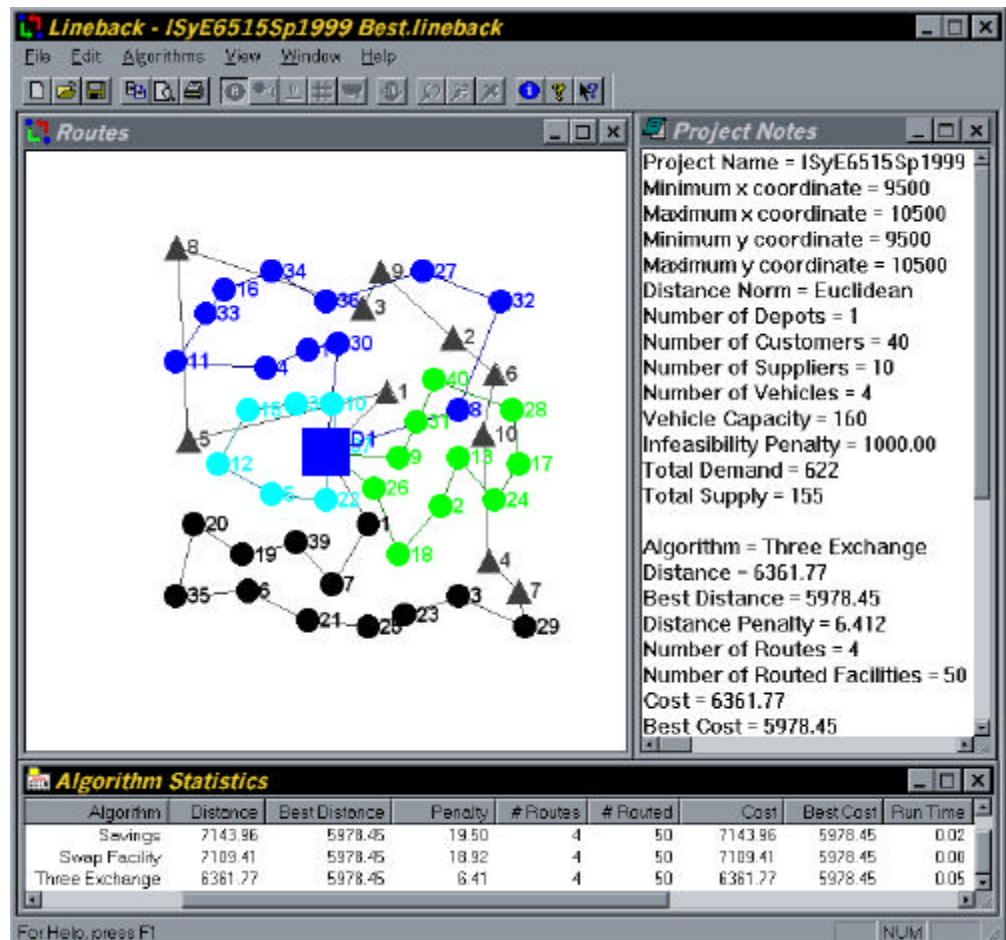


# Lineback

## User's Manual



By Marc Goetschalckx

***Lineback User's Manual, Version 1.04, June 20, 2000***  
***Copyright © 1987-2000, Marc Goetschalckx. All rights reserved.***

**Marc Goetschalckx**  
**4031 Bradbury Drive**  
**Marietta, GA 30062-6165**  
**+1-770-578-6148**  
**+1-770-565-3370**  
**Fax: +1-770-578-6148**  
**E-mail: [marc\\_goetschalckx@yahoo.com](mailto:marc_goetschalckx@yahoo.com)**  
**URL: [www.logisticsCAD.com](http://www.logisticsCAD.com)**

# Contents

- Disclaimer** **1**
  - Warranty ..... 1
  - Proprietary Notice..... 2
  - Version..... 2
  
- Chapter 1. Installation** **3**
  - Installing Lineback ..... 3
  - Removing Lineback..... 5
  
- Chapter 2. Tutorial** **7**
  - Creating a Small Tutorial Project ..... 7
  - Designing Vehicle Routes..... 17
  - Using Lineback Routes in Other Windows Programs ..... 25
  
- Chapter 3. Project Data** **31**
  - Specifying and Editing Project Data..... 31
  - Importing Data Files from Previous Versions ..... 39
  
- Chapter 4. Design Algorithms** **43**
  - Introduction..... 43
  - Construction Algorithms..... 44
  - Partial Routes Construction Algorithms ..... 45
  - Insertion Algorithms ..... 46
  - Improvement Algorithms..... 47
  - Evaluation Algorithms ..... 50
  
- Chapter 5. Command Reference** **51**
  - Menu Overview ..... 51
  - File Menu..... 51
  - Edit Menu ..... 66
  - Algorithms Menu..... 82
  - View Menu ..... 88
  - Windows Menu..... 91
  - Help Menu ..... 94
  
- References** **97**
  - Book and Journal References..... 97
  - World Wide Web Sites ..... 98
  
- Appendix: Sample Projects** **99**

Small.....	99
Vigo and Toth Case 51-80.....	101
<b>Glossary of Terms</b>	<b>105</b>
<b>Index</b>	<b>107</b>

# 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

Marc Goetschalckx owns both the **Lineback** software program and its documentation. Both the program and the documentation are copyrighted with all rights reserved by Marc Goetschalckx. No part of this publication may be produced, transmitted, transcribed, stored in a retrieval system, or translated into any language in any form without the written permission of Marc Goetschalckx.

---

## Version

Version 1.04, June 20, 2000.

# Chapter 1. Installation

---

## Installing Lineback

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

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

1. Insert the distribution disk 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 drive **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 **Lineback** using the installation instructions for Windows NT version 3.51.

## Installation Instructions for Windows NT 3.51

1. Insert the distribution disk 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*

**Lineback** 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 **Lineback** application directory. This will ensure that the **Lineback** 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 that you have write access privileges to the directory where **Lineback** 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. We recommend that you install **Lineback** 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 **Lineback** program. The automated **Setup** program copies and registers the grid control during its installation process. If you run **Lineback** 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 Small 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 **Lineback** and the grid control on this computer.



---

# Removing Lineback



You can remove the **Lineback** program and its application libraries from your computer, as well as remove its keys from the **Registry**. The exact method of removing **Lineback** 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 **Lineback** on your computer see the instructions in the section on Installing **Lineback**.

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

Select the **Add/Remove Programs** command in the **Control Panel** of your computer. **Lineback** will be listed as one of the applications that can be automatically removed. Select **Lineback** and press **Add/Remove**. All the files specific to **Lineback**, 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 **Lineback** 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. **Lineback** 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 **Lineback** from your computer was placed in the same program group that you selected for the **Lineback** icon. Press this **Uninstall** icon. All the files specific to **Lineback**, 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 **Lineback** 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. **Lineback** 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 project and design its linehaul-backhaul routes in an interactive manner. For large projects it might be more convenient to create the necessary data files outside **Lineback** with a file editor capable of creating pure ASCII files and then to import the project from these data files.

**Lineback** follows the standard Windows graphical user interface (GUI) conventions. This tutorial will focus on the features unique to the **Lineback** 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 chapters on Project Data and Design Algorithms. A summary of all the available actions in **Lineback** can be found in the section on the Command Reference. A list of traditional and WorldWide Web (WWW) references for further reading is given in the Reference section. Finally, we will be using the Small linehaul-backhaul 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 **Lineback** program. **Lineback** is compatible with long file names and long 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 subdirectory of the **Lineback** 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 **New** dialog window for this project is illustrated in Figure 2.1.

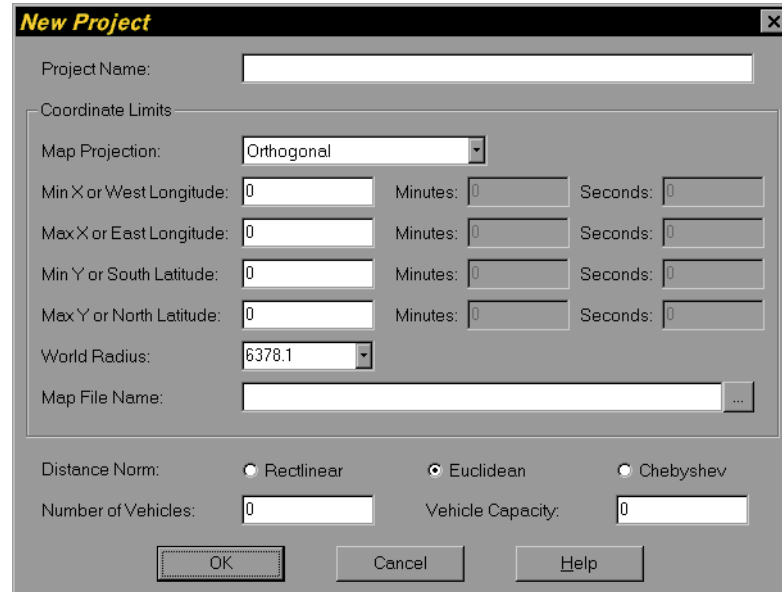


Figure 2.1. Initial New Project Dialog Window

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 16000, 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. We will be using the straight line or Euclidean distance norm in this tutorial. Finally, enter five for the number of vehicles and enter a vehicle capacity of 400. The final **New** dialog window for this project is illustrated in Figure 2.2. Press **OK** and a new project without any facilities will be created.

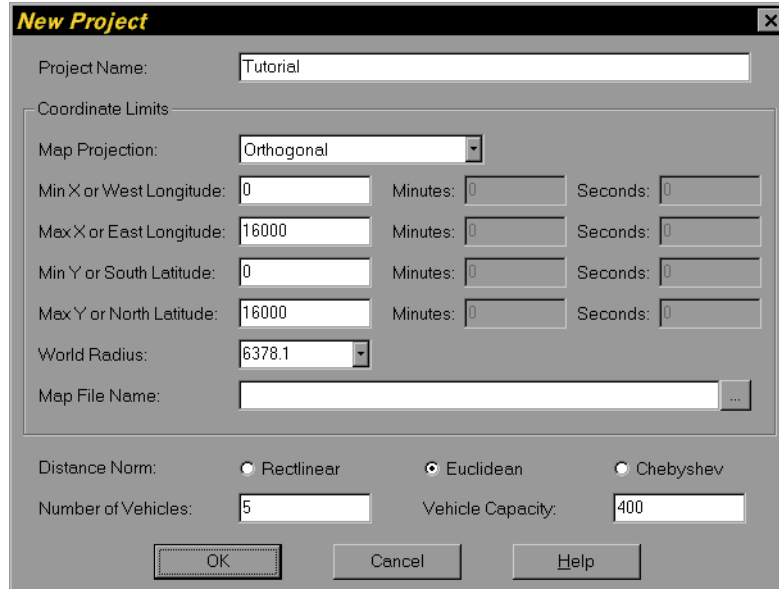


Figure 2.2. Final New Project Dialog Window

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** menus. 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.

The **Lineback** 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) **Routes** view. Changing the attribute of one view does not affect the same attribute in other views.

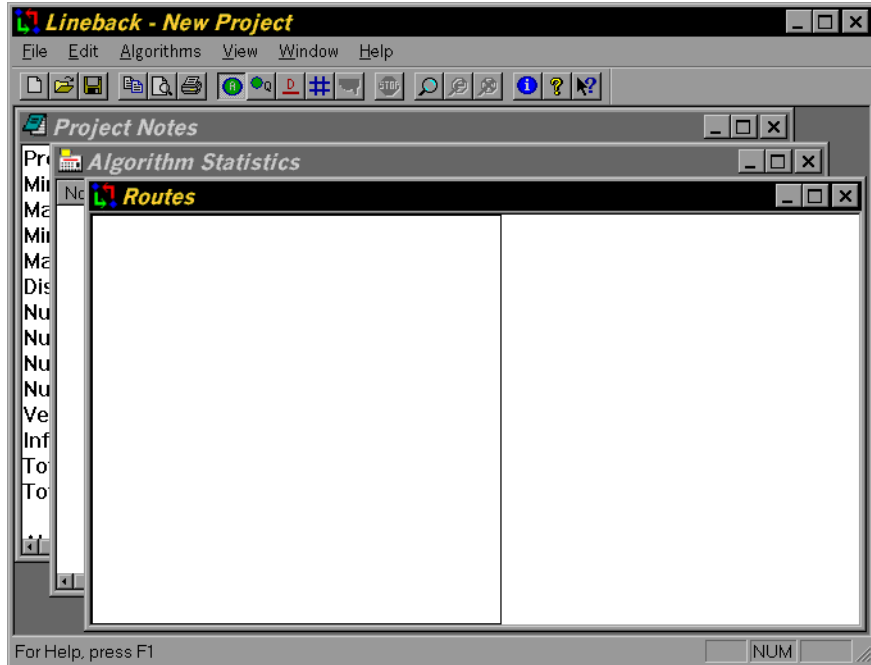


Figure 2.3. Initial Views for a New Project

### 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.

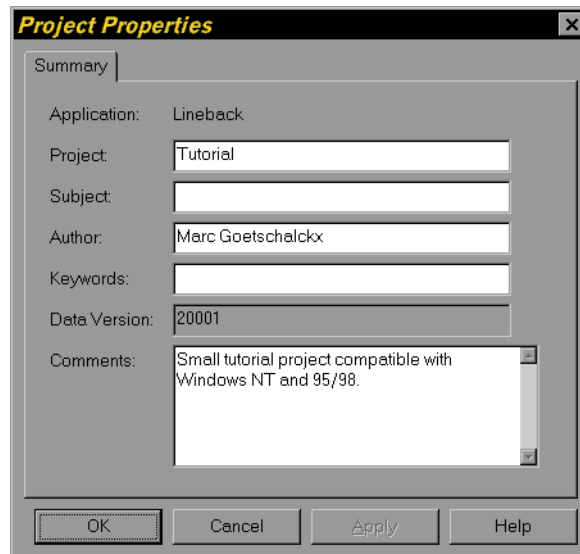


Figure 2.4. Properties Dialog Window for the Tutorial Project

## Controlling the display of the background map

Select the **Routes** view, which will display the facilities and the routes but does not contain any facilities 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, Outline: the boundaries of the map objects are shown only, and Area Filled: the boundaries of the map objects are 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 facilities. Usually, either the background map or the background grid is displayed. In this tutorial we will be using the grid, so select None to turn the display of the background map off.

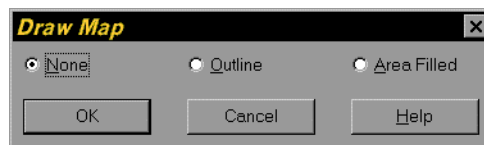


Figure 2.5. View Map Dialog Window

## 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 **Routes** view are now 1000 units apart.

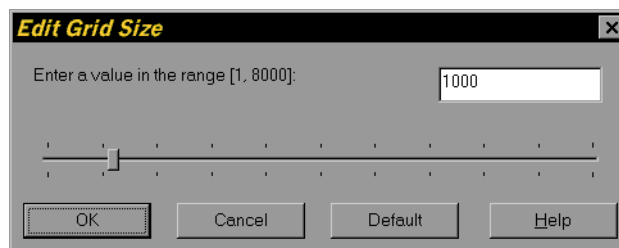


Figure 2.6. Edit Grid Size Dialog Window

If you have selected to display the grid in the **Routes** 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 **Lineback**. Specify a name for the project data file, which by default has the *lineback* extension. The **Save As** dialog box for this



tutorial project is illustrated in Figure 2.7. Press the **Save** button and the file for the current project will be saved to disk.

Some versions of the Windows operating environment will truncate the default *lineback* extension to the three letters *lin*, so we recommend that you explicitly add the *lineback* extension to the file name in the Save As dialog window when saving a project for the first time.

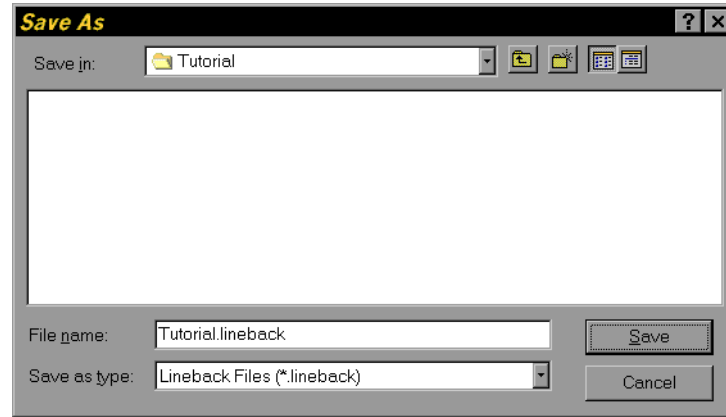


Figure 2.7. Save As Dialog Box for the Tutorial Project

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.

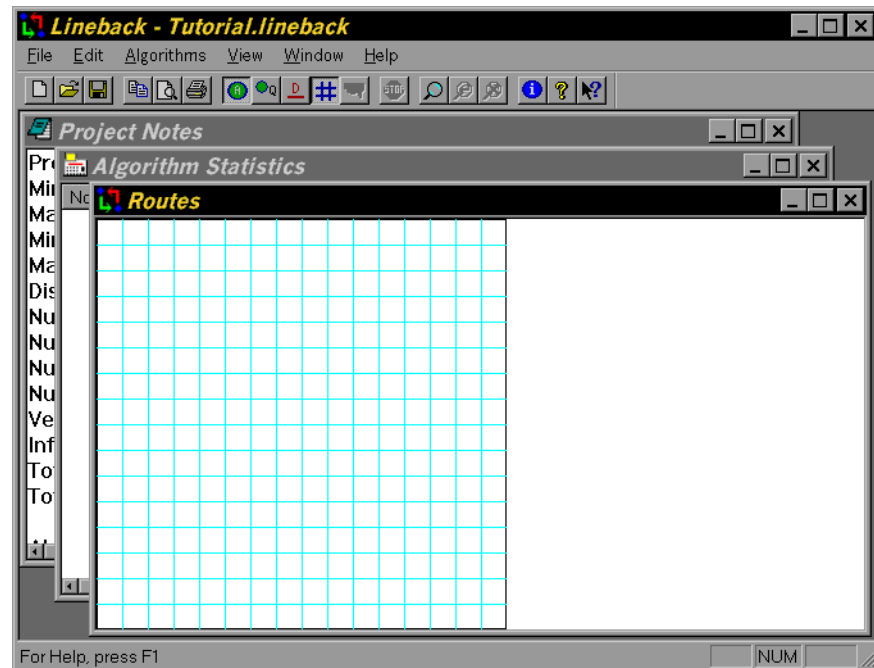


Figure 2.8. Initial Views of the Tutorial Project



## Adding Facilities to the Project

From the **Edit** menu, select the Add Facilities submenu and select the **Add Depot** command. The cursor will change to a crosshair shape. Move the cursor in the **Routes** view to the approximate location of the depot and click the left mouse button. The Add Depot dialog window will be displayed where you can specify additional data for this depot and refine the coordinates. The dialog window is shown in Figure 2.9.

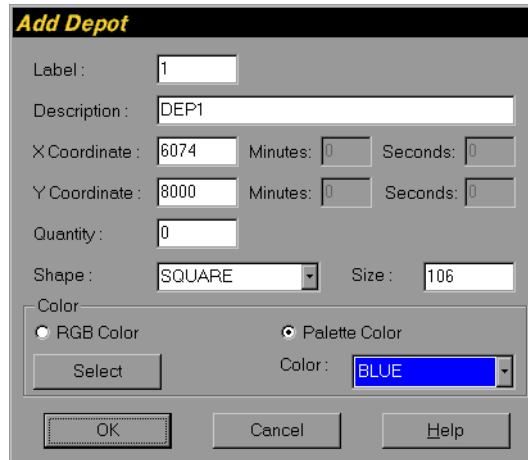


Figure 2.9. Add Depot Dialog Window

A facility label must consist of 1 to 7 characters. Only letters, numbers, and the underscore character are allowed. A facility has to have a unique label among all facilities of that type. You have only one customer with label 1, but you can have a customer and a supplier both with label 1. The label is a required data item; i.e., it cannot be empty. The label will be used to identify the facility in all screen dialog windows.

The facility name must consist of 1 to 31 characters. Only letters, numbers, and the underscore character are allowed. The name is a required data item; i.e., it cannot be empty.

The location of the facility is given by its latitude and longitude or x and y coordinates, depending on the map projection. The latitude or the y coordinate of a facility must fall inside the world latitude or y coordinate boundaries. The longitude or the x coordinate of a facility must fall inside the world longitude or x coordinate boundaries. The world coordinate boundaries were specified when the project was created in the **New** dialog window.

Each customer has a delivery quantity and each supplier has a pickup quantity that must be carried on the vehicle. The quantity must be a non-negative integer less than or equal to the vehicle capacity.

Each facility has a geometrical symbol or shape used to display the facility in the **Routes** views. Each facility has a color used to display the facility symbol in the **Routes** views. Each facility has a size of its geometric symbol displayed in the **Routes** views.

Specify 1 for the facility label, DEP1 for the depot facility description. Define the x and y coordinates to be 6074 and 8000, respectively. Since the depot has no delivery or pickup quantity, we specify a zero quantity. Specify a square shape, a display size of 106, and a blue palette color. Press **Ok** to add this depot to the project. The number of depots in the **Notes** view is updated to indicate one depot facility. You

can only have a single depot in a project, so the **Add Depot** command is now disabled.

From the **Edit** menu, select the Add Facilities submenu and then select the **Add Customer** command. The cursor will change to a crosshair shape. Move the cursor in the **Route** view to the approximate location of the customer and click the left mouse button. The Add Customer dialog window will be displayed where you can specify additional data for this customer and refine the coordinates. The dialog window is shown in Figure 2.10.

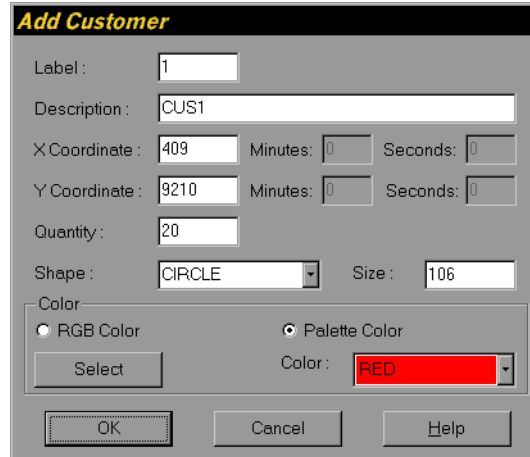


Figure 2.10. Add Customer Dialog Window

Specify 1 for the customer label, CUS1 for the customer facility description. Define the x and y coordinates to be 409 and 9210, respectively. Specify the quantity to be 20. Specify a circle shape, a display size of 106, and a red palette color. Press **Ok** to add this customer to the project. The number of customers in the **Notes** view is updated to indicate one customer facility and total linehaul or delivery quantity is updated to 20.

From the **Edit** menu, select the Add Facilities submenu and select the **Add Supplier** command. The cursor will change to a crosshair shape. Move the cursor in the **Route** view to the approximate location of the supplier and click the left mouse button. The Add Supplier dialog window will be displayed where you can specify additional data for this depot and refine the coordinates. The dialog window is shown in Figure 2.11.

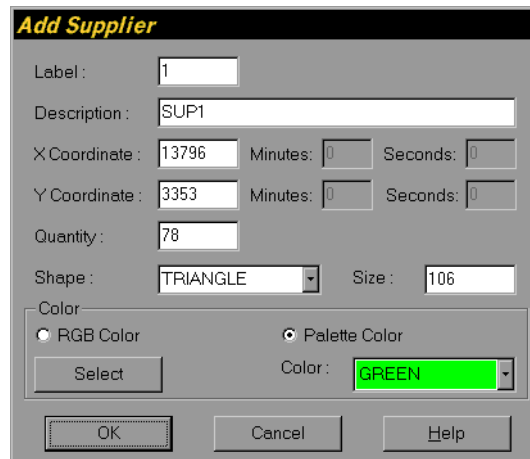


Figure 2.11. Add Supplier Dialog Window

Specify 1 for the supplier label, SUP1 for the supplier facility description. Define the x and y coordinates to be 13796 and 3353, respectively. Specify the quantity to be 78. Specify a triangle shape, a display size of 106, and a green palette color. Press **Ok** to add this supplier to the project. The number of suppliers in the **Notes** view is updated to indicate one supplier facility and the total backhaul or pickup quantity is updated to 78.

At this time the project contains three facilities and you can continue adding customers and suppliers in this way. The data for all the customers, suppliers and depot is given in the appendix as the Small project. You can also import all the facilities from a set of ASCII data files.



However, the **Import** command essentially creates a new project from the ASCII data files, so you have to specify again any information that you added to the project with the **Properties** command and adjust the settings of the **Routes** view to suite your taste. You have to use again the **Save As** command to save this project for the first time and specify manually the *lineback* extension, otherwise the ASCII data in the import data file will be overwritten with the binary data.

Select the **Import** command from the **File** menu and navigate to the Project\Tutorial subdirectory of the directory where you stored the **Lineback** program. Select the file **Tutorial.dat** in this directory to import the Tutorial project with all facilities defined. The **Import** dialog window is show in Figure 2.12.

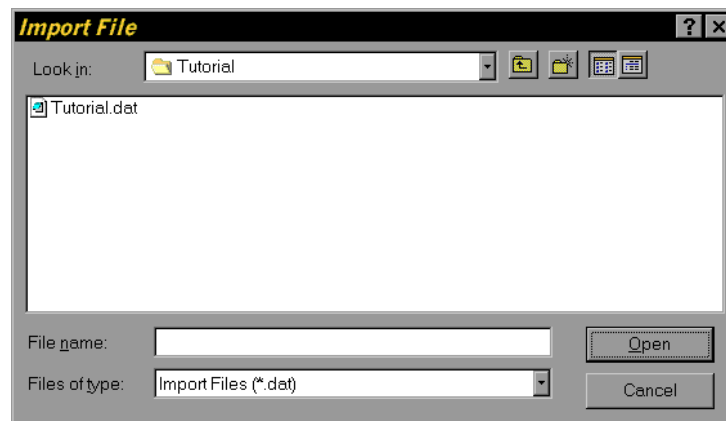


Figure 2.12. Import Dialog Window

The different views of the current project at this time are illustrated in Figure 2.13. The total delivery demand of all the customers is 830 and the total pickup supply of all the suppliers is 1030.

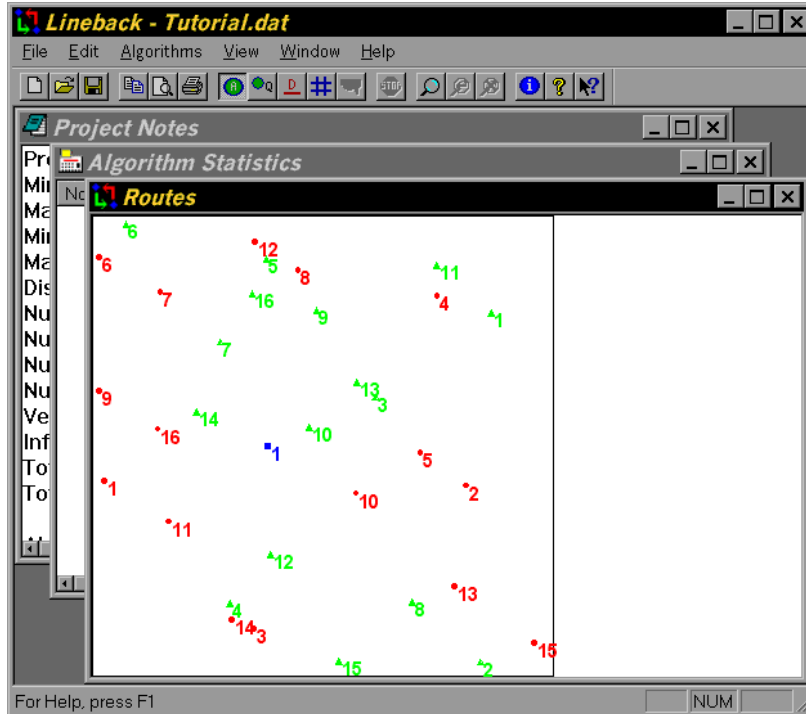


Figure 2.13. Tutorial Project Import Views

You can display the distances between each pair of facilities with the **All Distances** command of the Edit Links submenu of the **Edit** menu. The **All Distances** dialog window will be shown, which is illustrated in Figure 2.14. The distances are computed with the distance norm you specified when the project was created or imported. In this tutorial example the Euclidean or straight-line distance norm is used.

Select a Link to Edit

	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1		12532	7298	13205	11004	7750	6839	9927	3147	8740	2649	9806	12656	6531
2	12532		8864	6650	1944	14956	12529	9451	13109	3810	10357	11169	3506	9351
3	7298	8864		13177	8406	13946	12124	12523	9850	5887	4749	13423	7088	836
4	13205	6650	13177		5462	11782	9590	4895	12147	7396	12143	6584	10074	13295
5	11004	1944	8406	5462		13018	10589	7601	11323	2639	9032	9286	4770	8739
6	7750	14956	13946	11782	13018		2430	6906	4605	12079	9462	5423	16753	13367
7	6839	12529	12124	9590	10589	2430		4832	4010	9728	7961	3711	14400	11628
8	9927	9451	12523	4895	7601	6906	4832		8041	7980	9786	1796	12198	12327
9	3147	13109	9850	12147	11323	4605	4010	8041		9575	5141	7453	14027	9174
10	8740	3810	5887	7396	2639	12079	9728	7980	9575		6554	9394	4674	6149
11	2649	10357	4749	12143	9032	9462	7961	9786	5141	6554		10143	10123	4040
12	9806	11169	13423	6584	9286	5423	3711	1796	7453	9394	10143		13781	13129
13	12656	3506	7088	10074	4770	16753	14400	12198	14027	4674	10123	13781		7791
14	6531	9351	836	13295	8739	13367	11628	12327	9174	6149	4040	13129	7791	
15	15904	5938	9702	12483	7675	20121	17753	15271	17400	8047	13315	16926	3391	10493
16	2606	10843	7698	10711	9128	6271	4740	7327	2421	7217	3239	7299	11614	7098
1	14612	6068	13701	1935	5435	13688	11460	6820	13814	7814	13292	8519	9573	13931
2	14439	6125	7899	12767	7535	19233	16929	14949	16181	7241	11808	16507	2771	8713
3	9822	4411	9076	4109	2502	10692	8264	5116	9535	3411	8338	6784	7115	9187
4	6067	9146	1231	12856	8428	12803	11052	11779	8642	5810	3525	12561	7800	579

OK Cancel Help

Figure 2.14. Display All Distances Dialog Window

---

# Designing Vehicle Routes

## Setting Project Algorithm Parameters and Output Log File

Several parameters apply to all computer and interactive design algorithms.

### ***Setting the reporting level of detail and frequency of algorithm pauses***

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. Report level three shows the creation of all the routes without interruption. Report level four halts the algorithm after each route is created. Select the **Report Level** command from the Parameters submenu of the **Edit** menu and select level four. The Report Level dialog window is illustrated in Figure 2.15.

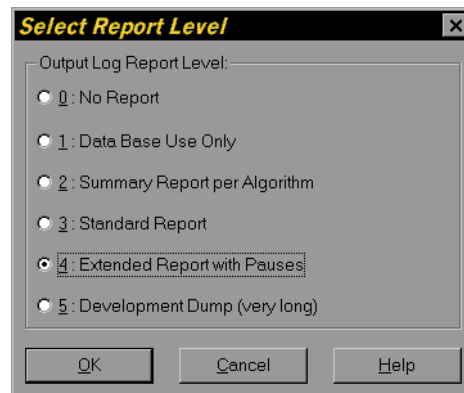


Figure 2.15. Report Level Dialog Window

Vehicle routing algorithms can be classified as construction, insertion, and improvement algorithms. The construction algorithms create a set of routes based solely on the facility locations and quantities. The insertion algorithms insert the unvisited facilities into a set of existing partial routes. The improvement algorithms attempt to reduce the total length of the routes by exchanging the sequence of facilities inside a route and by swapping facilities between routes. The **Lineback** program contains a number of vehicle routing algorithms, all of which are heuristic algorithms. An algorithm is a heuristic when it is not guaranteed to find the best or optimal solution, but rather determines a high quality solution.

### ***Specifying a Algorithm Log File***

**Lineback** will save the results of any algorithm in a log file, if such a file has been defined for the project. To specify a log file execute the **Output Log** command from the **File** menu. The **Output Log** dialog will be shown. The Output Log dialog is illustrated in Figure 2.18. Specify the name and location of the output log file and press **Ok** to accept the new values. The output log file has by default the extension *log* and is a pure ASCII file. The amount of information written to the output log file for each algorithm depends on the Report Level.

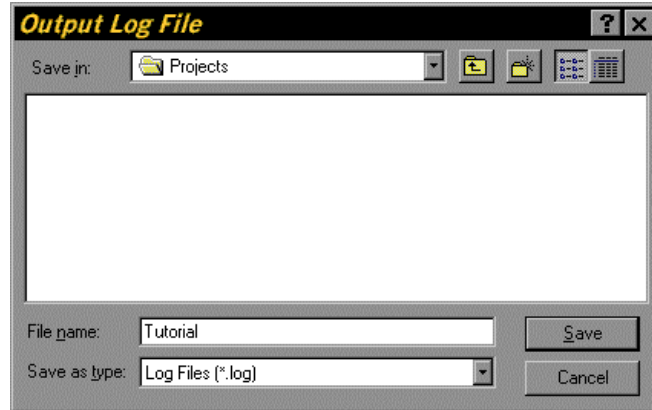


Figure 2.16. Open Output Log Dialog for the Tutorial Project

### **Specifying a maximum time limit for algorithm execution**

During the execution of algorithms, the program will periodically check if the algorithm has been computing longer than the allowable run time. If the limit has been exceeded, then the algorithm will be terminated and no results of the algorithms will be retained. To set the maximum amount of execution time for an algorithm, select the **Time Limit** command from the **Edit** menu. The **Time Limit** dialog window will be shown. The Time Limit dialog box is shown in Figure 2.17. Enter a limit of 120 seconds, which is more than sufficient for this small tutorial project. Press **OK** to accept the new value.

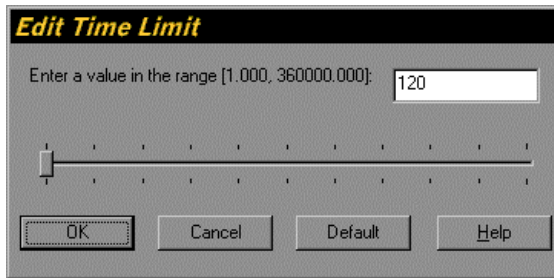


Figure 2.17. Time Limit Dialog

### Specifying a random number seed

Several algorithms must make random choices at certain points during their execution. For example, the simulated annealing algorithms select randomly two facilities to be tested for an exchange improvement. These random choices are made based on the next pseudo-random number. The sequence of these random numbers depends on the starting number of the sequence, which is called the seed. When started from the same state, algorithms will generate the same solution if it has been given the same seed. You can set the random number seed by executing the **Seed** command from the **Edit** menu. The **Seed** dialog will be shown. The Seed dialog is shown in Figure 2.18. Set the seed to 12345. Press **Ok** to accept the new value.

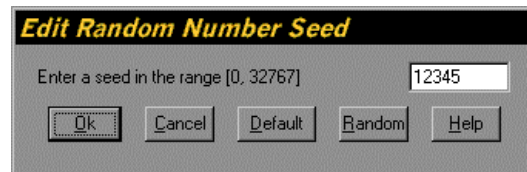


Figure 2.18. Seed Dialog

### Using construction algorithms

The Nearest Neighbor algorithm is a simple and intuitive construction algorithm. The Nearest Neighbor algorithm starts each route at the depot facility and then iteratively appends the nearest unvisited customer to the route. When appending the next customer would violate the truck capacity, the algorithm switches to iteratively appending the nearest unvisited supplier to the route. When appending the next supplier would violate truck capacity, the route returns to the depot. If unvisited customers or suppliers remain and the maximum number of routes has not been reached, then a new route is started.

Select the **Nearest Neighbor** command from the Construction submenu of the **Algorithms** menu. The algorithm will halt after the construction of every route. Press the **Continue** button to display the next route. The views after the Nearest Neighbor algorithm has created two routes are illustrated in Figure 2.19.

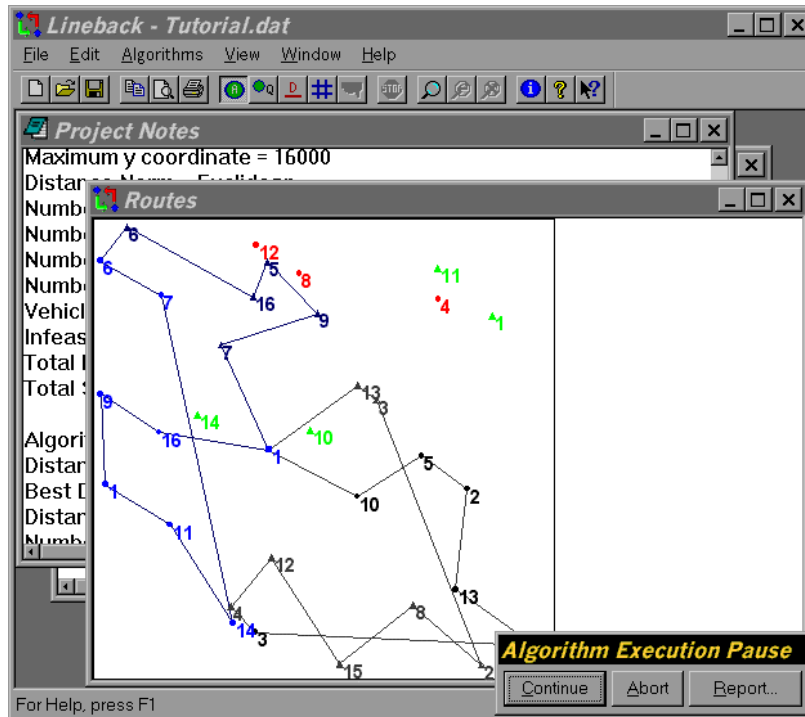


Figure 2.19. Nearest Neighbor Algorithm Pause

After the algorithm has complete successfully and has created a set of routes, the statistics of the routes are shown in the **Project Notes** and **Algorithm Statistics** views. The total distance of the routes, the number of routes, and the number of routed customers and suppliers is shown. The route cost is computed as the sum of the route distance and the infeasibility penalty. The infeasibility penalty itself is the product of any quantities on the routes over the truck capacity multiplied by the infeasibility penalty factor and the quantities of any unvisited facility multiplied by the infeasibility penalty factor. If all the routes are feasible and all facilities are visited then the route cost will be equal to the route distance. The Algorithm Statistics view is illustrated in Figure 2.20. Every time you execute an algorithm, a line with the new statistics will be shown in the Statistics view. The Statistics view also records and displays the best distance and the best cost that any algorithm has determined during this session.

Algorithm	Distance	Best Distance	Penalty	# Routes	# Routed	Cost	Best Cost
Nearest Neighbor	133529.52	133529.52	0.00	3	32	133529.52	133529.52

Figure 2.20. Algorithm Statistics View after Nearest Neighbor

## Using insertion algorithms

Insertion algorithms insert the facilities that are still unvisited into a set of existing partial routes. One hopes that the partial routes form a high quality initial route set. These initial partial routes are also called route skeletons or route primitives. The unvisited facilities are inserted so that the insertion cost is minimized. Insertion algorithms have to make two decisions: which facility to insert next and on which



link to insert this facility. Depending on the answer to those two questions, many different variations of insertion algorithms have been developed.

The Extreme Crossovers algorithm creates a number of partial routes with on each route a single customer and a single supplier. It matches the customer and supplier that have the largest savings if they were visited on single route instead of on separate routes. This is the same savings definition as used by Clarke and Wright in the Savings algorithm. The number of routes is equal to the minimum of the number of vehicles, the number of customers, and the number of suppliers.

$$s_{ij} = d_{i0} + d_{0j} - d_{ij} \quad i \in \mathbf{C}, j \in \mathbf{S}$$

The typical result of this algorithm is a number of routes with very acute aperture of the covering sector and with short crossover link far away from the depot.

The Cheapest Insertion algorithm inserts the remaining unvisited facilities into the currently existing partial routes. The algorithm finds for every unvisited facility the link on the routes with the smallest insertion penalty for that facility provided the facility insertion does not make the route infeasible. It then inserts next the facility with the smallest of these insertion penalties and then updates the insertion penalties. Let  $\mathbf{F}$  be the set of free facilities and  $\mathbf{T}$  be the set of links on the routes, then the facility  $k$  that is inserted next is given by

$$\min_{k \in \mathbf{F}} \left\{ \min_{(i,j) \in \mathbf{T}} (c_{ik} + c_{kj} - c_{ij}) \right\}$$

The algorithm thus executes a mini-min operation.

Select the **Extreme Crossovers** command from the Construction submenu of the **Algorithms** menu to create a set of partial routes. A set of five routes each with a single customer and supplier will be created. Then select the **Cheapest Insertion** command from the Insertion submenu of the **Algorithms** menu. The remaining unvisited facilities are inserted into these routes.

## Using improvement algorithms

Improvement algorithms attempt to reduce the length of the routes by changing the sequence of the facilities on a route and by exchanging facilities between routes, provided the exchanges do not make the routes infeasible. Improvement algorithms require the existence of a feasible set of routes. Depending on the number of facilities involved in the exchange and which exchange is selected, many different variations of improvement algorithms have been developed.

The Sweep algorithm creates a sweep tour by appending the facilities to the routes when they are traversed by a ray rotating around the depot. This construction algorithm is extended in a straightforward manner to the case of vehicle routing with backhauling. When a customer or supplier facility is traversed by the rotating ray, it is inserted between the most recently appended customer and supplier facility on that route. When the inserting either the next customer or supplier would violate truck capacity, then the route is terminated. If unvisited customers or suppliers remain and the maximum number of routes has not been reached, then a new route is started.

Select the **Sweep** algorithm from the Construction submenu of the **Algorithms** menu. Attempt to improve these routes with feasible intra-route exchanges by selecting the **Three Exchange** command of the Improvement submenu. Then attempt to improve these routes by feasible inter-route swaps by selecting the **Swap** command of the Improvement submenu. Finally, attempt again to improve these routes with feasible intra-route exchanges by selecting the **Three Exchange** command of the Improvement submenu.

## Manually constructing a set of routes

You can also manually construct a complete set or an initial set of partial routes. The **Manual** command allows you to construct a set of feasible routes in an interactive manner. When you execute the **Manual** command, the cursor changes to a vertical arrow indicating that you can now select facilities to be added to the routes and the algorithm **Abort** button on the tool bar is enabled. You start by clicking the left button of the mouse when it is over the depot and extend the routes by clicking the left button of the mouse when the cursor is over an unvisited or free facility. If you click for the second time on the depot, then the route is closed. Once a supplier or backhaul facility has been added to the route, you can not longer add customer or linehaul facilities to the current route. If adding the customer or supplier would violate the vehicle capacity, then that facility is not added to the route. The algorithm terminates when either all facilities have been placed on a route or the number of routes is equal to the number of vehicles. Not all facilities have to be included on the routes when the algorithm terminates, in other word, partial routes or route primitives are allowed.

If you click again on the last facility of the current route, then this facility is removed from the route and becomes free again and the next to last facility on the route becomes the last facility. If the depot is the only facility on the route and is clicked again, then the complete route is removed. Clicking then again on the depot will make the depot the starting facility of a new route. If the removed facility was the first backhaul supplier on the route, then you can add again linehaul customers to the route.

In general, clicking on an unvisited facility adds this facility to the route if this is a feasible extension of the route and clicking on the last facility of the route removes this facility from the route. Clicking on intermediate facilities of the route has no effect.

You can stop the process of manually creating routes by pressing the **Escape** key. If the last facility on the current route is the depot then the algorithm terminates and route statistics are computed. If the last facility was not the depot, then the current route is first closed by appending the depot facility and then the route statistics are computed.

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

Select the **Manual Build** command from the Construction submenu of the **Algorithms** menu and create a route with a number of customers and suppliers. The statistics for the route you are building are shown in the Add Report dialog window. The views while building the routes manually are illustrated in Figure 2.21.

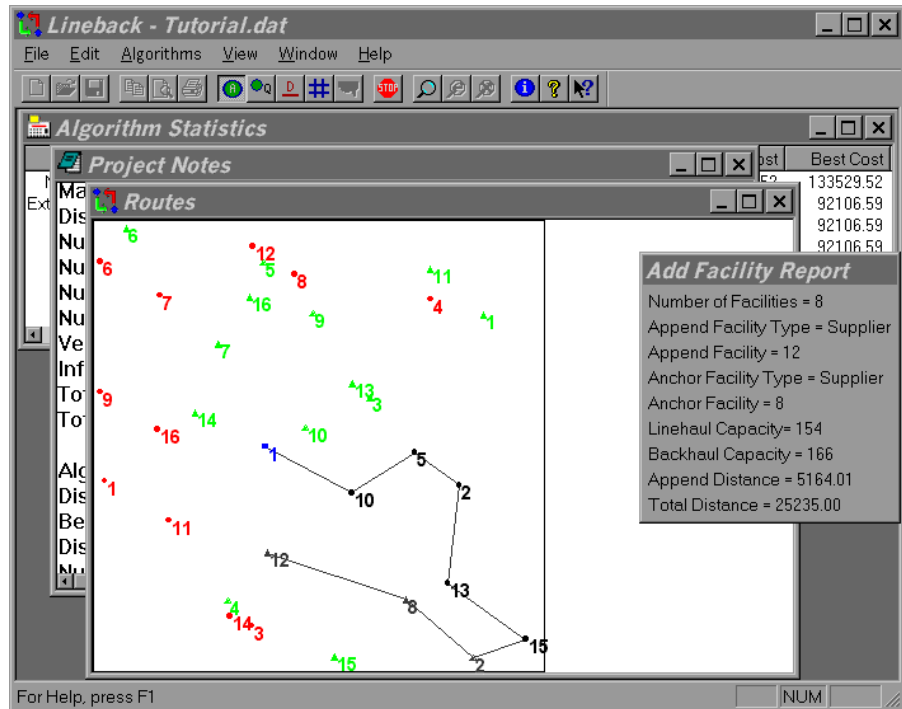


Figure 2.21. Manual Build Views

If you have manually created a set of partial routes, then you can select next one of the insertion algorithms to insert the remaining unvisited facilities.

## Sensitivity Analysis and Evaluating Routes

You can manually modify the data with the commands of the Edit Facilities and Edit Links submenus of the **Edit** menu. To evaluate the impact of you changes you use the **Evaluate** command of the **Algorithms** menu. The Evaluate command computes the total distance of the routes and the total cost of the routes, which includes the distance and the penalties for unvisited facilities, if routes have 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 quantities by you or after you have dragged a facility to a new location. The Evaluate command does not create a new set of routes, but rather computes the cost of the current set of routes based on the current distances and quantities. The command also displays in a dialog window the total distance, the total cost, and the number of linehaul and backhaul facilities and quantities on each route. The Evaluate dialog window is illustrated in Figure 2.22.

The route cost is computed as the sum of the route distance and the infeasibility penalty. The infeasibility penalty itself is the product of any quantities on the routes over the truck capacity multiplied by the infeasibility penalty factor and the quantities of any unvisited facility multiplied by the infeasibility penalty factor. If all the routes are feasible and all facilities are visited then the route cost will be equal to the route distance. You can change the infeasibility penalty factor with the **Infeasibility Penalty** command on the Parameters submenu of the **Edit** menu. A more positive value of the penalty factor tends to reduce the number and size of infeasibilities such as unvisited facilities and violated truck capacities. The penalty is set by default to 1000 when the project is created.

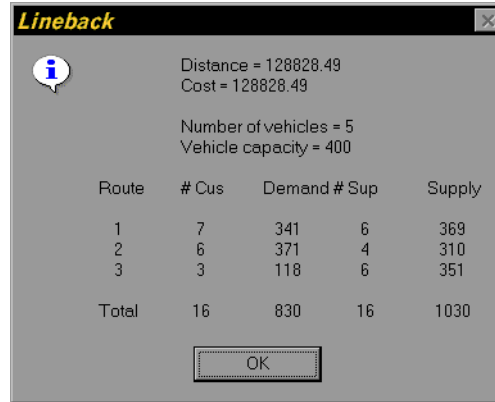


Figure 2.22. Evaluate Dialog Window

The **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 statistics of the different route algorithms that you have executed so far are illustrated in Figure 2.23. The corresponding **Routes** view is shown in Figure 2.24.

Algorithm	Distance	Best Distance	Penalty	# Routes	# Routed	Cost	Best Cost
Nearest Neighbor	133529.52	133529.52	0.00	3	32	133529.52	133529.52
Extreme Crossovers	92106.59	92106.59	0.00	5	10	92106.59	92106.59
Sweep	153551.28	92106.59	66.71	3	32	153551.28	92106.59
Three Exchange	128828.49	92106.59	39.87	3	32	128828.49	92106.59
Swap Facility	128828.49	92106.59	39.87	3	32	128828.49	92106.59
Three Exchange	128828.49	92106.59	39.87	3	32	128828.49	92106.59
Evaluate	128828.49	92106.59	39.87	3	32	128828.49	92106.59

Figure 2.23. Algorithm Statistics View for the Tutorial Project

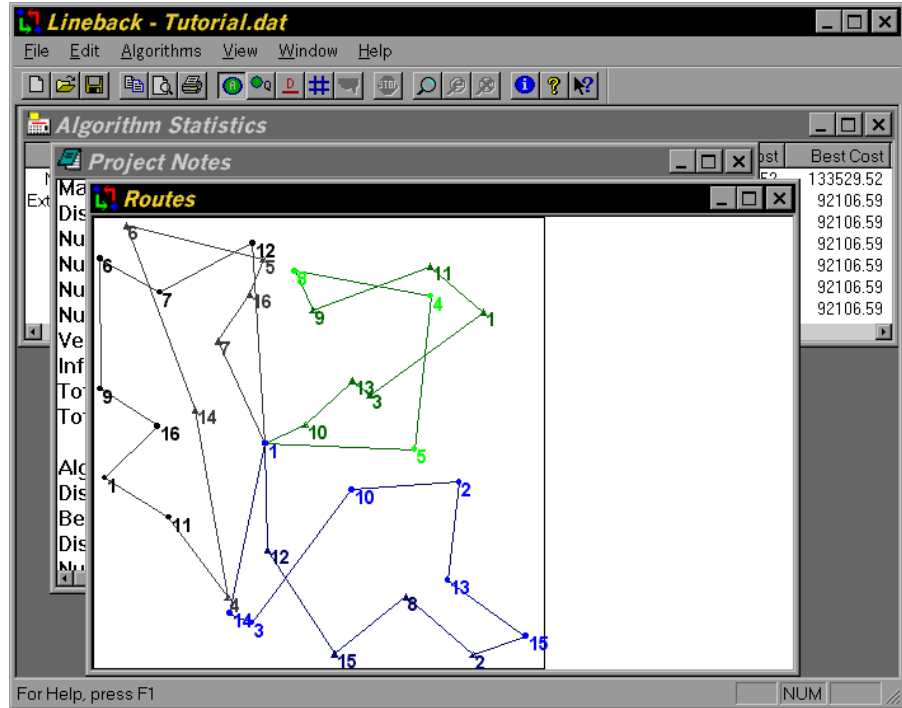


Figure 2.24. Final Views for the Tutorial Project

## Using Lineback Routes in Other Windows Programs

The results of the **Lineback** 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 **Lineback** views. First, we will print the **Lineback** views to any installed printer, and then we will copy and paste **Lineback** 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 Lineback Views

Select the printer on which you wish to print the **Lineback** 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 2.25.

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

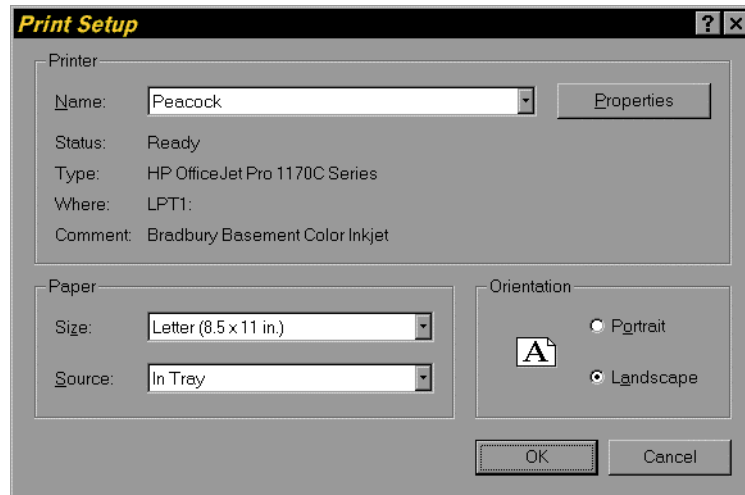


Figure 2.25. *Print Setup Dialog Box*

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 2.26.

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

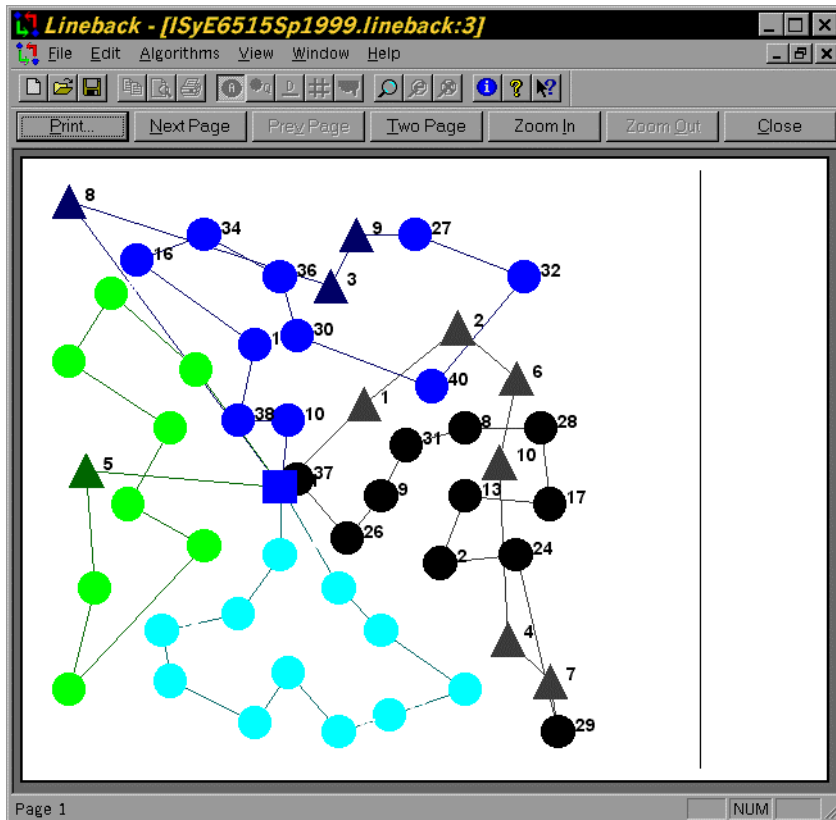


Figure 2.26. Print Preview Window

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 **Routes** views of large projects, printing such a complex view might require substantial processing times. The **Print** dialog box is illustrated in Figure 2.27.

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

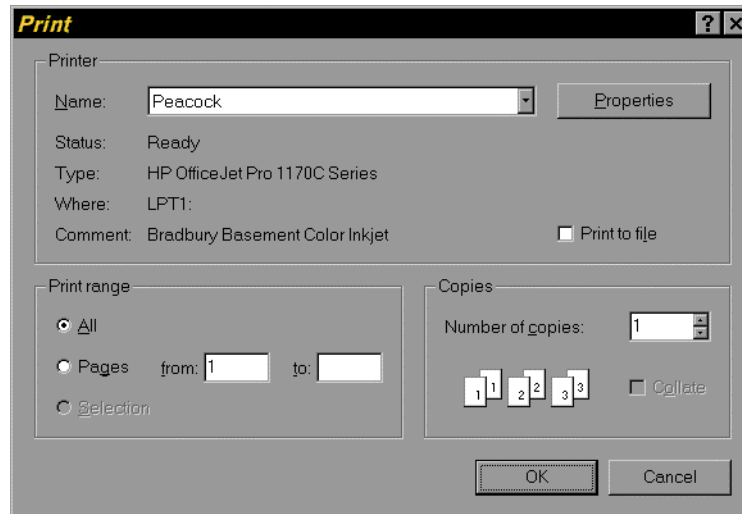


Figure 2.27. Print Dialog Box

## Copying and Pasting Lineback Views with the Clipboard

To paste the contents of one of the **Lineback** views in another application, select the view that you wish to paste then execute the **Copy** command from the **Edit** menu. For the **Routes** views, the section of the routes 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 **Lineback** program, the name of the project and the date the view was copied to the clipboard.

Select in turn the **Notes**, **Statistics**, and **Routes** 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 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 **Lineback** views, maximize this view and then capture the view with the **Alt-PrintScreen** command. For all views, the section of the routes or text currently displayed in the view will be copied in graphical format to the clipboard. No header indicating the **Lineback** 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 **Lineback** window as it is currently displayed. The same technique can also be used to capture images of the various dialog boxed used by **Lineback** 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 **Route** 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 Photo Editor.



### ***Sending the saved project as an electronic mail attachment***

You can send the current routing 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 correspond to the Small case and are listed in the appendix Sample Projects.

This page left intentionally blank.

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

The project name can be changed 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 of the name 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 project name.



### *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 the distance between two points and the 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 orthogonal projection is suitable only for small areas. 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, Western Europe, and areas at intermediate north latitudes.

In the following projection formulas, all longitudes and latitudes are expressed in degrees. The origin longitude and latitude are denoted by  $lon_0$  and  $lat_0$ . All points on the origin longitude have  $x = 0$ . The radius of the earth is denoted by  $R$  and is equal to 3955.8 miles or 6366.2 kilometers. Finally,  $deg\_per\_rad$  is the conversion factor that corresponds to the number of degrees per radians.

### Mercator Projection

$$x = \frac{R \cdot (lon - lon_0)}{deg\_per\_rad}$$

$$y = R \cdot \ln \left( \tan \left( \frac{45 + (lat/2)}{deg\_per\_rad} \right) \right)$$

### Albers Projection

The latitudes of the standard parallels are denoted by  $lat_1$  and  $lat_2$ . The standard parallels for the continental United States are typically 33 and 45 degrees North.

A number of calculations are identical for all the points in a particular Albers projection. These calculations have to be made only once for the projection.

$$n = \frac{\sin(lat_1) + \sin(lat_2)}{2}$$

$$C = \cos^2(lat_1) + 2n \sin(lat_1)$$

$$r_0 = \frac{\sqrt{C - 2n \sin(lat_0)}}{n}$$

The following formulas change with the longitude and latitude of the points to be projected.

$$q = n(lon - lon_0)$$

$$r = \frac{\sqrt{C - 2n \sin(lat)}}{n}$$

$$y = r_0 - r \cos q$$

$$x = r \sin q$$

### 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 map projections are used. Longitudes and latitudes are displayed and edited in three different fields, corresponding to the degrees, minutes, and seconds of the longitude and latitude. However, longitudes and latitudes are stored internally in a packed integer format. 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.

### ***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 west longitude is the rightmost boundary value of valid coordinates when the Mercator or Albers map projections are used. Longitudes and latitudes are displayed and edited in three different fields, corresponding to the degrees, minutes, and seconds of the longitude and latitude. However, longitudes and latitudes are stored internally in a packed integer format. 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 map projections are used. Longitudes and latitudes are displayed and edited in three different fields, corresponding to the degrees, minutes, and seconds of the longitude and latitude. However, longitudes and latitudes are stored internally in a packed integer format. 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 map projections are used. Longitudes and latitudes are displayed and edited in three different fields, corresponding to the degrees, minutes, and seconds of the longitude and latitude. However, longitudes and latitudes are stored internally in a packed integer format. 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 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*. The map file name must conform to the requirements for long file names in Windows. 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.

### ***Distance Norm***

When new facilities are located the distance between the various facilities is computed with a distance norm. For the orthogonal map projection, the available distance norms are rectilinear, Euclidean or Chebyshev. For the Mercator and Albers map projections, the great circle norm is used. The distance norm used with the orthogonal map projection can be changed with the **Distance Norm** command of the **Edit** menu.

### ***Number of Vehicles***

The number of vehicles available to execute routes. It is assumed that each vehicle can execute at most one route, so the number of vehicles is an upper bound on the number of routes. The number of vehicles can be changed with the **Number of Vehicles** command of the **Edit** menu.

### ***Vehicle Capacity***

The capacity of a single vehicle is expressed in units consistent with the demand and supply quantities of customers and suppliers. All vehicles are assumed to have the same capacity. The vehicle capacity can be changed with the **Vehicle Capacity** command of the **Edit** menu.

### ***Infeasibility Penalty***

The infeasibility penalty factor is multiplied by the quantities of any unvisited facilities and by the quantity on each route that exceeds the vehicle capacity to compute the total infeasibility penalty. The total infeasibility penalty is then added to the total route distance to yield the total route cost. The infeasibility penalty factor must a non-negative number. A more positive infeasibility factor induces the algorithms to create more feasible routes. The infeasibility penalty factor can be changed with the **Infeasibility Penalty** command of the **Edit** menu.

### ***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. 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 route construction or improvement algorithms often need to make a random choice among several alternative departments. 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 vehicle routes. 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.

### ***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 only used to terminate the exchange improvement algorithms if they have exceeded the time limit after one complete iteration, i.e. after all possible 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 routes. The higher the maximum number of

replications, the more likely a good set of vehicle routes will be constructed. Of course, more replications require more computation time. The default number of the maximum number of replications is equal to 20. In versions of **Lineback** before 4.0, the name of this field was **Number of Iterations**.

## Map Data

In **Lineback**, 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. **Lineback** 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 is 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 <u>all</u> 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 packed integer.

For example the file **atlanta.map** contains the 11 map objects with a total of 473 points. The first object contains 112 points, has a boundary color of OLIVE, and an area fill color of OLIVE. 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 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 if the Mercator or Albers map projections are used for the project. Longitudes and latitudes are displayed and edited in three different fields, corresponding to the degrees, minutes, and seconds of the longitude and latitude. The longitude is stored in the packed integer "(+/-)dddmmss" 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 if the Mercator or Albers map projections are used for the project. Longitudes and latitudes are displayed and edited in three different fields, corresponding to the degrees, minutes, and seconds of the longitude and latitude. The latitude is stored in the packed integer "(+/-)dddmmss" 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 **Lineback** 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 **Lineback** can be displayed with one of the following standard geometrical shapes.

Table 3.4. Valid Display Symbol Names

CIRCLE
SQUARE
TRIANGLE
DIAMOND
HEXAGON

## Facility Data

For the following data items, each facility can have a different individual value.

### Label

A facility label must consist of 1 to 7 characters. Only letters, numbers, and the underscore character are allowed. A facility has to have a unique label among all facilities of that type. You can have only one customer with label 1, but you can have a customer and a supplier both with label 1. The label is a required data item; i.e., it cannot be empty. The label will be used to identify the facility in all screen dialog and view windows.

### Name

The facility name must consist of 1 to 63 characters. Only letters, numbers, the underscore and space characters are allowed. The name is a required data item; i.e., it cannot be empty.

If the facility 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 facility name.

### Latitude

The location of the facility is given by its latitude and longitude or x and y coordinate, depending on the map projection. Longitudes and latitudes are displayed



and edited in three different fields, corresponding to the degrees, minutes, and seconds of the longitude and latitude. The latitude is stored in the packed integer (+/-)dddmmss format. For example, the longitude 23°20'15" W is represented as -0232015, and the latitude 35°20'43" N as 0352043. Latitude and longitude in the packed integer format are described in more detail in the section on **Common Data Items**. The latitude or the y coordinate of a facility must fall inside the world latitude or y coordinate boundaries, specified when the project was created.

### ***Longitude***

The location of the facility is given by its latitude and longitude or x and y coordinate, depending on the map projection. Longitudes and latitudes are displayed and edited in three different fields, corresponding to the degrees, minutes, and seconds of the longitude and latitude. The longitude is stored in the packed integer (+/-)dddmmss format. For example, the longitude 23°20'15" W is represented as -0232015, and the latitude 35°20'43" N as 0352043. Latitude and longitude in the packed integer format are described in more detail in the section on **Common Data Items**. The longitude or the x coordinate of a facility must fall inside the world longitude or x coordinate boundaries, specified when the project was created.

### ***Quantity***

Each customer has a delivery quantity and each supplier has a pickup quantity that must be carried on the vehicle. The quantity must be a non-negative integer less than or equal to the vehicle capacity. The quantity data field for depot facilities is not used and typically set to zero.

### ***Display Symbol***

The geometrical symbol or shape used to display the facility in the **Routes** views. A list of all valid symbol names is given in the **Display Symbols** section of the **Common Data Items**.

### ***Display Color***

The color used to display the facility symbol in the **Routes** views. A list of valid color names is given in the **Color** section of the **Common Data Items**.

### ***Display Size***

The size of the geometric symbol displayed in the **Routes** views.

---

## **Importing Data Files from Previous Versions**

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

A project is completely described by five files. A sixth file contains the optional map data. The first file is the **Project Data** file that holds all the scalar information about this project. The second through fourth files are the **Facility Data** files that contain all the data for the customer, depot, and supplier facilities, respectively. The optional fifth file contains all the data for the current set of routes. The fifth file is not required if no routes are currently defined in the project. The sixth file contains the data for the optional background map and is not required.

## 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 **Lineback** 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 **Lineback** 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:

```
[project_name]          CASEO1
[minimum_x_coordinate]  0
[maximum_x_coordinate] 32000
[minimum_y_coordinate]  0
[maximum_y_coordinate] 36000
[number_of_vehicles]    10
[vehicle_capacity]      5700
[infeasibility_penalty] 1000.000000
[distance_norm]         Euclidean
[number_of_customers]   100
[customer_file_name]    CaseO.cus
[number_of_suppliers]   100
[supplier_file_name]    CaseO.sup
[number_of_depots]      1
[depot_file_name]       CaseO.dep
[number_of_routes]      10
[route_file_name]       CaseO1.rou
[report_level]          3
[seed]                  12345
[number_of_iterations]  20
[time_limit]            30.000000
```

### ***Project 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 **Lineback** or higher the size of this data field has been expanded and space characters are allowed, however the smaller size and no space characters should be used in the import files.

### ***Customer File Name***

### ***Supplier File Name***

### ***Depot File Name***

### ***Route File Name***

The next items are the names of the customer, supplier, depot, and route data files, respectively. They contain all the data for facilities of that respective type or for the routes. The file names must satisfy the Windows long file name conventions. A path must precede the file name if it is not in the current directory or in the data path. Long file names may be used, but the file name and the path cannot contain spaces. An example of a customer data file is the file *Case5180.sup* that holds the supplier

information. This file is included in the appendix and on the distribution disk. The files must be created outside the **Lineback** program with an editor capable of generating pure ASCII files. The extensions *cus*, *sup*, *dep* and *rou* are solely a convention, other extensions may be used.

### **Seed**

The tour construction or improvement algorithms often need to make a random choice among several alternative departments. 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 vehicle routes. 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 only used to terminate the exchange algorithms if they have exceeded the time limit after one complete iteration, i.e. after all possible 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 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 set of vehicle routes will be constructed. Of course, more replications require more computation time. The default number of the maximum number of replications is equal to 20. In version 4.0 or higher of **Lineback**, 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.

## **Facility Data Files**

The Facility Data files 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 **Lineback** program, so special care should be taken when using a word processor to generate a pure ASCII file. The facility data files also should not contain any blank lines. Once an input files have been created, they can be used repeatedly by the **Lineback** program.

All facility data files have the same structure. The customer, depot, and supplier facility data are contained in a file with standard extension *cus*, *dep*, and *sup*,

respectively. They contain all the data for a single facility on a single line, which contains eight data fields. The fields contain the facility label, x coordinate, y coordinate, quantity, display shape, display size, display color, and facility name, respectively. A typical facility data line is shown below.

```
1      21524  24879  483   CIRCLE      250   RED   NONE
```

## Routes Data File

The Routes Data file contains a block of data lines for each currently defined route. The first line of each route block contains the number of customers on the route, the number of suppliers on the route, and the display color of the route. The lines after that contain the facilities on this route. Each route has to start with the depot facility and has to terminate with the depot facility. After the initial depot facility, all the customer facilities are listed followed by all the supplier facilities. A route may not have any customers or not any suppliers, but must include at least one facility different from the depot. For each facility the facility type and the facility label are given. The number of lines for the customer and supplier facilities has to match the number of customers and suppliers specified on the first line. A typical set of data lines for a single route is shown below.

```
3      1      BROWN
depot      D1
customer   20
customer   8
customer   12
supplier   5
depot      D1
```

# Chapter 4. Design Algorithms

---

## Introduction

### 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. The vehicle routing construction algorithms create a set of routes based solely on the facility locations and quantities. Algorithms are called improving (improvement algorithms) if they require an initial feasible solution and attempt to improve the quality of this solution. The vehicle routing improvement algorithms attempt to reduce the total length of the routes by exchange the sequence of facilities inside a route and by swapping facilities between routes.

The vehicle routing insertion algorithms insert the unvisited facilities into a set of existing partial routes. The **Lineback** program contains a number of vehicle routing algorithms, all of which are heuristic algorithms.

#### *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 or sets of facilities, solve the resulting problem to optimality, and then attempt to add the ignored constraints or facilities while maintaining optimality of the current solution with respect to the current constraints.

---

# Construction Algorithms

## Nearest Neighbor

The Nearest Neighbor algorithm starts each route at the depot facility and then iteratively appends the nearest unvisited customer to the route. When appending the next customer would violate the truck capacity, the algorithm switches to iteratively appending the nearest unvisited supplier to the route. When appending the next supplier would violate truck capacity, then the route returns to the depot. If unvisited customers or suppliers remain and the maximum number of routes has not been reached, then a new route is started.

This algorithm was originally described by Rosenkrantz et al. (1977) for the traveling salesman problem and is extended in a straightforward manner to the case of linehaul-backhaul vehicle routing.

## Sweep

The Sweep algorithm creates a sweep route by appending the points to the tour when they are traversed by a ray rotating around the depot. The relative polar coordinates of each point with respect to the depot are determined and the points are inserted in the route by increasing polar angle. The algorithm was first described by Gillet and Miller (1974) for the vehicle routing problem. The starting angle of the ray is an algorithm parameter.

This algorithm is extended in a straightforward manner to the case of vehicle routing with backhauls. When a customer or supplier facility is traversed by the rotating ray, it is inserted between the most recently appended customer and supplier facility on that route. When the inserting either the next customer or supplier would violate truck capacity, then the route is terminated. If unvisited customers or suppliers remain and the maximum number of routes has not been reached, then a new route is started.

## Clarke and Wright Savings Heuristic

Clarke and Wright (1964) developed a construction procedure that extends a partial route or route primitive at its two end points. Conceptually the algorithm defines the depot as the base point and constructs an Eulerian tour that visits each of the other customer or supplier points and the returns to the depot 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.

$$\max_{i,j} \{s_{ij} = c_{i0} + c_{0j} - c_{ij}\} \quad (4.1)$$



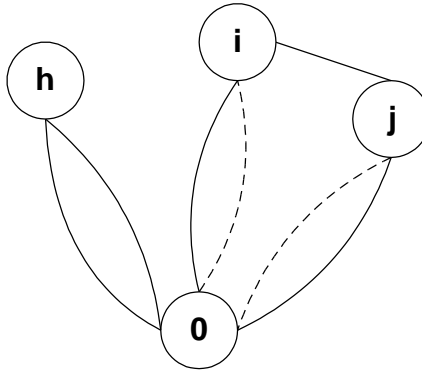


Figure 4.1. Clarke and Wright Tour Extension Illustration

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.

$$\max_{i,j} \left\{ \max_h \{ s_{ih} = c_{i0} + c_{h0} - c_{ih} \} \right\} \quad (4.2)$$

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

1. Start a new tour at the depot
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

This algorithm was adapted to the vehicle routing problem with backhauls by Deif and Bodin (1984). Once a partial tour contains both customer and suppliers, customers can only be added between the customer end point and the depot and suppliers can only be added between the supplier end point and the depot. Deif and Bodin delayed the crossover shortcut between customers and suppliers by reducing the savings of establishing a short cut between a customer and supplier by the product of a savings penalty factor  $\alpha$  multiplied by the largest savings of any points  $S$ . This savings penalty factor is an algorithm parameter set by the user. Deif and Bodin found that a good value for the savings penalty factor is 0.25

$$\max_{i,j} \left\{ \begin{array}{ll} s_{ij} = c_{i0} + c_{0j} - c_{ij} & \text{if } i, j \in \mathbf{C} \text{ or } i, j \in \mathbf{S} \\ s_{ij} = c_{i0} + c_{0j} - c_{ij} - \alpha S & \text{if } i \in \mathbf{C}, j \in \mathbf{S} \end{array} \right\} \quad (4.3)$$

---

## Partial Routes Construction Algorithms

The objective of partial routes construction algorithms is to create a number of routes that may not visit all the facilities but that are of high quality. The remaining unvisited facilities are then inserted into these routes. The partial routes are also called route skeletons or route primitives. Examples of similar partial tours for the Traveling Salesman problem are the convex hull and extreme point quadrilateral.

## Extreme Crossovers

The Extreme Crossovers algorithm creates a number of partial routes with on each route a single customer and a single supplier. It matches the customer and supplier that have the largest savings if they were visited on single route instead of on separate routes. This is the same savings definition as used by Clarke and Wright in the Savings algorithm. The number of routes is equal to the minimum of the number of vehicles, the number of customers, and the number of suppliers.

$$s_{ij} = d_{i0} + d_{0j} - d_{ij} \quad i \in \mathbf{C}, j \in \mathbf{S} \quad (4.4)$$

The typical result of this algorithm is a number of routes with very acute aperture of the covering sector and with short crossover link far away from the depot.

---

## Insertion Algorithms

Insertion algorithms insert the facilities that are still unvisited into a set of existing partial routes. One hopes that the partial routes form a high quality route set. These initial partial routes are also called route skeletons and route primitives. The unvisited facilities are inserted so that the insertion cost is minimized. They have to make two decisions: which facility to insert next and on which link to insert this facility. Depending on the answer to those two questions, many different variations of insertion algorithms have been developed.

### Cheapest Insertion

The Cheapest Insertion algorithm inserts the remaining free facilities into the currently existing partial routes. The algorithm finds for every free facility the link on the routes with the smallest insertion penalty for that facility, provided the insertion would not make the route infeasible. It then inserts next the facility with the smallest of these insertion penalties and then updates the insertion penalties. Let  $F$  be the set of free facilities and  $T$  be the set of links on the routes, then the point  $k$  that is inserted next is given by

$$\min_{k \in F} \left\{ \min_{(i,j) \in T} (c_{ik} + c_{kj} - c_{ij}) \right\}$$

The algorithm thus executes a mini-min operation.

### Priciest Insertion

The Priciest Insertion algorithm inserts the remaining free facilities into the currently existing partial routes. The algorithm finds for every free facility the link on the routes with the smallest insertion penalty for that facility, provided the insertion would not make the route infeasible. It then inserts next the facility with the largest of these insertion penalties and then updates the insertion penalties. Let  $F$  be the set of free facilities and  $T$  be the set of links on the routes, then the point  $k$  that is inserted next is given by

$$\max_{k \in F} \left\{ \min_{(i,j) \in T} (c_{ik} + c_{kj} - c_{ij}) \right\}$$

The algorithm thus executes a maxi-min operation.

---

# Improvement Algorithms

## Introduction

Improvement algorithms attempt to reduce the length of the routes by changing the sequence of the facilities on a route and by exchanging facilities between routes, provided the exchanges do not make the routes infeasible. Improvement algorithms require the existence of a feasible set of routes. Depending on the number of facilities involved in the exchange and which exchange is selected, many different variations of improvement algorithms have been developed.

## *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 a tour with the lowest possible length, the categories are

- First Descend
- Steepest Descend
- Simulated Annealing
- Tabu Search

### First Descend

All possible edge exchanges that can result in new routes are examined in a structured way until an exchange is found that reduces the route 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 length reduction is executed. The process terminates when no further exchanges can be found that yield a cost reduction.

### Steepest Descend

All possible edge exchanges that can result in new routes are examined in a structured way and the exchange that yielded the largest reduction in the route length is retained. If this exchange reduces the route 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 set of routes into specific final set of routes. Since they are heuristics, this final set may not be of high quality. To remedy this problem, a probabilistic 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 facilities 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  $\Delta$  and the temperature  $T$ .  $T$  is a search control parameter that is systematically reduced during the algorithm execution.

$$\begin{aligned}
 \text{if } \Delta < 0 \quad P[\text{Exch}] &= 1 \\
 \text{if } \Delta \geq 0 \quad P[\text{Exch}] &= e^{-\Delta/T}
 \end{aligned}
 \tag{4.5}$$

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 descend 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 route set if run from the same initial route set 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

The 2-opt Exchange is an intra-route edge-exchange improvement procedure that operates on a single route at the time. It attempts the change the sequence of the facilities in a route so that the route length is reduced by removing two links of the route and replacing them with two other links that connect the route again. All pairs of links that would yield a feasible exchange are tested and the exchange with the largest positive savings is executed. The algorithm thus belongs to the class of steepest descend improvement algorithms. An exchange is infeasible if it would violate the precedence constraint that all deliveries to customers have to be made before any pickup from a supplier can be made. A two exchange is illustrated in Figure 4.2.

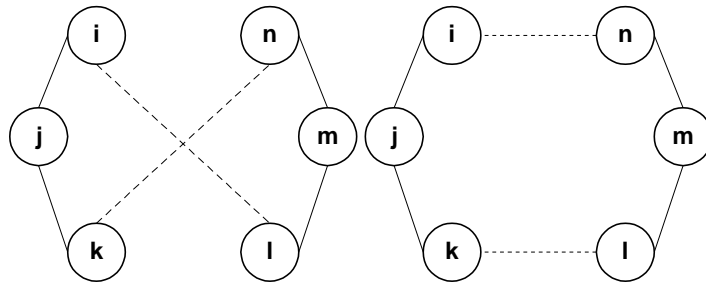


Figure 4.2. Two Exchange Improvement Illustration

## 3-Opt Exchange

The 3-opt Exchange is an intra-route edge-exchange improvement procedure that operates on a single route at the time. It attempts the change the sequence of the facilities in a route so that the route length is reduced by removing three links of the route and replacing them with three other links that connect the route again. All triplets of links that would yield a feasible exchange are tested and the exchange with the largest positive savings is executed. The algorithm thus belongs to the class of steepest descend improvement algorithms. An exchange is infeasible if it would violate the precedence constraint that all deliveries to customers have to be made before any pickup from a supplier can be made. A three exchange is illustrated in Figure 4.3.

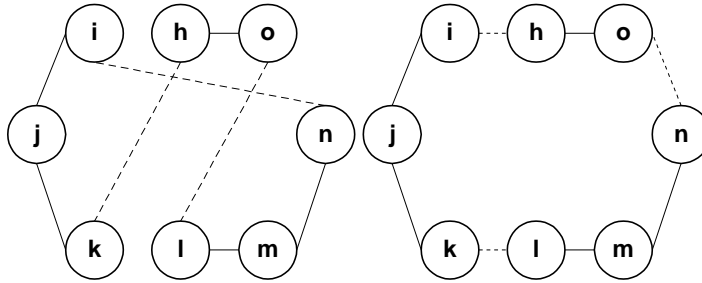


Figure 4.3. Three Exchange Improvement Illustration

## Or-Opt Exchange

The Or-opt Exchange is an intra-route edge-exchange improvement procedure that operates on a single route at the time. It attempts to change the sequence of the facilities in a route so that the route length is reduced by removing a section or chain of the route and inserting this chain in a different place on the route. Moving a chain involves deleting three links of the route and replacing them with three other links that connect the route again. The number of facilities on the chain is called the length of the chain and is a parameter that you can specify. All chains of a certain length that would yield a feasible exchange are tested and the exchange with the largest positive savings is executed. The algorithm thus belongs to the class of steepest descend improvement algorithms. Since the Or-exchange involves removing three links and substituting them with three other links, all chain exchanges of all lengths will also be tested by the three-exchange procedure. An exchange is infeasible if it would violate the precedence constraint that all deliveries to customers have to be made before any pickup from a supplier can be made. A chain exchange with chain length of two is illustrated in Figure 4.4.

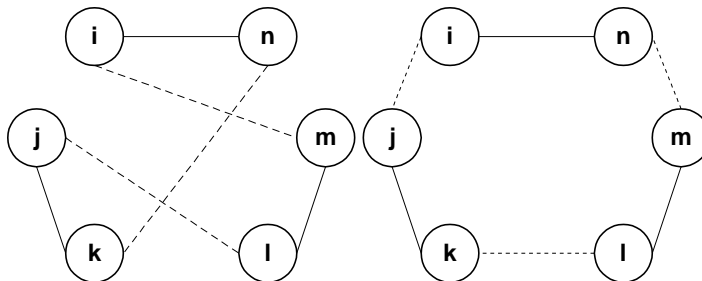


Figure 4.4. Or (Chain) Exchange Improvement Illustration

## Move Facility

The Move Facility procedure attempts to reduce the combined length of all the routes by moving a single facility from one route to another, provided the resulting routes remain feasible. A route can become infeasible if either the linehaul or backhaul quantity exceeds the truck capacity or if the precedence constraint is violated that states that all deliveries to customers have to be made before any pickup from a supplier can be made. All move exchanges are tested and the move exchange with the largest positive savings is executed. The algorithm thus belongs to the class of steepest descend improvement algorithms.

## Swap Facility

The Swap Facility procedure attempts to reduce the combined length of all the routes by moving one facility from one route to another and simultaneously move one facility from the second route to the first route, provided the resulting routes remain feasible. A route can become infeasible if either the linehaul or backhaul quantity exceeds the truck capacity or if the precedence constraint is violated that states that all deliveries to customers have to be made before any pickup from a supplier can be made. All swap exchanges are tested and the swap exchange with the largest positive savings is executed. The algorithm thus belongs to the class of steepest descend improvement algorithms.

---

## Evaluation Algorithms

### Evaluate

The Evaluate algorithm computes the length and the cost of the current routes. The Evaluate algorithm does not create new routes, but rather computes the length and the cost of the current routes based on the current distances, supply and demand data, number of vehicles and vehicle capacity, and infeasibility penalty. This algorithm is most frequently used to evaluate the impact of the changes you have made after you have edited interactively the distances, the facility, or the vehicle data.

# Chapter 5. Command Reference

---

## Menu Overview

An overview of the **Lineback** 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. **Lineback** Menu Structure

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



Figure 5.2. **Lineback** Toolbar

---

## File Menu

The **Lineback** 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 vehicle routing 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 facilities or routes. If you press **Cancel**, no new project will be created.

### New Project Shortcuts

Toolbar:   
Keys: CTRL+N

### Save Changes Dialog Window

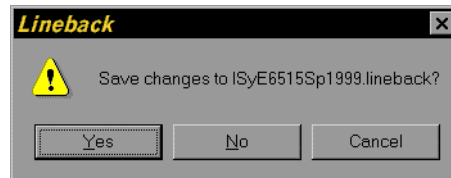


Figure 5.3. Save Changes Dialog Window

### New Project Dialog Window

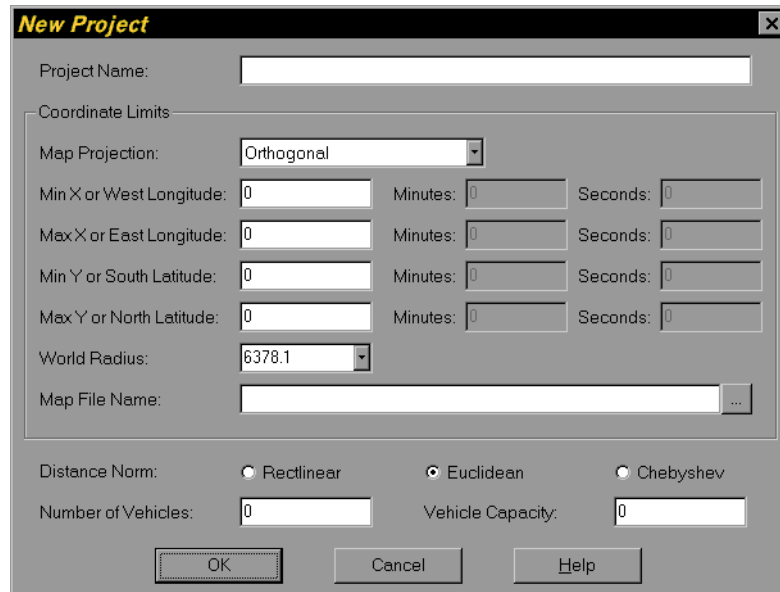


Figure 5.4. New Project Dialog Window



Information on the individual data fields is given in **New Project Data Items** and information on the general function of the command is given in **New** command of the **File** menu.

## ***New Project Data Items***

### **Project Name**

The project name or title refers to the title of the project to be used in reports and printouts. It consists of a maximum of 63 alphanumeric, space, 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 modified with the **Properties** command of the **File** menu.

### **Map Projection**

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 orthogonal projection is suitable only for small areas. 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, Western Europe, and areas at intermediate north latitudes.

Further information can be found in the Project Data section under **Map Projection**. The map projection cannot be changed once a project has been created or imported.

### **Minimum X or West Longitude**

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 map projections are used. Longitudes and latitudes are displayed and edited in three different fields, corresponding to the degrees, minutes, and seconds of the longitude and latitude. However, longitudes and latitudes are stored internally in a packed integer format. Longitudes can range from -180 to +180 degrees or from -1800000 to +1800000 in the packed integer latitude/longitude format.

Further information can be found in the Project Data section under **Minimum X or West Longitude**. The minimum x or west longitude cannot be changed once a project has been created or imported.

### **Maximum X or East Longitude**

The maximum x is the rightmost boundary value of valid coordinates if the orthogonal map projection is used. The maximum west longitude is the rightmost boundary value of valid coordinates when the Mercator or Albers map projections are used. Longitudes and latitudes are displayed and edited in three different fields, corresponding to the degrees, minutes, and seconds of the longitude and latitude. However, longitudes and latitudes are stored internally in a packed integer format. Longitudes can range from -180 to +180 degrees or from -1800000 to +1800000 in the packed integer latitude/longitude format.

Further information can be found in the Project Data section under **Maximum X or East Longitude**. The maximum x or east longitude cannot be changed once a project has been created or imported.

### **Minimum Y or South Latitude**

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 map projections are used. Longitudes and latitudes are displayed and edited in three different fields, corresponding to the degrees, minutes, and seconds of the longitude and latitude. However, longitudes and latitudes are stored internally in a packed integer format. Latitudes can range from -90 to +90 degrees or from -900000 to +900000 in the packed integer latitude/longitude format.

Further information can be found in the Project Data section under **Minimum Y or South Latitude**. The minimum y or south latitude cannot be changed once a project has been created or imported.

### **Maximum Y or North Latitude**

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 map projections are used. Longitudes and latitudes are displayed and edited in three different fields, corresponding to the degrees, minutes, and seconds of the longitude and latitude. However, longitudes and latitudes are stored internally in a packed integer format. Latitudes can range from -90 to +90 degrees or from -900000 to +900000 in the packed integer latitude/longitude format.

Further information can be found in the Project Data section under **Maximum Y or North Latitude**. The maximum y or north latitude cannot be changed once a project has been created or imported.

### **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.

Further information can be found in the Project Data section under **World Radius**. The world radius cannot be changed once a project has been created or imported.

### **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*. The map file name must conform to the requirements for long file names in Windows. Further information can be found in the Project Data section under **Map Data File Name**. 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.

### Distance Norm

When new facilities are located the distance between the various facilities is computed with a distance norm. For the orthogonal map projection, the available distance norms are rectilinear, Euclidean or Chebyshev. For the Mercator and Albers map projections, the great circle norm is used.

Further information can be found in the Project Data section under **Distance Norm**. The distance norm used with the orthogonal map projection can be changed with the **Distance Norm** command of the **Edit** menu.

### Number of Vehicles

The number of vehicles available to execute routes. It is assumed that each vehicle can execute at most one route, so the number of vehicles is an upper bound on the number of routes. Further information can be found in the Project Data section under **Number of Vehicles**. The number of vehicles can be changed with the **Number of Vehicles** command of the **Edit** menu.

### Vehicle Capacity


The capacity of a single vehicle is expressed in units consistent with the demand and supply quantities of customers and suppliers. All vehicles are assumed to have the same capacity. Further information can be found in the Project Data section under **Vehicle Capacity**. The vehicle capacity can be changed with the **Vehicle Capacity** command of the **Edit** menu.

## 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 **Lineback** program, the **Project Data File** is a binary file that can no longer be viewed or manipulated outside the **Lineback** program. Use the **Import** command to import project data created with previous versions of the **Lineback** program or with the **Export** command. The default extension for **Lineback** project files is *lineback*. 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

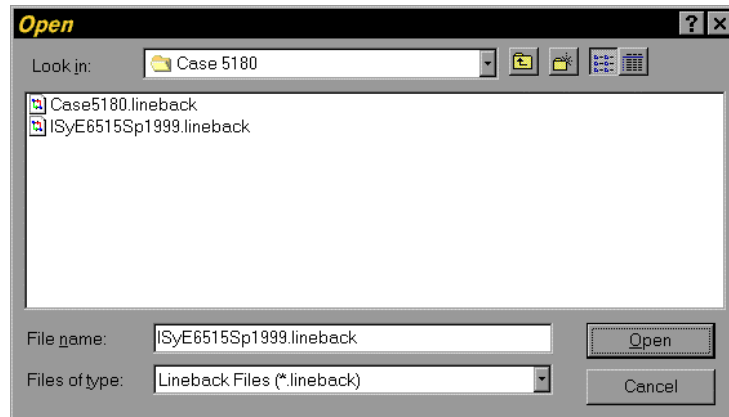


Figure 5.5. Open Project Dialog Window

## 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

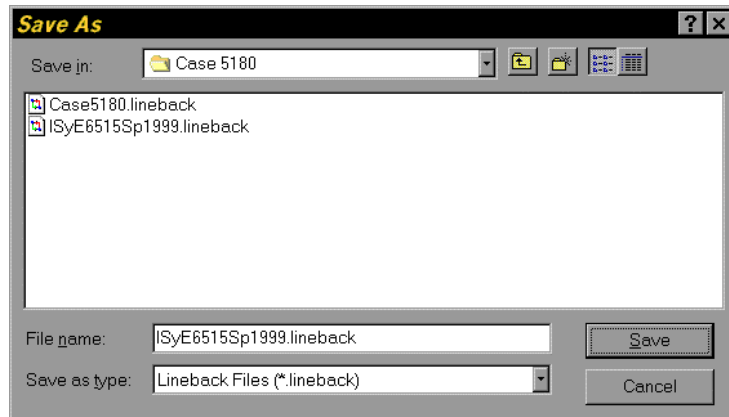


Figure 5.6. Project Save As Dialog Window

## Save As Data Fields

### File Name

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

### File Type

Starting with version 4.0 of the **Lineback** program, the **Project Data File** is a binary file that can no longer be viewed or manipulated outside the **Lineback** 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 **Lineback** program with an ASCII text editor and then imported again into the **Lineback** 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 **Lineback** program earlier than 4.0, created with the **Export** command, or created manually outside the **Lineback** program with an ASCII editor.

Starting with version 4.0 of the **Lineback** program, the **Project Data File** is a binary file that can no longer be viewed or manipulated outside the **Lineback** program. Use the **Open** command to read project data files created with the **Save** command of version 4.0 or higher of **Lineback**. The default extension for **Lineback** project files is *lineback*, the default extension for the Lineback 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 **Customer Data File** (*\*.cus*), **Supplier Data File** (*\*.sup*), **Depot Data File** (*\*.dep*), and **Route Data File** (*\*.rou*). It will also read the corresponding **Map Data File** (*\*.map*) if a map file has been specified.

The **Project Data File** and the **Facility, Route 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 **Lineback** 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 **Lineback** program.

### **Import Project Dialog Window**

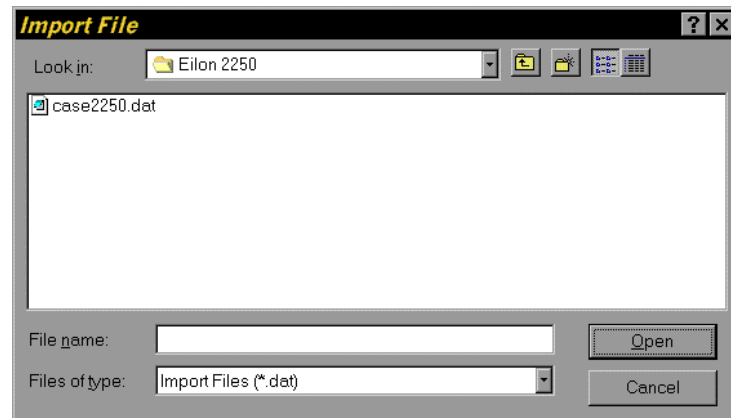


Figure 5.7. Import Project Dialog Window

### **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), **Customer Data File** (\*.cus), **Supplier Data File** (\*.sup), **Depot Data File** (\*.dep), and **Route Data File** (\*.rou). 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 facilities data.

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

## Export Dialog Window

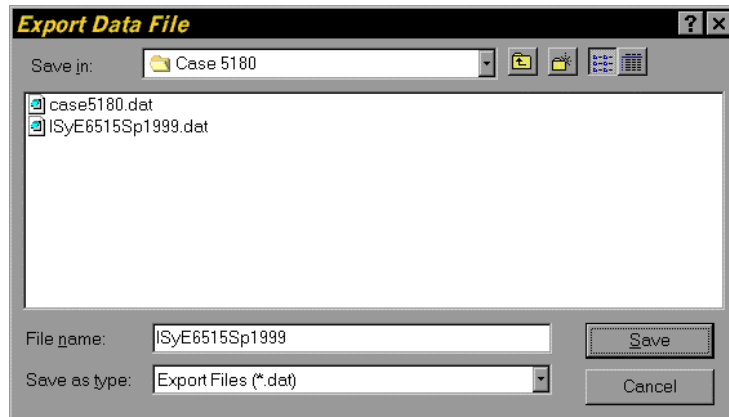


Figure 5.8. Export Dialog Window

## 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, keywords, and comments to the current project. The **Properties** command displays the **Properties Dialog Window** for the current document, which is illustrated in Figure 5.9. You can provide additional information in this dialog window about the current project.

## Properties Dialog Window

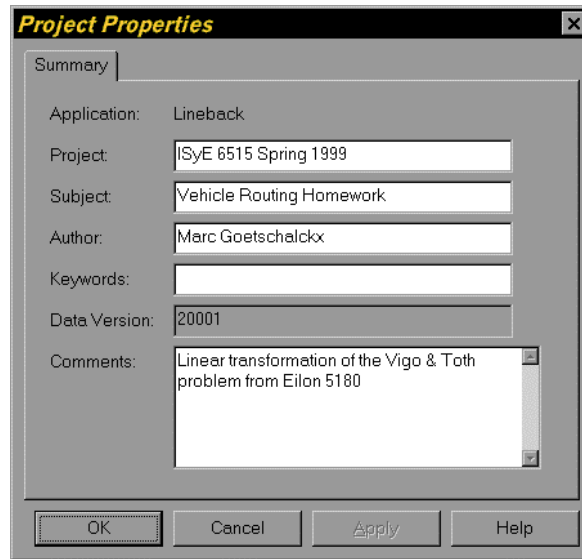


Figure 5.9. Properties Dialog Window

Information on the individual data fields is given in **Properties Data Items** and information on the general function of the command is given in **Properties** command of the **File** menu.

### Properties Data Items

#### Application

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

#### Project

The project name or title. This is the only place where the title 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 Title.

#### 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 **Lineback** program. **Lineback** 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

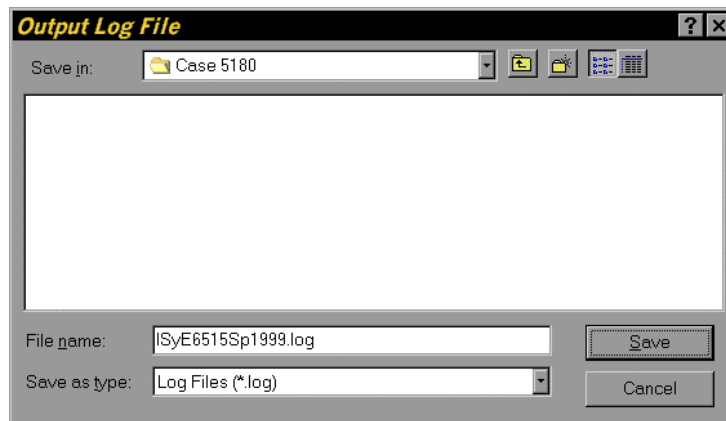


Figure 5.10. Output Log Selection Dialog Window

Information on the individual data fields is given in **Output Log Data Items** and information on the general function of the command is given in **Output Log** command of the **File** menu.

### Output Log Data Fields

#### File Name

The file name identifies the **Output Log File**. **Lineback** is compatible with long file and directory names that follow the Windows conventions.

#### File Type

The default output log file type is *log* and this type corresponds to a pure ASCII text file that can be viewed with a variety of text editors.

## 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

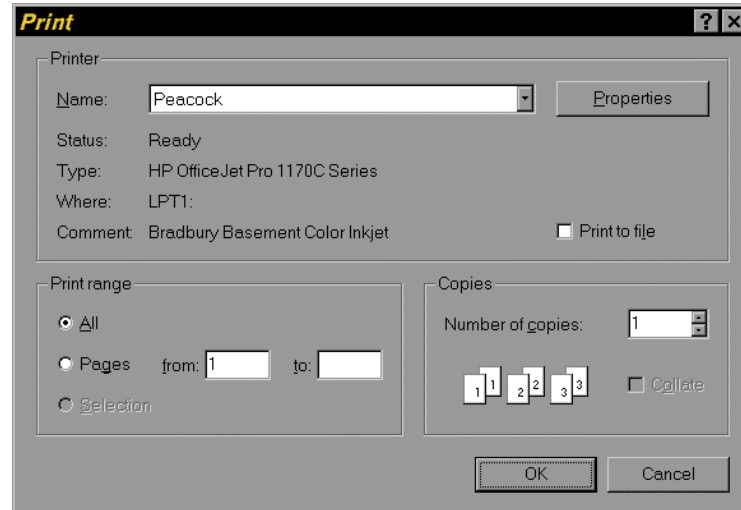


Figure 5.11. Print Dialog Window

### Print Data Items

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

#### Name

This is the active printer and printer connection. Choose the Setup option to change the printer and printer connection.

#### Properties

Displays a **Print Properties** dialog window, so you can select a printer and printer connection.

#### 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 **Lineback** 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

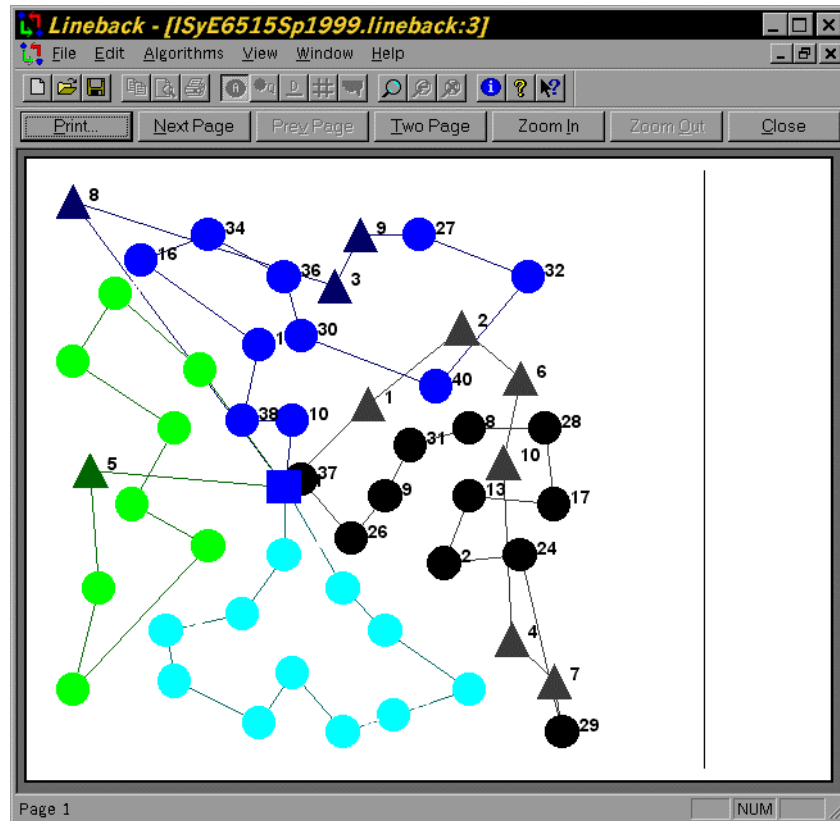


Figure 5.12. Print Preview Window

## 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*

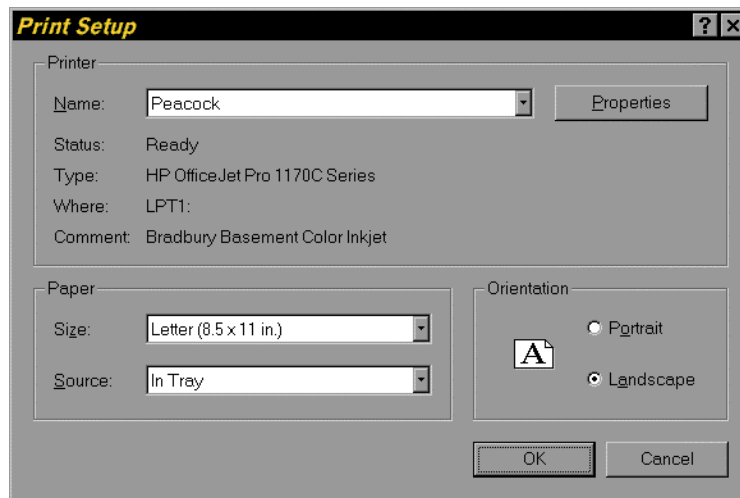


Figure 5.13. Print Setup Dialog Window

### *Print Setup Data Items*

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

#### **Name**

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.

#### **Properties**

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, **Lineback** 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 **Lineback** 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 **Lineback** 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

All the data items of the current project can be modified while executing the **Lineback** program. 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.

## Add Depot

The Add Depot command adds a depot to the current project. A project can only have a single depot, so if there is already a depot facility included in the project, then this command is disabled. Select the Add Depot command and the cursor will change to a crosshair shape. Move the cursor in the **Routes** view to the approximate location of the depot and click the left mouse button. The **Add Depot** dialog window will be displayed where you can specify additional data for this depot and refine the coordinates. The dialog window is shown in Figure 5.14. Press **Ok** to add this depot to the project. The number of depots in the **Notes** view is updated and the **Routes** view displays the depot.

## Add Depot Dialog Window

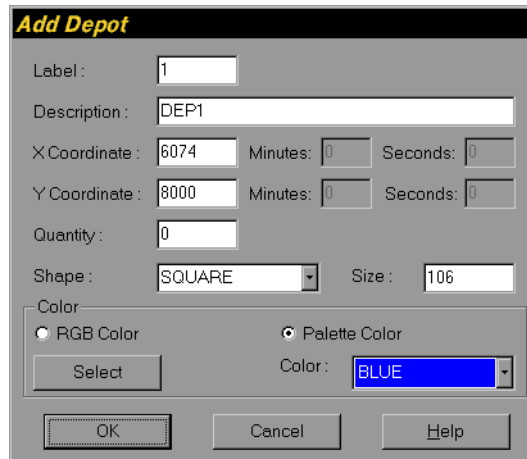


Figure 5.14. Add Depot Dialog Window

Information on the individual data fields is given in **Depot Data Items** and information on the general function of the command is given in **Add Depot** command of the **Edit** menu.

## Depot Data Items

### Label

Each depot has a unique label. The label can only contain letters, digits, and underscore characters. The maximum length of the label is seven characters. The label is a required data item; i.e., it cannot be empty. The label will be used to identify the depot in all screen and dialog windows.

### Description

Each depot has a description or name. The description can contain only letters, underscore characters, and spaces and the maximum length of the description is 63 characters. The description is a required data item; i.e., it cannot be empty.

### Y Coordinate

The location of the supplier is given by its x and y coordinate or by its latitude and longitude, depending on if the orthogonal map projection is used for the project. The latitude is stored in the packed integer (+/-)ddmmss format. For example, the longitude 23°20'15" W is represented as -0232015, and the latitude 35°20'43" N as 0352043. The degrees, minutes, and seconds of the latitude or longitude are unpacked and then shown and edited separately in the first, second, and third field of the latitude or longitude line, respectively. Latitude and longitude in the packed integer format are described in more detail in the section on **Common Data Items**.

### X Coordinate

The location of the supplier is given by its x and y coordinate or by its latitude and longitude, depending on if the orthogonal map projection is used for the project. The longitude is stored in the packed integer (+/-)ddmmss format. For example, the longitude 23°20'15" W is represented as -0232015, and the latitude 35°20'43" N as 0352043. The degrees, minutes, and seconds of the latitude or longitude are unpacked and then shown and edited separately in the first, second, and third field of

the latitude or longitude line, respectively. Latitude and longitude in the packed integer format are described in more detail in the section on **Common Data Items**.

### Quantity

The customer delivery and supplier pickup quantity must be non-negative quantity less than the vehicle capacity. The depot has neither a linehaul delivery or a backhaul pickup quantity, so typically its quantity is set to zero. The depot quantity is not used in any of the route design algorithms.

### Display Shape

The geometrical symbol or shape used to display the supplier in the **Routes** views. A list of all valid symbol names is given in the **Display Symbols** section of the **Common Data Items**.

### Display Size

The size of the geometric symbol displayed in the **Routes** views.

### Display Color

The color used to display the supplier symbol in the **Routes** views. A list of valid color names is given in the **Color** section of the **Common Data Items**.

## Add Customer

The Add Customer command adds a customer to the current project. The maximum number of customers in a project depends on the version of **Lineback**. The addition process, dialog windows, and data items are complete analogous to the **Add Depot** command.

## Add Supplier

The Add Supplier command adds a supplier to the current project. The maximum number of suppliers in a project depends on the version of **Lineback**. The addition process, dialog windows, and data items are complete analogous to the **Add Depot** command.

## Add Random Facilities

The Add Random Facilities command allows you to add quickly a large number of facilities to the current project with random locations and random quantities. The **Add Random Facilities Dialog Window** will be displayed which allows you to specify the **Add Random Facilities Data Items**. The maximum number of new facilities for each type is the difference between the maximum number of facilities of that type in your version of **Lineback** and the number of facilities of that type already in the project. The facilities will be uniformly located inside the world boundary coordinates. You can specify a maximum quantity for each facility type. The facilities of that type will have quantities uniformly distributed between one and the respective maximum quantity. The random locations and quantities are generated with the random number seed that you specify.

Press **Ok** to add the indicated number of new facilities to the project. Press **Cancel** to cancel the dialog window and *not* to add facilities to the project.



## Add Random Facilities Dialog Window

The screenshot shows a dialog window titled "Add Random Facilities". It contains three sections for inputting facility data:

- Customers:** Number: 16, Max Quantity: 25
- Depots:** Number: 1, Max Quantity: 0
- Suppliers:** Number: 16, Max Quantity: 100

Below these sections is a "Seed" field with the value 12345. At the bottom of the dialog are three buttons: "OK", "Cancel", and "Help".

Figure 5.15. Add Random Facilities Dialog Window

Information on the individual data fields is given in **Add Random Facilities Data Items** and information on the general function of the command is given in **Add Random Facilities** command of the **Edit** menu.

### Add Random Facilities Data Items

#### Number of Customers

The number of customers that you want to add to this project. The minimum number you can add is zero or no new customers. The maximum number you can add is the difference between the maximum number of customers for your version of **Lineback** and the number of customers already in the project.

#### Maximum Customer Quantity

The maximum delivery quantity for the customers that you want to add to this project. The customer's quantity will be uniformly distributed between one and the maximum quantity you specified. The maximum quantity can be no larger than the vehicle capacity.

#### Number of Depots

The number of depots that you want to add to this project. The minimum number you can add is zero or no new depots. The maximum number you can add is the difference between the maximum number of depots for your version of **Lineback** and the number of depots already in the project. Currently **Lineback** allows only a single depot in a project.

#### Maximum Depot Quantity

The maximum depot quantity is not used in the project and will be set to zero.

#### Number of Suppliers

The number of suppliers that you want to add to this project. The minimum number you can add is zero or no new suppliers. The maximum number you can add is the difference between the maximum number of suppliers for your version of **Lineback** and the number of suppliers already in the project.

## Maximum Supplier Quantity

The maximum delivery quantity for the suppliers that you want to add to this project. The supplier's quantity will be uniformly distributed between one and the maximum quantity you specified. The maximum quantity can be no larger than the vehicle capacity.

## Seed

This data item is the seed for the random number generator, which is used to generate the random samples of customer and supplier locations and quantities. The Add Random command will always generate identical additional facilities if you select the same number and maximum quantities of the facilities and set the same random number seed.

## Edit Depot

The Edit Depot command allows you to change the data items of a depot already included in the current project. Select the Edit Depot command and the cursor will change to an up-arrow shape. Move the cursor in the **Routes** view to the depot symbol and click the left mouse button. The **Edit Depot** dialog window will be displayed where you can modify the data for this depot. The dialog window is shown in Figure 5.16. Please see the description of the **Depot Data Items** under the **Add Depot** command for further information on the data fields associated with this depot. Press **Ok** to confirm the changes you have made for this depot and update the project. Press **Cancel** to cancel the dialog window and *not* to change the facility data.

### *Edit Depot Dialog Window*

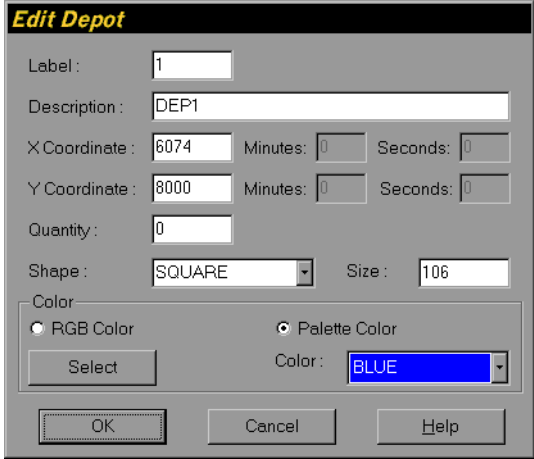


Figure 5.16. *Edit Depot Dialog Window*

Information on the individual data fields is given in **Depot Data Items** and information on the general function of the command is given in **Edit Depot** command of the **Edit** menu.

## Edit Customer

The Edit Customer command allows you to modify the data characteristic of an existing customer. The modification process, dialog windows, and data items are complete analogous to the **Edit Depot** command.

## Edit Supplier

The Edit Customer command allows you to modify the data characteristic of an existing customer. The modification process, dialog windows, and data items are complete analogous to the **Edit Depot** command.

## Edit Facility

The Edit Facility command allows you to modify the data characteristic of an existing facility regardless of its type. The modification process, dialog windows, and data items are complete analogous to the **Edit Depot**, **Edit Customer**, or **Edit Supplier** command, depending on the type of facility that you select.

## Edit Depot by Label

The Edit Depot by Label command will display a list with all the labels of the current depot facilities in the project. The **Select Depot by Label Dialog Window** is illustrated in Figure 5.17. After a depot has been selected with its label, the depot data can be edited just like in the **Edit Depot** command by pressing the **Edit** button. While the **Edit Depot** dialog is shown, choose **OK** to retain the modified depot data and choose **Cancel** to cancel the dialog window and *not* to change the depot data. In either case, you are returned to the **Select Depot by Label** dialog window and can then select the next depot to be edited by its label. Pressing the **Done** button will terminate the depot selection dialog window.

### Select Depot by Label Dialog Window

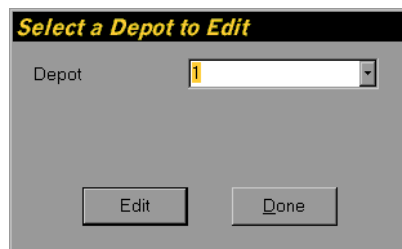


Figure 5.17. Select the Depot by Label to Edit Dialog Window

Information on the individual data fields is given in **Select Depot by Label Data Items** and information on the general function of the command is given in the **Edit Depot by Label** command of the **Edit** menu.

### Select Depot by Label Data Items

#### Depot

The label of the depot currently selected for the editing. You can specify this label by typing it directly into the field or by selecting it from the drop-down list of labels of the current depot facilities in the project.

## Edit Customer by Label

The Edit Customer by Label command allows you to select an existing customer by its label and then to modify the data characteristic of this customer. The customer selection process, modification process, dialog windows, and data items are complete analogous to the **Edit Depot by Label** command.

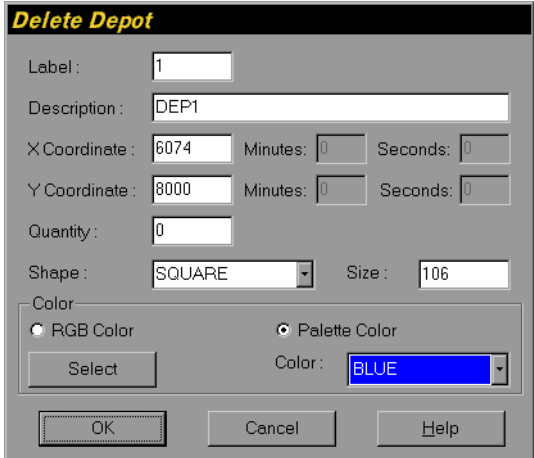
## Edit Supplier by Label

The Edit Supplier by Label command allows you to select an existing supplier by its label and then to modify the data characteristic of this supplier. The supplier selection process, modification process, dialog windows, and data items are complete analogous to the **Edit Depot by Label** command.

## Delete Depot

The Delete Depot command allows you to delete a depot already included in the current project. Select the Delete Depot command and the cursor will change to an up-arrow shape. Move the cursor in the **Routes** view to the depot symbol and click the left mouse button. The **Delete Depot** dialog window will be displayed where you can verify that this is the correct depot to be deleted. The dialog window is shown in Figure 5.18. Press **Ok** to confirm the deletion of this depot and update the project. Press **Cancel** to cancel the dialog window and *not* to delete the depot.

### Delete Depot Dialog Window



The screenshot shows the 'Delete Depot' dialog window with the following fields and values:

- Label: 1
- Description: DEP1
- X Coordinate: 6074, Minutes: 0, Seconds: 0
- Y Coordinate: 8000, Minutes: 0, Seconds: 0
- Quantity: 0
- Shape: SQUARE
- Size: 106
- Color: Palette Color (selected), Color: BLUE

Buttons: Select, OK, Cancel, Help

Figure 5.18. Delete Depot Dialog Window

Information on the individual data fields is given in **Depot Data Items** and information on the general function of the command is given in **Delete Depot** command of the **Edit** menu.

## Delete Customer

The Delete Customer command allows you to delete an existing customer from the project. The deletion process, dialog windows, and data items are complete analogous to the **Delete Depot** command.

## Delete Supplier

The Delete Supplier command allows you to delete an existing supplier from the project. The deletion process, dialog windows, and data items are complete analogous to the **Delete Depot** command.

## Delete Facility

The Delete Facility command allows you to delete an existing facility from the project regardless of its type. The deletion process, dialog windows, and data items are complete analogous to the **Delete Depot**, **Delete Customer**, or **Delete Supplier** command, depending on the type of facility that you select.

## Delete Depot by Label

The Delete Depot by Label command will display a list with all the labels of the current depot facilities in the project in the **Select Depot by Label Dialog Window**. This dialog windows is illustrated in Figure 5.19. After you have selected a depot with its label, the depot can be deleted just like in the **Delete Depot** command by pressing the **Delete** button. When the **Delete Depot Dialog Window** is shown for the selected depot, choose **OK** to delete the depot and choose **Cancel** to cancel the dialog window and *not* to delete the depot. In either case, you are returned to the **Select Depot by Label Dialog Window** and can then select the next depot to be deleted by its label. Pressing the **Done** button will terminate the depot selection process.

### Select Depot by Label Dialog Window

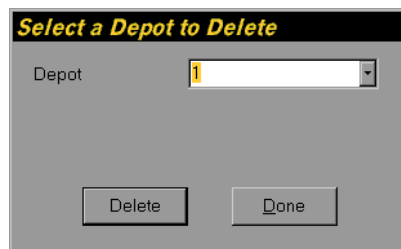


Figure 5.19. Select the Depot by Label to Delete Dialog Window

Information on the individual data fields is given in **Select Depot by Label Data Items** and information on the general function of the command is given in the **Delete Depot by Label** command of the **Edit** menu.

### Select Depot by Label Data Items

#### Depot

The label of the depot currently selected for the deletion. You can specify this label by typing it directly into the field or by selecting it from the drop-down list of labels of the current depot facilities in the project.

## Delete Customer by Label

The Delete Customer by Label command allows you to select an existing customer by its label and then to delete this customer. The customer selection process, deletion process, dialog windows, and data items are complete analogous to the **Delete Depot by Label** command.

## Delete Supplier by Label

The Delete Supplier by Label command allows you to select an existing supplier by its label and then to delete this supplier. The supplier selection process, deletion

process, dialog windows, and data items are complete analogous to the **Delete Depot by Label** command.

## Delete All Facilities

The Delete All Facilities command allows you to delete all the facilities from the current project. All the routes and distance information will also be deleted from the project. Because of its significant impact on the current project, a warning dialog box is displayed to verify that you really want to delete all facilities. The **Delete All Facilities Confirmation Dialog Window** is illustrated in Figure 5.20.

### Delete All Facilities Confirmation Dialog Window

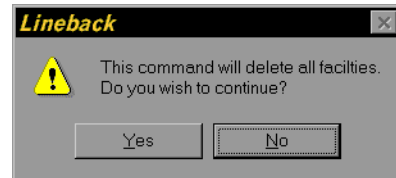


Figure 5.20. Delete All Facilities Confirmation Dialog Window

## Edit Link

The Edit Link command displays the **Edit Link Dialog Window**, which allows you to modify the characteristics of a single link between two facilities. At the current time, the only attribute you can change is the distance between the two facilities. The algorithms from then on will use the distance that you specified until the distances are recomputed with the **Compute Distances** command. Press **Ok** to confirm the changes you have made to the distances and update the project. Press **Cancel** to cancel the dialog window and *not* to change the distances for this link.

### Edit Link Dialog Window

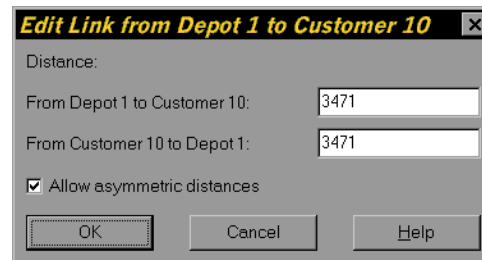


Figure 5.21. Edit Link Distances Dialog Window

Information on the individual data fields is given in **Edit Link Data Items** and information on the general function of the command is given in the **Edit Link** command of the **Edit** menu.

### Edit Link Data Items

#### Distance

The distances from the first facility of the pair to the second facility and from the second facility to the first facility. These distances may be different if you enabled asymmetric distances in the project. If asymmetric distances are disabled, then the

two distances will always be identical. The distance between a pair of facilities must be non-negative.

### **Allow Asymmetric Distances**

If the distance between two facilities is different depending on the direction of the travel between the facilities, the distances are said to be asymmetrical. You must enable asymmetrical distances for the project, in order to allow the distances between a pair of facilities to be different.

## **All Distances**

Use this command to display or edit all the distances between the facilities. 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 5.22.

This dialog window shows the distances between the facilities. Each row represents an origin facility and each column represents a destination facility. You can use the scrollbars to move around the distance matrix, if there are more facilities 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 distances, 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

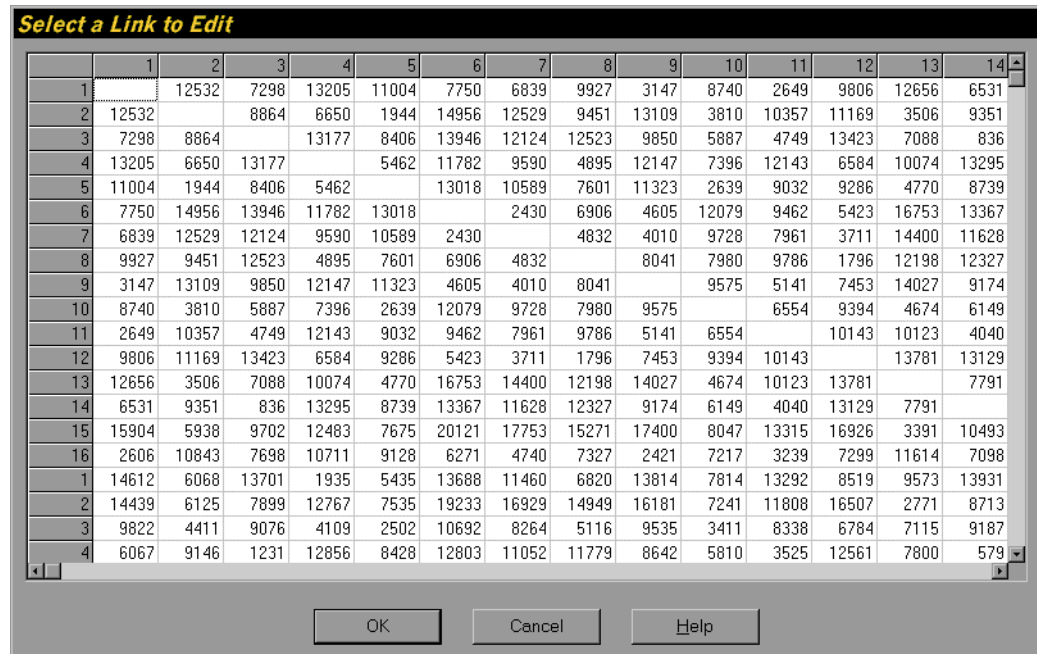


Figure 5.22. Edit All Distances Dialog Window

Information on the general function of the command is given in the **All Distances** command of the **Edit** menu.

### Compute Distances

The Compute Distances command updates the distance between each pair of facilities based on the current facility location and using the currently specified distance norm. This command is most often used after you have modified the location of one or more facilities and you want to update the inter-facility distance so that they correspond to the new facility locations.



If you have modified manually the distance between one or more pair of facilities, then these modifications will be replaced by the newly computed distance.

### Symmetric Distances

The Symmetric Distances command will make the distance between a pair of facilities symmetrical by copying the pairwise distance from above the main diagonal of the distance matrix to the corresponding element below the main diagonal.



If you have modified manually a distance between a pair of facilities below the main diagonal of the distance matrix, then these modifications will be replaced by the corresponding distance from above the main diagonal.

### Delete All Routes

The Delete All Routes command allows you to delete all the routes from the current project. Because of its significant impact on the current project, a warning dialog



box is displayed to verify that you really want to delete all routes. The Delete All Routes Confirmation Dialog Window is illustrated in Figure 5.23.

### Delete All Routes Confirmation Dialog Window

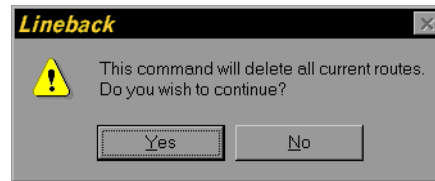


Figure 5.23. Delete All Routes Confirmation Dialog Window

### Number of Vehicles...

The Number of Vehicles command displays the **Number of Vehicles Dialog Window**, which allows you to edit the number of vehicles available in the project to execute routes. It is assumed that each vehicle can execute at most one route, so the number of vehicles is an upper bound on the number of routes. The maximum number of vehicles is dependent on your version of **Lineback**.

### Number of Vehicles Dialog Window

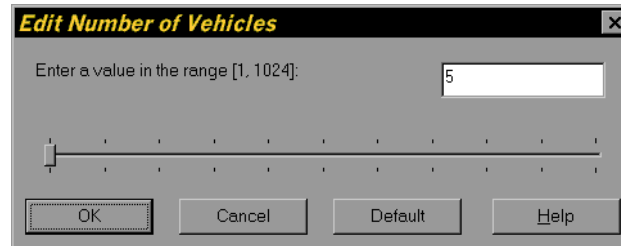


Figure 5.24. Number of Vehicles Dialog Window

Information on the general function of the command is given in the **Number of Vehicles** command of the **Edit** menu.

### Vehicle Capacity...

The Vehicle Capacity command displays the **Vehicle Capacity Dialog Window**, which allows you to edit the capacity of all the vehicles in the project. The vehicle capacity is expressed in units consistent with the demand and supply quantities of customers and suppliers. All vehicles are assumed to have the same capacity.

### Vehicle Capacity Dialog Window

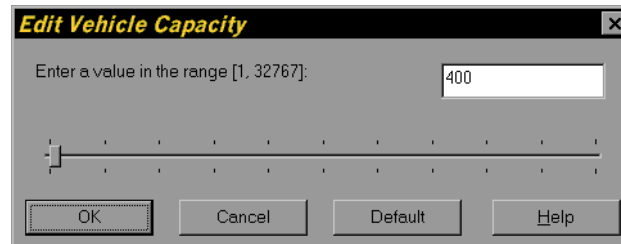


Figure 5.25. Vehicle Capacity Dialog Window

Information on the general function of the command is given in the **Vehicle Capacity** command of the **Edit** menu.

## Infeasibility Penalty...

The Infeasibility Penalty command displays the **Infeasibility Penalty Dialog Window**, which allows you to edit the infeasibility penalty factor. The infeasibility penalty factor is multiplied by the quantities of any unvisited facilities and by the quantity on each route that exceeds the vehicle capacity to compute the total infeasibility penalty. The total infeasibility penalty is then added to the total route distance to yield the total route cost. The infeasibility penalty factor must be a non-negative number. A more positive infeasibility factor increases the tendency of the algorithms to create more feasible routes.

### *Infeasibility Penalty Factor Dialog Window*

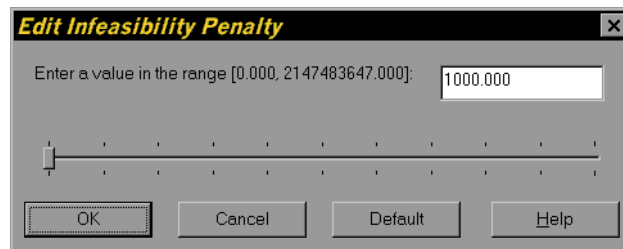


Figure 5.26. *Infeasibility Penalty Factor Dialog Window*

Information on the general function of the command is given in the **Infeasibility Penalty** command of the **Edit** menu.

## Distance Norm...

If the project uses the orthogonal map projection, then the Distance Norm command displays the **Distance Norm Dialog Window**, which allows you to specify the distance norm used to compute the distance between every pair of facilities. For the orthogonal map projection, the available distance norms are rectilinear, Euclidean or Chebyshev. For the Mercator and Albers map projections, the great circle norm is always used and the Distance Norm command is disabled.

### *Distance Norm Dialog Window*

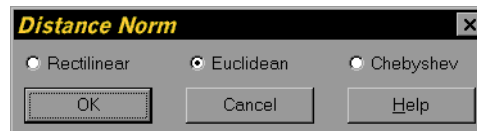


Figure 5.27. *Distance Norm Dialog Window*

Information on the general function of the command is given in the **Distance Norm** command of the **Edit** menu.

## 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**

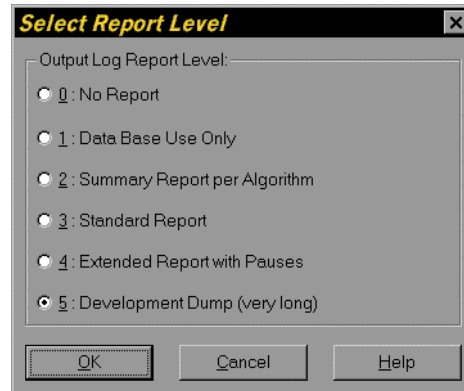


Figure 5.28. Report Level Dialog Window

Information on the individual data fields is given in the **Report Level Data Items** and information on the general function of the command is given in **Report Level** command of the **Edit** menu.

### **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.
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. The route improvement algorithms based on simulated annealing use random numbers to select the next two facilities to be tested for a possible exchange. An algorithm will always produce the same identical results if you select the same algorithm settings and set 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 5.29. Seed Dialog Window

Information on the individual data fields is given in the **Seed Data Items** and information on the general function of the command is given in **Seed** command of the **Edit** menu.

### 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 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

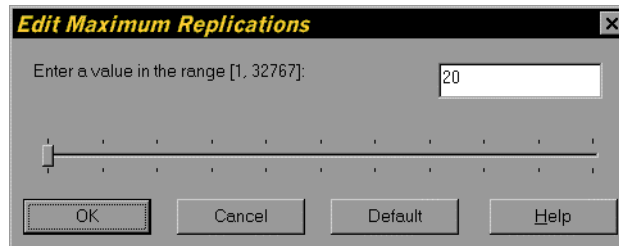


Figure 5.30. Maximum Replications Dialog Window

Information on the individual data fields is given in the **Maximum Replications Data Items** and information on the general function of the command is given in **Maximum Replications** command of the **Edit** menu.

## Maximum Replications Data Items

Some non-deterministic route improvement algorithms need to make repeated random choices to identify facilities to be tested for a possible exchange. Different replications of the same algorithm can thus provide different routes. The higher the number of replications, the more likely it is that a good set of routes 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 can either abort or continue the algorithm with the Time Expiration dialog window. At that time you have also the option to set a new time limit.

## Time Limit Dialog Window

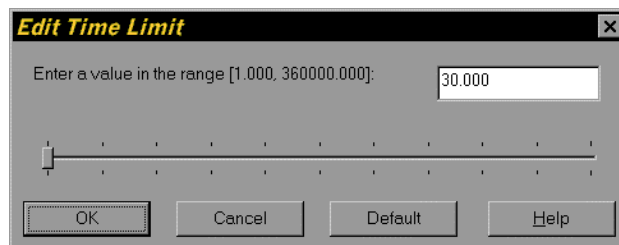


Figure 5.31. Time Limit Dialog Window

Information on the individual data fields is given in the **Time Limit Data Items** and information on the general function of the command is given in **Time Limit** command of the **Edit** menu.

## 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***

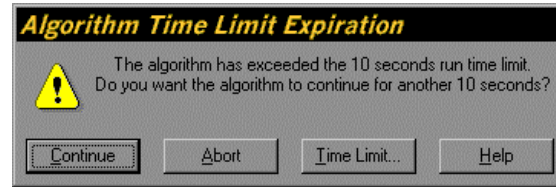


Figure 5.32. *Time Limit Expiration Dialog Window*

## **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 Visio to include the routes in reports and presentations. The **Routes** 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 each route at the depot facility and then iteratively appends the nearest unvisited customer to the route. When appending the next customer would violate the truck capacity, the algorithm switches to iteratively appending the nearest unvisited supplier to the route. When appending the next supplier would violate truck capacity, then the route returns to the depot. If unvisited customers or suppliers remain and the maximum number of routes has not been reached, then a new route is started and the above process is repeated.

### ***Nearest Neighbor Algorithm Shortcuts***

Keys: CTRL+ALT+N

### **Sweep**

The Sweep algorithm creates a sweep tour by appending the facilities to the tour when they are traversed by a ray rotating around the depot. The relative polar coordinates of each point with respect to the depot are determined and the points are inserted in the tour by increasing polar angle. The algorithm was first described by

Gillet and Miller (1974) for the vehicle routing problem. The starting angle of the ray is an algorithm parameter.

This algorithm is extended in a straightforward manner to the case of vehicle routing with backhauls. When a customer or supplier facility is traversed by the rotating ray it is inserted between the most recently appended customer and supplier facility on that route. When the inserting either the next customer or supplier would violate truck capacity, then the route is terminated. If unvisited customers or suppliers remain and the maximum number of routes has not been reached, then a new route is started and the process is repeated.

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.

### ***Sweep Algorithm Shortcuts***

Keys: CTRL+ALT+S

## **Savings**

The Savings algorithm executes the Clarke and Wright (1964) construction procedure that extends a partial route or route primitive at its two end points. Conceptually the algorithm constructs an Eulerian tour that visits each of the customer and suppliers and the returns to the depot. The Eulerian tour is then reduced in length by finding and executing the shortcut with the largest savings. Deif and Bodin (1984) delayed the crossover link between customers and suppliers on each route by reducing the savings of establishing a short cut between a customer and supplier by the product of the savings penalty factor and the largest savings of any pair of points.

This savings penalty factor is an algorithm parameter set by the user. Deif and Bodin found that a good value for the savings penalty factor is 0.25. You can set the savings penalty factor on the **Savings** tab of the **Select** command of the **Algorithms** menu.

### ***Savings Algorithm Shortcuts***

Keys: CTRL+ALT+V

## **Extreme Crossovers**

The Extreme Crossovers algorithm creates a number of partial routes with on each route a single customer and a single supplier. It matches the customer and supplier that have the largest savings if they were visited on single route instead of on separate routes. This is the same savings definition as used by Clarke and Wright in the **Savings** algorithm. The number of routes is equal to the minimum of the number of vehicles, the number of customers, and the number of suppliers.

### ***Extreme Crossovers Algorithm Shortcuts***

Keys: CTRL+ALT+X

## **Manual**

The **Manual** command allows you to construct a set of feasible routes in an interactive manner. When you execute the **Manual** command, the cursor changes to a vertical arrow indicating that you can now select facilities to be added to the routes

and the algorithm Abort button on the tool bar is enabled. You start by clicking the left button of the mouse when it is over the depot and extend the routes by clicking the left button of the mouse when the cursor is over an unvisited or free facility. If you click for the second time on the depot, then the route is closed. Once a supplier or backhaul facility has been added to the route, you can not longer add customer or linehaul facilities to the current route. If adding the customer or supplier would violate the vehicle capacity, then that facility is not added to the route. The algorithm terminates when either all facilities have been placed on a route or the number of routes is equal to the number of vehicles. Not all facilities have to be included on the routes when the algorithm terminates, in other word, partial routes or route primitives are allowed.

If you click again on the last facility of the current route, then this point is removed from the route and becomes free again and the next to last facility on the route becomes the last facility. If the depot is the only facility on the route and is clicked again, then the complete route is removed. Clicking then again on the depot will make the depot the starting point of a new route. If the removed facility was the first backhaul supplier on the route, then you can add again linehaul customers to the route.


In general, clicking on a free facility adds this facility to the route if this is a feasible extension of the route and clicking on the last point of the route removes this facility from the route. Clicking on intermediate points of the route has no effect.

You stop the process of manually creating routes by pressing the **Escape** key. If the last facility on the current route is the depot then the algorithm terminates and route statistics are computed. If the last facility was not the depot, then the current route is closed by appending the depot facility and finally the route statistics are computed.

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

### Manual Algorithm Shortcuts

Keys: Ctrl+Alt+U (to start the construction)  
 Escape (to terminate the construction)

Toolbar:  (to abort the construction)

### Cheapest Insertion

The Cheapest Insertion algorithm inserts the remaining free facilities into the currently existing partial routes. The algorithm finds for every free facility the link on the routes with the smallest insertion penalty for that facility, provided the insertion would not make the route infeasible. It then inserts next the facility with the smallest of these insertion penalties and then updates the insertion penalties. Let  $F$  be the set of free facilities and  $T$  be the set of links on the routes, then the point  $k$  that is inserted next is given by

$$\min_{k \in F} \left\{ \min_{(i,j) \in T} (c_{ik} + c_{kj} - c_{ij}) \right\}$$

The algorithm thus executes a mini-min operation.



## ***Cheapest Insertion Algorithm Shortcuts***

Keys: Ctrl+Alt+C

## **Priciest Insertion**

The Priciest Insertion algorithm inserts the remaining free facilities into the currently existing partial routes. The algorithm finds for every free facility the link on the routes with the smallest insertion penalty for that facility, provided the insertion would not make the route infeasible. It then inserts next the facility with the largest of these insertion penalties and then updates the insertion penalties. Let  $F$  be the set of free facilities and  $T$  be the set of links on the routes, then the point  $k$  that is inserted next is given by

$$\max_{k \in F} \left\{ \min_{(i,j) \in T} (c_{ik} + c_{kj} - c_{ij}) \right\}$$

The algorithm thus executes a maxi-min operation.

## ***Priciest Insertion Algorithm Shortcuts***

Keys: Ctrl+Alt+P

## **Two Exchange**

The 2-opt Exchange algorithm is an intra-route edge-exchange improvement procedure that operates on a single route at the time. It attempts the change the sequence of the facilities in a route so that the route length is reduced by removing two links of the route and replacing them with two other links that connect the route again. All pairs of links that would yield a feasible exchange are tested and the exchange with the largest positive savings is executed. The algorithm thus belongs to the class of steepest descend improvement algorithms. An exchange is infeasible if it would violate the precedence constraint that all deliveries to customers have to be made before any pickup from a supplier can be made.

## ***Two Exchange Algorithm Shortcuts***

Keys: Ctrl+Alt+2

## **Or Exchange**

The Or-opt Exchange algorithm is an intra-route edge-exchange improvement procedure that operates on a single route at the time. It attempts the change the sequence of the facilities in a route so that the route length is reduced by removing a section or chain of the route and inserting this chain in a different place on the route. Moving a chain involves deleting three links of the route and replacing them with three other links that connect the route again. The number of facilities on the chain is called the length of the chain and is a parameter that you can specify. All chains of a certain length that would yield a feasible exchange are tested and the exchange with the largest positive savings is executed. The algorithm thus belongs to the class of steepest descend improvement algorithms. Since the Or-exchange involves removing three links and substituting them with three other links, all chain exchanges of all lengths will also be tested by the three-exchange procedure. An exchange is infeasible if it would violate the precedence constraint that all deliveries to customers have to be made before any pickup from a supplier can be made.

## ***Or Exchange Algorithm Shortcuts***

Keys: Ctrl+Alt+O

## **Three Exchange**

The 3-opt Exchange algorithm is an intra-route edge-exchange improvement procedure that operates on a single route at the time. It attempts the change the sequence of the facilities in a route so that the route length is reduced by removing three links of the route and replacing them with three other links that connect the route again. All triplets of links that would yield a feasible exchange are tested and the exchange with the largest positive savings is executed. The algorithm thus belongs to the class of steepest descend improvement algorithms. An exchange is infeasible if it would violate the precedence constraint that all deliveries to customers have to be made before any pickup from a supplier can be made.

## ***Three Exchange Algorithm Shortcuts***

Keys: Ctrl+Alt+3

## **Move**

The Move Facility procedure attempts to reduce the combined length of all the routes by moving a single facility from one route to another, provided the resulting routes remain feasible. A route can become infeasible if either the linehaul or backhaul quantity exceeds the truck capacity or if the precedence constraint is violated that states that all deliveries to customers have to be made before any pickup from a supplier can be made. All move exchanges are tested and the move exchange with the largest positive savings is executed. The algorithm thus belongs to the class of steepest descend improvement algorithms.

## ***Move Algorithm Shortcuts***

Keys: Ctrl+Alt+M

## **Swap**

The Swap Facility procedure attempts to reduce the combined length of all the routes by moving one facility from one route to another and simultaneously move one facility from the second route to the first route, provided the resulting routes remain feasible. A route can become infeasible if either the linehaul or backhaul quantity exceeds the truck capacity or if the precedence constraint is violated that states that all deliveries to customers have to be made before any pickup from a supplier can be made. All swap exchanges are tested and the swap exchange with the largest positive savings is executed. The algorithm thus belongs to the class of steepest descend improvement algorithms.

## ***Swap Algorithm Shortcuts***

Keys: Ctrl+Alt+W

## **Select**

The Select command allows you to specify an algorithm, to set the parameters for this algorithm, and to execute this algorithm. Only algorithms that take additional

parameters will be displayed as one of the tabs in this dialog window. At the current time the **Sweep**, **Or Exchange**, and **Savings** algorithms have parameters that can be changed by you and are displayed as tabs.

For the Sweep algorithm you can specify the starting angle of the rotating line segment. For the Or Exchange algorithm you can specify the number of facilities on the route chain that will be evaluated for relocation. For the Savings algorithm you can specify the penalty factor by which the savings are reduced when going from a delivery customer to a pick-up supplier.

### Select Algorithm Shortcuts

Keys: CTRL+ALT+A

### Select Algorithm Dialog Window Sweep Page

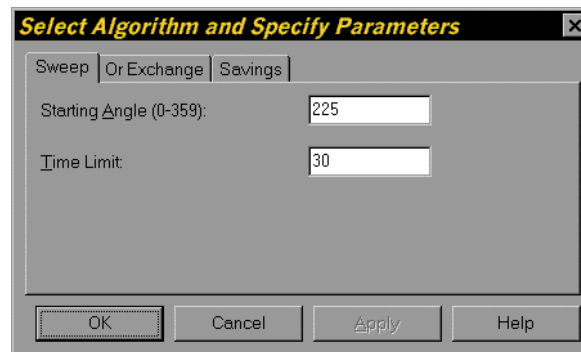


Figure 5.33. Select Algorithm Window Sweep Tab

### Select Algorithm Dialog Window Or Exchange Page

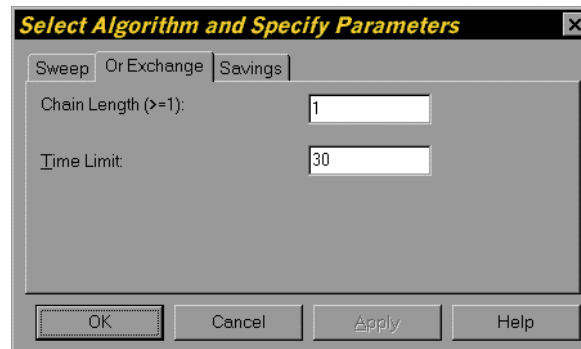


Figure 5.34. Select Algorithm Window Or Exchange Tab

## Select Algorithm Dialog Window Savings Page

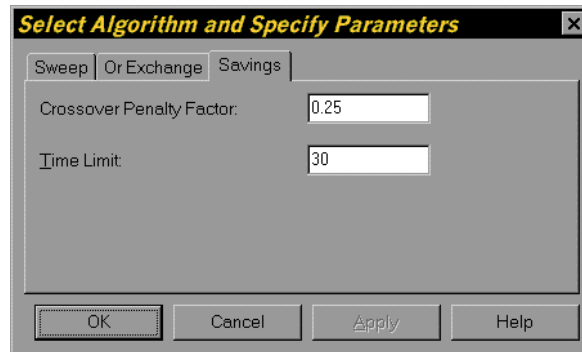


Figure 5.35. Select Algorithm Window Savings Tab

## Evaluate

The **Evaluate** algorithm computes the length and the cost of the current routes. The **Evaluate** command does not create new routes, but rather computes the length and the cost of the current routes based on the current distances, supply and demand data, number of vehicles and vehicle capacity, and infeasibility penalty. The results are displayed in the **Notes** and **Statistics** views and in the Evaluate algorithm dialog window. This algorithm is most frequently used after you have edited interactively the distances, the facility, or the vehicle data to evaluate the impact of the changes you have made.

## Evaluate Algorithm Dialog Window

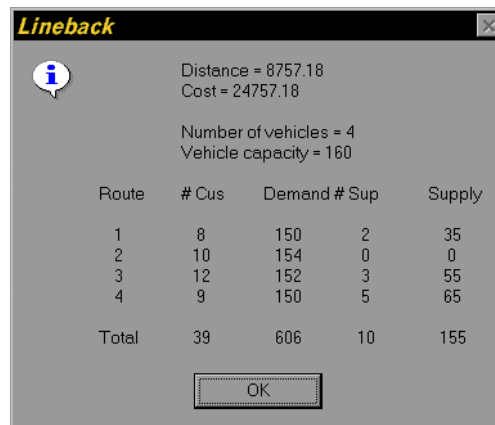


Figure 5.36. Evaluate Algorithm Result Window

## Evaluate Algorithm Shortcuts

Keys: CTRL+ALT+E

---

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

### Grid Project Shortcuts

Toolbar: 

Keys: CTRL+SHIFT+G

## 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

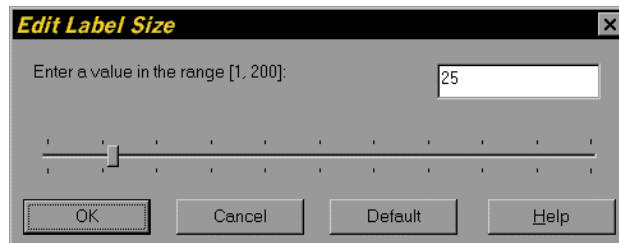


Figure 5.37. Edit Label Size Dialog Window

Information on the general function of the command is given in **Label Size** command of the **View** menu.

## 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

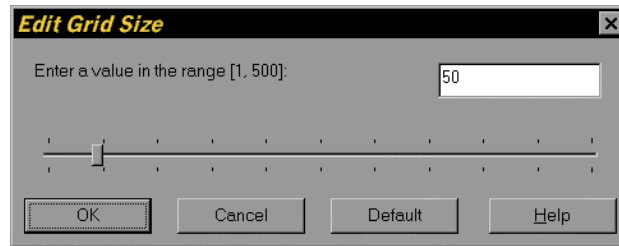


Figure 5.38. Edit Grid Size Dialog Window

Information on the general function of the command is given in **Grid Size** command of the **View** menu.

## 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 **Routes** 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



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



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



Keys: CTRL+SHIFT+O

## Redraw

The **Redraw** command of the **Display** menu redraws immediately the currently active view, be it either a **Routes** 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 facility in the current view.

### Redraw Shortcuts

Keys: CTRL+R

---

## Windows Menu

The **Windows** menu allows the opening, closing, arrangement, and selection of the **Routes** 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.

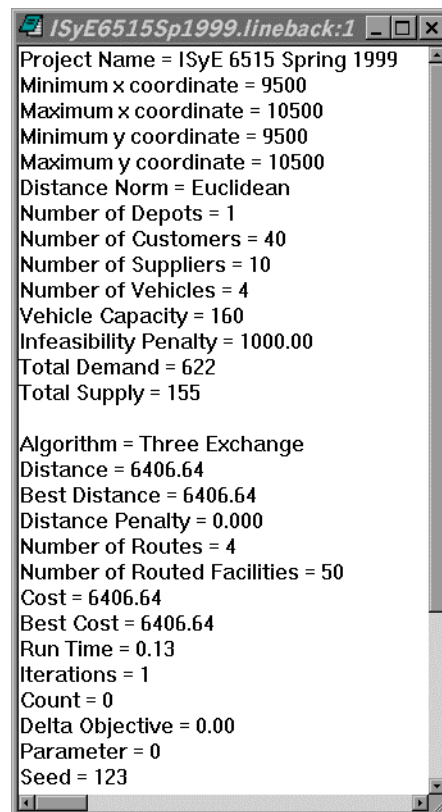
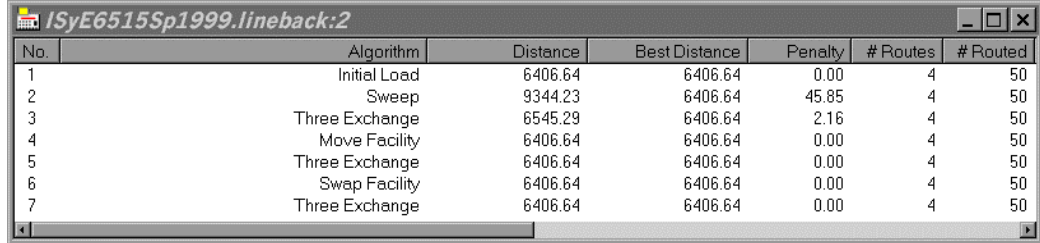


Figure 5.39. Notes View

## 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.



No.	Algorithm	Distance	Best Distance	Penalty	# Routes	# Routed
1	Initial Load	6406.64	6406.64	0.00	4	50
2	Sweep	9344.23	6406.64	45.85	4	50
3	Three Exchange	6545.29	6406.64	2.16	4	50
4	Move Facility	6406.64	6406.64	0.00	4	50
5	Three Exchange	6406.64	6406.64	0.00	4	50
6	Swap Facility	6406.64	6406.64	0.00	4	50
7	Three Exchange	6406.64	6406.64	0.00	4	50

Figure 5.40. Statistics View

## New Routes Window

The **New Routes View** command adds a new window that displays the hexagonal adjacency graph 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.

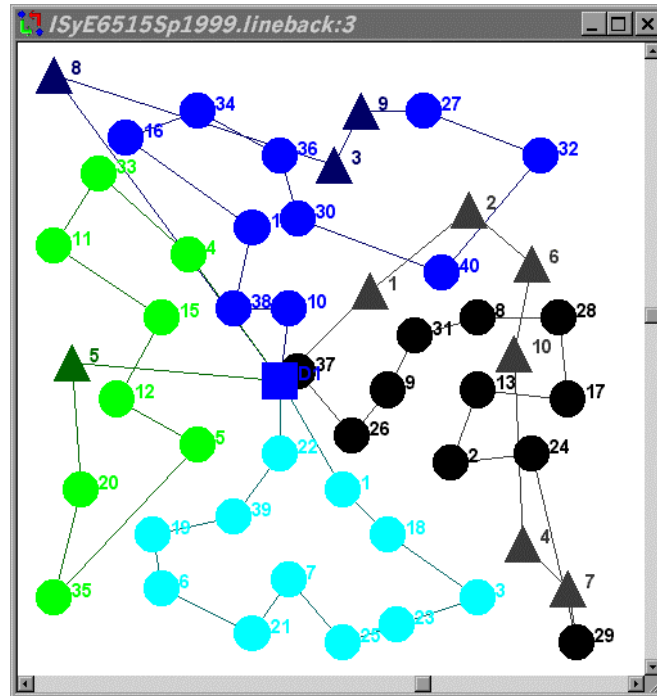


Figure 5.41. Routes View

## 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 **Lineback** 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 **Lineback** 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 5.42. *Lineback* Toolbar

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



Figure 5.43. *Dockable Lineback* Toolbar

## Status Bar

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



Figure 5.44. *Status Bar*

---

# Help Menu

## Help Topics

This command displays the Contents page of the **Lineback** interactive help system as shown in Figure 5.45.

The **Lineback** 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

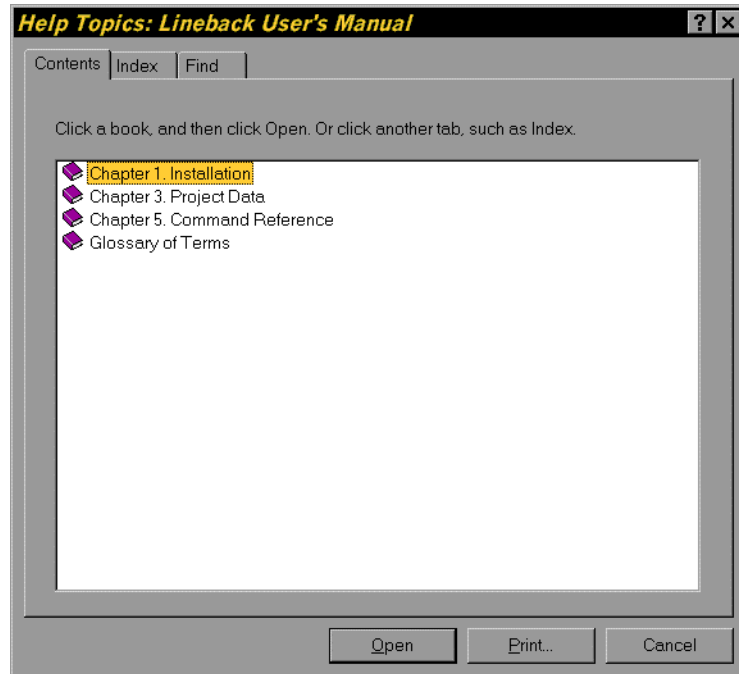



Figure 5.45. Help Contents Window

Information on the general function of the command is given in **Help Topics** command of the **Help** menu.

## 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 **Lineback** 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**



Keys: Shift+F1

## **About Lineback**

The **About** command of the **Help** menu shows the **About Lineback Dialog Window**, which displays the **Lineback** 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 Lineback** dialog window is illustrated in Figure 5.46.

### **About Lineback Dialog Window**

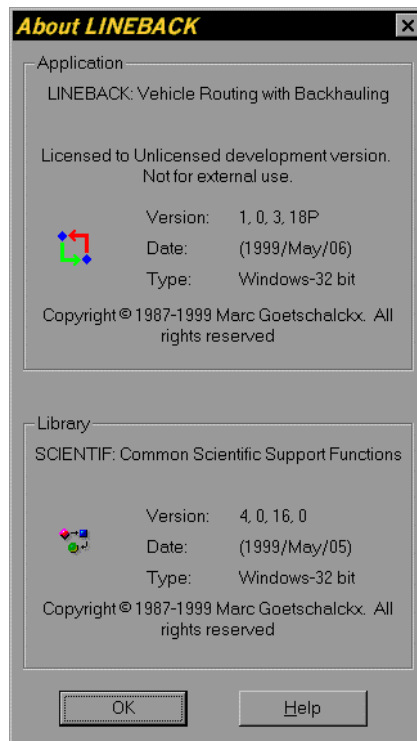


Figure 5.46. About Dialog Window

Information on general function of the command is given in the **About Lineback** command of the **Help** menu.

## ***About Shortcuts***

Toolbar: 

Keys: CTRL+A

# References

---

## Book and Journal References

1. Ball, M., T. Magnanti, C. Monma, and G. Nemhauser, (1995), **Handbooks in Operations Research and Management Science 8: Network Routing**, Elsevier, Amsterdam.
2. Bodin L. and B. Golden, (1981), "Classification in Vehicle Routing and Scheduling," *Networks*, Vol. 11, No. 2, pp. 97-108.
3. 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.
4. Deif, I. and L. D. Bodin (1984). "Extension of the Clarke-Wright Algorithm for Solving the Vehicle Routing Problem with Backhauling". Proceedings of the Babson Conference on Software Uses in Transportation and Logistics Management, A. E. Kidder, Ed., Babson Park, MA, pp. 75-96.
5. Duhamel, C. J.-Y. Potvin, and J.-M. Rousseau, (1997), "A Tabu Search Heuristic for the Vehicle Routing Problem with Backhauls and Time Windows," *Transportation Science*, Vol. 31, No. 1, pp. 49-59.
6. Fishetti, M., P. Toth, and D. Vigo, (1994), "A Branch-and-Bound Algorithm for the Capacitated Vehicle Routing Problem on Directed Graphs," *Operations Research*, Vol. 42, No. 5, pp. 846-859.
7. Gendreau, M., G. Laporte and A. Hertz, (1997), "An Approximation Algorithm for the Traveling Salesman Problem with Backhauls," *Operations Research*, Vol. 45, No. 4, pp. 639-641.
8. Gillett, B. and L. Miller, (1974), "A Heuristic Algorithm For the Vehicle Dispatch Problem," *Operations Research*, Vol. 22, pp 340-349.
9. Goetschalckx, M. and C. Jacobs-Blecha, (1989). "Vehicle Routing Problem with Backhauls". *European Journal of Operational Research*, Vol. 42, No. 1, pp. 39-51.
10. Golden, B. L. and Assad, A. A., (Eds), 1988, **Vehicle Routing: Methods and Studies**, Elsevier Science Publishers, New York, New York.

11. Golden, B. L. and Bodin, L., (1986), "Microcomputer-Based Vehicle Routing and Scheduling Software," *Computers & Operations Research*, Vol. 13, No. 2, pp. 277-285.
12. Golden, B. L., Baker, E. K., Alfaro, J. L., and Schaffer, J. R., (1985), "The Vehicle Routing Problem With Backhauling: Two Approaches," Working Paper Series MS/S 85-037, College of Business and Management, University of Maryland.
13. Jacobs C. D. and M. Goetschalckx, (1988). "The Vehicle Routing Problem with Backhauls". Material Handling Research Center Technical Report TR 88-00, Georgia Institute of Technology, Atlanta, Georgia.
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. 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.
16. Toth, P. and D. Vigo, (1996), "A Heuristic Algorithm for the Vehicle Routing Problem with Backhauls", in *Advanced Methods in Transportation Analysis*, L. Bianco and P Toth (eds.), Springer-Verlag, Berlin, pp. 585-608.
17. Toth, P. and D. Vigo, (1997), "An Exact Algorithm for the Vehicle Routing Problem with Backhauls", *Transportation Science*, Vol. 31, No. 4, pp. 372-386.
18. Yano, C. A., T. J. Chan, L. Richter, T. Culter, K. G. Murty, and D. McGettigan, (1987). "Vehicle Routing at Quality Stores". *Interfaces*, Vol. 17, No. 2, pp. 52-63.

---

## World Wide Web Sites

19. Goetschalckx, Marc. [www.isye.gatech.edu/~mgoetsch/index.html](http://www.isye.gatech.edu/~mgoetsch/index.html)
20. LogisticsCAD, [www.logisticscad.com](http://www.logisticscad.com)

# Appendix: Sample Projects

---

## Small

This case is compatible with all versions of the Lineback program, both Windows NT and Windows 95 and 98, and educational and industrial versions. It contains 16 customers and 16 suppliers and one depot. It is primarily used to illustrate the workings of the program and the design algorithms. It is the case used in the tutorial.

### Small.dat Project Data File Example

```
[data_version]          20000
[project_name]          Small
[minimum_x_coordinate]  0
[maximum_x_coordinate] 16000
[minimum_y_coordinate]  0
[maximum_y_coordinate] 16000
[number_of_vehicles]     5
[vehicle_capacity]      400
[infeasibility_penalty] 1000.000
[number_of_customers]   16
[customer_file_name]    Small.cus
[number_of_suppliers]   16
[supplier_file_name]    Small.sup
[number_of_depots]      1
[depot_file_name]       Small.dep
[number_of_routes]      3
[route_file_name]       Small.rou
[report_level]          3
[seed]                  12345
[number_of_iterations]  20
[time_limit]            30.000000
```

### Small.cus Customer Data File Example

1	409	9210	20	CIRCLE	106	RED	CUS1
2	12940	9360	48	CIRCLE	106	RED	CUS2
3	5604	14335	83	CIRCLE	106	RED	CUS3
4	11946	2785	86	CIRCLE	106	RED	CUS4
5	11368	8216	31	CIRCLE	106	RED	CUS5

6	239	1462	37	CIRCLE	106	RED	CUS6
7	2357	2654	99	CIRCLE	106	RED	CUS7
8	7131	1905	1	CIRCLE	106	RED	CUS8
9	251	6067	54	CIRCLE	106	RED	CUS9
10	9139	9628	61	CIRCLE	106	RED	CUS10
11	2659	10609	46	CIRCLE	106	RED	CUS11
12	5634	912	61	CIRCLE	106	RED	CUS12
13	12533	12842	52	CIRCLE	106	RED	CUS13
14	4831	14016	73	CIRCLE	106	RED	CUS14
15	15294	14811	54	CIRCLE	106	RED	CUS15
16	2277	7393	24	CIRCLE	106	RED	CUS16

### Small.sup Supplier Data File Example

1	13796	3353	78	TRIANGLE	106	GREEN	SUP1
2	13422	15466	100	TRIANGLE	106	GREEN	SUP2
3	9784	6279	27	TRIANGLE	106	GREEN	SUP3
4	4756	13442	3	TRIANGLE	106	GREEN	SUP4
5	6014	1482	68	TRIANGLE	106	GREEN	SUP5
6	1163	283	92	TRIANGLE	106	GREEN	SUP6
7	4414	4366	59	TRIANGLE	106	GREEN	SUP7
8	11059	13402	73	TRIANGLE	106	GREEN	SUP8
9	7759	3285	75	TRIANGLE	106	GREEN	SUP9
10	7495	7327	95	TRIANGLE	106	GREEN	SUP10
11	11911	1732	60	TRIANGLE	106	GREEN	SUP11
12	6163	11760	61	TRIANGLE	106	GREEN	SUP12
13	9158	5781	16	TRIANGLE	106	GREEN	SUP13
14	3601	6802	81	TRIANGLE	106	GREEN	SUP14
15	8519	15434	76	TRIANGLE	106	GREEN	SUP15
16	5529	2703	66	TRIANGLE	106	GREEN	SUP16

### Small.dep Depot Data File Example

1	6074	8000	0	SQUARE	106	BLUE	DEP1
---	------	------	---	--------	-----	------	------

### Small.rou Route Data File Example

7	5	DARKGRAY
Depot	1	
Customer	12	
Customer	7	
Customer	6	
Customer	9	
Customer	1	
Customer	11	
Customer	16	
Supplier	14	
Supplier	7	
Supplier	6	
Supplier	16	
Supplier	5	
Depot	1	
7	6	NAVY
Depot	1	
Customer	8	



Customer	14	
Customer	3	
Customer	10	
Customer	2	
Customer	13	
Customer	15	
Supplier	2	
Supplier	8	
Supplier	15	
Supplier	4	
Supplier	12	
Supplier	9	
Depot	1	
2	5	FOREST
Depot	1	
Customer	4	
Customer	5	
Supplier	10	
Supplier	13	
Supplier	3	
Supplier	1	
Supplier	11	
Depot	1	

---

## Vigo and Toth Case 51-80

This case was first described by Eilon in the framework of the classical Vehicle Routing Problem (VRP). The problem contained 50 linehaul customers and a depot. This corresponds to a total of 51 facilities which is the first segment of the number in the case name. Vigo and Toth converted this problem to a linehaul-backhaul or Vehicle Routing Problem with Backhauls (VRPB) by randomly assigning 80 % of the points to be linehaul customers and 20 % of the points to be backhaul suppliers. The 80 % corresponds to the second part of the number in the project name.

### Case5180.dat Project Data File Example

```
[data_version]          20000
[project_name]          Case5180
[minimum_x_coordinate]  15700
[maximum_x_coordinate]  16100
[minimum_y_coordinate]  17500
[maximum_y_coordinate]  18100
[number_of_vehicles]     4
[vehicle_capacity]      160
[infeasibility_penalty] 1000
[distance_norm]         Euclidean
[number_of_customers]   40
[customer_file_name]    case5180.cus
[number_of_suppliers]   10
[supplier_file_name]    case5180.sup
[number_of_depots]      1
[depot_file_name]       case5180.dep
[number_of_routes]      4
[route_file_name]       case5180.rou
```

[report_level]	3
[seed]	123
[number_of_iterations]	20
[time_limit]	30

### Case5180.cus Customer Data File Example

1	16007	18012	7	CIRCLE	20	RED	NONE
2	16019	18009	30	CIRCLE	20	RED	NONE
3	16022	18024	16	CIRCLE	20	RED	NONE
4	15990	17986	9	CIRCLE	20	RED	NONE
5	15991	18007	15	CIRCLE	20	RED	NONE
6	15987	18023	19	CIRCLE	20	RED	NONE
7	16001	18022	23	CIRCLE	20	RED	NONE
8	16022	17993	11	CIRCLE	20	RED	NONE
9	16012	18001	19	CIRCLE	20	RED	NONE
10	16001	17992	29	CIRCLE	20	RED	NONE
11	15975	17985	23	CIRCLE	20	RED	NONE
12	15982	18002	21	CIRCLE	20	RED	NONE
13	16022	18001	15	CIRCLE	20	RED	NONE
14	15997	17983	3	CIRCLE	20	RED	NONE
15	15987	17993	41	CIRCLE	20	RED	NONE
16	15983	17973	9	CIRCLE	20	RED	NONE
17	16032	18002	8	CIRCLE	20	RED	NONE
18	16012	18017	8	CIRCLE	20	RED	NONE
19	15986	18017	16	CIRCLE	20	RED	NONE
20	15978	18012	10	CIRCLE	20	RED	NONE
21	15997	18028	7	CIRCLE	20	RED	NONE
22	16000	18008	15	CIRCLE	20	RED	NONE
23	16013	18027	14	CIRCLE	20	RED	NONE
24	16028	18008	6	CIRCLE	20	RED	NONE
25	16007	18029	11	CIRCLE	20	RED	NONE
26	16008	18006	12	CIRCLE	20	RED	NONE
27	16016	17970	23	CIRCLE	20	RED	NONE
28	16031	17993	26	CIRCLE	20	RED	NONE
29	16033	18029	6	CIRCLE	20	RED	NONE
30	16002	17982	9	CIRCLE	20	RED	NONE
31	16015	17995	15	CIRCLE	20	RED	NONE
32	16029	17975	14	CIRCLE	20	RED	NONE
33	15980	17977	27	CIRCLE	20	RED	NONE
34	15991	17970	13	CIRCLE	20	RED	NONE
35	15975	18024	11	CIRCLE	20	RED	NONE
36	16000	17975	16	CIRCLE	20	RED	NONE
37	16002	17999	5	CIRCLE	20	RED	NONE
38	15995	17992	25	CIRCLE	20	RED	NONE
39	15995	18015	17	CIRCLE	20	RED	NONE
40	16018	17988	18	CIRCLE	20	RED	NONE

### Case5180.sup Supplier Data File Example

1	16010	17990	21	TRIANGLE	20	GREEN	NONE
2	16021	17981	5	TRIANGLE	20	GREEN	NONE
3	16006	17976	10	TRIANGLE	20	GREEN	NONE
4	16027	18018	28	TRIANGLE	20	GREEN	NONE
5	15977	17998	28	TRIANGLE	20	GREEN	NONE

6	16028	17987	19	TRIANGLE	20	GREEN NONE
7	16032	18023	17	TRIANGLE	20	GREEN NONE
8	15975	17966	7	TRIANGLE	20	GREEN NONE
9	16009	17970	10	TRIANGLE	20	GREEN NONE
10	16026	17997	10	TRIANGLE	20	GREEN NONE

### Case5180.dep Depot Data File Example

D1	16000	18000	0	SQUARE	40	BLUE NONE
----	-------	-------	---	--------	----	-----------

### Case5180.rou Route Data File Example

9	5	NAVY
depot		D1
customer		33
customer		15
customer		38
customer		5
customer		26
customer		1
customer		18
customer		13
customer		29
supplier		7
supplier		4
supplier		10
supplier		6
supplier		2
depot		D1
10	0	OCEAN
depot		D1
customer		11
customer		4
customer		14
customer		37
customer		31
customer		8
customer		2
customer		7
customer		39
customer		12
depot		D1
12	1	OLIVE
depot		D1
customer		16
customer		34
customer		30
customer		9
customer		32
customer		28
customer		17
customer		24
customer		3
customer		23
customer		25

customer	21
supplier	5
depot	D1
9 4	DARKGRAY
depot	D1
customer	20
customer	35
customer	6
customer	19
customer	22
customer	10
customer	36
customer	27
customer	40
supplier	1
supplier	3
supplier	9
supplier	8
depot	D1

# 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. **Lineback** 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 **Lineback** 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 **Lineback** 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 **Lineback** program, help, libraries, and example data files to your hard disk. In addition, it installs **Lineback** 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.

## Win32

**Lineback** 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) and Windows 95 and 98 (all versions). This version of **Lineback** 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

## 2

2-Opt Exchange 48

## 3

3-Opt Exchange 48

## A

About Lineback 95  
Add Customer 68  
Add Depot 66  
Add Random Facilities 68  
Add Supplier 68  
Algorithms 82  
    2-Opt Exchange 48  
    3-Opt Exchange 48  
    Cheapest Insertion 46  
    Evaluate 50  
    Move Facility 49  
    Nearest Neighbor 44, 82  
    Or-Opt Exchange 49  
    Priciest Insertion 46  
    Swap Facility 50  
All Distances 75  
Arrange Icons 93  
ASCII 105  
atlanta.map 36

## C

Cascade 92  
Case5180.cus 102  
Case5180.dat 101  
Case5180.dep 103  
Case5180.rou 103  
Case5180.sup 40, 102  
Cheapest Insertion 46  
Close 56  
Close Log 61

Color 38

## Commands

About Lineback 95  
Add Customer 68  
Add Depot 66  
Add Random Facilities 68  
Add Supplier 68  
All Distances 75  
Arrange Icons 93  
Cascade 92  
Close 56  
Close Log 61  
Compute Distances 76  
Context Sensitive Help 94  
Copy 82  
Delete All Facilities 74  
Delete All Routes 76  
Delete Customer 72  
Delete Customer by Label 73  
Delete Depot 72  
Delete Depot by Label 73  
Delete Facility 73  
Delete Supplier 72  
Delete Supplier by Label 73  
Distance Norm 78  
Edit Customer 70  
Edit Customer by Label 71  
Edit Depot 70  
Edit Depot by Label 71  
Edit Facility 71  
Edit Supplier 71  
Edit Supplier by Label 72  
Evaluate 88  
Exit 66  
Export 58  
Extreme Crossovers 83  
Grid 89  
Grid Size 89  
Help Topics 94  
Import 57  
Infeasibility Penalty 78  
Label Size 89  
Manual 83  
Max. Replications 80  
Most Recently Used Files 66  
New 52  
New Notes Window 91  
New Routes Window 92  
New Statistics Window 92  
Number of Vehicles 77  
Open 55  
Opened Windows 93  
Output Log 61  
Print 62  
Print Preview 63  
Print Setup 65

- Properties 59
- Redraw 91
- Report Level 78
- Save 56
- Save As 56
- Savings 83
- Seed 80
- Send 59
- Status Bar 93
- Sweep 82
- Symmetric Distances 76
- Tile 93
- Time Limit 81
- Toolbar 93
- Vehicle Capacity 77
- Zoom 90
- Zoom Original 90
- Zoom Previous 90
- Common Data Items 37
  - Color 38
  - Display Symbol 38
  - Latitude 37
  - Longitude 37
- Compute Distances 76
- Context Sensitive Help 94
- Copy 82
- Customer File Name 40

## D

- Delete All Facilities 74
- Delete All Routes 76
- Delete Customer 72
- Delete Customer by Label 73
- Delete Depot 72
- Delete Depot by Label 73
- Delete Facility 73
- Delete Supplier 72
- Delete Supplier by Label 73
- Depot File Name 40
- Display Color 39
- Display Size 39
- Display Symbol 38, 39
- Distance Norm 34, 78

## E

- Edit 66
- Edit Customer 70
- Edit Customer by Label 71
- Edit Depot 70
- Edit Depot by Label 71
- Edit Facility 71
- Edit Supplier 71
- Edit Supplier by Label 72
- Evaluate 50, 88

- Exit 66
- Export 58
- Extreme Crossovers 83

## F

- Facility Data
  - Display Color 39
  - Display Size 39
  - Display Symbol 39
  - Label 38
  - Latitude 38
  - Longitude 39
  - Name 38
  - Quantity 39
- Facility Data File 41
- File 51
- Files
  - atlanta.map 36
  - Case5180.cus 102
  - Case5180.dat 101
  - Case5180.dep 103
  - Case5180.rou 103
  - Case5180.sup 40, 102
  - Facilities Data 41
  - msflxgrd.ocx 4
  - Project Data 40
  - Routes Data 42
  - scienmfc.dll 4
  - Scienmfc.dll 106
  - setup.exe 106
  - Setup.exe 3
  - Small.cus 99
  - Small.dat 99
  - Small.dep 100
  - Small.rou 100
  - Small.sup 100
  - South America North.map 37
  - Tutorial.dat 15
  - usa.map 7, 36

## G

- Grid 89
- Grid Size 89
- GUI 105

## H

- Help 94
- Help Topics 94

## I

- Import 57
- Infeasibility Penalty 34, 78



Installation 3

## K

Kirkpatrick 48

## L

Label 38

Label Size 89

Latitude 37, 38

Longitude 37, 39

## M

Manual 83

Map Data 36

Map Data File Name 33

Map File Format 36

Map Projection 31

Maximum Replications 80

Maximum X or East Longitude 33

Maximum Y or North Latitude 33

Menu 51

    Algorithms 82

    Edit 66

    File 51

    Help 94

    Overview 51

    Toolbar 51

    View 88

    Windows 91

Minimum X or West Longitude 32

Minimum Y or South Latitude 33

Most Recently Used Files 66

Move Facility 49

MRU 105

msflxgrd.ocx 4

## N

Name 38

Nearest Neighbor 44, 82

New 52

New Notes Window 91

New Routes Window 92

New Statistics Window 92

Number of Iterations 41

Number of Replications 35

Number of Vehicles 34, 77

## O

Open 55

Opened Windows 93

Or-Opt Exchange 49

Output Log 61

## P

Priciest Insertion 46

Print 62

Print Preview 63

Print Setup 65

Project Data File 40

Project Name 31, 40

Properties 59

## Q

Quantity 39

## R

Redraw 91

Registry 5

Removal 5

Report Level 34, 41, 78

RGB 105

Route File Name 40

Routes Data File 42

## S

Save 56

Save As 56

Savings 83

scienmfc.dll 4

Scienmfc.dll 106

Scientif 3, 5, 106

Seed 35, 41, 80

Send 59

Setup 3, 106

setup.exe 106

Setup.exe 3

Small.cus 99

Small.dat 99

Small.dep 100

Small.rou 100

Small.sup 100

South America North.map 37

Status Bar 93

Supplier File Name 40

Swap Facility 50

Sweep 82

Symmetric Distances 76

## T

Tile 93

Time Limit 35, 41, 81

Tolerance 41

Toolbar 93  
Tutorial.dat 15

## **U**

Uninstall 5  
usa.map 7, 36

## **V**

Vechi 48  
Vehicle Capacity 34, 77  
View 88

## **W**

Win32 106  
Windows 91  
World Radius 34  
WWW 106

## **Z**

Zoom 90  
Zoom Original 90  
Zoom Previous 90